Modeling and Analysis of High-Frequency Microprocessor Clocking Networks

A Dissertation

Presented to

the Academic Faculty

by

Martin Saint-Laurent

in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy in Electrical Engineering

Georgia Institute of Technology

School of Electrical and Computer Engineering

August 2005

Modeling and Analysis of High-Frequency Microprocessor Clocking Networks

Approved by:

Prof. Madhavan Swaminathan, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Prof. James D. Meindl
School of Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Jeffrey A. Davis
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: July 14, 2005

Acknowledgments

I would like to express my gratitude to Professor Madhavan Swaminathan for his support as my dissertation advisor. I would also like to thank Professors Jeffrey A. Davis and James D. Meindl for their comments and suggestions.

Table of Contents

List of Tables

List of Figures

Summary

Integrated systems with billions of transistors on a single chip are a now reality. These systems include multi-core microprocessors and are built today using deca-nanometer devices organized into synchronous digital circuits. The movement of data within such systems is regulated by a set of predictable timing signals, called clocks, which must be distributed to a large number of sequential elements. Collectively, these clocks have a significant impact on the frequency of operation and, consequently, on the performance of the systems. The clocks are also responsible for a large fraction of the power consumed by these systems.

The objective of this dissertation is to better understand clock distribution in order to identify opportunities and strategies for improvement by analyzing the conditions under which the optimal tradeoff between power and performance can be achieved, by modeling the constraints associated with local and global clocking, by evaluating the impact of noise, and by investigating promising new design strategies for future integrated systems.

CHAPTER 1

Introduction and Motivation

1.1 Background and Objective

Integrated systems with billions of transistors on a single chip are now a reality. These systems are constructed using large arrays of memory cells and banks of sequential elements separated by logic gates. In such systems, the movement of data is regulated by a set of predictable timing signals. These signals are called clocks. Collectively, these clocks have a significant impact on the frequency of operation and, consequently, on the performance of the systems. The clocks are also responsible for a large fraction of the power consumption of these systems.

The timing of the clocks is very important. Ideally, the relative phase of each clock should be well known. Each clock should have a predefined relationship specified with respect to an absolute time reference or virtual global signal called *the* clock.

The problem is that, in practice, these clocks cannot always maintain a perfectly predictable relationship with respect to each other. In high-frequency microprocessors, as well as in many other integrated systems, an analog circuit called a phase-locked loop (PLL) is used almost all the time for clock generation. The PLL receives a low-frequency clock, generated off-chip, from which it produces the frequency-multiplied on-chip clock. Because the sequential elements are typically scattered over the entire die, an interconnection network and a set of local clock buffers (LCBs) are required to distribute this on-chip clock. Unfortunately, the distribution network unavoidably introduces some undesirable random delay distortions. Because of these distortions, the clocks received by the sequential elements can never be predicted with certainty. This clock inaccuracy is very undesirable because it limits system performance and can cause functional failures.

In addition to timing, the power consumption associated with the clocks is of primary importance. For instance, for the 180-nm microprocessor described in [1.1], it represents 33% of the total power (leakage is negligible). In [1.2], for a 130-nm technology, the clocks are reported to consume 42% of the total dynamic power. The main components of the clock power are the power of the global clock distribution network, the power of the buffers and interconnects used for local clocking, and the power of the sequential elements. The clock distribution literature has historically focused on the first component. Now, however, the last two usually dominate [1.2], [1.3].

The objective of this dissertation is to better understand clock distribution in order to identify opportunities and strategies for improvement by analyzing the conditions under which the optimal tradeoff between power and performance can be achieved (Chapter 3), by modeling the constraints associated with local and global clocking (Chapters 4 and 5), by evaluating the impact of noise (Chapter 6), and by investigating promising new design strategies for future integrated systems (Chapter 7). For this, modeling and analyzing clocking in high-frequency microprocessors, where clock distribution has traditionally been the most challenging, is a natural choice.

1.2 Clock Inaccuracy and Clock Distribution

In this dissertation, the inaccuracy of a clock is defined in the phase domain, as the sum of the two components shown in Figure 1.1. There, the inaccuracy of the non-ideal clock Φ is determined with respect to an ideal clock having a period T_{cycle}. The first component is the constant phase error and is called the skew. The second component, the jitter, is the time-varying phase error. Mathematically, the phase of the non-ideal clock Φ can be expressed as:

$$\Phi(t) = \frac{2\pi}{T_{cycle}} t + \Phi_{skew} + \Phi_{jitter}(t) \tag{1.1}$$

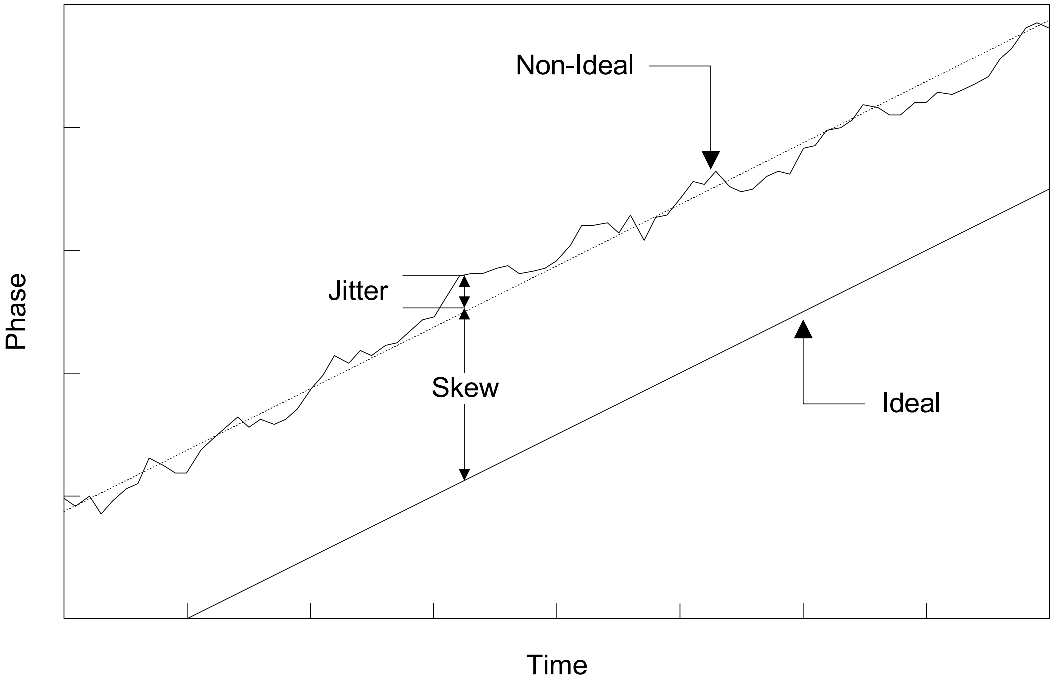


Figure 1.1: Definition of skew and jitter.

where the first term of the equation describes the phase of the ideal clock. The inaccuracy between a *pair* of clocks Φ_i and Φ_j is:

$$\Phi_i(t) - \Phi_j(t) = \Phi_{skew,i} - \Phi_{skew,j} + \Phi_{jitter,i}(t) - \Phi_{jitter,j}(t) \tag{1.2}$$

The pairwise clock inaccuracy thus has a skew and a jitter component as well.

It is sometimes convenient to distinguish between the rising and falling transitions of a clock. This can be done using vectors to represent Φ_{skew} and Φ_{jitter}:

$$\begin{gathered} \Phi_{skew} = \begin{bmatrix} \Phi_{skew,rise} \\ \Phi_{skew,fall} \end{bmatrix} \\ \Phi_{jitter}(t) = \begin{bmatrix} \Phi_{jitter,rise}(t) \\ \Phi_{jitter,fall}(t) \end{bmatrix} \end{gathered} \tag{1.3}$$

This definition for clock inaccuracy is general enough to describe clocks where the high and low phases have different durations. It is also sufficiently general to describe phase jitter in addition to cycle-to-cycle jitter.

1.2.1 Sources of Clock Inaccuracy

There are several sources of clock inaccuracy for integrated systems. Some are related to manufacturing variations. Others are due to the noise generated by the chips themselves or caused by the random conditions in which the chips operate.

The manufacturing process is not perfect, and various effects cause device and interconnect parameters to deviate from their expected values. The most important device parameters include the channel length, the threshold voltage, and the gate oxide thickness. For interconnects, the important parameters include the metal width, metal thickness, and interlevel dielectric (ILD) thickness.

For MOSFETs where the gate oxide is only a few atoms thick, the atomic scale roughness of the gate-to-oxide and oxide-to-silicon interfaces can lead to significant oxide thickness variation (OTV), as discussed in [1.4] by Asenov *et al*. This produces variations in threshold voltage and carrier mobility. The roughness is usually characterized by an amplitude and a correlation length. The amplitude describes how far one side of the interface can penetrate the other. The correlation length describes the rate at which the amplitude spatially varies. When the interface correlation length is small compared to the device dimensions, the oxide thickness variations tend to partially cancel each other out. The variations then produce a quasi-Gaussian threshold voltage distribution that is relatively tight. However, when the device dimensions shrink and become comparable to the correlation length, the oxide thickness becomes more uniform over the gate region while its average value fluctuates more. This leads to greater threshold voltage fluctuations. Asenov *et al*. point out that correlation lengths of 1.0 to 3.0 nm are reported based on transmission electron microscopy measurements and that apparently conflicting correlation lengths of 10.0 to 30.0 nm are also reported based on atomic force microscopy measurements. This uncertainly makes the impact of OTV more difficult to predict. As discussed in [1.5], surface roughness also produces scattering which reduces effective carrier mobility and prevents ballistic transport.

The random placement of the atoms used for channel doping also causes fluctuations in threshold voltage and saturation current [1.6], [1.7]. The MOSFET depletion region can be modeled as an array of small cubic MOS capacitors separating the source from the drain. Since the doping concentration for each capacitor is different, each capacitor has a different threshold voltage. For a given gate voltage, only the MOS capacitors having a sufficiently low doping concentration will be inverted. A conductive current path from the source to the drain may or may not exist. The actual gate voltage at which current percolation begins is determined by the random position of the cubes containing the fewer impurities. Even for double-gate devices, where channel doping is not required, the random placement of dopants in the source and drain regions can create fluctuations of the effective channel length along the width of the device [1.8].

Gate line edge roughness (LER), the random deviation of the gate line edges from their average position, is another source of parameter fluctuations [1.9], [1.10]. LER is caused by tolerances inherent to materials and tools used during lithography. It has not scaled down with line width and only reduces with improved process conditions. Like OTV, LER is characterized using an amplitude standard deviation and a correlation length. The amplitude is typically of the order of 5 nm [1.10], but can be reduced to at least 2 nm with a well controlled process [1.9]. For deca-nanometer MOSFETs, according to Asenov *et al*., LER produces threshold voltage and on and off current fluctuations on a scale similar to the fluctuations introduced by random dopant placement. LER could potentially be a major problem for double-gate MOSFETs [1.8].

The multilevel interconnect structures used in integrated circuits are fabricated layer by layer. Chemical-mechanical polishing (CMP) is used for planarization [1.11]. After the metal interconnects are patterned and the interlevel dielectric (ILD) is deposited, the wafer is pressed against a polishing pad. A slurry of fluid and particles is added as a chemical abrasive to soften the surface of the dielectric. The combined rotation of the wafer and the pad produces wear that planarizes the dielectric. However, even after CMP, non uniformities remain at the surface of the die. These non uniformities are strongly correlated with the pattern density of the mask used to produce the metal interconnects.

Sparse regions polish faster than dense ones. The result is that dense regions have a thicker ILD and, therefore, less capacitance. Although ILD thickness variations can be reduced using metal fill, they cannot be completely eliminated. Moreover, metal fill patterns tend to increase capacitance (when the additional metals are grounded) or crosstalk (when the additional metals are left floating) [1.12].

The dual damascene (DD) approach is commonly used to pattern the metals. With DD, the line trenches and vias are filled simultaneously in a single copper deposition step. First, the dielectric is etched (via first or trench first). Thin diffusion barriers are required to prevent the copper from contaminating the silicon devices and also from degrading the insulating performance of the intermetal and interlayer dielectrics. The barriers are constructed with materials like tantalum, tantalum nitride, or titanium nitride, using either physical vapor deposition (PVD) or chemical vapor deposition (CVD). Following the liner deposition, the trenches and vias are filled with copper using electroplating, or CVD, or a seed layer followed by electrochemical deposition (ECD). This step becomes increasingly difficult as the aspect ratio of the trenches and vias increases. CMP is then used for metal planarization. The line width and pattern density dependencies are different for copper and oxide CMP. Since copper is softer than silicon dioxide, it gets polished faster. Over regions with a high line density, this additional erosion can significantly reduce the metal thickness. In addition, wide lines are susceptible to become thinner than expected in the center (i.e. to dishing), regardless of the local line density. Copper CMP can therefore introduce significant sheet resistance variations across the die [1.13].

Line width is another factor that influences copper resistivity. As discussed in [1.14], metal interconnects that have any of their dimensions reduced to a length comparable to the mean free path of the electrons get significantly more resistive. The increase is due to surface scattering and to grain-boundary scattering. As wire widths scale down, the number of surface collisions experienced by the electrons becomes a significant fraction of the total number of collisions. The grain boundaries also contribute to the resistivity increase by acting like partially reflecting planes. Any random variation in the width of a narrow wire can therefore produce a considerable resistance variation.

During lithography, the proximity effects caused by local layout pattern differences can create significant channel length variations. Optical proximity correction (OPC) is a technique that adjusts the mask patterns to make the printed features closer to the desired shapes. Typical modifications include lengthening a feature, displacing the edge of a pattern, introducing assist features for image quality improvement, and creating serifs to reduce corner rounding [1.15]. But, as discussed in [1.16], perfect mask correction is impossible because of resolution limits preventing the application of arbitrary correction amounts and because of focal plane variations. There exist several correction algorithms that use different strategies to classify the local layout patterns. Some perform better than others, but not on all patterns. Algorithms have also been proposed to supplement OPC by correcting certain fluctuations occurring on a larger scale, such as the systematic intrafield channel length variability. One such algorithm is described in [1.17]. It is based on the empirical characterization of the spatial properties of the channel length variations for different layout patterns. The results, reported for a 180-nm technology, indicate that performing spatial correction can drastically improve circuit speed over OPC alone.

The random and systematic variations introduced during manufacturing are not the only sources of clock inaccuracy for integrated systems. Another source is the noise produced by the chips while they operate. Unlike manufacturing variations, this noise is time-dependent. It is a form of self-interference that can considerably perturb the dynamic characteristics of certain circuits. PLLs for instance are particularly sensitive to digital switching noise in general, and to substrate noise in particular. The end result is additional jitter [1.18], [1.19]. Furthermore, continued device scaling could introduce new problems like, perhaps, random telegraph signal (RTS) noise. RTS noise, observed as a discrete drain current fluctuation occurring at random times and resulting from the trapping and escaping of a single charge by a defect near the silicon-oxide interface, could become a serious issue for analog circuits with MOSFETs operating in weak inversion [1.20], [1.21].

The power delivery system is non-ideal and has voltage fluctuations that randomly affect device delays. These voltage fluctuations include the ripple of the voltage regulation module (VRM). The impedance of the package combined with the impedance of the on-chip power distribution network inevitably produces resistive (*IR*) voltage drops. These *IR* drops reduce the supply voltage reaching the devices, making them slower. Since the current drawn by a chip varies according to the data that it processes, the *IR* drops are time-dependent. The *IR* drops are also position-dependent. The chip regions having the highest activity tend to have larger drops than the more quiet regions. Inductance causes additional noise whenever and wherever the current drawn by the chip varies. The switching activity of the core contributes to this ΔI noise. Even more ΔI noise is produced when a large number of off-chip drivers can switch simultaneously. Given the complex frequency-dependent impedance of the package and of the on-chip power distribution network, it is currently very difficult to determine the timing impact of power-supply noise as a function of the switching activity of the devices, when realistic vectors are used. Although techniques exist to identify the vectors maximizing the noise [1.22], these vectors are not necessarily the ones impacting timing the most.

Furthermore, chip temperature is never uniform. It varies with time according to the data being processed and spatially according to the functional units that are active. Temperature gradients perturb the resistance of the clock interconnects and, consequently, their delay. Random capacitive and inductive coupling with data signals can change the delay of the clock wires as well.

The environmental conditions in which a chip operates also create clock inaccuracy. The variable ambient temperature and the properties of the system used for cooling the chip can significantly alter the average chip temperature. Even with no on-chip temperature gradient, changing the average temperature can still significantly impact interconnect performance. A single chip can also run at various supply voltage levels. Device and interconnect delays are generally difficult to match across a wide range of operating conditions.

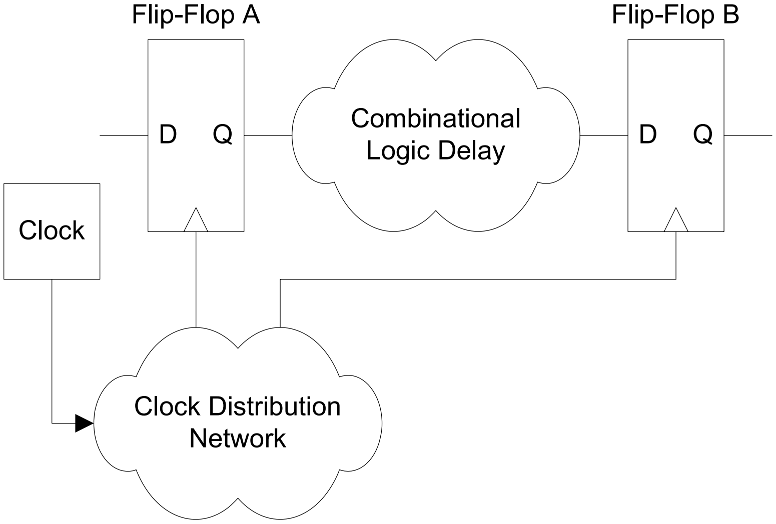


Figure 1.2: General flip-flop circuit.

1.2.2 Impact of Clock Inaccuracy on Frequency and Functional Failures

Clock inaccuracy limits performance and can cause functional failures [1.23]. To show how, the general flip-flop circuit of Figure 1.2 can be analyzed with the assumption that the combinational logic delay T_{data} is bounded by T_{min} and T_{max}. T_{data} includes any interconnect delay present in the combinational logic network. The time required by a flip-flop to respond to a rising edge of the clock and launch its data is called T_{launch}.

Let δ represent the maximum clock inaccuracy introduced by the clock distribution network between Φ_a, the clock generating the data, and Φ_b, the clock sampling the data. In other words, let δ be the maximum absolute time difference between the rising edge of Φ_a and the rising edge of Φ_b. By definition, δ is always positive. The circuit will operate properly if and only if the following two conditions are met:

$$T_{cycle} - T_{launch} - T_{max} - T_{setup} - \delta > 0 \tag{1.4}$$

and

$$T_{launch} + T_{min} - T_{hold} - \delta > 0 \tag{1.5}$$

The first condition is a maximum delay (setup time) check. It expresses that the data produced by flip-flop A must be valid at the input of flip-flop B soon enough to be

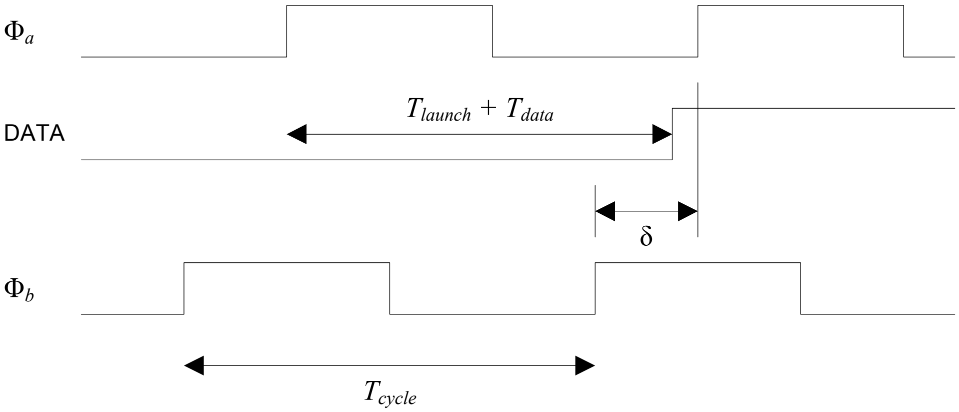


Figure 1.3: Maximum delay constraint.

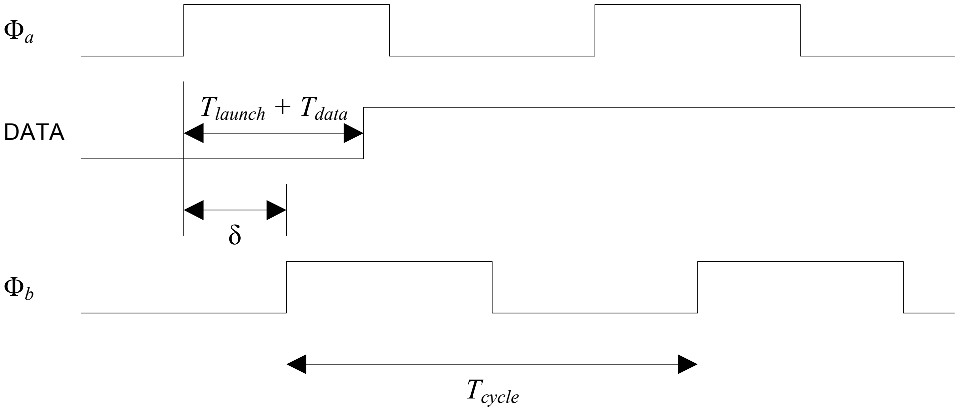


Figure 1.4: Minimum delay constraint.

sampled correctly, even when the sampling event occurs earlier than it should. As shown in Figure 1.3, if Φ_a lags Φ_b, the data will become valid at the input of flip-flop B (T_{launch} + T_{max}) after the rise of Φ_a. The earliest sampling instant can occur ($T_{cycle} - \delta$) after the rise of Φ_a. Since the data must be stable T_{setup} before the sampling event, ($T_{cycle} - \delta$) – (T_{launch} + T_{max}) must be greater than T_{setup} as stated in (1.4).

The second condition is a minimum delay (hold time) check. It expresses that the data produced by flip-flop A for a given clock cycle must be held long enough at the input of flip-flop B to be sampled correctly before being overwritten by the data produced for the next cycle. As shown in Figure 1.4, if Φ_a leads Φ_b, the new data generated by flip-flop A can become valid at the input of flip-flop B (T_{launch} + T_{min}) after the rise of Φ_a. Because the old data was sampled δ after the rise of Φ_a, it must remain stable until ($\delta + T_{hold}$) after the rise of Φ_a. Equivalently, ($T_{launch} + T_{min}$) must be greater than ($\delta + T_{hold}$) as stated in (1.5).

Each constraint gets more difficult to satisfy as the clock inaccuracy δ increases. In practice, violations of the maximum-delay constraint are undesirable but can be solved by increasing T_{cycle}. The cost is performance. However, because hold time violations cause functional failures at all frequencies, the minimum-delay timing parameters tend to be very conservative. If the clock inaccuracy is such that a minimum-delay violation is possible, T_{min} is increased (often by adding buffers) until the violation disappears. Unfortunately, increasing T_{min} frequently increases T_{max} and brings performance degradation.

For a more general system containing a set of clocks C = $\{\Phi_1, \Phi_2, \ldots, \Phi_k\}$ and including a combination of flip-flops and latches, the timing analysis problem is more difficult, especially when the skew between any two pairs of clocks is arbitrary [1.24]. Given a pair of sequentials, the solution proposed by Harris *et al.* for the maximum-delay checks is to make the receiving clock early with respect to the launching clock. It is assumed, without loss of generality, that all latches are transparent when the controlling clock is high. Edge-triggered flip-flops are simpler to handle because they do not allow time borrowing. If a latch is transparent when its input arrives, the data should propagate through the latch and continue through the succeeding stages of combinational logic with respect to the clock that initially launched the path. If a latch is opaque when its input arrives, the path from the launching clock will never constrain timing. A new path should then be started with the opaque latch's clock. Harris *et al.* point out that the timing analysis problem can be simplified and solved approximately by grouping clocks having a small mutual skew into a single domain and by treating the entire domain as a single clock.

For minimum-delay checks, Harris *et al.* make the receiving clock late relative to the launching clock. The system is free from minimum-delay races if, for every consecutive

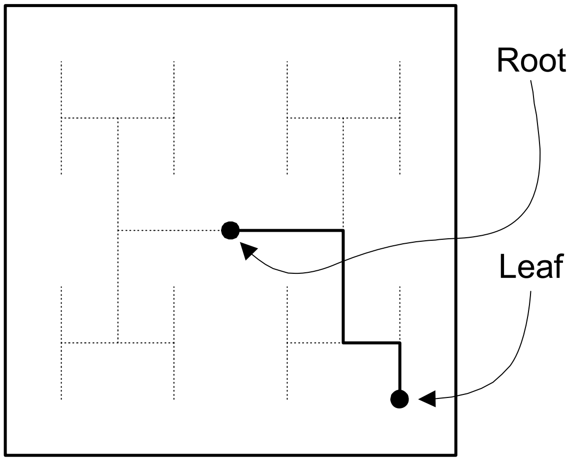


Figure 1.5: Clock distribution tree.

pair of sequentials, the data from the sender cannot arrive at the receiver until some hold time after the previous sampling edge of the receiver. Since the minimum-delay analysis must only check one element and its successor, it is fundamentally easier than the maximum-delay analysis.

1.2.3 Clock Distribution

Two global clock distribution topologies are typically used to attempt to minimize clock inaccuracy and power consumption: clock trees and clock grids. These two basic topologies are often combined to form more complex hierarchical networks [1.1], [1.3], [1.25]-[1.27].

A clock distribution tree is a symmetrical interconnect structure like the one shown in Figure 1.5. The clock is driven from the root and toward the leaves, where loads are attached. Signal repeaters are normally inserted to break long segments. If the delay through each of the branches is equal, the clock pulses generated at the root will arrive simultaneously at every receiver. Generally, the branches do not have to have the same length. To keep the structure electrically symmetrical, only the delays have to be matched.

Layout is usually very critical for clock trees. Every wire has to be carefully shielded to protect the clock signal from capacitive and inductive coupling noise. If the size or the

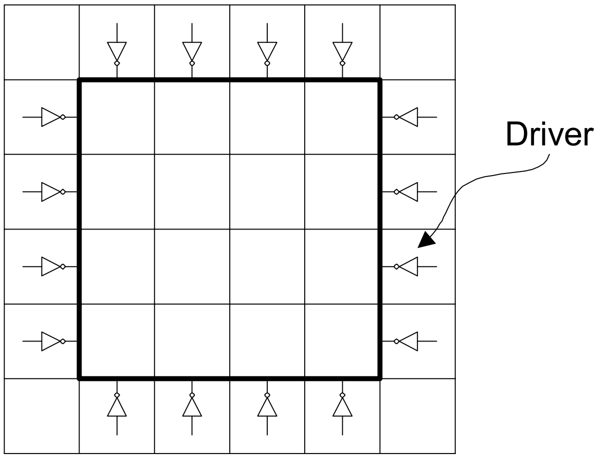


Figure 1.6: Clock distribution grid.

position of a clock receiver changes, it is common to have to rebalance the entire tree. Sometimes, rebalancing the tree requires adding, deleting, and moving a large number of wires, which can be hard to do in metal-congested areas. If the tree cannot be implemented exactly as desired, the clock inaccuracy can increase significantly. Another disadvantage of clock trees is that, to some extent, they constrain the position of the receivers and force them to have a similar capacitive load.

Compared to other clock distribution structures, clock trees are relatively easy to analyze. However, providing early clock inaccuracy estimations is difficult if the number, size, and position of the receivers are unknown. As a result, the early estimates are often conservative and may place unnecessary constraints on the combinational logic separating the sequential elements.

Clock trees are practical at multi-GHz frequencies. For example, the clock distribution network described in [1.26] operates at 2.0 GHz.

A simple clock grid structure is shown in Figure 1.6. Grids have the advantage of being relatively insensitive to the number, size, and position of the clock receivers. Grids can be designed with limited knowledge of the architecture and implementation of the subsystems to be clocked. For grids, size and density largely determine the clock inaccuracy.

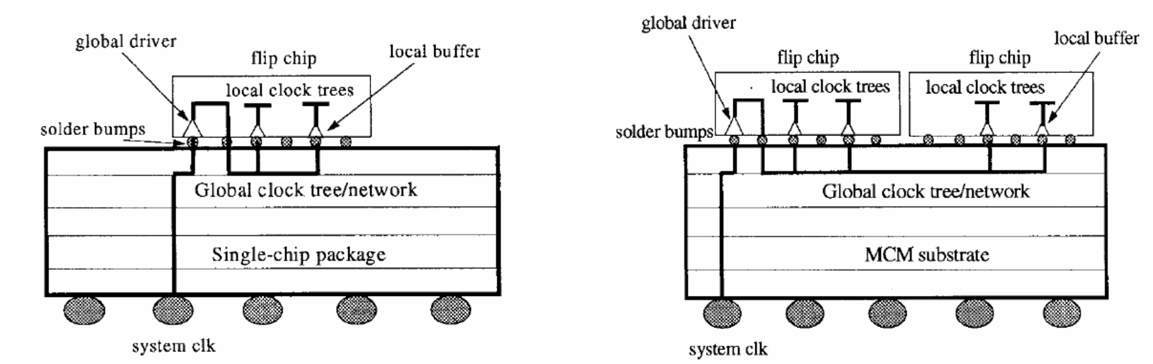


Figure 1.7: In-package clock distribution [1.28].

For high-density grids, the delay variation caused by the change of a receiver load is small. However, dense grids can cost a lot of power if they are not designed carefully [1.25]. By nature, they contain a large number of redundant wires and thus have a large capacitance. As pointed out in [1.3], the power dissipated for global clock distribution is typically rather small compared to the power dissipated for local clock distribution, even when clock gating is used for the local clocks. Therefore, despite their high capacitance, grids remain attractive. Their layout is fairly easy because it is regular. Moreover, if a few grid wires are not perfectly positioned, the impact on clock inaccuracy is often negligible. Clock grids can tolerate a larger amount of electrical noise because local perturbations are rapidly averaged over their entire structure. They often yield a lower clock inaccuracy, simpler architecture, and simpler implementation than clock trees.

1.2.4 In-Package Clock Distribution

Another strategy for global clock distribution in high-frequency microprocessors is in-package clocking. The interconnects available for routing within the package tend to be much wider and thicker than the ones found on-chip. For the same length, the in-package interconnects can be 1000 times less resistive and 10 times less capacitive [1.28]. To take advantage of the better electrical performance of the in-package interconnects, Zhu and Tam have proposed to use them for global clock distribution. Their proposal is illustrated in Figure 1.7 for a flip-chip packaging technology. First, the clock produced by the off-chip clock generator is routed through the package to the chip PLL (not shown in the

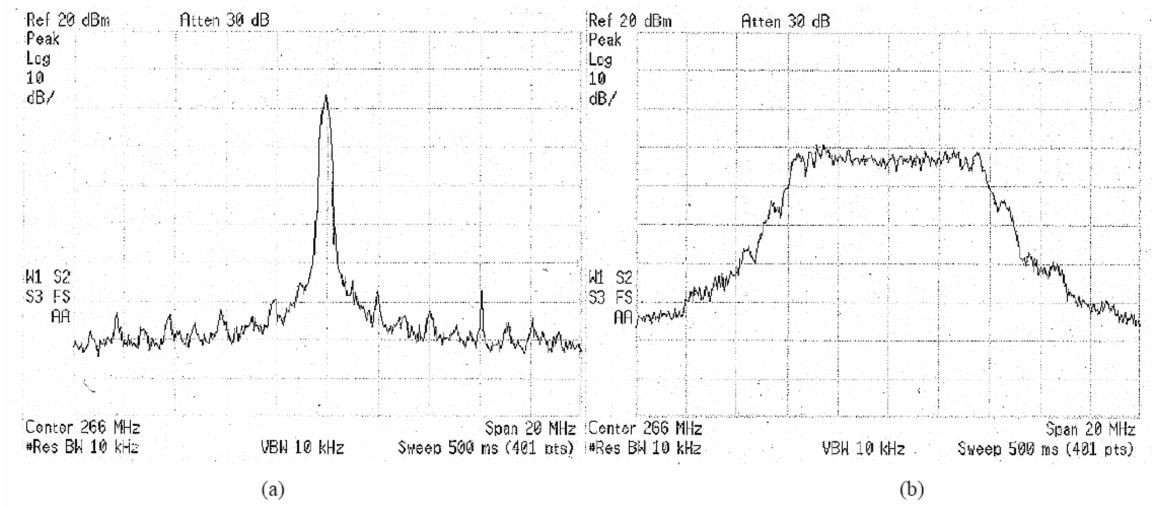


Figure 1.8: Measured clock spectrums (a) without and (b) with spreading [1.31].

figure), which in turns feeds a single global driver. This global driver then connects to the in-package interconnects to feed several local clock buffers. Electrostatic discharge protection is used for the local clock buffers because they could otherwise be damaged during package assembly. According to Zhu and Tam, the potential benefits of in-package clock distribution include: a latency reduction for the global clock distribution network, a decrease in global clock power dissipation, and a chip area improvement obtained by removing the global clock network from the chip. However, they point out in-package clock distribution complicate testing for chips that are not yet packaged because multiple clock sources are needed to feed the unconnected local clock buffers.

The best way to take advantage of the better electrical performance of the in-package interconnects remains a research topic however. At this point, it is unclear if the strategy proposed by Zhu and Tam has any significant power or performance advantage.

1.2.5 Spread-Spectrum Clocking

Clock inaccuracy, although undesirable for timing, is sometimes introduced on purpose and under highly controlled conditions. Complex systems usually require several clocks on their printed-circuit board. These clocks are typically generated by a dedicated chip connected to a reference crystal. For better noise immunity, these clocks are distributed using differential signaling. However, the frequency of the off-chip clocks is commonly modulated by a small amount to reduce electromagnetic interference. This modulation technique, which creates intentional jitter, is called spread-spectrum clocking. It is discussed in [1.29], [1.30], and [1.31]. Figure 1.8 shows the results reported in [1.31].

1.3 Overview

The dissertation is organized as follows. Chapter 2 introduces a new quasi-linear device model whose mathematical simplicity later enables the resolution of several differential equations involving interconnects. Chapter 3 derives the conditions under which the optimal tradeoff between power and performance is achieved in synchronous digital systems. Chapter 4 discusses local clocking, including local clock buffers and sequential elements, as well as interconnect bandwidth and crosstalk jitter. Chapter 5 discusses global clocking and the power impact of the placement of the local clock buffers. It also introduces the concept of interlevel coupling noise and a novel model for it. Chapter 6 analyzes the timing impact of power-supply noise. Chapter 7 finally proposes a new multi-PLL clock distribution architecture that addresses some of the limitations associated with conventional clock distribution strategies and provides additional opportunities for saving power.

CHAPTER 2

A Quasi-Linear MOSFET Model

The MOSFET device model proposed by Sakurai and Newton about 15 years ago [2.1] is widely used in the literature. The model is empirical. It is also compact. It has been successfully used to mathematically analyze the properties of several circuits. Due to some of its non-linearities, however, the model is difficult to apply to differential equations involving interconnects. This chapter proposes a novel quasi-linear MOSFET model better suited for solving interconnect problems, such as the ones that arise during the analysis of high-frequency microprocessor clocking networks.

2.1 Introduction

Several MOSFET models having various degrees of accuracy and complexity have been proposed in the literature over the years [2.1]-[2.5]. Due to its mathematical simplicity, the α-power law model proposed by Sakurai and Newton in [2.1] is still widely used to analytically derive circuit properties.

The model of Sakurai and Newton describes the behavior of the drain current I_D for three regions of operations. In the cutoff region, when the gate-to-source voltage V_{GS} of the device is below the threshold voltage V_T, no current is flowing:

$$I_D = 0 \tag{2.1}$$

In the saturation region, when the drain-to-source voltage V_{DS} is greater than the drain saturation voltage V_{DS0}, the drain current is entirely controlled by the gate voltage:

$$I_D = \frac{W}{L} P_C (V_{GS} - V_T)^{\alpha} \tag{2.2}$$

where

$$V_{DS0} = P_V (V_{GS} - V_T)^{\frac{\alpha}{2}} \tag{2.3}$$

and where α is the velocity saturation index. P_C and P_V are empirical constants. The channel width and length of the device are described by W and L.

In the linear region, when $V_{DS} < V_{DS0}$,

$$I_D = \frac{W}{L}\frac{P_C}{P_V}(V_{GS} - V_T)^{\frac{\alpha}{2}} V_{DS} \tag{2.4}$$

The drain current in saturation when V_{GS} is equal to the supply voltage V_0 is:

$$I_{D0} = \frac{W}{L} P_C (V_0 - V_T)^{\alpha} \tag{2.5}$$

The drain saturation voltage V_{D0} at $V_{GS} = V_0$ is:

$$V_{D0} = P_V (V_0 - V_T)^{\frac{\alpha}{2}} \tag{2.6}$$

Using (2.5) and (2.6), the saturation current of (2.2) and the linear current of (2.4) respectively become:

$$I_D = I_{D0}\left(\frac{V_{GS} - V_T}{V_0 - V_T}\right)^{\alpha} \tag{2.7}$$

and

$$I_D = I_{D0}\frac{V_{DS}}{V_{D0}}\left(\frac{V_{GS} - V_T}{V_0 - V_T}\right)^{\frac{\alpha}{2}} \tag{2.8}$$

which is the form in which the model is typically used. Of course, I_{D0} and V_{D0} can be treated as model parameters, but they are only valid when the supply voltage is V_0. In other words, I_{D0} and V_{D0} are not independent variables, but a function of V_0.

The model proposed in [2.2] by Miura-Mattausch *et al.* is intended for circuit simulations and is far more complex. All device characteristics are described as a function of the drain and source surface potentials, which are calculated iteratively from the applied voltages under the charge-sheet approximation. Because of its iterative nature, the model only implicitly defines the drain current as a function of the applied voltages. The model accounts for carrier drift and diffusion and is valid for all regions of operations. Given the source and drain surface potentials φ_{S0} and φ_{SL}, the drain current is obtained as follows:

$$\begin{aligned} I_D = \frac{W}{L}\frac{\mu}{\beta}\Bigg[& C_{ox}(1+\beta V'_G)(\varphi_{SL}-\varphi_{S0}) - \frac{\beta}{2}C_{ox}(\varphi_{SL}^2-\varphi_{S0}^2) \\ & -\frac{2\sqrt{2}}{3}qN_{sub}L_D\left((\beta\varphi_{SL}-1)^{\frac{3}{2}}-(\beta\varphi_{S0}-1)^{\frac{3}{2}}\right)+\sqrt{2}qN_{sub}L_D\left((\beta\varphi_{SL}-1)^{\frac{1}{2}}-(\beta\varphi_{S0}-1)^{\frac{1}{2}}\right)\Bigg] \end{aligned} \tag{2.9}$$

where q, C_{ox}, N_{sub}, L_D, and β are respectively the electronic charge, the gate oxide capacitance per unit area, the substrate doping concentration, the Debye length, and the inverse of the thermal voltage. V'_G represents the effective gate voltage. The carrier mobility μ is determined using an empirical function of the applied voltages and is assumed independent of the position in the channel.

Later, Bowman *et al*. combined α-power law model of Sakurai and Newton with the physical model proposed in [2.3] in order to enable projections of circuit performance for future technology generations.

The expression for α derived by Bowman *et al*. in [2.4] is:

$$\alpha = \frac{1}{\ln(2)}\ln\left(\frac{2V_{D0}(V_0-V_T-\frac{\eta}{2}V_{D0})}{V_{Da}(V_0-V_T-\eta V_{Da})}\right) \tag{2.10}$$

where V_{D0} and V_{Da} are constants:

$$V_{D0} = E_C(V_0)L\left(\sqrt{1+\frac{2}{E_C(V_0)L}\frac{V_0-V_T}{\eta}}-1\right) \tag{2.11}$$

and

$$V_{Da} = E_C(\tfrac{1}{2}(V_0+V_T))L\left(\sqrt{1+\frac{2}{E_C(\frac{1}{2}(V_0+V_T))L}\frac{\frac{1}{2}(V_0+V_T)-V_T}{\eta}}-1\right) \tag{2.12}$$

The critical electric field E_C for carrier velocity saturation is:

$$E_C(V_{GS}) = \frac{v_{sat}}{\mu_0}(1+\theta(V_{GS}-V_T)) \tag{2.13}$$

The parameter η represents the subthreshold slope factor while θ represents the vertical high field mobility degradation factor.

By definition, α is constant with respect to V_{GS} and V_{DS}.

For I_{D0}, the expression is:

$$I_{D0} = \frac{W}{L}\mu_0 C_{ox} \frac{V_{D0}(V_0 - V_T - \frac{\eta}{2}V_{D0})}{(1+\theta(V_{GS}-V_T))\sqrt{1+\frac{2}{E_C(V_{GS})L}\frac{V_{GS}-V_T}{\eta}}} \tag{2.14}$$

where C_{ox} is the gate oxide capacitance per unit area, as before.

The models proposed in [2.1]-[2.4] all assume that the channel charge is able to respond instantaneously to all voltage changes at the terminals of the device. However, as pointed out in [2.5] by Roy, Vazi, and Patil, this quasi-static assumption starts to be inaccurate when the transition time of the terminal voltages become comparable to the transit time of the carriers through the channel. They account for the non-quasi-static effects by numerically solving the following equation:

$$\frac{dQ_{qs}}{dt} + \frac{dQ_{nqs}}{dt} = k\left[\left(\frac{\partial Q_{nqs}}{\partial x}\right)^2 + Q_{nqs}\frac{\partial^2 Q_{nqs}}{\partial x^2}\right] + k\left[Q_{nqs}\frac{\partial^2 Q_{qs}}{\partial x^2} + 2\frac{\partial Q_{nqs}}{\partial x}\frac{\partial Q_{qs}}{\partial x} + Q_{qs}\frac{\partial^2 Q_{nqs}}{\partial x^2}\right] \tag{2.15}$$

where Q_{qs} is the channel charge under the quasi-static assumption and Q_{nqs} is the transient charge that appears under non-quasi-static conditions. The model is intended for implementation in a circuit simulator. Like the models proposed in [2.2] and [2.3], it is not suitable for the manual derivation of circuit properties.

There are two problems with the α-power law model of Sakurai and Newton. The first one is the velocity saturation index. In saturation, it makes the relationship between V_{GS} and I_D non-linear. For the 180-nm devices of [2.6], the 130-nm devices of [2.7], the 90-nm devices of [2.8] and [2.9], and the 65-nm devices of [2.10], $\alpha \approx 1$ and that non-linearity is an unnecessary mathematical burden.

The other problem with the α-power law model is that it ignores channel-length modulation. In other words, it assumes *flat* I-V curves in saturation. Channel-length modulation occurs when a V_{DS} increase reduces the effective channel length. In saturation, this effective channel length reduction increases the W/L ratio, which increases the drain current. For the devices of [2.6]-[2.10] that span four technology generations, this effect is *not* negligible. The model extension presented in [2.11] includes a channel-length modulation parameter. Unfortunately, this extension makes the model even more difficult to apply to problems involving interconnects. The additional complexity stems from the fact that in saturation, I_D becomes a non-linear function of two variables (V_{GS} and V_{DS}) instead of one.

The physical α-power law model of Bowman *et al*. suffers from the same two problems: α is still an unnecessary mathematical burden and channel-length modulation is still ignored. Furthermore, its mathematical complexity is even higher because it makes I_{D0} a function of the gate voltage. The model is unfortunately too complex to be useful to analytically derive circuit properties. In [2.12] Bowman *et al*. are forced to make the simplifying assumption that I_{D0} is constant in order to be able to analytically determine the delay of an inverter (α is always constant with respect to the gate and drain voltages). With α and I_{D0} treated as constants, albeit physical constants, the physical α-power law model of Bowman *et al*. is no more accurate than the one proposed by Sakurai and Newton in [2.1].

This chapter introduces a novel quasi-linear device model that is mathematically easier to use and also better suited for solving interconnect problems. The model is derived in Section 2.2 and validated in Section 2.3 for a 130-nm device. Section 2.4 uses the model to derive new expressions for the delay required by a device to discharge a capacitive load. The delay predicted using the new expressions for a 130-nm device is compared to the delay obtained using a circuit simulator. For typical input transition times, the error with the quasi-linear device model does not exceed 5%. The accuracy of the model is maintained when the experiment is repeated with a 90-nm device.

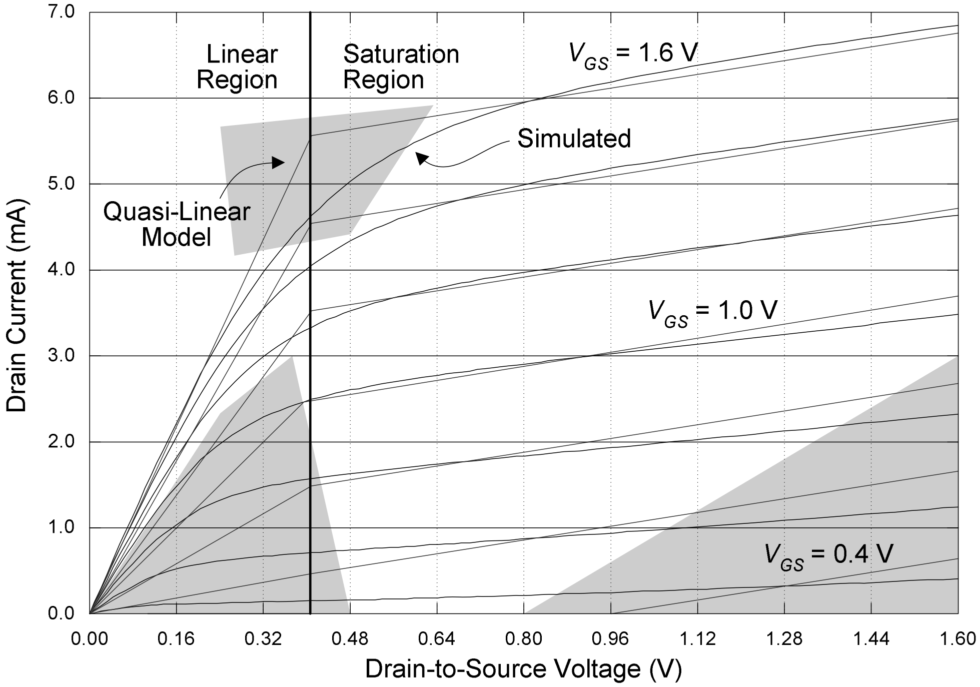


Figure 2.1: I-V curves.

2.2 Derivation

The proposed quasi-linear MOSFET model includes the three regions of operation used in [2.1]: saturation, linear, and cutoff. In saturation, the drain current is given by the following linear relationship:

$$I_D = g_G(V_{GS} - V_T) + g_D(V_{DS} - V_{DS0}) \tag{2.16}$$

where g_G is the gate transconductance, g_D is the drain conductance, and V_{DS0} is the drain-to-source voltage separating the saturation and linear regions. In (2.16), V_{DS0} is a constant parameter. For a device having a width w, it is convenient to express g_G as $g_G = g_{G0}\,w$, where g_{G0} is the gate transconductance per micron. Similarly, $g_D = g_{D0}\,w$.

Equation (2.16) is valid when V_{GS} is sufficient to make I_D positive and when $V_{DS} \geq V_{DS0}$. I_D will be positive when:

$$V_{GS} \geq V_T - g_D(V_{DS} - V_{DS0}) / g_G \tag{2.17}$$

The mathematical simplicity of (2.16) is partially achieved due to the saturation condition defined in (2.17). This saturation condition is more complex than the one of the α-power law MOSFET model.

In the linear region, when $V_{GS} \geq V_T$ and $V_{DS} \leq V_{DS0}$:

$$I_D = g_G (V_{GS} - V_T) \frac{V_{DS}}{V_{DS0}} = \frac{V_{DS}}{R_{lin}} \tag{2.18}$$

where R_{lin} is modulated by V_{GS}:

$$R_{lin} = \frac{1}{g_G} \frac{V_{DS0}}{V_{GS} - V_T} \tag{2.19}$$

If V_{GS} is constant then (2.18) becomes linear. Finally, when the device is neither in saturation nor in the linear region, it is in cutoff. The drain current is then approximated as zero.

2.3 Validation

Figure 2.1 shows that the I-V curves of [2.7] are generally well approximated by the quasi-linear model. The device width is 5.00 µm. V_{GS} varies from 0.4 V to 1.6 V in steps of 0.2 V. The separation between the linear and saturation regions of operation is a simple vertical line. In saturation, the curves are strictly parallel (but generally not flat).

The parameters are chosen to optimize the model accuracy at large drain currents, when V_{GS} is between 1.0 V and 1.6 V.

The model is least accurate when V_{GS} is close to V_T and when V_{DS} is small. In most problems involving an inverter driving a lumped or distributed load (e.g. an interconnect), the reduced accuracy in the linear region at small gate-to-source voltages is irrelevant. The inverter almost never operates there, unless it's input transition time is very slow and its output load very small. The reduced accuracy in the saturation region when the gate-to-source voltage is small is usually not important either. There, the drain currents are small. With realistically fast input transition times, V_{GS} tends to rise rapidly enough to make the error a small fraction of the total charge flowing through the drain.

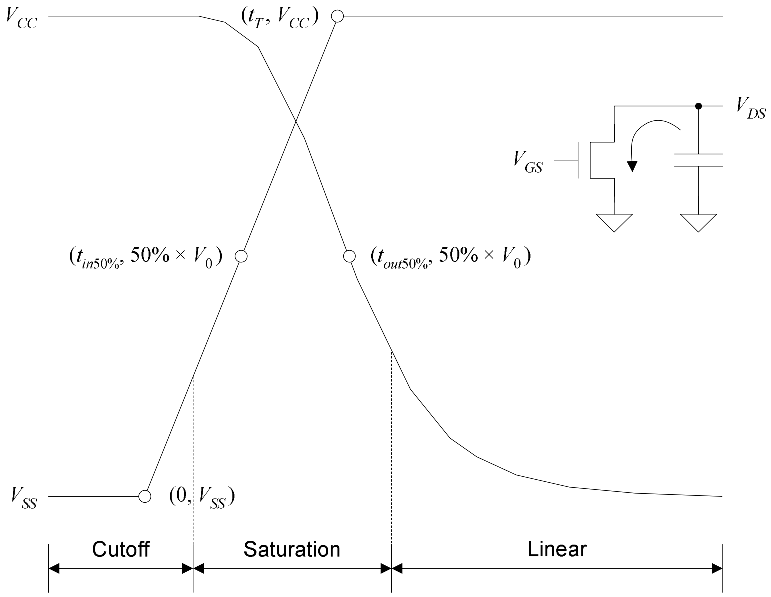


Figure 2.2: Capacitive load discharge.

Finally, the drain current is overestimated around the boundary separating the linear and saturation regions when V_{GS} is higher than the nominal supply voltage. If this region is important, the error there can be minimized by properly extracting the model parameters.

2.4 Application

This section uses the quasi-linear MOSFET model to derive the output waveform of a device discharging a capacitive load *C*. The input waveform applied to the device is shown in Figure 2.2.

The gate-to-source voltage is given by:

$$V_{GS}(t) = V_0\,(t / t_T) \tag{2.20}$$

The time at which the device starts conducting (i.e. the time at which the device goes from cutoff to saturation) is when V_{GS} begins to satisfy (2.17). This occurs at:

$$t_{sat} = \frac{V_T - g_D(V_{DS} - V_{DS0}) / g_G}{V_0} t_T \tag{2.21}$$

Since V_{DS} is initially equal to V_0,

$$t_{sat} = \frac{V_T - g_D(V_0 - V_{DS0})/g_G}{V_0} t_T \tag{2.22}$$

The output voltage after time t_{sat} obeys the following differential equation, for as long at the device remains in saturation:

$$-CV'_{DS}(t) = I_D(V_{GS}(t), V_{DS}(t)) \tag{2.23}$$

For quasi-linear MOSFETs, the drain current is given by (2.16). The differential equation for the output voltage thus becomes:

$$-CV'_{DS} = g_G(V_{GS} - V_T) + g_D(V_{DS} - V_{DS0}) \tag{2.24}$$

Since $V_{GS} = V_0\ (t\ /\ t_T)$,

$$-CV'_{DS} = g_G(V_0(t/t_T) - V_T) + g_D(V_{DS} - V_{DS0}) \tag{2.25}$$

or

$$-\frac{C}{g_D}V'_{DS} - V_{DS} = mt - b \tag{2.26}$$

where m and b are time-invariant:

$$\begin{aligned} m &= \frac{g_G}{g_D}\frac{V_0}{t_T} \\ b &= \frac{g_G}{g_D}V_T + V_{DS0} \end{aligned} \tag{2.27}$$

The solution with initial condition $V_{DS}(t_{sat}) = V_0$ is:

$$V_{DS}(t) = \left(m\left(t_{sat} - \frac{C}{g_D} \right) - b + V_0 \right) e^{-\frac{g_D}{C}(t - t_{sat})} - \left(m\left(t - \frac{C}{g_D} \right) - b \right) \tag{2.28}$$

The device is expected to remain in saturation as the output waveform crosses the 50% point. Based on the I-V curves published in [2.6]-[2.10], this is reasonable since the drain-to-source voltage must fall well below 50% × V_0 for the device to enter its linear region of operation.

When the input transition time is slow, the output waveform may cross the 50% point while the input is still rising. In this case, the time at which the output crosses the 50% point can be obtained from (2.28) by solving $V_{DS}(t) = 50\% \times V_0$:

$$t_{out50\%} = \frac{b - V_0/2}{m} + \frac{C}{g_D} + \frac{C}{g_D} W\left(\frac{g_D(mt_{sat} - b + V_0) - mC}{mC} e^{\frac{g_D}{mC}(mt_{sat} - b + V_0/2) - 1} \right) \tag{2.29}$$

where $W(\cdot)$ is the Lambert W function [2.13]. For negative arguments, the following approximation is reasonably accurate:

$$W(x) \approx \sqrt{2}(\sqrt{ex+1} - \tfrac{3}{4}x - 1) \tag{2.30}$$

At $t = t_T$, the input voltage stabilizes to V_0. The remainder of the derivation assumes that the crossing event occurs after that time. The output voltage at t_T is:

$$V_{DS}(t_T) = \left(m\left(t_{sat} - \frac{C}{g_D} \right) - b + V_0 \right) e^{-\frac{g_D}{C}(t_T - t_{sat})} + m\frac{C}{g_D} - (mt_T - b) \tag{2.31}$$

For $t > t_T$, the output voltage decreases according to the following differential equation:

$$-CV'_{DS} = g_G(V_0 - V_T) + g_D(V_{DS} - V_{DS0}) \tag{2.32}$$

or

$$-\frac{C}{g_D}V'_{DS} - V_{DS} = mt_T - b \tag{2.33}$$

The solution to (2.33) satisfying (2.31) is:

$$V_{DS}(t) = (mt_T - b + V_{DS}(t_T))e^{-\frac{g_D}{C}(t - t_T)} - (mt_T - b) \tag{2.34}$$

Using (2.31) for $V_{DS}(t_T)$ and solving (2.34) for 50% × V_0 yields $t_{out50\%}$:

$$t_{out50\%} = t_T + \frac{C}{g_D} \log \frac{\left(m\left(t_{sat} - \frac{C}{g_D} \right) - b + V_0 \right) e^{-\frac{g_D}{C}(t_T - t_{sat})} + m\frac{C}{g_D}}{mt_T - b + \frac{V_0}{2}} \tag{2.35}$$

By definition, the delay d of the device is $t_{out50\%} - t_{in50\%}$. From Figure 2.2, the input ramp crosses its 50% point at $t_T / 2$. From (2.29) and (2.35), the delay is therefore:

$$d = \begin{cases} \dfrac{b - V_0/2}{m} + \dfrac{C}{g_D} + \dfrac{C}{g_D} W\left(\dfrac{g_D(mt_{sat} - b + V_0) - mC}{mC} e^{\frac{g_D}{mC}(mt_{sat} - b + V_0/2) - 1} \right) - \dfrac{t_T}{2} & (t_T \text{ slow}) \\ \dfrac{t_T}{2} + \dfrac{C}{g_D} \log \dfrac{\left(m\left(t_{sat} - \dfrac{C}{g_D} \right) - b + V_0 \right) e^{-\frac{g_D}{C}(t_T - t_{sat})} + m\dfrac{C}{g_D}}{mt_T - b + \dfrac{V_0}{2}} & (t_T \text{ fast}) \end{cases} \tag{2.36}$$

where the transition time is considered slow if $t_{out50\%} < t_T$ when using (2.29) to compute $t_{out50\%}$.

The transition time of the output waveform is obtained by extrapolating its slope at the 50% point:

$$t_{Tout} = -\frac{V_0}{V'_{DS}(t_{out50\%})} \tag{2.37}$$

From (2.28) and (2.34),

$$t_{Tout} = \begin{cases} \dfrac{C}{g_D} \dfrac{V_0}{mt_{out50\%} - b + V_{DS}(t_{out50\%})} & (t_T \text{ slow}) \\ \dfrac{C}{g_D} \dfrac{V_0}{\frac{g_G}{g_D} V_0 - b + V_{DS}(t_{out50\%})} & (t_T \text{ fast}) \end{cases} \tag{2.38}$$

Using (2.29) for $t_{out50\%}$ and the fact that, by definition, $V_{DS}(t_{out50\%}) = V_0 / 2$:

$$t_{Tout} = \begin{cases} \dfrac{\frac{1}{m} V_0}{W\left(\dfrac{g_D(mt_{sat} - b + V_0) - mC}{mC} e^{\frac{g_D}{mC}(mt_{sat} - b + V_0/2) - 1} \right) + 1} & (t_T \text{ slow}) \\ \dfrac{C}{g_D} \dfrac{V_0}{\frac{g_G}{g_D} V_0 - b + \frac{1}{2} V_0} & (t_T \text{ fast}) \end{cases} \tag{2.39}$$

2.5 Comparison to α-Power Law Model

Table 2.1 gives the simulated delays for a 1.00-μm device discharging a 40-fF load for various input transition times. The technology parameters corresponding to [2.7] are: $g_G = 1.13$ mA/V, $g_D = 0.22$ mA/V, $V_T = 0.60$ V, and $V_{DS0} = 0.40$ V. The supply voltage is set to 1.5 V. The first three input transition times meet the condition required to be considered slow. The remaining ones are fast. The simulated delays are clearly very close

Table 2.1: Delay versus transition time for a 130-nm device.

Input Transition Time (ps)	Simulated Delay (ps)	Quasi-Linear Model (ps)	Error (%)	α-Power Law Model (ps)	Error (%)
20.0	29.5	28.2	-4.4	26.1	-11.5
40.0	32.0	30.9	-3.4	28.8	-9.9
60.0	34.6	33.7	-2.6	31.6	-8.7
80.0	37.1	36.5	-1.6	34.3	-7.4
100.0	38.9	38.6	-0.8	37.1	-4.5
120.0	40.0	40.2	+0.5	39.9	-0.4

Table 2.2: Delay versus transition time for a 90-nm device.

Input Transition Time (ps)	Simulated Delay (ps)	Quasi-Linear Model (ps)	Error (%)	α-Power Law Model (ps)	Error (%)
20.0	24.2	23.1	-4.7	22.1	-8.8
40.0	27.1	26.3	-2.8	24.9	-8.1
60.0	30.0	29.7	-1.0	27.7	-7.6
80.0	32.2	32.5	+0.9	30.5	-5.5
100.0	33.7	34.7	+2.8	33.3	-1.3
120.0	34.6	36.3	+5.0	36.1	+4.3

(within 5%) to the ones predicted by the quasi-linear MOSFET model in (2.36). The error is highest when the input transition time is very fast or very slow, for the reasons discussed in Section 2.3. Table 2.1 also shows the delays calculated for the α-power law model, using the expression given in [2.1]. The error is higher. The parameters α, V_T, and I_{D0} are extracted from Figure 2.3 using the method given in [2.1].

Table 2.2 shows that the quasi-linear MOSFET model is still valid for the 90-nm technology described in [2.14]. The technology features strained-silicon devices having a 45-nm gate length. The load is still 40 fF, but the supply voltage is reduced to 1.2 V. The technology parameters are: $g_G = 1.54$ mA/V, $g_D = 0.25$ mA/V, $V_T = 0.50$ V, and $V_{DS0} = 0.40$ V. The error with the quasi-linear device model is still under 5%. The error with the α-power law model and the parameters extracted based on Figure 2.4 remains higher.

2.6 Summary

The novel quasi-linear MOSFET model described in this chapter is mathematically simpler than the one proposed by Sakurai and Newton about 15 years ago [2.1]. When the device is in saturation, the new model is more accurate because it does not neglect

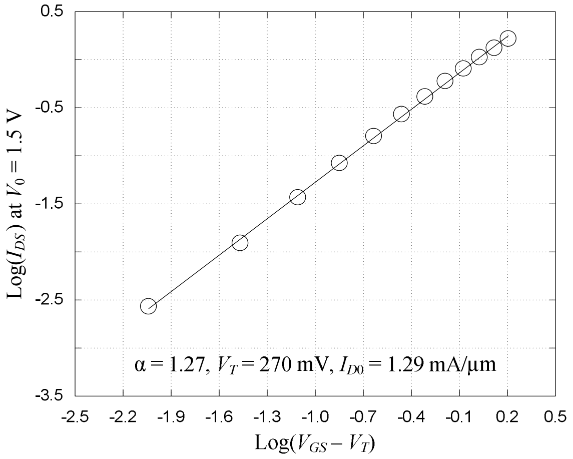


Figure 2.3: 130-nm parameters for the α-power law model.

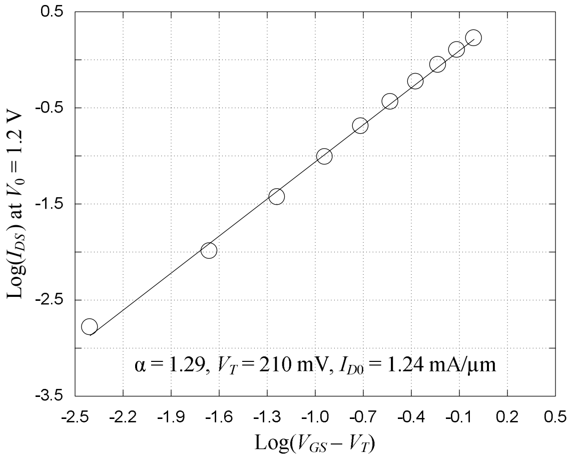


Figure 2.4: 90-nm parameters for the α-power law model.

channel-length modulation. There, the drain current is modeled as a *linear* function of the gate-to-source and drain-to-source voltages. The fixed drain-to-source voltage separating the linear and saturation regions of operation also contributes to the model's mathematical simplicity.

Expressions are derived for the delay required by a 130-nm device to discharge a capacitive load. For typical input transition times, the error with the quasi-linear device model is under 5%. The accuracy of the model is maintained when the experiment is repeated with a 90-nm device.

CHAPTER 3

Power-Efficient Clocking

The tradeoff between the energy required for a computation and the time required to perform it has traditionally been the focus of the low-power circuit design literature. A system can be made faster (up to a limit) if its designer is willing to increase its power consumption. This chapter proposes a model for the relationship between the power consumption and the supply voltage of synchronous digital systems. It also models how the frequency changes when the voltage changes. The general conditions under which the tradeoff between power and performance is optimal are derived. These conditions are then applied to determine how much energy should be allocated for sequencing in these systems, in particular for clock distribution.

3.1 Introduction

The focus of the low-power circuit design literature has traditionally been to improve energy efficiency, usually under some performance constraints, by trying to minimize switching activity and leakage or by optimizing device parameters. In [3.1], Gonzalez and Horowitz argue that for microprocessors, power alone is not a good metric for energy efficiency because it is proportional to clock frequency. Simply reducing the clock speed does reduce power, but does not make the microprocessor better. They point out that the average energy per instruction is a metric that also has problems. Although this metric improves when the supply voltage is lowered or when smaller devices are used (for a given process technology), both of these changes reduce the performance of the microprocessor. They conclude that power and performance must be considered simultaneously and that the simplest metric that does that is the energy-delay product. The energy efficiency of a microprocessor thus improves when either the performance increases or the power decreases without adversely affecting the other quantity.

In [3.2], Zyuban and Kogge explore architectural tradeoffs for reducing power without compromising performance. They start by keeping track of the events generated during the simulation of instruction traces and by associating an energy dissipation to those events. They then observe that the energy per instruction typically grows exponentially with the number of instructions issued per cycle (IPC). This growth is attributed to speculative execution, where instructions are sometimes executed for nothing and where the number of instructions discarded tends to increase with the issue width. They call a superscalar architecture energy-efficient if its energy-delay product does not grow with increasing architecture performance (i.e. IPC).

The methodology proposed by Zyuban and Kogge to optimize the energy efficiency of superscalar architectures requires a systematic exploration of all possible power-performance tradeoffs. In [3.2], the design space is limited to the size of the instruction issue window, the size of the physical register file, and the size of the load-store issue window. First, a performance target is set. This target is directly related to the average instruction runtime R of the architecture. Then, the optimization variables are manipulated to find the implementation with the lowest energy E for that performance. Finally, another performance target is set and the process is repeated. Each step yields a point representing the lowest possible energy achievable for a given average instruction runtime. That point is optimal in the sense that it minimizes ER^w for some w. The resulting power-performance curve represents the set of all energy-efficient implementations for the architecture. Unfortunately, the methodology proposed by Zyuban and Kogge is only practical when the number of optimization variables is relatively small. Increasing the number of variables expands the design space and rapidly increases the time required to perform the optimization. Furthermore, the methodology only deals with architecture variables. Other important optimization variables, most notably the supply voltage, are ignored.

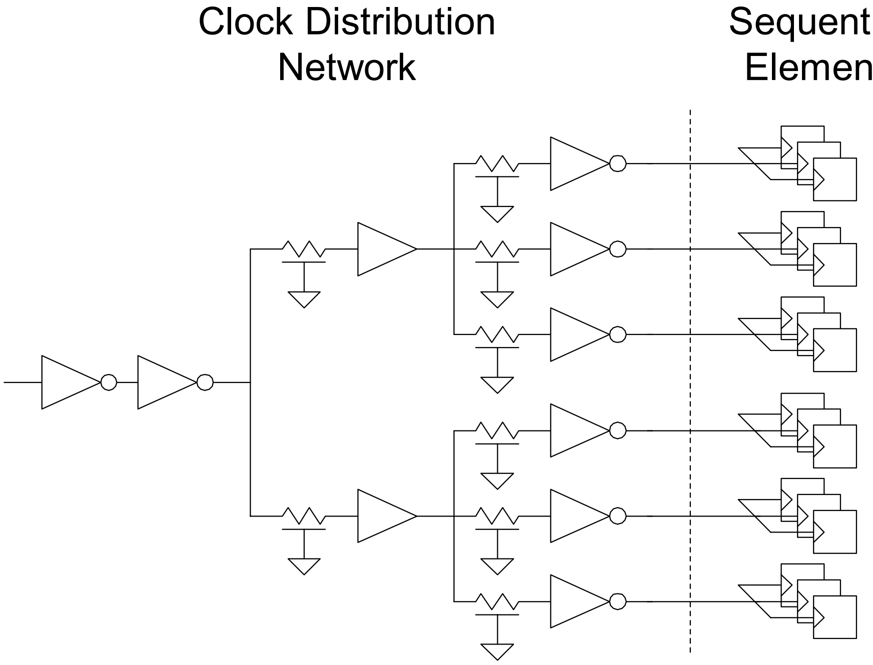


Figure 3.1: Sequencing in synchronous digital systems.

For synchronous digital systems, where the clock is responsible for a large fraction of the power consumed, making the optimal power-performance tradeoff requires the analysis of the overhead associated with sequencing. This overhead includes the skew and the jitter of the clock. It also includes the delay of the sequential elements (e.g. the setup and launch times for flip-flops or the data-to-output delay for latches). The components typically required for sequencing are shown in Figure 3.1. The clock distribution network is one of them.

When designing a synchronous digital system, choosing the right amount of sequencing overhead is non-trivial. It is intrinsically a power-performance tradeoff. For instance, how should the clock inaccuracy targets be set? Should the clock skew between adjacent functional units be 10% of the clock cycle? Or should it be relaxed to 15% to reduce the clock distribution power? Alternatively, maybe the delay of some sequential elements can be improved by making their clocked transistors bigger. If so, should the delay be improved and by how much? The problem addressed here is to define a systematic way of making these kinds of choices in order to design systems that operate where the power-performance tradeoff is optimal.

In the clock distribution literature, the problem of determining the right amount of skew and jitter is typically ignored [3.3]. Instead, the focus is on meeting certain clock inaccuracy targets while dissipating as little power as possible [3.4] or on meeting certain power targets while minimizing the clock inaccuracy [3.5]. In both cases however, these design targets are set using *ad-hoc* methodologies, which may or may not be energy-efficient.

For sequential elements, a power-delay product minimization strategy is often used to choose the setup and the launch times [3.6]. This approach is reasonable, but it has its limits. It is reasonable because it considers the *local* clock power of the sequential elements. The primary limit of this approach is that it ignores the impact of the sequential elements on the rest of the clock distribution network. In general, the load that the clock distribution network must drive has a significant impact on its structure. Therefore, the load strongly influences the power and clock inaccuracy of the network. The fact that this dependency is ignored is a limit of the power-delay product minimization strategy.

This chapter defines how to make the optimal power-performance tradeoff by expanding the solution first introduced in [3.7]. It proposes a new methodology for choosing the right amount of sequencing overhead in a system from an energy-efficiency standpoint. The concept of hardware intensity, introduced later in [3.8], is based on a similar idea. The new methodology is systematic and general. It can be used to jointly optimize the clock distribution network and the sequential elements. It can even be used to make decisions about the architecture of the system. First, the relationship between supply voltage, clock frequency, and power dissipation is examined. New models are developed and validated. It is argued that reducing the sequencing overhead is equivalent to saving power. The equivalence is applied to derive, for the first time, an expression for the optimal allocation of the energy used for sequencing. Then, based on this optimum, the new methodology to design energy-efficient systems is developed. The methodology ensures that the amount of energy allocated for sequencing is appropriate for the *context*

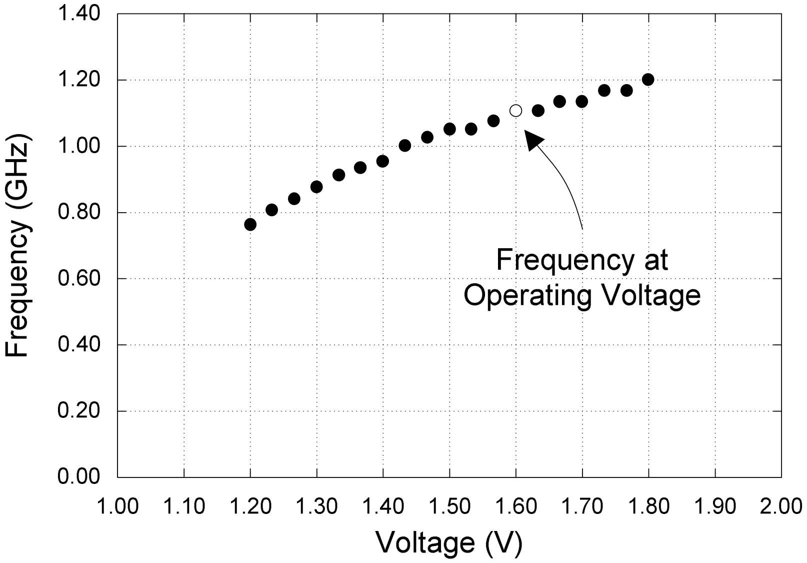


Figure 3.2: Measured frequency-versus-voltage relationship for the SPARC V9.

of each system. In other words, the methodology ensures that the amount of energy allocated for sequencing is appropriate for each system's frequency and power characteristics.

3.2 Relationship Between Voltage, Power Dissipation, and Frequency

Most CMOS integrated circuits can be classified as either frequency-limited or power-limited. A chip is frequency-limited if its maximum operating frequency is determined by its slowest timing path at the maximum supply voltage allowed. Conversely, a chip is power-limited if it reaches the maximum power dissipation allowed when its supply voltage, its clock frequency at this voltage, or both, are below their upper limit.

From a sequencing energy allocation standpoint, frequency-limited chips are not very interesting. There is no power tradeoff to make: they should simply be made as fast as possible. The power dissipated is unimportant because it does not exceed the budget. Power-limited chips however are more interesting and also more common. They may still be designed for high frequency, but not for high frequency at any price. The rest of this chapter only considers power-limited chips. Frequency-limited chips are increasingly rare.

The frequency-voltage relationship of a chip is conceptually straightforward to determine empirically. First, the supply voltage is fixed to a certain value. Then, a sequence of test vectors is applied to the chip while its response is monitored. Finally, the frequency of operation is increased until the chip no longer produces the correct response. This is the maximum frequency at that voltage.

Figure 3.2 shows the measured frequency-voltage relationship for the 64-bit SPARC V9 microprocessor [3.9]. The chip is built using a 150-nm technology that has 7 layers of aluminum interconnects. It is designed to operate at 1.0 GHz with a supply voltage of 1.6 V. The temperature is maintained at 60 °C. The measurements indicate that the frequency-versus-voltage curve is monotonic, smooth, and roughly linear. The measurements also indicate a 10% frequency margin at the nominal supply voltage. This chip is power-limited based on the definition given above since its frequency can be increased if the power consumption is allowed to increase.

3.2.1 Power Dissipation

The power dissipation P of any microprocessor as a function of the supply voltage V and the operating frequency f can be approximated by:

$$P = P_d + P_s + P_{leak} \tag{3.1}$$

where:

$$P_d = C_d V^2 f \tag{3.2}$$

$$P_s = Q_s V f \tag{3.3}$$

$$P_{leak} = I_{leak} V \tag{3.4}$$

The first term of (3.1) is the dynamic power. In (3.2), C_d represents the effective capacitance switched per cycle and is assumed independent of the supply voltage.

Table 3.1: Frequency and power dissipation for a 130-nm ring oscillator.

Supply Voltage (V)	Frequency (GHz)	Power (µW)	Effective Switching Capacitance (fF)
1.0	6.185	177.7	28.73
1.1	7.184	250.1	28.77
1.2	8.115	337.9	28.91
1.3	8.972	441.8	29.14
1.4	9.761	560.6	29.30
1.5	10.480	695.3	29.49
1.6	11.131	850.3	29.84
1.7	11.729	1012.3	29.86
1.8	12.267	1199.8	30.19

The second term of (3.1) is the short-circuit power. In (3.3), Q_s is the average charge flowing directly from the positive to the negative power rails during a clock period. According to the model proposed in [3.10], Q_s is proportional to V. With this model, it is convenient to express Q_s as:

$$Q_s = C_s V \tag{3.5}$$

where C_s is a design-dependant constant. C_s decreases as the average relative capacitance (i.e. fanout) driven by each device increases. C_s also increases with the average signal transition time.

Equation (3.5) implies that the dynamic power and the short-circuit power have the same behavior: they both increase quadratically with the supply voltage and linearly with the frequency. Mathematically,

$$P_d + P_s = C_d V^2 f + Q_s V f = (C_d + C_s) V^2 f \tag{3.6}$$

This is confirmed by the data of Table 3.1 where the total power dissipated by a 130-nm ring oscillator is shown. The oscillator is simulated at various voltages. As the voltage increases, its frequency of oscillation and its power consumption both increase. The effective switching capacitance ($C_d + C_s$) is obtained from the frequency and power data by computing the following ratio: $P / (V^2 f)$. The effective switching capacitance is almost constant.

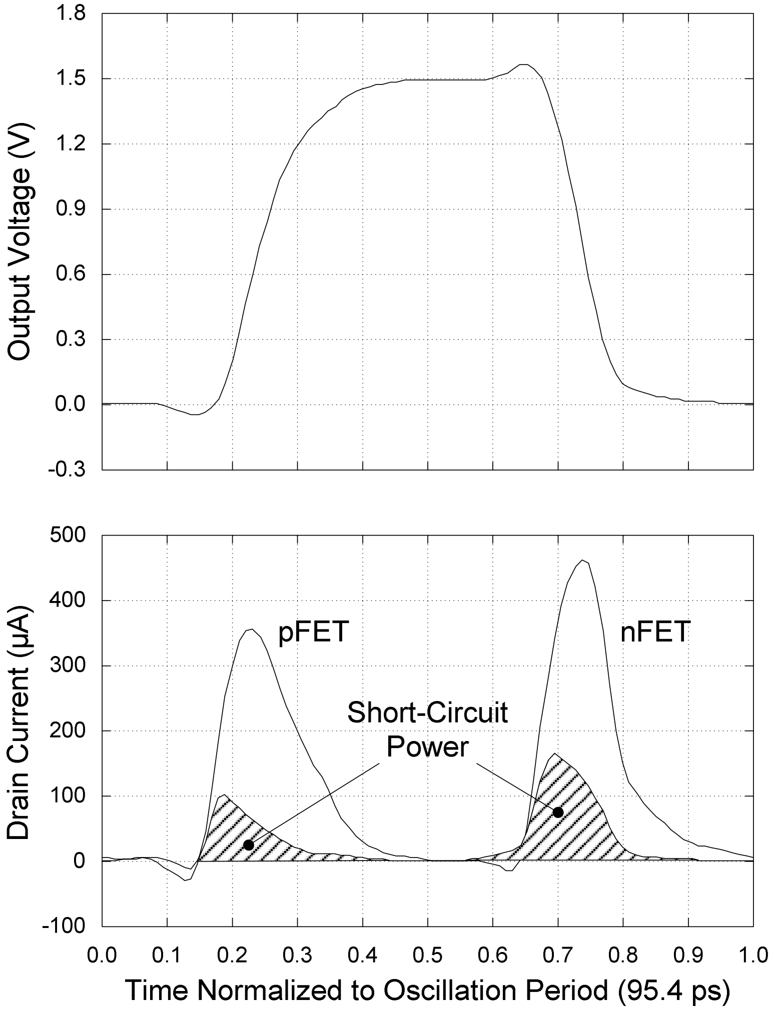


Figure 3.3: Voltage and current waveforms for the 130-nm ring oscillator.

Figure 3.3 shows the drain currents of the devices of one of the stages of the oscillator in conjunction with the output voltage of that stage. The short-circuit current is clearly significant during switching. When the output voltage is rising, the short-circuit power is dissipated through the n-device. On the falling edge, the power is dissipated through the p-device.

The behavior predicted by (3.6) is also confirmed by the power measurements taken on the 200-nm G4 microprocessor [3.11]. Figure 3.4 shows that the power increases linearly with frequency. The curves, when extrapolated, do indicate some small residual power

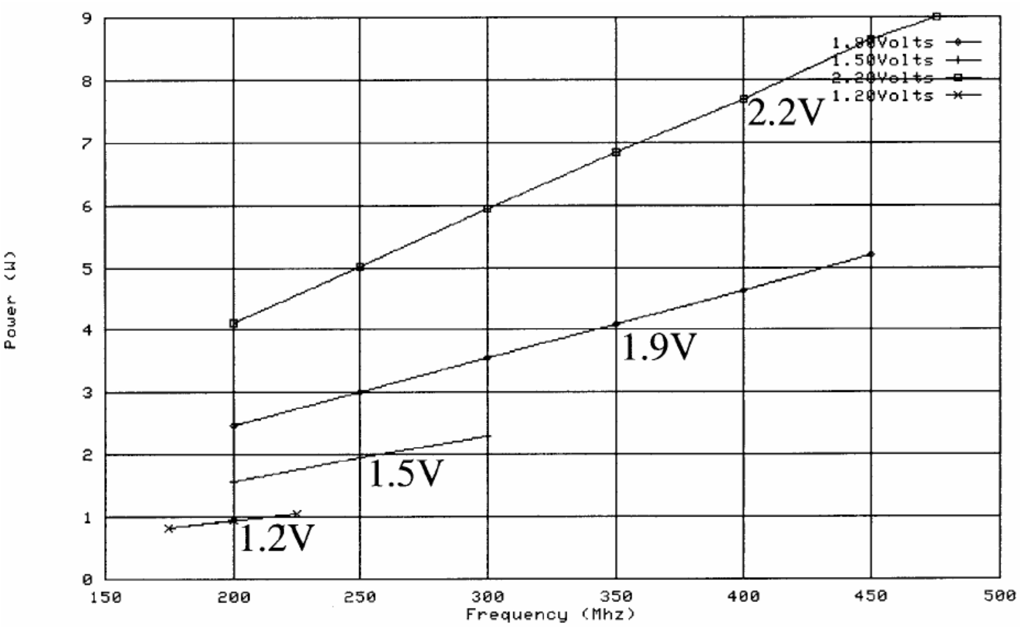


Figure 3.4: Power versus frequency for the G4 microprocessor.

dissipation when the frequency is zero. This residual power can be attributed in part to the biasing circuitry used for the input-output buffers of the chip. According to [3.12], each buffer can dissipate 429 μW.

The last term of (3.1) is the leakage power. Several mechanisms contribute to this term, including reverse-biased diode junction leakage, gate oxide leakage, and subthreshold leakage [3.13]. It is assumed here that subthreshold leakage dominates. The subthreshold leakage current for an n-device is given by [3.14]:

$$I_{leak} = \kappa(1 - e^{-\beta V_{DS}})e^{\frac{\beta}{\eta}(V_{GS} - V_T)} \tag{3.7}$$

where κ is a technology and design dependent constant, β is the inverse of the thermal voltage, η is the subthreshold swing coefficient, and V_T is the threshold voltage. For a device leaking in its cutoff region of operation, V_{GS} is close to zero and V_{DS} is close to the supply voltage. The supply voltage itself is always much larger than the thermal voltage. Thus,

$$I_{leak} \approx \kappa\, e^{-\frac{\beta}{\eta}V_T} \tag{3.8}$$

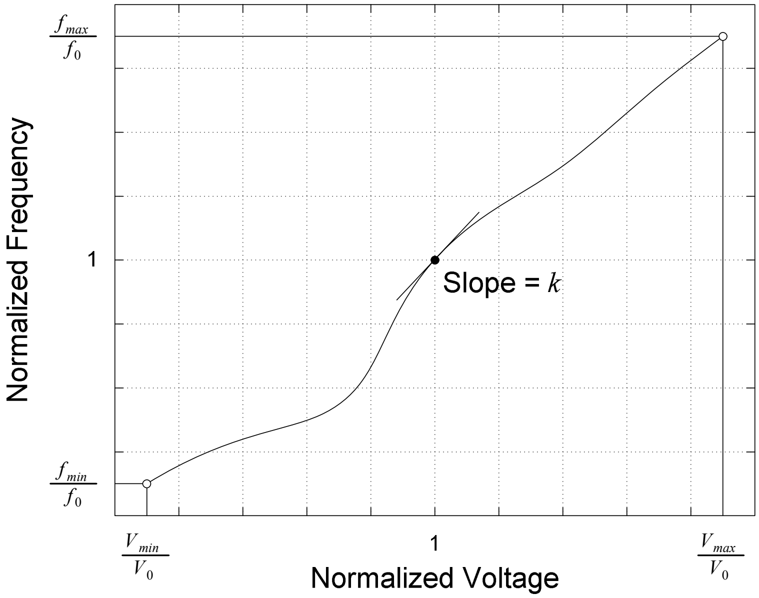


Figure 3.5: Normalized frequency-versus-voltage curve.

As the supply voltage increases, the drain-induced barrier lowering (DIBL) effect linearly lowers V_T. Thus, it introduces an exponential relationship between I_{leak} and V. However, it is assumed here that the DIBL effect is small and that I_{leak} is independent of the supply voltage. Consequently, I_{leak} is treated as a constant in (3.4).

3.2.2 Frequency

To analyze how the frequency of operation is related to the supply voltage, it is useful to consider the logic path limiting the frequency. Let its delay be D. Its sequencing overhead S can be written as the sum of the latency L of its sequential elements and the clock inaccuracy y:

$$S = L + y \tag{3.9}$$

For flip-flops, L is the setup time plus the data-to-output delay. It is worth noting that certain skew-tolerant design techniques can partially hide the clock inaccuracy [3.15]. These techniques are very important. They are treated here as a way of making the effective clock inaccuracy smaller. The clock inaccuracy is the sum of the skew and the jitter.

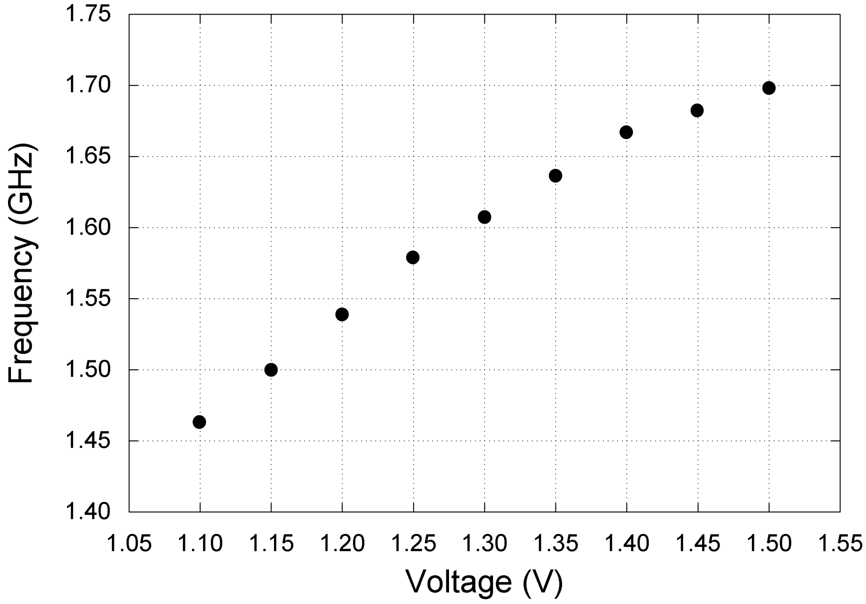


Figure 3.6: Frequency-versus-voltage curve for the 130-nm Itanium microprocessor.

The supply voltage affects D and S in a complicated way. But based on the empirical data shown in Figure 3.2, on the ring oscillator data of Table 3.1, and on the short-channel transistor model proposed in [3.16], it is reasonable to assume a simple linear relationship between f and V when V is near some nominal value V_0:

$$f = \frac{1}{D+S}\left(1 + k\frac{V - V_0}{V_0}\right) \tag{3.10}$$

In (3.10), k is a unit-less frequency-scaling factor that measures the ease of increasing the clock frequency by increasing the supply voltage. The value of k is defined as the slope of the general frequency-versus-voltage curve of Figure 3.5 at $x = y = 1$. The x-axis represents the supply voltage normalized to its nominal value whereas the y-axis gives the normalized frequency of the system. With this definition, the relative frequency increase $\Delta f / f_0$ corresponding to a relative voltage increase $\Delta V / V_0$ is $k \times (\Delta V / V_0)$.

It is important to note that in the frequency scaling model defined by (3.10), D and S are independent of V. When the supply voltage and the sequencing overhead are nominal, so is the frequency.

The validity of (3.10) is further confirmed by the frequency-versus-voltage relationship measured for the 130-nm Itanium microprocessor [3.17] and shown in Figure 3.6. The nominal supply voltage is 1.3 V. At that point, the value of the frequency-scaling factor *k* is 0.49. For the SPARC V9 microprocessor of Figure 3.2, the frequency-scaling factor at 1.6 V is 0.80.

3.3 Benefit of Reducing the Sequencing Overhead

Increasing the supply voltage of a chip and reducing its sequencing overhead are two equivalent ways of improving its frequency. In each case, the power dissipation increases. But generally, the power increase associated with a supply voltage increase is greater than the power increase associated with an ideal sequencing overhead reduction. Because of this, the ideal sequencing overhead reduction is equivalent to a power savings at the new frequency.

To quantify this benefit, it is necessary to compare the cost of increasing frequency by increasing the supply voltage to the cost of increasing frequency by reducing the sequencing overhead. The two are generally not equal. For the supply voltage increase ΔV and the sequencing overhead reduction ΔS to produce the same given frequency improvement Δf, the following equality must hold:

$$f(V+\Delta V,S)=f(V,S-\Delta S)=f(V,S)+\Delta f \tag{3.11}$$

Beyond this point, $f(V, S)$ will be denoted f for conciseness. From (3.10),

$$f(V+\Delta V,S)=\frac{1}{D+S}\left(1+k\frac{V+\Delta V-V_0}{V_0}\right) \tag{3.12}$$

and

$$f(V,S-\Delta S)=\frac{1}{D+S-\Delta S}\left(1+k\frac{V-V_0}{V_0}\right) \tag{3.13}$$

Using (3.12) and (3.13) to solve (3.11) for ΔV yields:

$$\Delta V = \frac{V - V_0 + V_0 / k}{D + S - \Delta S} \Delta S \tag{3.14}$$

Equivalently,

$$\Delta S = \frac{D + S}{V + \Delta V - V_0 + V_0 / k} \Delta V \tag{3.15}$$

Also,

$$\Delta f = \frac{k}{D + S} \frac{\Delta V}{V_0} = f \frac{\Delta S}{D + S - \Delta S} \tag{3.16}$$

If the supply voltage is increased, the power dissipated at the new frequency ($f + \Delta f$) is $P(V + \Delta V, S)$:

$$P(V + \Delta V, S) = (C_d + C_s)(V + \Delta V)^2 (f + \Delta f) + I_{leak} (V + \Delta V) \tag{3.17}$$

After simplifications,

$$\begin{aligned} P(V + \Delta V, S) = P(V, S) + (C_d + C_s)V^2 \Delta f + 2(C_d + C_s)Vf\Delta V + I_{leak}\Delta V \\ + 2(C_d + C_s)V\Delta V\Delta f + (C_d + C_s)\Delta V^2 f + (C_d + C_s)\Delta V^2 \Delta f \end{aligned} \tag{3.18}$$

But if the sequencing overhead is decreased, the power dissipated at the new frequency is $P(V, S - \Delta S)$:

$$P(V, S - \Delta S) = (C_d + C_s)V^2 (f + \Delta f) + I_{leak} V \tag{3.19}$$

In (3.19), the sequencing overhead reduction ΔS is assumed *ideal*. This means that the design change reducing the sequencing overhead costs *no* power besides the one associated with the frequency increase that ΔS creates. Equation (3.19) simplifies to:

$$P(V, S - \Delta S) = P(V, S) + (C_d + C_s)V^2 \Delta f \tag{3.20}$$

In (3.18) and (3.20), the power dissipation increases because the new frequency is higher. However, the voltage is not increased when the sequencing overhead is reduced. Therefore, $P(V + \Delta V, S)$ is greater than $P(V, S - \Delta S)$ and the sequencing overhead improvement is equivalent to a savings in power. This can be shown mathematically. The difference is:

$$\Delta P = P(V + \Delta V, S) - P(V, S - \Delta S) \tag{3.21}$$

After simplification, ΔP becomes:

$$\Delta P = 2(C_d + C_s)Vf\Delta V + I_{leak}\Delta V + 2(C_d + C_s)V\Delta V\Delta f + (C_d + C_s)\Delta V^2 f + (C_d + C_s)\Delta V^2\Delta f \tag{3.22}$$

From (3.14) and (3.16), ΔV and Δf can both be considered a function of ΔS. Therefore, ΔP can be explicitly written as a function of ΔS:

$$\begin{aligned}\Delta P = {} & (2C_d Vf + 2C_s Vf + I_{leak})\frac{V_0}{k} f(V, S - \Delta S)\Delta S \\ & + \frac{C_d + C_s}{D + S}\left(1 + k\frac{3V - V_0}{V_0}\right)\left(\frac{V_0}{k}\right)^2 f^2(V, S - \Delta S)\Delta S^2 \\ & + \frac{C_d + C_s}{D + S}\left(\frac{V_0}{k}\right)^2 f^3(V, S - \Delta S)\Delta S^3\end{aligned} \tag{3.23}$$

Since ΔS cannot be less than the initial sequencing overhead S, $f(V, S - \Delta S)$ is always positive. ΔP is guaranteed to be positive because by definition $\Delta \mathrm{S} > 0$. Equation (3.23) quantifies the power savings associated with an *ideal* sequencing overhead reduction.

The SPARC V9 microprocessor described in [3.9] is useful to illustrate the concept. Given its technology, the leakage power is assumed negligible. A nominal dynamic and short-circuit power dissipation of 80.0 W is reported when the supply voltage is 1.60 V and the frequency is 1105 MHz. The measurements of Figure 3.2 indicate that increasing the supply voltage to 1.70 V increases the frequency to 1135 MHz. This is equivalent to decreasing the clock cycle by 24 ps. This increase in voltage increases the power consumption to:

$$(80 \text{ W}) \times (1.70 \text{ V} / 1.60 \text{ V})^2 \times (1135 \text{ MHz} / 1105 \text{ MHz}) = 92.8 \text{ W}$$

That represents a power increase of 92.8 W – 80.0 W = 12.8 W. If the sequencing overhead for this chip was somehow reduced by 24 ps, the frequency would still be 1135 MHz. But then, the power dissipation would only be:

$$(80.0 \text{ W}) \times (1135 \text{ MHz} / 1105 \text{ MHz}) = 82.2 \text{ W}$$

Reducing the sequencing overhead by 24 ps is thus equivalent to a power savings of 92.8 – 82.2 = 10.6 W. This is the benefit of reducing the sequencing overhead.

3.4 Optimal Energy Allocation

The problem of optimally allocating energy for sequencing is conceptually a three-step problem.

1. Define the performance target for the entire system.
2. Determine how much energy should be allocated for sequencing, given the overall system context.
3. Allocate this energy between the various sequencing subsystems to achieve the lowest timing overhead (i.e. the highest frequency).

The first step is intrinsically application-specific. For a microprocessor, it defines the relative importance of minimizing the runtime R versus minimizing the overall system energy E required for the task. There is no universally optimal tradeoff for this. The choice is driven by the usage model of the system.

The second step of defining how much energy should be allocated for sequencing is *not* arbitrary. It has to be done such that the system performance target is satisfied at the lowest power possible.

The last step, where the energy allocation takes place, is not arbitrary either. If it makes the system exceed its performance target, the unwanted additional performance can be translated into a power savings by adjusting the supply voltage.

3.4.1 Defining the Optimization Problem

The simplest heuristic for resolving the application-specific power-performance tradeoff is to minimize the energy-delay product ER, as in [3.1]. However, it is assumed here that the preference is resolved in a more general way by an arbitrary weight w ($w \geq 0$) and that the goal of the optimization problem is to minimize ER^w. When $w = 0$, the problem becomes a pure energy minimization problem where the runtime is irrelevant. When $w \rightarrow \infty$, the problem becomes a runtime minimization problem where the energy required for the computation is ignored.

For a microprocessor operating at frequency f and processing an average of ζ instructions per cycle (IPC), the runtime R of a program having n instructions is:

$$R = \frac{n}{f\zeta} \tag{3.24}$$

The total number of instructions generally includes some no-operation instructions. These instructions produce dead clock cycles where no work is done. They are needed whenever the microprocessor must wait for the memory to return something, typically when a branch is not predicted correctly or when a new block of data must be fetched. The number of clock cycles spent waiting is a function of the frequency of the microprocessor and of the latency of the memory. Although n is generally a function of f, n is treated as a constant here. The number of instructions per cycle ζ is also considered a constant.

The energy E required to execute the program is:

$$E = PR \tag{3.25}$$

where P is its average power dissipation during execution, as defined earlier.

From (3.24) and (3.25), ER^w can be expressed as:

$$ER^w = PR^{w+1} = P\left(\frac{n}{f\zeta}\right)^{w+1} \tag{3.26}$$

Therefore, minimizing the weighted energy-delay product ER^w is equivalent to minimizing P/f^{w+1} because n and ζ are fixed.

The minimization problem has two independent optimization variables: the supply voltage V and the implementation Ψ of the sequencing circuitry. Ψ affects the total power dissipation P because it defines C_d, C_s, and I_{leak}:

$$P_\Psi = C_d(\Psi)V^2 f + C_s(\Psi)V^2 f + I_{leak}(\Psi)V \tag{3.27}$$

The implementation also affects the sequencing overhead and, thus, the frequency:

$$f = \frac{1}{D + S_\Psi}\left(1 + k\frac{V - V_0}{V_0}\right) \tag{3.28}$$

The notation P_Ψ and S_Ψ is used to explicitly indicate that the power dissipation and the sequencing overhead are both a function of Ψ.

In term of the optimization variables, the ratio to minimize (P / f^{w+1}) can finally be expressed as:

$$\frac{P}{f^{w+1}} = \frac{C_d(\Psi) + C_s(\Psi)}{f^w}V^2 + \frac{I_{leak}(\Psi)V}{f^{w+1}} \tag{3.29}$$

3.4.2 Solving the Optimization Problem

Figure 3.7 gives an overview of the strategy used in this section for solving the sequencing energy allocation problem. This strategy was first outlined in [3.7]. The concept of hardware intensity (introduced later in [3.8]) is based on a similar idea. The solution begins with an initial design. The frequency of the design can be improved by decreasing the sequencing overhead (i.e. by changing the implementation) or by increasing the supply voltage. The best choice is the one resulting in the lower power. If the higher frequency is not desired, the supply voltage can be lowered to return to the initial performance while saving power.

The proposed optimal energy allocation solution is based on the marginal benefit (measured in mW / ps) associated with one unit of *ideal* sequencing overhead reduction ΔS. Again, a sequencing overhead reduction is said to be ideal if it costs no power besides the power associated with the frequency increase that it creates. The marginal benefit B of that ideal sequencing overhead reduction is defined as the normalized savings in power resulting from it:

$$B_\Psi \equiv \lim_{\Delta S \to 0} \frac{P_\Psi(V + \Delta V, S_\Psi) - P_\Psi(V, S_\Psi - \Delta S)}{\Delta S} \tag{3.30}$$

The implementation Ψ is fixed. The first term of the numerator represents the power dissipation for that implementation when the voltage is increased by ΔV. The second term represents the power dissipation after the ideal sequencing overhead reduction with no

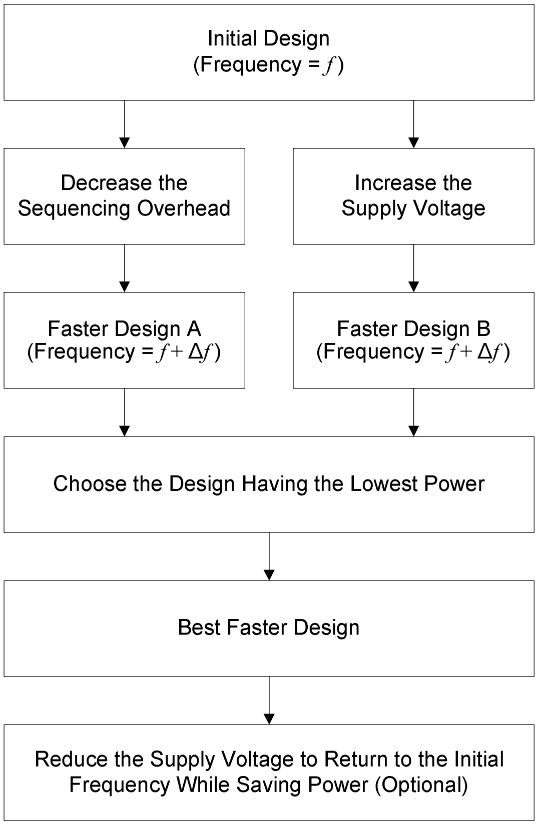

Initial Design
(Frequency = f)
Decrease the Sequencing Overhead
Increase the Supply Voltage
Faster Design A
(Frequency = $f + \Delta f$)
Faster Design B
(Frequency = $f + \Delta f$)
Choose the Design Having the Lowest Power
Best Faster Design
Reduce the Supply Voltage to Return to the Initial Frequency While Saving Power (Optional)

Figure 3.7: Optimization methodology for sequencing energy allocation.

voltage increase. ΔV and ΔS must be such that they produce exactly the same frequency improvement. In other words, they must obey (3.14). The subscript Ψ is added to B to explicitly indicate that the benefit of an ideal sequencing overhead reduction is a function of the implementation.

The expression for the numerator is given by ΔP in (3.23). Substituting into (3.30) yields:

$$B_\Psi = \frac{1}{k}(2C_d Vf + 2C_s Vf + I_{leak})V_0 f \tag{3.31}$$

Given that ΔP is always positive, B_Ψ is always positive.

When the leakage power is small and V is around the nominal supply voltage V_0,

$$B_\Psi \approx \frac{2P_0 f_0}{k} \tag{3.32}$$

where P_0 and f_0 are the nominal power and frequency corresponding to that voltage.

A design change $\Delta\Psi$ aimed at reducing the sequencing overhead (and thus increasing the frequency) should be implemented only if its marginal cost is below B_Ψ. The cost C of the design change $\Delta\Psi$ applied to the given implementation Ψ is denoted by $C_{\Delta\Psi|\Psi}$. The expression for $C_{\Delta\Psi|\Psi}$ is:

$$C_{\Delta\Psi|\Psi} \equiv \lim_{\Delta S \to 0} \frac{P_{\Psi+\Delta\Psi}(V, S_{\Psi+\Delta\Psi}) - P_\Psi(V, S_\Psi - \Delta S)}{\Delta S} \tag{3.33}$$

under the condition that $S_{\Psi+\Delta\Psi} = S_\Psi - \Delta S$.

This condition ensures that the design change $\Delta\Psi$ produces exactly the same frequency improvement as the ideal sequencing overhead reduction ΔS does for implementation Ψ. Therefore,

$$C_{\Delta\Psi|\Psi} \equiv \lim_{\Delta S \to 0} \frac{P_{\Psi+\Delta\Psi}(V, S_\Psi - \Delta S) - P_\Psi(V, S_\Psi - \Delta S)}{\Delta S} \tag{3.34}$$

$C_{\Delta\Psi|\Psi}$ includes the change in power due to the higher frequency produced by the design change. It also includes the change in power due to the design change itself (which would have been zero for an *ideal* design change).

If its marginal cost is above B_Ψ, the change requires too much power. It is cheaper to increase the supply voltage to achieve the same frequency improvement.

Furthermore, a design change $\Delta\Psi$ should always be implemented if its marginal cost $C_{\Delta\Psi|\Psi}$ is below B_Ψ, even if the higher frequency is not desired. Although implementing the change costs some power, it increases frequency. But because its marginal cost is below B_Ψ, even more power can be recovered by reducing the supply voltage to return to the initial frequency.

The argument for a design change aimed at reducing power is analogous. Its cost in frequency should not exceed its benefit. If it does not, it should be implemented. For a given system, the sequencing energy allocation is optimal if and only if the marginal cost of any design change $\Delta\Psi$ aimed at reducing the sequencing overhead or the power dissipation matches the benefit:

$$B_{\Psi} = C_{\Delta\Psi|\Psi} \tag{3.35}$$

From (3.32), it is clear that the frequency-scaling factor k is one of the key parameters defining the optimal energy allocation point.

Theorem 2.1: Any design change $\Delta\Psi$ is useful at reducing power if $C_{\Delta\Psi|\Psi} < B_{\Psi}$.

The change is useful at reducing power if, at all supply voltages, the following statement is true:

- If ΔV is such that $f(V, S_{\Psi+\Delta\Psi}) = f(V + \Delta V, S_{\Psi})$ then $P_{\Psi+\Delta\Psi}(V, S_{\Psi+\Delta\Psi})$ is less than $P_{\Psi}(V + \Delta V, S_{\Psi})$.

In other words, if a design with the change and a design without the change both operate at same the frequency (but at different voltages), the one with the change will always consume less power.

Proof: By definition, $C_{\Delta\Psi|\Psi} < B_{\Psi}$ implies:

$$\lim_{\Delta S \to 0} \frac{P_{\Psi+\Delta\Psi}(V, S_{\Psi+\Delta\Psi}) - P_{\Psi}(V, S_{\Psi} - \Delta S)}{\Delta S} < \lim_{\Delta S \to 0} \frac{P_{\Psi}(V + \Delta V, S_{\Psi}) - P_{\Psi}(V, S_{\Psi} - \Delta S)}{\Delta S}$$

where $S_{\Psi+\Delta\Psi} = S_{\Psi} - \Delta S$ and where ΔV is such that $f(V, S_{\Psi+\Delta\Psi}) = f(V + \Delta V, S_{\Psi})$. Since both numerators have one term in common,

$$\lim_{\Delta S \to 0} \frac{P_{\Psi+\Delta\Psi}(V, S_{\Psi+\Delta\Psi})}{\Delta S} < \lim_{\Delta S \to 0} \frac{P_{\Psi}(V + \Delta V, S_{\Psi})}{\Delta S}$$

Therefore, $P_{\Psi+\Delta\Psi}(V, S_{\Psi+\Delta\Psi}) < P_{\Psi}(V + \Delta V, S_{\Psi})$.

Lemma 2.1: If $C_{\Delta\Psi|\Psi} > B_{\Psi}$, then the design change increases power at all supply voltages. In other words, if $C_{\Delta\Psi|\Psi} > B_{\Psi}$ then the following statement is true:

- If ΔV is such that $f(V, S_{\Psi+\Delta\Psi}) = f(V + \Delta V, S_{\Psi})$ then $P_{\Psi+\Delta\Psi}(V, S_{\Psi+\Delta\Psi})$ is more than $P_{\Psi}(V + \Delta V, S_{\Psi})$.

Proof: Analogous to the proof of Theorem 2.1.

Theorem 2.2: Any design change $\Delta\Psi$ is useful at increasing frequency if $C_{\Delta\Psi|\Psi} < B_{\Psi}$.

The change is useful at increasing frequency if, at all supply voltages, the following statement is true.

- If ΔV is such that $P_{\Psi+\Delta\Psi}(V, S_{\Psi+\Delta\Psi}) = P_{\Psi}(V + \Delta V, S_{\Psi})$ then $f(V, S_{\Psi+\Delta\Psi})$ is more than $f(V + \Delta V, S_{\Psi})$.

In other words, if a design with the change and a design without the change both consume the same amount of power (but at different voltages), the one with the change will always be faster.

Proof: From Theorem 2.1, if Δv_1 is such that $f(v, S_{\Psi+\Delta\Psi}) = f(v + \Delta v_1, S_{\Psi})$ then:

$$P_{\Psi+\Delta\Psi}(v, S_{\Psi+\Delta\Psi}) < P_{\Psi}(v + \Delta v_1, S_{\Psi})$$

Let Δv_2 be the voltage that makes $P_{\Psi+\Delta\Psi}(v + \Delta v_2, S_{\Psi+\Delta\Psi})$ equal to $P_{\Psi}(v + \Delta v_1, S_{\Psi})$. From the fact that P increases monotonically with the supply voltage, Δv_2 is known to exist and to be greater than zero. Because f is also monotonic with respect to the supply voltage, $f(v + \Delta v_2, S_{\Psi+\Delta\Psi}) > f(v, S_{\Psi+\Delta\Psi})$. From the assumption that $f(v, S_{\Psi+\Delta\Psi}) = f(v + \Delta v_1, S_{\Psi})$, it follows that:

$$f(v + \Delta v_2, S_{\Psi+\Delta\Psi}) > f(v + \Delta v_1, S_{\Psi})$$

Consequently,

$$P_{\Psi+\Delta\Psi}(v + \Delta v_2, S_{\Psi+\Delta\Psi}) = P_{\Psi}(v + \Delta v_1, S_{\Psi}) \Rightarrow f(v + \Delta v_2, S_{\Psi+\Delta\Psi}) > f(v + \Delta v_1, S_{\Psi})$$

Let $V = v + \Delta v_2$ and $\Delta V = \Delta v_1 - \Delta v_2$. Then, $v + \Delta v_1 = V + \Delta V$.

$$P_{\Psi+\Delta\Psi}(V, S_{\Psi+\Delta\Psi}) = P_{\Psi}(V + \Delta V, S_{\Psi}) \Rightarrow f(V, S_{\Psi+\Delta\Psi}) > f(V + \Delta V, S_{\Psi})$$

This completes the proof.

Lemma 2.2: If $C_{\Delta\Psi|\Psi} > B_{\Psi,}$ then the design change reduces frequency at all supply voltages. In other words, if $C_{\Delta\Psi|\Psi} > B_{\Psi}$ then the following statement is true:

- If ΔV is such that $P_{\Psi+\Delta\Psi}(V, S_{\Psi+\Delta\Psi}) = P_{\Psi}(V + \Delta V, S_{\Psi})$ then $f(V, S_{\Psi+\Delta\Psi})$ is less than $f(V + \Delta V, S_{\Psi})$.

Proof: Analogous to the proof of Theorem 2.2.

Theorem 2.3: The weighted energy-delay product ER^w is locally minimum when $C_{\Delta\Psi|\Psi} = B_\Psi$ for any possible design changes $\Delta\Psi$. Again, minimizing ER^w is equivalent to minimizing P / f^{w+1}:

$$\frac{P}{f^{w+1}} = \frac{C_d(\Psi) + C_s(\Psi)}{f^w} V^2 + \frac{I_{leak}(\Psi) V}{f^{w+1}}$$

Proof: If $C_{\Delta\Psi|\Psi} < B_\Psi$, then, from Theorem 2.1, P can be decreased without changing f by implementing $\Delta\Psi$ and adjusting the supply voltage V. Equivalently, from Theorem 2.2, f can be increased without changing P by implementing $\Delta\Psi$. Therefore, if $C_{\Delta\Psi|\Psi} < B_\Psi$ then P / f^{w+1} can be reduced. In other words, if $C_{\Delta\Psi|\Psi} < B_\Psi$ then the weighted energy-delay product is non-minimum. From Lemma 2.1, if $C_{\Delta\Psi|\Psi} > B_\Psi$ then the design change $\Delta\Psi$ increases power at all voltages if the frequency is kept the same. Equivalently, from Lemma 2.2, the change reduces frequency if the power is maintained constant. Consequently, if $C_{\Delta\Psi|\Psi} > B_\Psi$ then $\Delta\Psi$ makes P / f^{w+1} increase. The ratio P / f^{w+1} (and the weighted energy-delay product) is therefore locally minimum when $C_{\Delta\Psi|\Psi} = B_\Psi$.

For the SPARC V9 microprocessor considered earlier, allocating more power for clock distribution could hypothetically reduce the sequencing overhead by 24 ps and increase the frequency. This extra power could be used to drive more global clock grid wires for instance. If the additional power does not exceed 10.6 W, then it is cheaper to reduce the sequencing overhead (as opposed to increasing the voltage) to get the frequency increase. The design change should be implemented. If the cost of reducing the sequencing overhead exceeds 10.6 W, increasing the supply voltage costs less and achieves the same result.

3.5 Summary

The analysis of the tradeoff between power and performance is based on two generic power and frequency models.

The first model describes how the dynamic power, the short-circuit power, and the subthreshold leakage power of a synchronous digital system vary with the supply voltage.

Its validity is confirmed by ring oscillator simulations and by measurements taken on a 200-nm G4 microprocessor [3.11].

The frequency scaling model only captures the first-order supply voltage dependence, but can still describe accurately the measurements taken on a 150-nm SPARC V9 microprocessor [3.9] and on a 130-nm Itanium microprocessor [3.17].

The notions of the power-performance cost and benefit are then formally defined. It is mathematically shown that the tradeoff between power and performance is locally optimal if and only if the cost of any design change equals its benefit.

CHAPTER 4

Local Clocking

In high-frequency microprocessors, most of the clock distribution power is due to local clocks [4.1]. So is a significant fraction of the clock inaccuracy [4.2]. Yet, in the literature, the issues related to local clocking are rarely discussed in a quantitative manner [4.3]. This chapter proposes several new models to better understand those issues.

First, this chapter discusses the sequencing overhead associated with flip-flops, which are arguably the most common type of sequential element. A generalized optimal setup time that takes into account clock inaccuracy is defined. The data-dependant clock jitter resulting from the switching activity of sequentials with naked clocks is quantified. Next, local clock buffers (LCBs) are discussed. A model is derived to analyze the properties of the topology used on the 90-nm microprocessor of [4.4] and commonly used elsewhere. The model is applied to better understand how the delay tracks the supply voltage. It shows that the impact of device sizing on tracking is relatively small and that sizing the devices for equal rise and fall delays is not mandatory. The model is also used to analyze the gain of the LCBs. Finally, a non-linear model is proposed for the bandwidth of a clock buffer driving a local interconnect. The non-linearity is required to adequately capture the behavior of the devices. The model is applied to analyze crosstalk jitter and to examine the conditions where clock shielding is advantageous from a power standpoint.

4.1 Overview

Clock distribution networks always include several buffering stages. The first few stages are responsible for global clock distribution. The goal of the global clock distribution

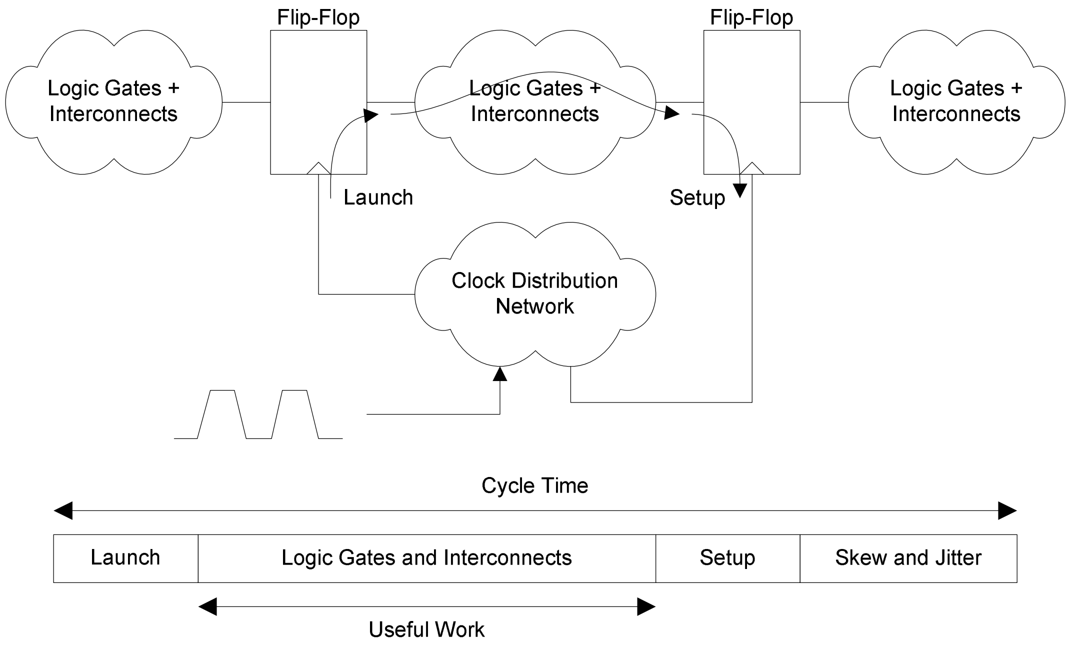

Flip-Flop
Flip-Flop
Logic Gates + Interconnects
Logic Gates + Interconnects
Logic Gates + Interconnects
Launch
Setup
Clock Distribution Network
Cycle Time
Launch
Logic Gates and Interconnects
Setup
Skew and Jitter
Useful Work

Figure 4.1: Sequencing overhead.

network is to deliver a low skew and low jitter clock to a large number of local clock buffers. The distinction between the global and local portions of the clock distribution network is somewhat arbitrary. The notion of regional clock distribution is sometimes introduced to refer to the intermediate buffering stages located between the global and local clock buffers. Each local clock buffer usually feeds several sequential elements and defines a clock domain. The local clock buffers can typically be disabled to save clock power. The clock control logic decides when certain clock domains are turned on or off. To prevent deadlocks, the clock control logic usually requires some free-running clocks.

4.2 Sequential Elements

4.2.1 Sequencing Overhead

Flip-flops are arguably the most common type of sequential element in high-frequency microprocessors. They are widely used for logic synthesis and in data path designs. The sequencing overhead of a flip-flop includes two components, as shown in Figure 4.1: the delay required for launching the data when the clock switches (i.e. the clock-to-output delay) and the setup time required to properly sample the data. The clock-to-output delay L_0 associated with launching a zero is generally different from the delay L_1 required to launch a one. Similarly, the setup time S_0 required to correctly sample a zero is usually different from the setup time S_1 required to sample a one. In a typical structured or random-logic block, the polarity of the critical paths tends to be evenly distributed (between positive and negative) and often changes as the design evolves. For a positive path, the flip-flop delay overhead is:

$$D_{pos} = \max(L_0 + S_0, L_1 + S_1) \tag{4.1}$$

For a negative path, the overhead is:

$$D_{neg} = \max(L_0 + S_1, L_1 + S_0) \tag{4.2}$$

For a path whose polarity cannot be determined *a priori* (e.g. a path through an xor gate or through the select input of a multiplexer), the overhead is $D = \max(D_{pos}, D_{neg})$. Substituting the expressions for D_{pos} and D_{neg} yields:

$$D = \max(L_0 + S_0, L_1 + S_1, L_0 + S_1, L_1 + S_0) \tag{4.3}$$

Mathematically, this simplifies to:

$$D = \max(L_0, L_1) + \max(S_0, S_1) \tag{4.4}$$

In other words, the sequencing overhead associated with a general-purpose flip-flop is the worst-case launch time plus the worst-case setup time.

At any given setup time, the clock-to-output delay that matters from a sequencing overhead perspective is always the largest. Thus, at the setup time producing the smallest flip-flop delay, any power dissipated to make one of the launch times faster than the other is wasted because it does not improve the sequencing overhead. It is therefore advantageous to balance the two launch times. In practice, the launch times cannot be balanced for every setup time, even when the output load, the clock transition time, and data transition time are fixed. The importance of balancing is highest around the setup time producing the optimal flip-flop delay, under typical conditions.

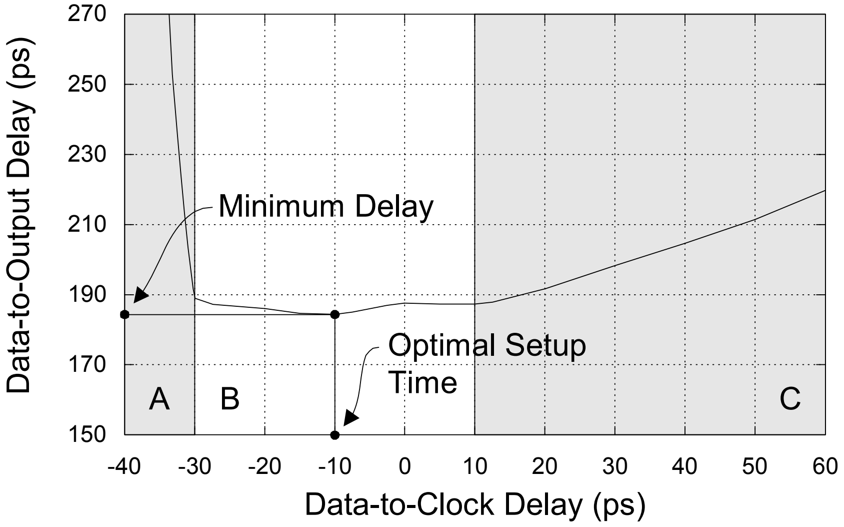


Figure 4.2: Data-to-output delay characteristic.

4.2.2 Generalized Optimal Setup Time

It is observed in [4.5] that some flip-flops exhibit a desirable property: under certain circumstances, the clock inaccuracy has very little effect on their delay.

Figure 4.2 shows the data-to-output delay of such a flip-flop, the hybrid latch-flip-flop (HLFF) used in [4.6]. The horizontal axis represents the time difference between the data and the clock arrival. A negative value means the data arrives after the clock (the HLFF has a negative setup time). Figure 4.2 has three distinct regions. If the data arrives later than 30 ps after the clock, the HLFF fails. This corresponds to region A. If the data arrives much earlier than the clock, it is blocked. This time spent waiting for the clock effectively makes the data-to-output delay larger. This behavior occurs in region C. It is worth noting that in region B, the data-to-output characteristic is fairly flat. When the data arrives from 10 ps to –30 ps before the clock, the data-to-output delay is almost constant. This flatness has two interpretations. The first is data-centric. It assumes that the delay of the pipeline stage producing the data received by the HLFF exceeds one clock cycle. The flatness of region B allows the extra delay to be passed to the next pipeline stage. In the design community, this is often called transparency or time borrowing. The second interpretation is clock-centric: in the flat region, the clock can move relatively freely without affecting the data-to-output delay. The size of the flat region defines how much clock inaccuracy can be absorbed.

When there is no clock inaccuracy, the best arrival time for the data is the one minimizing the data-to-output delay [4.7]. This arrival time is called the optimal setup time and is widely used in practice. It is referred to here as the traditional optimal setup time.

Determining the best arrival time for the data in the presence of clock inaccuracy is a more general problem. Obviously, it is dangerous to allow data to arrive too close to the failure region. There, a small amount of clock inaccuracy could make the clocked storage element unreliable. A generalized optimal setup time is proposed here for a given worst-case clock inaccuracy y. It is assumed that the launch times $L_0(S)$ and $L_1(S)$ of the clocked storage element are known as a function of the setup time S. The proposed definition for the generalized optimal setup time $\hat{S}_i$ is:

$$\hat{S}_i = \left\{ S : \left(S + \max_{t \in [-y, +y]} L_i(S+t) \right) \text{ is minimum} \right\} \tag{4.5}$$

where i is an element of $\{0, 1\}$.

Equation (4.5) defines an interval around the nominal arrival time of the clock. $\hat{S}_i$ is selected assuming that the clock can switch at any time during this interval. $\hat{S}_i$ can be interpreted as the setup time that minimizes the delay through the sequential element for all the clock arrival times possible under a given clock inaccuracy limit. Clearly, when the clock inaccuracy is zero, this generalized optimal setup time becomes equal to the traditional optimal setup time, i.e. the one minimizing: $S + L_i(S)$. The failure region of Figure 4.2 is implicitly taken into account by (4.5). This is because the delay of a clocked storage element for which the setup constraint is not satisfied is infinite.

In Figure 4.2, the generalized optimal setup time is equal to the traditional optimal time (i.e. –10 ps) when the clock inaccuracy is under 20 ps. Making the data nominally arrive in the middle of the flat region allows the clock to move without affecting the system's performance. When the worst-case clock inaccuracy increases to 30 ps, the generalized optimal setup time becomes 0 ps to avoid the failure region.

General-purpose flip-flops are designed once and instantiated a large number of times. The same design is shared by a diverse set of timing paths. Some have a well-known polarity. Others, like the paths going through an xor gate or through the select input of a multiplexer, have a polarity that is *a priori* unknown. Designing a general-purpose flip-flop with a fixed power budget requires minimizing (4.4) to achieve the lowest possible sequencing overhead.

If the data polarity is unknown, the worst-case launch time corresponding to a given setup time S is $\max[L_0(S), L_1(S)]$. In the presence of clock inaccuracy, the worst-case launch time L_{max} becomes:

$$L_{max} = \max_{t \in [-y,+y]} [L_0(S+t), L_1(S+t)] \tag{4.6}$$

The setup time $\hat{S}_{GP}$ minimizing the sequencing overhead of a general-purpose flip-flop is the one minimizing $S + L_{max}$.

Thus:

$$\hat{S}_{GP} = \left\{ S : \left(S + \max_{t \in [-y,+y]} [L_0(S+t), L_1(S+t)] \right) \text{ is minimum} \right\} \tag{4.7}$$

For the timing paths whose polarity is unknown, increasing the power consumption of the flip-flop in order to make one of the setup times better than the other without increasing the worst-case launch time provides no sequencing overhead improvement. Similarly, making one of the launch times better than the other without degrading the setup times offers no advantage. Timing only improves for *some* of the paths having a well-known polarity.

$\hat{S}_{GP}$ always minimizes the worst-case sequencing overhead of the timing paths sharing the flip-flop. Furthermore, if most of these paths have an unknown polarity, $\hat{S}_{GP}$ closely minimizes the average sequencing overhead of the set.

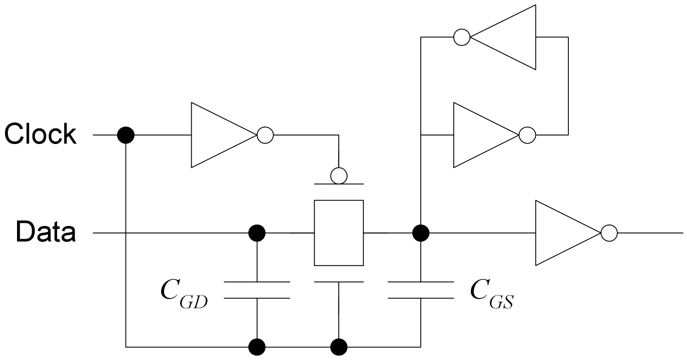


Figure 4.3: Basic sequential element.

4.2.3 Data-Dependant Jitter

The capacitance of the clock input of a sequential element is, in general, not constant. It can vary with the internal switching activity of the sequential, which itself varies with the switching activity of the data input. For some sequential topologies, the capacitance variation can be significant. If the capacitance of the clock input increases, the local clock buffer producing that clock slows down. Conversely, the clock speeds up when the capacitance decreases. As a result, the switching activity of the data input can cause clock jitter.

For the sequential element shown in Figure 4.3, the capacitance of the clock input C_{clock} includes the gate-to-drain C_{GD}, gate-to-source C_{GS}, and gate-to-channel C_{GC} capacitances of the n-device of the transmission gate. It also includes the capacitance of the clock inverter and some interconnect capacitance:

$$C_{clock} = C_{wire} + C_{inv} + C_{GC} + C_{GD} + C_{GS} \tag{4.8}$$

Because the clock input is not buffered and directly connected to the transmission gate, it is qualified of naked. When the data rises, the drain and the source of the n-device also rise. The effective capacitance of the clock input decreases approximately to:

$$C_{clock} = C_{wire} + C_{inv} + C_{GC} \tag{4.9}$$

When the data falls, the capacitance of the clock input increases to:

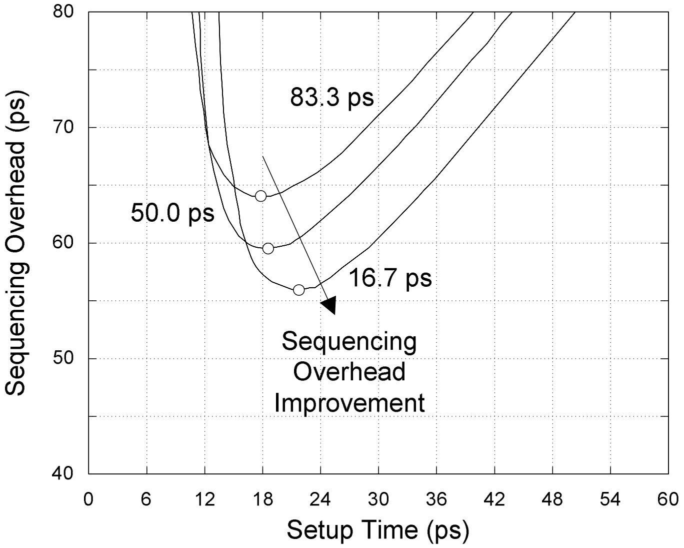


Figure 4.4: Impact of clock transition time on sequencing overhead for a flip-flop.

$$C_{clock} = C_{wire} + C_{inv} + C_{GC} + 2(C_{GD} + C_{GS}) \tag{4.10}$$

The switching activity of the data signal therefore impacts the capacitance. The variation is approximately:

$$\frac{\Delta C_{clock}}{C_{clock}} = \frac{2(C_{GD} + C_{GS})}{C_{wire} + C_{inv} + C_{GC} + C_{GD} + C_{GS}} \tag{4.11}$$

For a structure like the one shown in Figure 4.3, the transmission gate is typically larger than the clock inverter and the capacitance variation is not negligible.

4.2.4 Impact of Clock Transition Time of Sequencing Overhead

The important performance impact of the transition time of the clock is usually ignored in the clock distribution literature. Faster clocks are produced by larger local clock buffers and require more power. For flip-flops and latches, they tend to degrade the setup time, but improve the launch time. The net effect on the sequencing overhead is shown in Figure 4.4 for a 90-nm flip-flop. At each setup time, the time to launch a one is measured and the sequencing overhead is computed. The minimum of each curve gives the traditional optimal setup time. On average, the sequencing increases by 0.12 ps when the

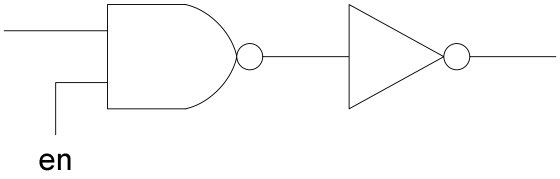


Figure 4.5: Two-stage local clock buffer.

clock transition time increases by 1.0 ps. The clock transition time is measured from 0% to 100% of the supply voltage.

4.3 Local Clock Buffers

In high-frequency microprocessors, local clock buffers are typically used for amplification and for clock gating. A local clock buffer usually feeds a relatively small number of sequential elements: about 10 to 100. For static logic, the path from the local clock buffer to the sequential elements often consists of just a local route. For domino logic, the path usually includes additional devices that produce delayed clocks. An ideal local clock buffer introduces no clock inaccuracy, consumes no power, has an infinite gain, and has an infinite bandwidth.

Several local clock buffer topologies have been used for high-frequency microprocessor clocking [4.8]-[4.10]. The simplest and most common is arguably the one shown in Figure 4.5. The nand gate is controlled by an enable signal that is used to prevent the output of the local clock buffer from switching. This topology has been recently used on a 90-nm microprocessor [4.4]. Its properties are analyzed in this section.

4.3.1 Delay Modeling

The impact of device sizing on the delay of the local clock buffer of Figure 4.5 is modeled in this section.

The delay d of a device discharging a capacitive load C is a strong function of the supply voltage V_0. It is also a strong function of the transition time t_T of the waveform applied to the input of the device. As shown in Chapter 2,

$$d = \begin{cases} \dfrac{b - V_0/2}{m} + \dfrac{C}{g_D} + \dfrac{C}{g_D} W\left(\dfrac{g_D(mt_{sat} - b + V_0) - mC}{mC} e^{\frac{g_D}{mC}(mt_{sat} - b + V_0/2) - 1} \right) - \dfrac{t_T}{2} & (t_T \text{ slow}) \\ \dfrac{t_T}{2} + \dfrac{C}{g_D} \log \dfrac{\left(m\left(t_{sat} - \dfrac{C}{g_D} \right) - b + V_0 \right) e^{-\frac{g_D}{C}(t_T - t_{sat})} + m\dfrac{C}{g_D}}{mt_T - b + \dfrac{V_0}{2}} & (t_T \text{ fast}) \end{cases} \tag{4.12}$$

where t_{sat} is the time at which the device starts conducting

$$t_{sat} = \frac{V_T - \frac{g_D}{g_G}(V_0 - V_{DS0})}{V_0} t_T \tag{4.13}$$

and

$$\begin{aligned} m &= \frac{g_G}{g_D} \frac{V_0}{t_T} \\ b &= \frac{g_G}{g_D} V_T + V_{DS0} \end{aligned} \tag{4.14}$$

The four remaining parameters (g_G, g_D, V_T, and V_{DS0}) characterize the device and are technology-dependant.

In a chain of devices, the transition time and the load are not independent variables. They are closely related because the output load of a device determines the input transition time of the next. Both the transition time and the load are controlled by the size of the devices. To simplify the analysis of the delay of the local clock buffer, it is assumed that there exists an inverter producing approximately the same output current as the nand gate.

From Figure 4.6, the capacitance C_1 driven by this hypothetical inverter is:

$$C_1 = c_{diff}(w_{cp1} + w_{cn1} + w_{ep1}) + c_{gate}(w_{p2} + w_{n2}) + C_{wire} \tag{4.15}$$

where c_{diff} and c_{gate} are the unit-width diffusion and gate capacitances and C_{wire} is the capacitance of the local interconnect between the two inverters. The diffusion capacitance between devices cn_1 and en_1 does not affect C_1.

A simple, but relatively inaccurate, way to analytically compute the equivalent pull-down strength of the inverter is to assume that w_{cn1} and w_{en1} behave like resistors in series.

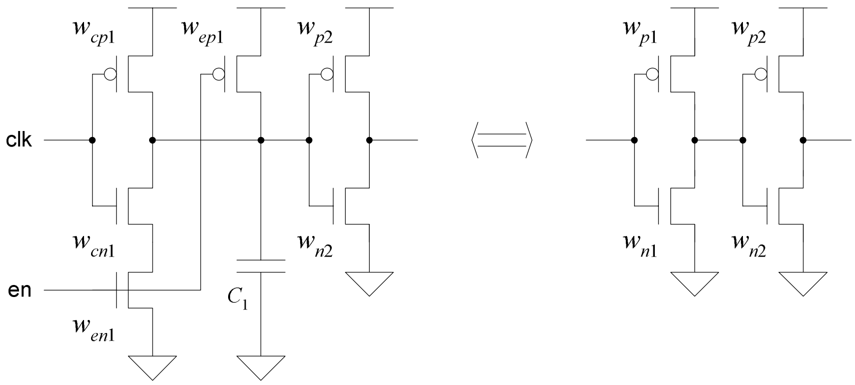


Figure 4.6: Devices of two-stage local clock buffer.

This results in:

$$\begin{aligned} w_{p1} &= w_{cp1} \\ w_{n1} &= \left(\frac{1}{w_{cn1}} + \frac{1}{w_{en1}} \right)^{-1} \end{aligned} \tag{4.16}$$

For the second inverter,

$$C_2 = c_{diff}(w_{p2} + w_{n2}) + C_{load} \tag{4.17}$$

where C_{load} the external load driven by the local clock buffer.

When the input of the clock buffer rises, the delay d_{n1} to discharge C_1 through n_1 is given by (4.12) with $C = C_1$ and:

$$\begin{aligned} g_G &= w_{n1} g_{G0} \\ g_D &= w_{n1} g_{D0} \end{aligned} \tag{4.18}$$

where g_{G0} is the unit-width gate transconductance and g_{D0} is the unit-width drain conductance. The output transition time t_{Tn1} is given in Chapter 2 by (2.39):

$$t_{Tn1} = \begin{cases} \dfrac{\frac{1}{m} V_0}{W\left(\dfrac{g_D(mt_{sat} - b + V_0) - mC_1}{mC_1} e^{\frac{g_D}{mC_1}(mt_{sat} - b + V_0/2) - 1} \right) + 1} & (t_T \text{ slow}) \\ \dfrac{C_1}{g_D} \dfrac{V_0}{\frac{g_G}{g_D} V_0 - b + \frac{1}{2} V_0} & (t_T \text{ fast}) \end{cases} \tag{4.19}$$

Table 4.1: Typical 130-nm β ratios.

Device Threshold Voltage	Supply Voltage (V)	n-Device Drain Current (mA/µm)	p-Device Drain Current (mA/µm)	β
Low	0.7	0.37	0.19	1.95
High	0.7	0.32	0.16	2.00
Low	1.4	1.30	0.66	1.97
High	1.4	1.14	0.56	2.04

The second stage charges the output load through p_2. For simplicity, it is assumed that all p-devices behave like n-devices with a lower drain current. Under this condition, the delay d_{p2} for the second stage is also given by (4.12) with $t_T = t_{Tn1}$, $C = C_2$ and:

$$\begin{aligned} g_G &= \frac{w_{p2} g_{G0}}{\beta} \\ g_D &= \frac{w_{p2} g_{D0}}{\beta} \end{aligned} \tag{4.20}$$

where β is the n-device to p-device drain current ratio.

The drain currents reported for the technology described in [4.11] are shown in Table 4.1. They indicate that β = 2.0. The value of β does not change significantly between 0.7 V and 1.4 V, validating the assumption behind (4.20).

The total local clock buffer delay for an input rising transition is therefore:

$$d_{rise} = d_{n1}(t_T, C_1) + d_{p2}(t_{Tn1}, C_2) \tag{4.21}$$

The derivation of the delay for a falling transition is analogous. The result is:

$$d_{fall} = d_{p1}(t_T, C_1) + d_{n2}(t_{Tp1}, C_2) \tag{4.22}$$

Equations (4.21) and (4.22) describe how the behavior of the local clock buffer is affected by the device sizes.

4.3.2 Gain Analysis

The gain of the local clock buffer is defined as the ratio of its output load to its input capacitance. It is an important property because it has a large impact on the load that the global clock distribution network must drive.

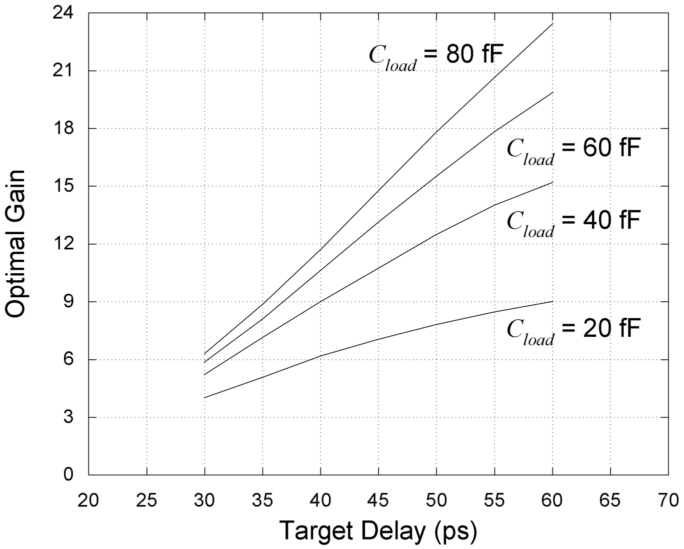


Figure 4.7: Optimal local clock buffer gain versus target delay.

The input capacitance for the nand gate of Figure 4.6 is:

$$C_{in} = c_{gate}(w_{cp1} + w_{cn1}) + C_{wire} \tag{4.23}$$

where the input interconnect capacitance of the clock input is assumed equal to the internal interconnect capacitance defined earlier. The gain G is therefore:

$$G = \frac{C_{load}}{C_{in}} \tag{4.24}$$

The gain of a buffer driving a fixed output load can only be improved by reducing its input capacitance. The minimum interconnect capacitance of the input is usually determined by layout constraints such as port placement and accessibility. This capacitance is typically small, but can never be completely eliminated. It is treated here as a constant. The gate capacitance of the nand gate can be reduced however. As it gets smaller, the nand gate gets slower and delay of the LCB increases.

This relationship can be understood quantitatively using the delay model derived in the previous section, with 130-nm technology parameters. The two stages of the local clock buffer are sized for equal rise and fall delays. In other words, the ratios w_{p1} / w_{n1} and w_{p2} / w_{n2} are both constrained to the value β. The input transition time is fixed at 60 ps.

An output load is selected. A target LCB delay is also selected. Next, the stage ratio w_{n2} / w_{n1} is adjusted to maximize the gain of the LCB. Then, the process is repeated for a new target delay until the optimal gain curve is completely traced for that load.

The results are shown in Figure 4.7 for several output loads. The gain always increases as the target delay increases. But as the devices get smaller, the fixed parasitic interconnect capacitances begin to dominate. Eventually, the gain improvement resulting from a target delay relaxation starts to diminish. This phenomenon is clearly visible when the output load is 20 fF because that load corresponds to the local clock buffer having the smallest devices.

At any given target delay, the gain improves as the output load increases. This is shown in Figure 4.8 for a 50-ps target delay. The optimization makes the unrealistic assumption that the devices have no minimum width. In reality, when the input devices reach their minimum width, the input capacitance ceases to decrease. The gain then begins to degrade even faster as the output load keeps decreasing.

Figure 4.9 shows the actual gain curves for the local clock buffers used in repeater bays on a 90-nm microprocessor. The drive strength is a measure of the size of the output driver and correlates with the output load producing a 60-ps delay. The curves for two topologies are shown. The first topology has three inverters in series and the highest gain. The second topology has also three stages, but the middle inverter is replaced by a nand gate for clock gating. Two layout styles are employed for each topology, one for the horizontal repeater bays and one for the vertical repeater bays. The shape of the actual gain curves roughly resembles the theoretical optimal gain curve. Clearly, the layout style has a significant impact on the gain. It explains the gain drop at large drive strengths, where the local clock buffers no longer fit in a single layout slot and must be split over two slots.

4.3.3 Voltage Tracking

Different local clock buffers have different delay responses to supply voltage variations. Understanding these responses is important to improve supply voltage tracking and

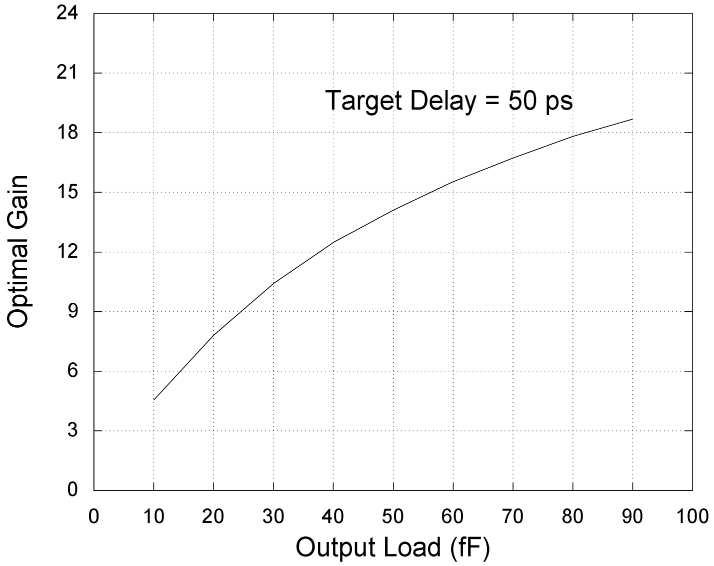


Figure 4.8: Optimal local clock buffer gain versus load.

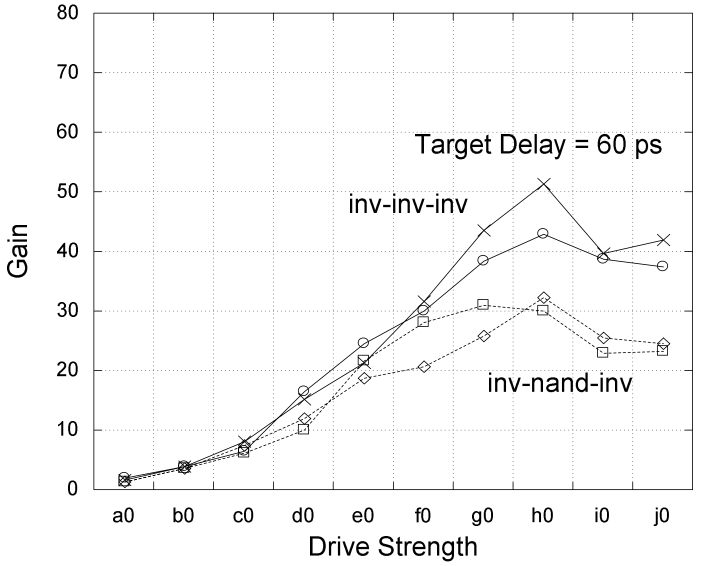


Figure 4.9: Actual local clock buffer gain versus drive strength.

minimize skew. Generally, both the rising and falling output transitions are important. If their delays are different, the local clock buffer will produce an output clock with a distorted duty cycle. This is not necessarily undesirable, in particular for domino logic where non-50% duty cycle clocks are routinely used.

The concept of supply voltage tracking requires the formal definition of frequency loss for the two sequential elements shown in Figure 4.1. For that, the events triggered by the rising edge of the clock need to be examined.

The events involve two clock paths and one data path. Collectively, they are assumed to limit the frequency of operation. The first clock path is called the generating path. It includes the delay of the local clock buffer driving the flip-flop generating the data. The data path includes the flip-flop's clock-to-output delay and the delay through the combinational logic and interconnects. The second clock path is the sampling path. It includes the delay of the other clock buffer and the delay of the devices clocked inside the receiver. When the supply voltage changes, the delays of the clock and data paths also change. The delay changes are assumed monotonic.

When the supply voltage increases, the delay of the data path improves. In turn, this reduces the cycle time. The cycle time improvement is expected to match or exceed the data delay improvement. If not, the supply voltage increase is said to produce a frequency loss. Conversely, the cycle time is expected to degrade when the supply voltage decreases. However, the degradation should not exceed the delay degradation of the data path.

The generating and sampling clock paths are said to *track* the supply voltage if changing it never results in a frequency loss.

Theorem: A supply voltage change ΔV produces no frequency loss if and only if the change has the same proportional effect on the delay d_{data} of the data path and on the delay difference ($d_{gen} - d_{sam}$) between the generating and sampling paths.

Proof: Initially, the cycle time T is given by $T = d_{gen} + d_{data} - d_{sam}$. If $\Delta V > 0$, then all paths become faster because their delay is monotonic. The delays become d'_{gen}, d'_{data}, and d'_{sam}. The new cycle time is:

$$T' = d'_{gen} + d'_{data} - d'_{sam} \tag{4.25}$$

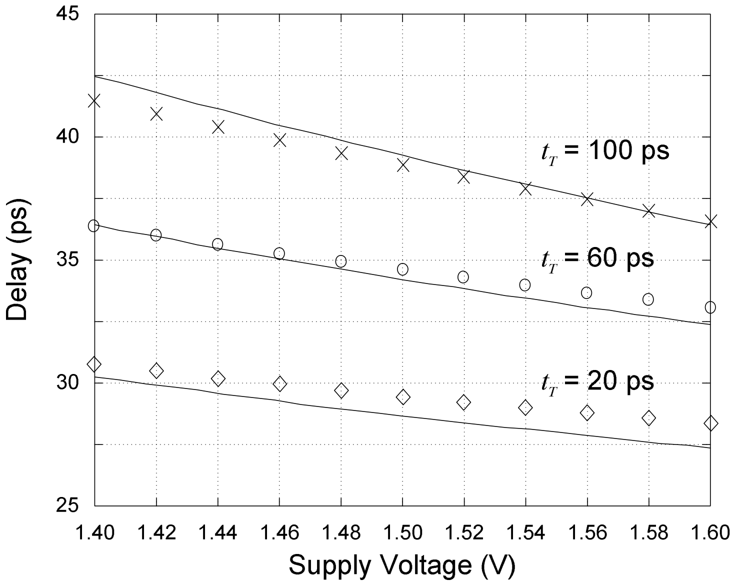


Figure 4.10: Delay versus supply voltage for various input transition times.

To avoid a frequency loss, the cycle time reduction must match or exceed the delay improvement of the data path:

$$\frac{T'}{T} \leq \frac{d'_{data}}{d_{data}} \tag{4.26}$$

Substituting the expression for T and T' and simplifying yields:

$$d'_{gen} - d'_{sam} \leq \frac{d'_{data}}{d_{data}}(d_{gen} - d_{sam}) \tag{4.27}$$

If the supply voltage drops back to initial value, the delays of the generating, data, and sampling paths will return to d_{gen}, d_{data}, and d_{sam}. With respect to the cycle time at ΔV, the degradation is T / T'. To avoid a frequency loss, the cycle time increase should not exceed the delay degradation of the data path:

$$\frac{T}{T'} \leq \frac{d_{data}}{d'_{data}} \tag{4.28}$$

Substituting (4.25) in (4.28) yields:

$$d'_{gen} - d'_{sam} \geq \frac{d'_{data}}{d_{data}}(d_{gen} - d_{sam}) \tag{4.29}$$

Equations (4.27) and (4.29) imply that to avoid a frequency loss,

$$d'_{gen} - d'_{sam} = \frac{d'_{data}}{d_{data}}(d_{gen} - d_{sam}) \tag{4.30}$$

This concludes the proof.

The supply voltage tracking condition given by (4.30) relates the behavior of the clock paths to the behavior of the data path. If the generating and sampling paths have roughly the same delay (i.e. $d_{gen} \approx d_{sam}$), the data dependence can be neglected.

Corollary: A sufficient condition for voltage tracking is:

$$\frac{d'_{gen}}{d_{gen}} = \frac{d'_{data}}{d_{data}} = \frac{d'_{sam}}{d_{sam}} \tag{4.31}$$

Proof: Since $d'_{gen} = d_{gen} (d'_{data} / d_{data})$ and $d'_{sam} = d_{sam} (d'_{data} / d_{data})$, $d'_{gen} - d'_{sam}$ is equal to $d_{gen} (d'_{data} / d_{data}) - d_{sam} (d'_{data} / d_{data})$ and (4.30) is satisfied.

Figure 4.10 shows the relationship between the delay d of a device discharging an output load and the supply voltage V_0 for various input transition times. The relationship is based on the device delay model of (4.12). The size of the device is 1.00 µm. The load is 40 fF. The points are simulated and are considered exact. The solid lines are generated using the model. The *sensitivity* of the delay to the supply voltage increases with the input transition time. When the supply voltage varies by ±100 mV, the simulation results indicate that the delay varies by ±4.0% when the transition time is 20 ps. The delay varies by ±6.3% at 100 ps. Figure 4.11 shows a similar graph for various capacitive loads when the input transition time is fixed to 60 ps. The sensitivity of the delay to the supply voltage increases as the output load decreases. With a load of 20 fF, the sensitivity is ±7.1%. At 60 fF, it is ±4.4%.

Figure 4.10 and Figure 4.11 indicate that the input transition time and the output load have a significant impact on the voltage tracking behavior of the device. They also indicate that the device delay model correlates well with the simulation results.

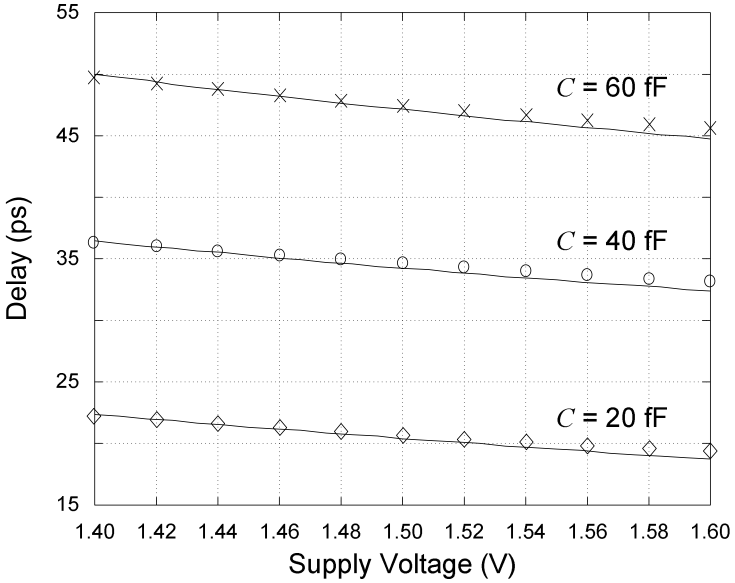


Figure 4.11: Delay versus supply voltage for various output loads.

The delay model of Section 4.3.1 can be used to estimate the voltage tracking behavior of local clock buffers in order to better understand how this behavior is affected by device sizing.

Figure 4.12 shows how the rise and fall delays track the supply voltage when $w_{p1} / w_{n1} = w_{p2} / w_{n2} = \beta$ and when each stage has a fanout of 4. The input transition time is fixed at 60 ps. The output load is fixed at 40 fF. The rise and fall delay curves are both normalized to the delays at $V_0 = 1.5$ V. The model predicts that both transitions scale the same. The simulation results show a minor difference between the two. They also show that the model slightly overestimates the supply voltage sensitivity.

Figure 4.13 shows how the rise and fall delays behave when they are asymmetrical, with $w_{p1} / w_{n1} = w_{p2} / w_{n2} = ¾ \beta$. Each stage still has a fanout of 4. The input transition time and the output load are kept constant. The model predicts that the rise and fall transitions track the supply voltage almost identically. The simulation results correlate well with the model's predictions. The impact of the *p*-to-*n* ratio of each stage on the delay-versus-voltage relationship is small. In all cases, the responses shown in Figure 4.12 and Figure 4.13 are nearly identical.

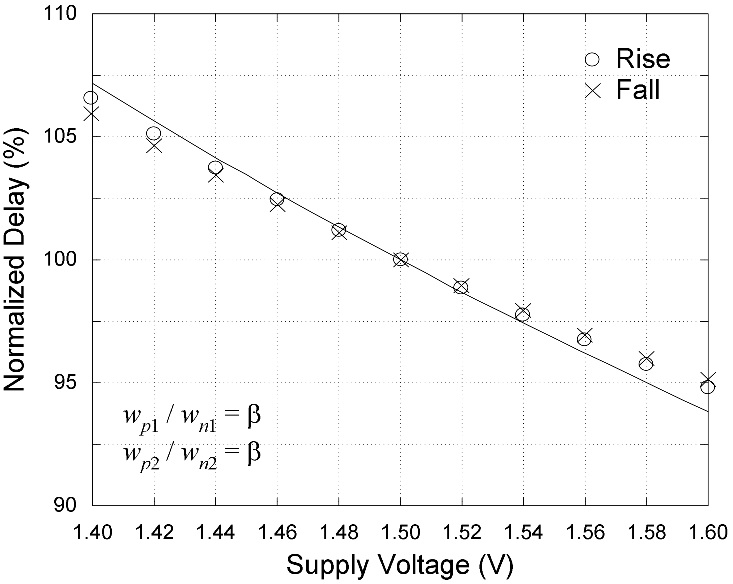


Figure 4.12: Voltage tracking with each stage having equal rise and fall delays.

For the same output load, the symmetrical design with $w_{p1} / w_{n1} = w_{p2} / w_{n2} = \beta$ consumes more power because its device capacitance is higher. Its gain is also poorer. For $\beta = 2.0$, the device capacitance difference is $(\beta + 1) / (\frac{3}{4}\beta + 1) = 20\%$. However, the simulation results indicate that its output rising transition time is about 20% faster. As discussed earlier, this improves flip-flop performance. It also results in more clock bandwidth.

Figure 4.14 shows how the asymmetrical design performs when the threshold voltages are skewed in different directions for the p-devices and the n-devices. The rise and fall delays still track the supply voltage reasonably well.

4.4 Device and Local Interconnect Bandwidth Models

The device bandwidth of a local clock buffer is the maximum clock frequency that it can reach when it has no output load. The maximum frequency is limited by the time required to charge or discharge the internal capacitances of the local clock buffer. As discussed in the previous section, these internal capacitances tend to be dominated by the gate and diffusion capacitances of its devices. Consequently, the strategy used for sizing the

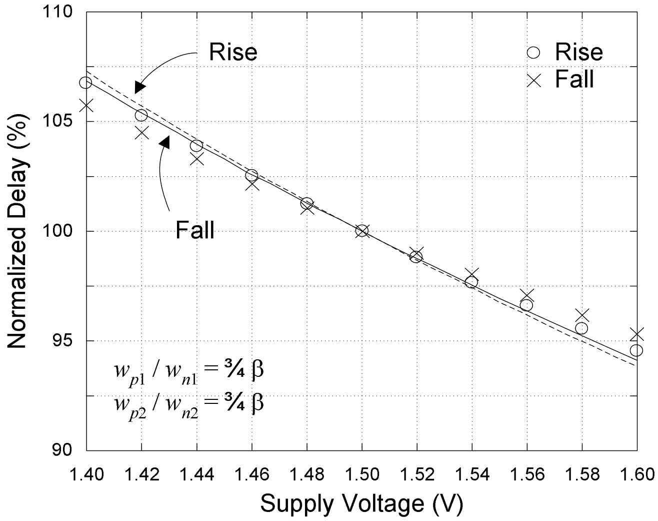


Figure 4.13: Voltage tracking with each stage having unequal rise and fall delays.

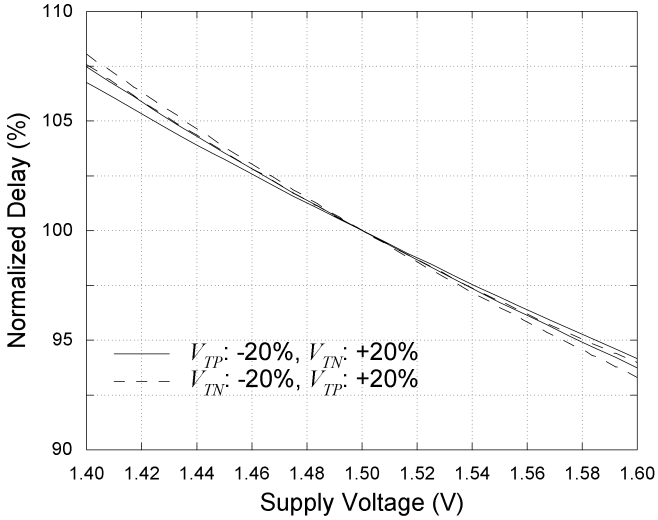


Figure 4.14: Voltage tracking with threshold voltage mismatches.

devices, in particular the fanout of each device, largely determines the device bandwidth of the buffer.

The notion of interconnect bandwidth is analogous: it is the maximum frequency at which an interconnect can switch. Interconnect bandwidth is traditionally defined as the bandwidth of a first-order linear system [4.12]:

$$f_{max} = \frac{1}{2\pi RC} \tag{4.32}$$

where R is the resistance and C is the capacitance of the interconnect. This approach is simple, but it ignores the interconnect driver. This is a problem because the driver, when undersized, can severely limit the maximum frequency at which the interconnect can operate.

In [4.13], Sakurai derives a model for the far-end voltage that attempts to take the behavior of the driver into account. The non-linear driver is modeled as a linear circuit having a resistance R_T. The far end of the interconnect is terminated by a load C_T. Sakurai approximates the step response at the far by:

$$v(t) = V_0 - V_0 \exp\left(- \frac{\frac{t}{RC} - 0.1}{\frac{R_T C_T}{RC} + \frac{R_T}{R} + \frac{C_T}{C} + 0.4} \right) \tag{4.33}$$

where V_0 is the supply voltage.

From this expression, the time t_ε required for the far-end voltage to reach $(1 - \varepsilon)V_0$ is given by:

$$t_\varepsilon = RC\left[0.1 - \log(\varepsilon)\left(\frac{R_T C_T}{RC} + \frac{R_T}{R} + \frac{C_T}{C} + 0.4 \right) \right] \tag{4.34}$$

Time t_ε is useful to establish a better definition of interconnect bandwidth. The definition proposed here is based on Figure 4.15 where a square wave is applied to the input of the interconnect driver. The interconnect bandwidth is defined as the highest square wave frequency f_{max} at which the far end still swings within ε of the power rails. If the input square wave rises at time zero, then time t_ε is the earliest time at which the input can switch again. Assuming that the driver is sized for equal rise and fall delays, the far-end voltage will reach $(1 - \varepsilon)V_0$ at time $2t_\varepsilon$. The corresponding bandwidth is therefore $1 / (2t_\varepsilon)$. It is worth noting that the proposed definition is valid for lumped and distributed interconnect models. The definition allows for loads to be attached at arbitrary points

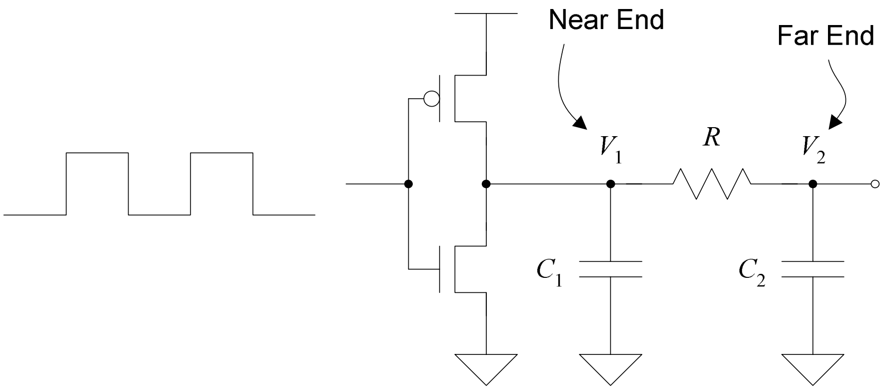


Figure 4.15: Definition of interconnect bandwidth.

along the interconnect. The device bandwidth of the driver is obtained by setting the interconnect resistance to zero.

Instead of assuming a linear driver, a more accurate expression for t_ε and for the corresponding square-wave bandwidth can be derived based on the device model proposed in Chapter 2.

4.4.1 Device Bandwidth Derivation

For this derivation, $R = 0$. At time zero, the driver input instantaneously rises from V_{SS} to V_{CC}. Immediately, the p-device turns off and the n-device enters saturation. Although the p-device supplies no current to the output, it still impacts the bandwidth because its drain capacitance must be discharged by the n-device. From Figure 4.15, the output voltage $v_1(t) = v_2(t)$ obeys the following differential equation:

$$Cv_2' + g_D v_2 = -b \tag{4.35}$$

where $C = C_1 + C_2$ and $b = g_G(V_0 - V_T) - g_D V_{DS0}$. The solution to (4.35) that satisfies $v_2(0) = V_0$ is:

$$v_2(t) = \left(\frac{b}{g_D} + V_0\right)e^{-\frac{g_D}{C}t} - \frac{b}{g_D} \tag{4.36}$$

where $V_0 = V_{CC} - V_{SS}$ is the nominal supply voltage.

Eventually, the drain-to-source voltage reaches V_{DS0} and the device enters its linear region of operation. This occurs at time t_{lin}, when $v_1(t_{lin}) = V_{DS0}$. Solving for t_{lin} yields:

$$t_{lin} = \frac{C}{g_D} \log \frac{g_D V_0 + b}{g_D V_{DS0} + b} \tag{4.37}$$

In the linear region of operation, the drain current is given by (2.18). R_{lin} is:

$$R_{lin} = \frac{1}{g_G} \frac{V_{DS0}}{V_0 - V_T} \tag{4.38}$$

After time t_{lin}, v_2 obeys:

$$R_{lin} C v_2' + v_2 = 0 \tag{4.39}$$

The solution that makes v_2 and v'_2 continuous at t_{lin} is:

$$v_2(t) = V_{DS0} e^{-\frac{1}{R_{lin}C}(t - t_{lin})} \tag{4.40}$$

The output voltage will reach εV_0 at time t_ε. Solving for t_ε yields:

$$t_\varepsilon = t_{lin} - R_{lin} C \log \frac{\varepsilon V_0}{V_{DS0}} \tag{4.41}$$

The square-wave device bandwidth f_{max} for the n-device is 1 / ($2t_\varepsilon$). The result, after substituting the expressions for t_{lin} and R_{lin} derived earlier, is:

$$f_{max} = \left(\frac{2C}{g_D} \log \left(\frac{g_D}{g_G} \frac{V_0 - V_{DS0}}{V_0 - V_T} + 1 \right) - \frac{2C}{g_G} \frac{V_{DS0}}{V_0 - V_T} \log \frac{\varepsilon V_0}{V_{DS0}} \right)^{-1} \tag{4.42}$$

The derivation for the p-device is analogous. The device bandwidth associated with the driver is limited by the device having the lowest bandwidth.

4.4.2 Interconnect Bandwidth Derivation

Like before, the driver input instantaneously rises from V_{SS} to V_{CC} at time zero. From Figure 4.15, the far-end voltage obeys the following differential equation:

$$k_2 v_2'' + k_1 v_2' - g_D v_2 = b \tag{4.43}$$

where $k_2 = -RC_1C_2$, $k_1 = -g_DRC_2 - C_1 - C_2$, and $b = g_G(V_0 - V_T) - g_DV_{DS0}$. The initial conditions are $v_2(0) = V_0$ and $v'_2(0) = 0$, where $V_0 = V_{CC} - V_{SS}$ is the nominal supply voltage. The solution to (4.43) is:

$$v_2(t) = p_1 e^{-\tau_1 t} + p_2 e^{-\tau_2 t} - \frac{b}{g_D} \tag{4.44}$$

where $\tau_i = (d_i + k_1) / (2\, k_2)$, $d_i = (-1)^{i+1} (k_1^2 + 4\, g_D\, k_2)^{½}$, and

$$p_i = \frac{d_i - k_1}{2 g_D d_i}(g_D V_0 + b) \tag{4.45}$$

Since $-\tau_1$ and $|p_1|$ are typically much larger than $-\tau_2$ and $|p_2|$,

$$v_2(t) \approx p_1 e^{-\tau_1 t} - \frac{b}{g_D} \tag{4.46}$$

Eventually, V_{DS} reaches V_{DS0} and the device enters its linear region of operation. This occurs at time t_{lin}, when $v_1(t_{lin}) = V_{DS0}$. From Figure 4.15, $v_1(t) = v_2(t) + RC_2v'_2(t)$:

$$v_1(t) = p_1(1 - RC_2\tau_1)e^{-\tau_1 t} + p_2(1 - RC_2\tau_2)e^{-\tau_2 t} - \frac{b}{g_D} \approx p_1(1 - RC_2\tau_1)e^{-\tau_1 t} - \frac{b}{g_D} \tag{4.47}$$

Solving for t_{lin} yields:

$$t_{lin} = \frac{1}{\tau_1}\log\left(g_D p_1 \frac{1 - RC_2\tau_1}{g_D V_{DS0} + b}\right) \tag{4.48}$$

In the linear region of operation, the drain current is given by (2.18). R_{lin} is:

$$R_{lin} = \frac{1}{g_G}\frac{V_{DS0}}{V_0 - V_T} \tag{4.49}$$

After time t_{lin}, v_2 obeys:

$$R_{lin}k_2v''_2 - k_3v'_2 - v_2 = 0 \tag{4.50}$$

where $k_3 = RC_2 + R_{lin}(C_1 + C_2)$. The solution that makes v_2 and v'_2 continuous at t_{lin} is:

$$v_2(t) = q_1 e^{\frac{k_3 - \delta_1}{2k_2R_{lin}}(t - t_{lin})} + q_2 e^{\frac{k_3 - \delta_2}{2k_2R_{lin}}(t - t_{lin})} \tag{4.51}$$

where $\delta_i = (-1)^{i+1}\,(k_3^2 + 4\,R_{lin}\,k_2)^{½}$ and

$$q_i = \frac{\delta_i + k_3}{2\delta_i} v_2(t_{lin}) - \frac{k_2 R_{lin}}{\delta_i} v_2'(t_{lin}) \tag{4.52}$$

Since the first term of (4.51) typically dominates, the far-end voltage can be approximated as follows:

$$v_2(t) \approx q_1 e^{\frac{k_3 - \delta_1}{2k_2 R_{lin}}(t - t_{lin})} \tag{4.53}$$

In (4.52), q_i is defined in terms of $v_2(t_{lin})$ and $v'_2(t_{lin})$. Equation (4.44) gives the expression for $v_2(t_{lin})$:

$$v_2(t_{lin}) = p_1 e^{-\tau_1 t_{lin}} + p_2 e^{-\tau_2 t_{lin}} - \frac{b}{g_D} \approx p_1 e^{-\tau_1 t_{lin}} - \frac{b}{g_D} \tag{4.54}$$

Taking the derivative yields $v'_2(t_{lin})$:

$$v_2'(t_{lin}) = -\tau_1\, p_1 e^{-\tau_1 t_{lin}} - \tau_2\, p_2 e^{-\tau_2 t_{lin}} \approx -\tau_1\, p_1 e^{-\tau_1 t_{lin}} \tag{4.55}$$

The far-end voltage will reach εV_0 at time t_ε. Solving for t_ε yields:

$$t_\varepsilon = t_{lin} + \frac{2k_2 R_{lin}}{k_3 - \delta_1} \log \frac{\varepsilon V_0}{q_1} \tag{4.56}$$

The square-wave bandwidth f_{max} of the interconnect is therefore 1 / ($2t_\varepsilon$):

$$f_{max} = \frac{1}{2}\left(t_{lin} + \frac{2k_2 R_{lin}}{k_3 - \delta_1} \log \frac{\varepsilon V_0}{q_1} \right)^{-1} \tag{4.57}$$

4.4.3 Validation and Examples

The mathematical structure of $v_2(t)$ is shown in Figure 4.16. The first part of the solution is given by (4.44) while the second is given by (4.51). The solid lines represent the exact solutions. The dotted lines represent the approximate solutions that were used to define t_ε.

The device enters its linear region of operation at $t_{lin} \approx 95$ ps when the near-end voltage (not shown) is equal to V_{DS0}. The exact and the approximate solutions for part 1 are almost identical, except at time zero where the approximate solution does not satisfy the

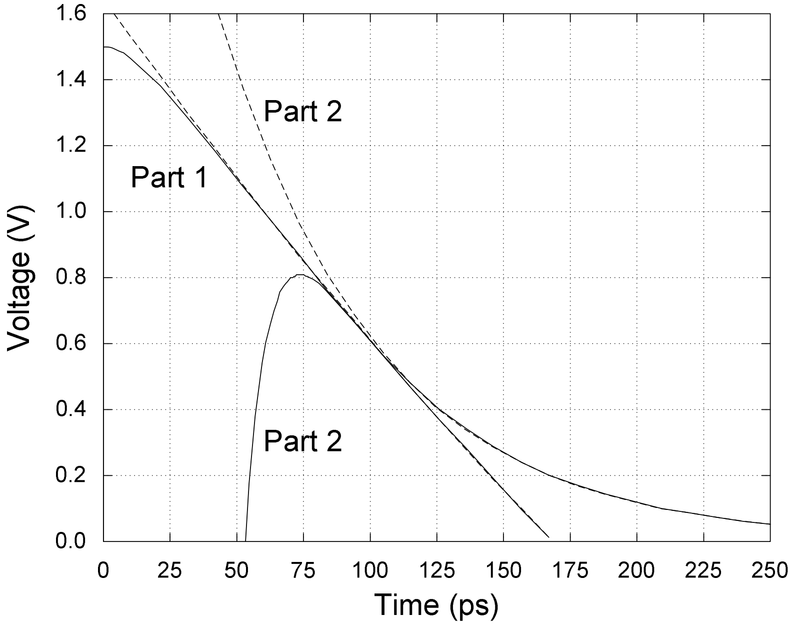


Figure 4.16: Mathematical structure of solution.

boundary conditions $v_2(0) = V_0$ and $v'_2(0) = 0$. For t_ε, this initial inaccuracy is irrelevant. Only the voltage at time t_{lin} matters. The exact and approximate solutions for part 2 are very different before t_{lin}, but quickly converge afterwards. They are indistinguishable at time t_ε.

Figure 4.17 shows how the square-wave bandwidth of a metal-5 copper interconnect is affected by the size of its driver. Its length is 2400 μm. The width and spacing are both 0.70 μm or 1.25 times the minimum. The thickness is 0.90 μm. The corresponding interconnect resistance and capacitance are 40.5 Ω/mm and 225.7 fF/mm, respectively. ε is set to 1%. The curve is plotted using the f_{max} equation. The points represent the simulated bandwidth and are considered exact. The lumped model of Figure 4.15 is used for the wire during simulation. Clearly, the model is in agreement with the simulation results.

The relationship between the length of the interconnect and its bandwidth is shown in Figure 4.18. Again, the model closely matches the simulation results. The bandwidth

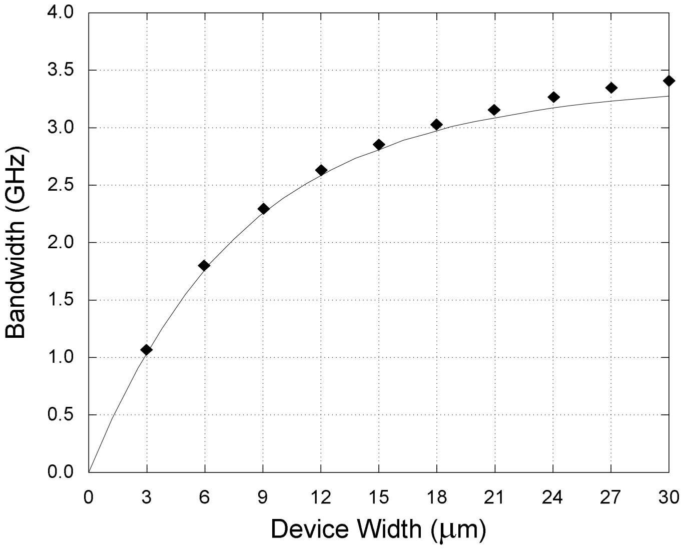


Figure 4.17: Bandwidth versus driver size.

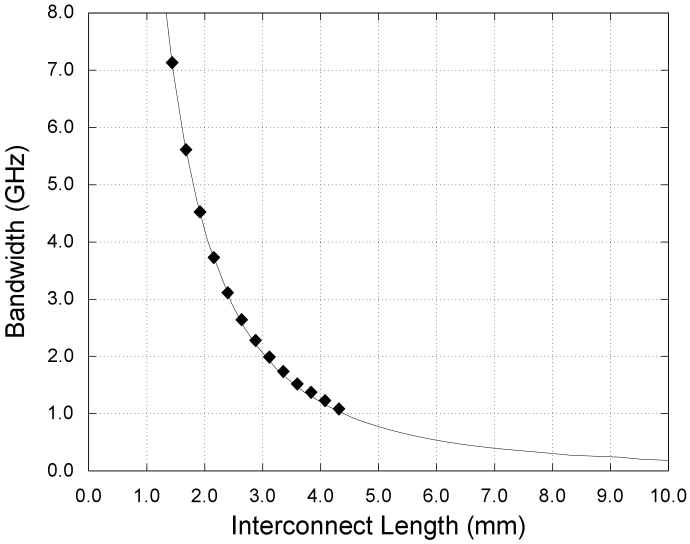


Figure 4.18: Bandwidth versus interconnect length.

rapidly decreases as the length increases. The n-device of the driver is fixed at 20 μm. ε is still 1%.

The impact of ε on the bandwidth is shown in Figure 4.19. Increasing the allowed voltage offset significantly increases the bandwidth. When ε is zero, so is f_{max}. Because it delays exponentially, the far-end voltage takes an infinite amount of time to reach zero.

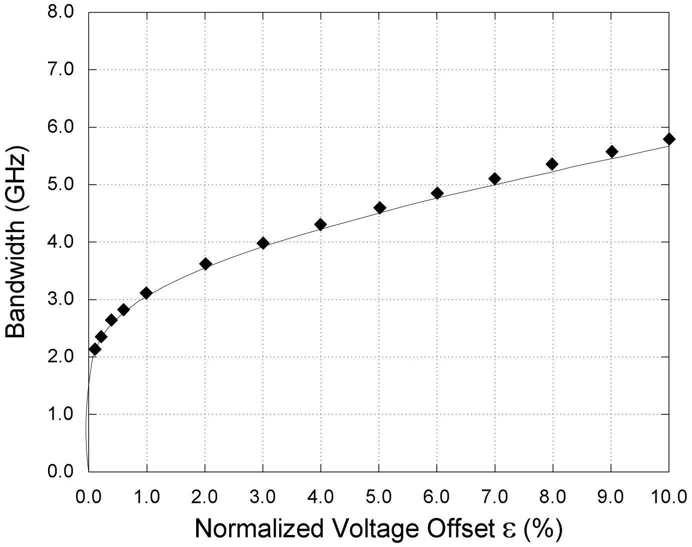


Figure 4.19: Bandwidth versus ε.

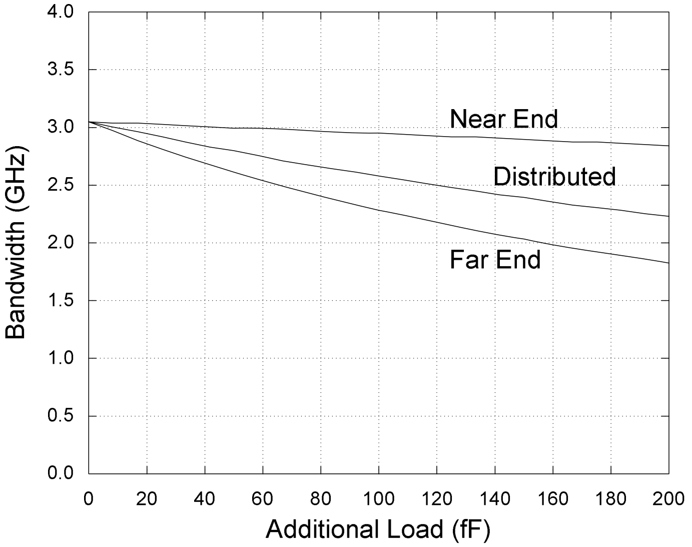


Figure 4.20: Bandwidth versus load profile.

Finally, Figure 4.20 quantifies the effect of load distribution on interconnect bandwidth when the entire load is placed at the near end, when the entire load is placed at the far end, and when the load is distributed evenly. Placing the load at the far end greatly diminished the bandwidth.

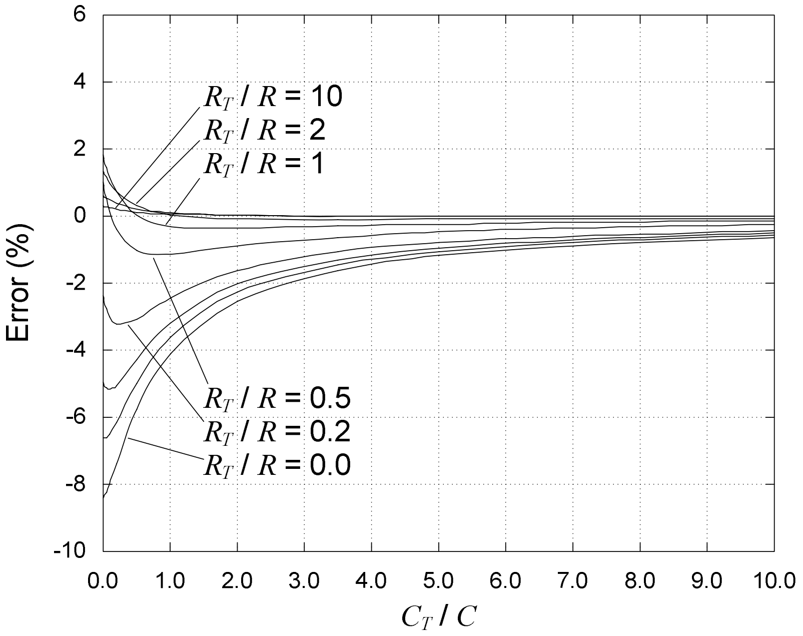


Figure 4.21: Delay error with lumped interconnect model.

The interconnect bandwidth given by (4.57) is derived assuming the lumped interconnect model of Figure 4.15, instead of a distributed one.

The error introduced by this approximation is analyzed in [4.14]. The interconnect driver is modeled as a linear circuit having a resistance R_T. The far end is terminated by a load C_T. It is shown that the error with a 3-step π ladder model is highest when R_T and C_T are both zero. For the 90% response time, the error reaches 2.3%.

When a single ladder step is used, Figure 4.21 shows that the error can reach 8.4% for the 50% response time. Again, the maximum error occurs when R_T and C_T are both zero. The error decreases rapidly when R_T or C_T increase.

Table 4.2 shows the bandwidth error associated with the 1-step π ladder approximation when the driver is represented using the quasi-linear MOSFET model instead of a simple resistor. With the lumped interconnect model, the bandwidth is underestimated by up to 8.9%. The accuracy of the1-step π ladder approximation can be improved by allowing the interconnect capacitance to be split unevenly between C_1 and C_2 [4.15].

Table 4.2: Bandwidth error due to lumped interconnect model.

Interconnect Length (mm)	R (Ω)	C (fF)	Simulated Bandwidth With Lumped Model (GHz)	Bandwidth Simulated With Distributed Model (GHz)	Error (%)
1.44	58.4	325.0	7.143	7.358	-2.9%
1.68	68.1	379.1	5.629	5.856	-3.9%
1.92	77.8	433.3	4.538	4.763	-4.7%
2.16	87.6	487.4	3.733	3.949	-5.5%
2.40	97.3	541.6	3.123	3.325	-6.1%
2.64	107.0	595.8	2.649	2.837	-6.6%
2.88	116.8	649.9	2.275	2.449	-7.1%
3.12	126.5	704.1	1.975	2.135	-7.5%
3.36	136.2	758.2	1.730	1.877	-7.9%
3.60	146.0	812.4	1.527	1.663	-8.2%
3.84	155.7	866.6	1.358	1.484	-8.5%
4.08	165.4	920.7	1.216	1.332	-8.7%
4.32	175.1	974.9	1.094	1.202	-8.9%

4.5 Crosstalk Jitter

Local clock interconnects are often (but not always) shielded to reduce cross-coupling jitter. Complete shielding of every interconnect segment is not necessarily desirable. The reason is that it tends to be excessively costly in term of metal usage and possibly power. A region where vertical metal congestion is aggravated by complete shielding is shown in Figure 4.22. This section derives a model to quantify the impact of crosstalk on jitter in order to better understand when shielding is appropriate.

4.5.1 Derivation

The model is based on the derivation of the square-wave bandwidth expression of Section 4.4. First, expressions for the 50% delay at the far-end of the interconnect are derived. Then, the timing impact of an attacker switching at the same time as the clock is quantified using the methodology described in [4.16]. For simplicity, the proposed jitter model assumes that the input transition time of the interconnect driver is infinitely fast. The error introduced by this assumption is small, as discussed later.

If the interconnect resistance is sufficiently low, the far-end voltage can reach its 50% point while the driver is still in saturation. When this is the case, $v_2(t)$ is given by (4.44):

$$v_2(t) \approx p_1 e^{-\tau_1 t} - \frac{b}{g_D} \tag{4.58}$$

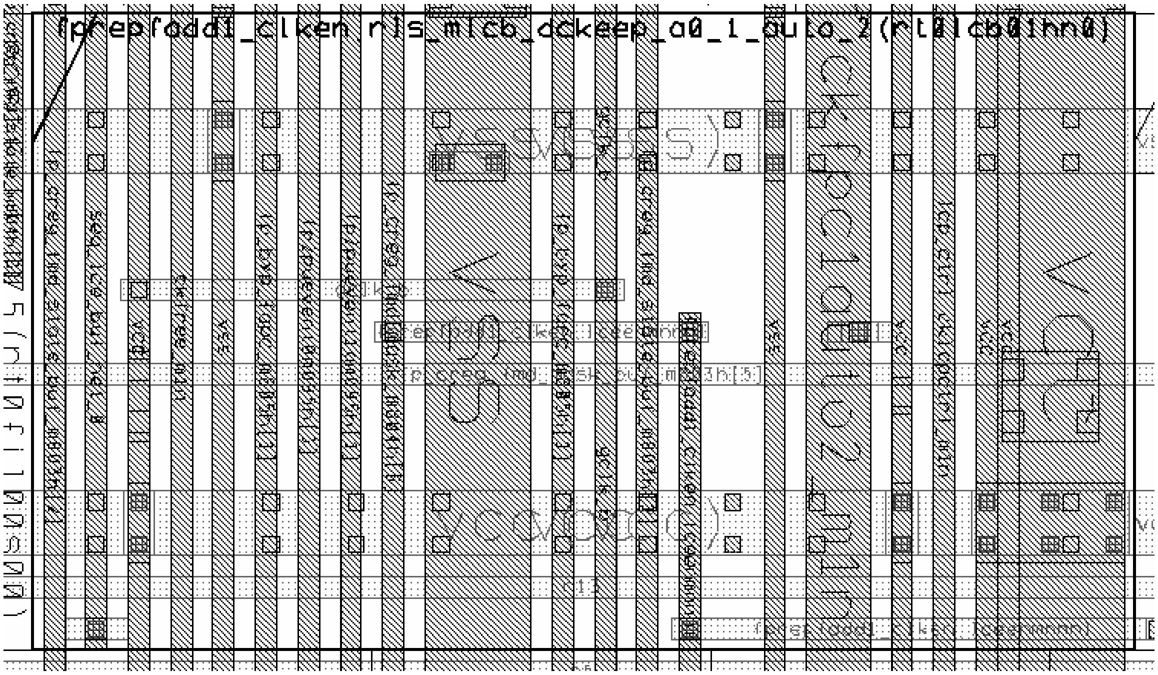


Figure 4.22: Metal congestion aggravated by complete shielding.

The 50% delay d_{sat} is obtained by solving $v_2(t) = 50\% \times V_0$. The result is:

$$d_{sat} = \frac{1}{\tau_1} \log\left(\frac{2g_D p_1}{g_D V_0 + 2b} \right) \tag{4.59}$$

The subscript emphasizes the saturation assumption.

It is also possible for the far-end voltage to reach its 50% point after time t_{lin}. Then, from (4.51),

$$v_2(t) \approx q_1 e^{\frac{k_3 - \delta_1}{2k_2 R_{lin}}(t - t_{lin})} \tag{4.60}$$

Solving for the 50% point when the driver is in its linear region of operation yields:

$$d_{lin} = t_{lin} + \frac{2k_2 R_{lin}}{k_3 - \delta_1} \log\left(\frac{V_0}{2q_1} \right) \tag{4.61}$$

It is important to keep in mind that d_{sat} and d_{lin} are functions of C_1 and C_2 and that d_{lin} must be greater than t_{lin} to be valid.

C_1 and C_2 can be adjusted to model several sources of capacitance like the line-to-ground capacitance C_{LG}, the line-to-line capacitance C_{LL}, and the capacitance of the devices receiving the clock. C_1 can also include the diffusion capacitance of the driver.

If the clock net is not fully shielded, then its delay will change (increase or decrease) when attacked by its parallel neighbors. This delay variation is traditionally estimated by appropriately scaling the line-to-line capacitance and treating it as if it was grounded [4.17]. If the clock and an attacker are switching in opposite directions, the coupling capacitance is multiplied by two. If both are switching in the same direction, the coupling capacitance is set to zero. C_1 and C_2 can therefore be expressed as follows:

$$\begin{aligned} C_1 &= C_{NE} + \tfrac{1}{2} C_{LG} + \alpha C_{LL} \\ C_2 &= C_{FE} + \tfrac{1}{2} C_{LG} + \alpha C_{LL} \end{aligned} \tag{4.62}$$

where the near-end capacitance C_{NE} includes the diffusion capacitance of the driver and the near-end loads, the far-end capacitance C_{FE} includes the far-end loads, and α is the switch factor used to scale the line-to-line capacitance based on the behavior of the attackers. When the neighbors on both sides of the clock are quiet, $\alpha = 1$ and the total wire capacitance is $C_{LG} + 2C_{LL}$.

The far-end delay $d_{50\%}$ is therefore an implicit function of the switch factor α:

$$d_{50\%}(\alpha) = \begin{cases} d_{sat} & (d_{sat} \le t_{lin}) \\ d_{lin} & (d_{sat} \ge t_{lin}) \end{cases} \tag{4.63}$$

The crosstalk jitter g is the change in delay resulting from the activity of the attackers and is modeled as:

$$g = d_{50\%}(\alpha_{max}) - d_{50\%}(\alpha_{min}) \tag{4.64}$$

As pointed out in [4.16], the delay with $\alpha = 0$ is not guaranteed to be the fastest. Similarly, the delay with $\alpha = 2$ is not guaranteed to be the slowest. When the attacker switches much faster than the victim, using –1 and 3 better reflects the worst-case variation. However, since the clock usually switches relatively fast, it is assumed here

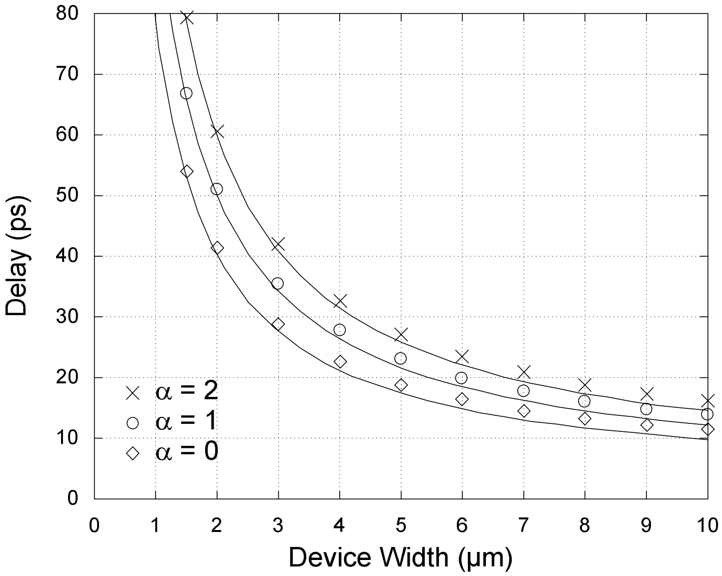


Figure 4.23: Interconnect jitter assuming step input.

that using α_{min} = 0 and α_{max} = 2 for the switch factor yields a realistic worst-case variation.

4.5.2 Validation

Figure 4.23 shows the delay curves $d_{50\%}(\alpha)$ corresponding to a local clock interconnect typical for a 130-nm technology [4.11]. The size of the n-device of the driver is varied. The net uses the metal-3 layer. It is unshielded and minimum-pitch. Its width and spacing are both 224 nm while its thickness is 360 nm. The length is 200 μm. The total device load is 100 fF (C_{NE} = C_{FE} = 50 fF). The crosstalk jitter is the difference between the curves corresponding to $\alpha = 0$ and $\alpha = 2$. The points are simulated and are considered exact. Clearly, they are in good agreement with the delay curves.

The jitter model is derived assuming a step-input for the interconnect driver, which is *a priori* unrealistic. Nevertheless, Figure 4.24 shows that the model is still reasonably accurate when a 50-ps input ramp is used instead. The slower input rise time increases the delay of the driver, but the increase tends to be relatively independent of α. The error increases with the input transition time. In practice, the fact that the clocks tend to be the signals with the fastest transition times helps keep the model accurate.

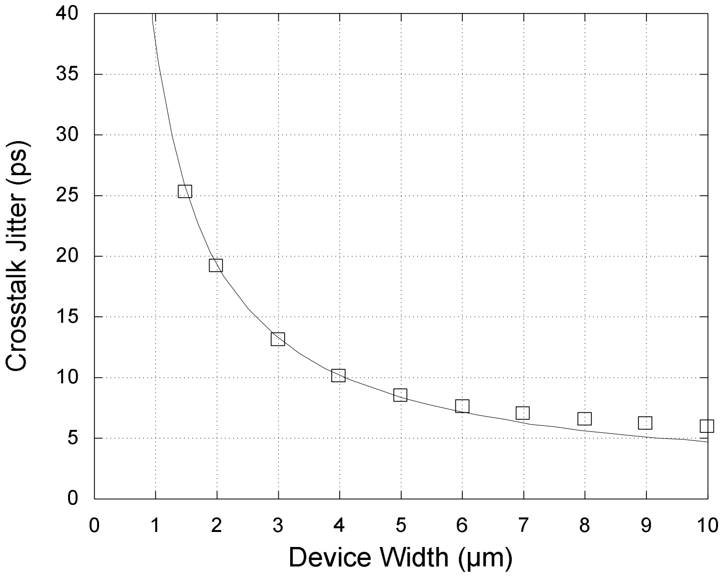


Figure 4.24: Interconnect jitter assuming 50-ps ramp input.

4.6 Shielding, Performance, and Power

The jitter model can be used to quantify the clock inaccuracy reduction that shielding can provide. It can also be used to analyze the power-performance tradeoff of removing existing shields to have more interconnect spacing.

4.6.1 Capacitance Modeling

As shown in Figure 4.25, C_{LG} and C_{LL} are determined by the thickness *T*, the width *W*, the spacing *S*, the interlayer dielectric (ILD) thickness *H*, and the length *L* of the interconnect. According to the curve-fitted numerical simulation results presented in [4.18], the line-to-ground capacitance is

$$\frac{C_{LG}}{k\varepsilon_0 L} = \frac{2W}{H} + 4.080 \times \left(\frac{T}{T + 4.531 \times H}\right)^{0.071} \left(\frac{S}{S + 0.536 \times H}\right)^{1.773} \tag{4.65}$$

and the line-to-line capacitance is

$$\frac{C_{LL}}{k\varepsilon_0 L} = 1.412 \times \frac{T}{S} e^{-\frac{4S}{S + 8.014 \times H}} + 2.370 \times e^{-\frac{2S}{S + 6H}} \left(\frac{W}{W + 0.308 \times S}\right)^{0.257} \left(\frac{H}{H + 8.961 \times S}\right)^{0.757} \tag{4.66}$$

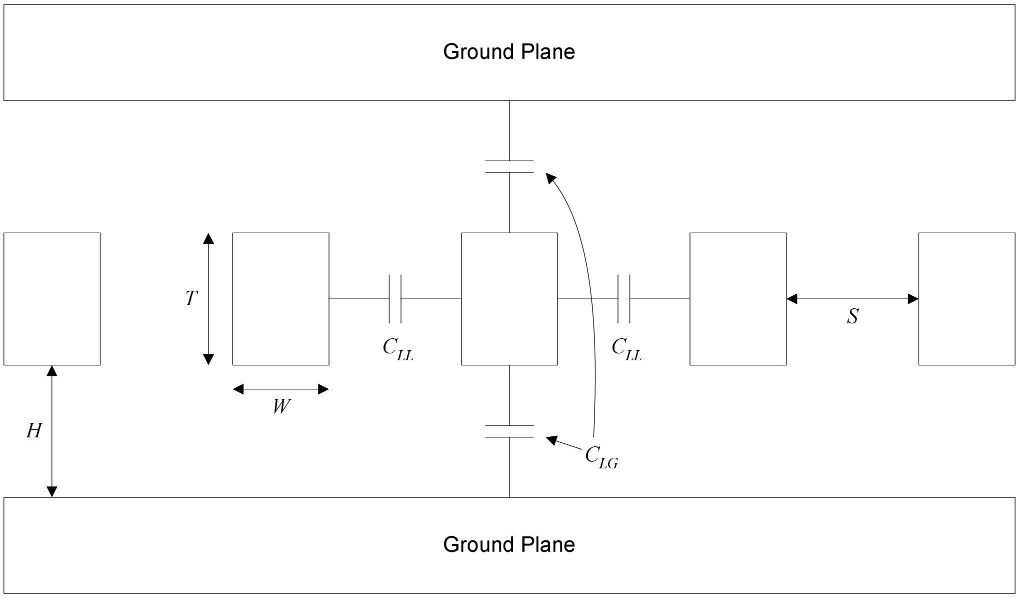


Figure 4.25: interconnect capacitance components.

where $\varepsilon_0 = 8.854 \times 10^{-3}$ fF/μm and k is the relative permittivity of the ILD. For silicon dioxide, $k = 3.9$. T and W are assumed to take into account the metal liner thickness.

The ground planes of Figure 4.25 are an accurate model for the interlayer capacitance provided than the metal density of the layers they represent exceeds 33% [4.17].

Figure 4.26 shows how the capacitance components of the M3 wire discussed earlier ($W = 224$ nm, $T = 360$ nm, $L = 200$ μm) vary when the spacing is increased. The line-to-line capacitance initially decreases very rapidly as the spacing increases. The additional spacing progressively diverts the electric field away from the adjacent lines to the upper and lower ground planes. This, in turn, increases the line-to-ground capacitance.

4.6.2 Jitter Versus Spacing and Shielding

Figure 4.27 shows how the clock jitter gets smaller as the spacing increases for the metal-3 interconnect ($W = 224$ nm, $T = 360$ nm, and $L = 200$ μm). The size of the driver is 6.0 μm. Furthermore, $C_{NE} = C_{FE} = 50$ fF. When the spacing is minimum and the clock interconnect is unshielded, the jitter is 7.4 ps. Doubling the space cuts the jitter in more than half, to 3.1 ps. Although the additional spacing stops reducing the total capacitance

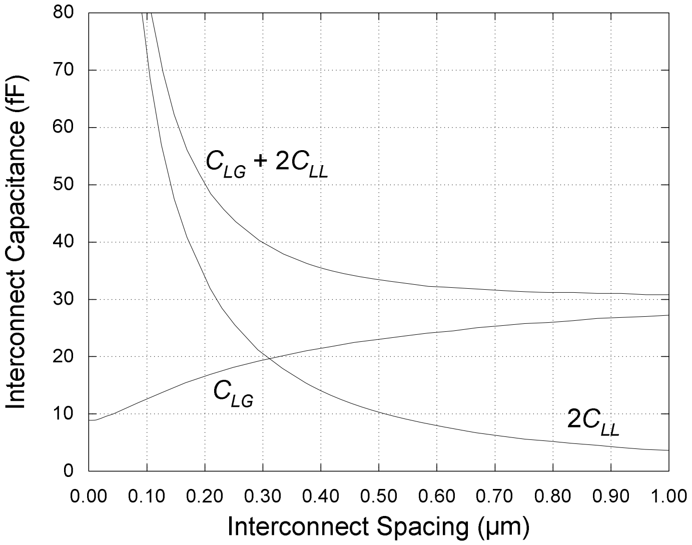


Figure 4.26: Capacitance versus spacing for minimum-width M3 interconnect.

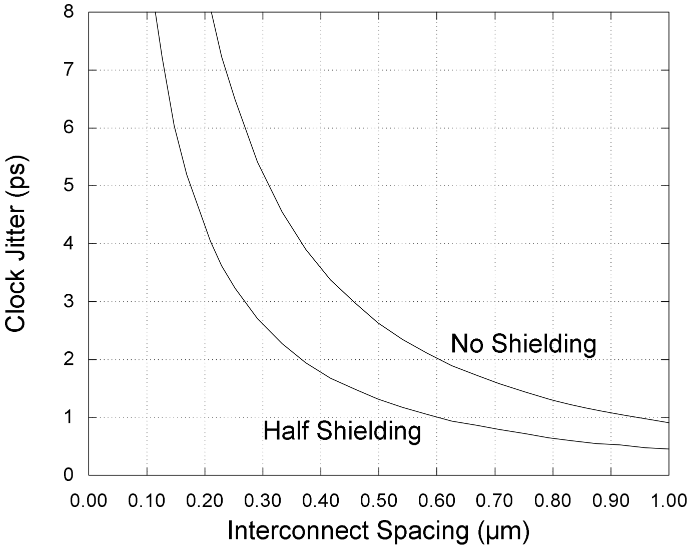


Figure 4.27: Jitter versus spacing for minimum-width M3 interconnect.

beyond a certain point (at about twice the minimum spacing, according to Figure 4.26), it continues to reduces the line-to-line capacitance and thus the jitter.

A fully shielded wire at minimum pitch occupies the same area (i.e. 3 tracks) as an unshielded wire at 3 times the minimum space. The difference in jitter is 1.7 ps. The difference in capacitance (which translates to a difference in power) is 14.9 fF. Removing one of the shields and centering the clock wire makes the spacing twice the minimum. Compared to the fully-shielded clock where the spacing is minimal, this reduces the capacitance by 12.3 fF and makes the jitter 3.1 ps.

With half shielding, the spacing between the clock and the shield does not have to equal the spacing between the clock and its switching neighbor. Moving the clock closer to the shield makes the spacing asymmetrical and can further reduce jitter.

The cost in power of choosing the fully shield all the clocks instead of making the space 3 times the minimum is easy to estimate. For an hypothetical 1.2-V chip running at 5.0 GHz with parameters roughly similar to the one described in [4.4], the cost is: CV^2f = 14.9 fF × $(1.2\ \text{V})^2$ × 5.0 GHz = 107 μW per clock. If the chip has 20000 clock wires, the cost is 20000 × 107 μW = 2.14 W for a 1.7-ps clock inaccuracy improvement. Depending on the frequency-versus-voltage relationship of the chip, removing the shields may or may not be advantageous, as discussed in Chapter 2.

4.7 Summary

For flip-flops, the setup time required for capturing a zero is generally different from the setup time for capturing a one. So are the times required for launching a zero and a one. For paths whose polarity cannot be determined *a priori*, the sequencing overhead is the worst-case setup time plus the worst-case launch time. The optimal setup time is traditionally defined as the setup time minimizing the sequencing overhead. The generalized definition proposed in Section 4.2 minimizes the sequencing overhead under all possible clock arrival times. Unlike the traditional definition, the generalized one accounts for flip-flop transparency and for clock inaccuracy. A model for the data-dependant clock jitter resulting from the switching activity of sequentials with naked clocks is proposed. It is shown that the impact of the clock transition time on the sequencing overhead is significant.

Section 4.3 derived a model to analyze the properties of the local clock buffers used in [4.4]. The model is applied to better understand how the delay tracks the supply voltage.

It shows that the impact of device sizing on tracking is relatively small and that sizing the devices for equal rise and fall delays is not mandatory. The model is also used to analyze the gain of the local clock buffer. It quantifies the relationship between gain and delay. It shows that local clock buffers driving small loads can suffer a significant gain loss due to the parasitic wire capacitance of their internal nodes.

Finally, a non-linear model is proposed for the bandwidth of a local clock buffer driving a local interconnect. The non-linearity is required to adequately capture the behavior of the devices. The model is used to analyze crosstalk jitter and to examine the conditions where clock shielding is advantageous from a power standpoint. A typical 130-nm local clock interconnect is analyzed, with and without shielding. The results show that the crosstalk jitter increases by 1.7 ps when the shields are removed to save power.

CHAPTER 5

Global Clocking

The subject of this chapter is global clocking. First, the power required for global clock distribution is discussed. It is shown to be fairly small, but not negligible. Then, the impact of the placement of the loads that the global clock distribution network must drive is studied quantitatively. Like in [5.1], the analysis uses minimum rectilinear Steiner trees (MRSTs) to connect randomly placed loads. Here, however, the MRSTs are constructed using an exact algorithm (instead of heuristics) and are applied, for the first time, to examine three specific structures often used for global clocking: a full grid, a partial grid, and an H-tree. It is shown that when the number of loads is very small, the fixed interconnect cost associated with a full or partial clock grid makes H-trees a better choice for minimizing power. However, as the number of loads increases, the difference between the three structures is found to practically vanish. It is also shown that the dispersion of the loads significantly impacts the total length of the wires required to connect them, and therefore, the power required for global clock distribution.

Next, several skew and jitter compensation strategies are reviewed and analyzed in terms of architecture, power dissipation, clock inaccuracy, and ease of implementation. Their compatibility with established design-for-testability (DFT) and design-for-debugability (DFD) techniques is also evaluated.

The last section discusses interconnect noise. Interconnect noise has traditionally been, and continues to be, a significant issue for global clock distribution as well as for skew and jitter compensation. All require sending signals over long distances and across multilevel interconnect structures. In these structures, the interconnect layers are

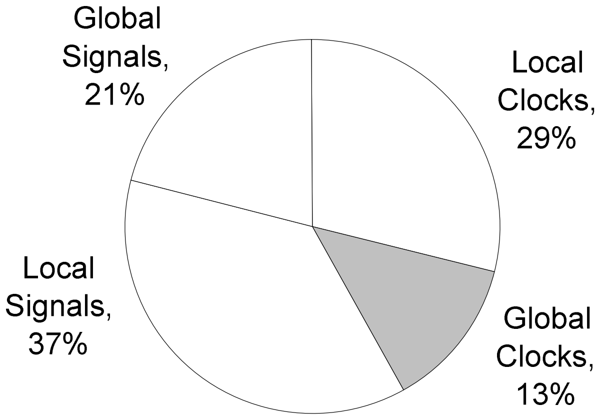


Figure 5.1: Dynamic power for global clocking.

practically always perpendicular to each other. Due to the capacitive coupling between adjacent layers, the switching activity in one layer produces noise in the others. Nevertheless, interlevel coupling noise is typically ignored and the orthogonal layers are assumed to behave as quiet metal planes. The problem is that the error due to this assumption is unclear, which significantly complicates clock jitter estimation. The last section examines and tests, for the first time, the limits of the quiet metal plane assumption. It analyses the capacitive interlevel coupling noise present at the far end of a victim line when a large number of perpendicular attackers are randomly switching. Each attacker is modeled as a Markov chain and the victim is modeled as a resistance-inductance-capacitance (RLC) transmission line. The result is a novel closed-form expression for the power spectral density of the interlevel coupling noise. The expression is used to rigorously show that the assumption is then statistically very good.

5.1 Introduction

The role of the clock distribution network is to take a reference clock signal and distribute it to a very large number of receivers scattered spatially. The input of the global clock distribution network is usually driven by a PLL. Consequently, its capacitance is required to be very small. On the other hand, the total capacitance for the receivers is typically very large because there are so many of them. The reference clock, in addition to being distributed, must therefore be amplified by several orders of magnitude. This

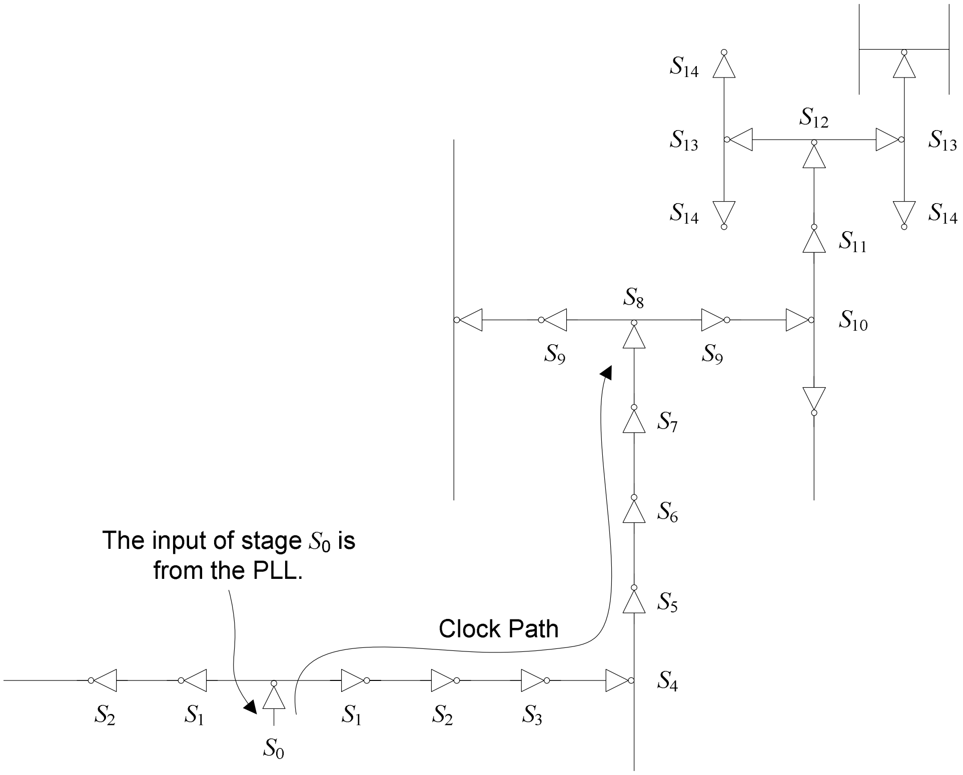


Figure 5.2: A 15-stage H-tree.

amplification process occurs in the current domain and does not alter the amplitude of the clock voltage waveform. The process can be also be analyzed in the power domain.

It is argued in [5.2] that, for microprocessors, the power required for global clock distribution is typically a fairly small fraction of the total power. This is verified in Figure 5.1 for a 130-nm microprocessor optimized for mobile applications, where only 13% of the dynamic power is used for global clocking [5.3]. After reaching the input of the local clock buffers, the clock no longer needs to be distributed over long distances. Local interconnects are used for routing to the sequentials. The primary function of the local clock buffers is to amplify the signal to drive the load presented by the sequential elements.

5.1.1 Distribution Versus Amplification

Figure 5.2 shows part of a 15-stage H-tree feeding a set of loads that are uniformly distributed over a 16-by-16 mm^2 die. The first stage, labeled S_0, drives the second stage

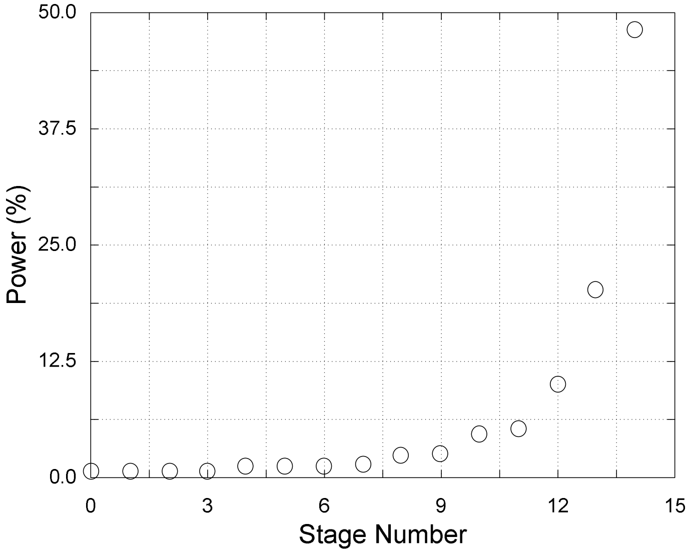


Figure 5.3: Power per stage.

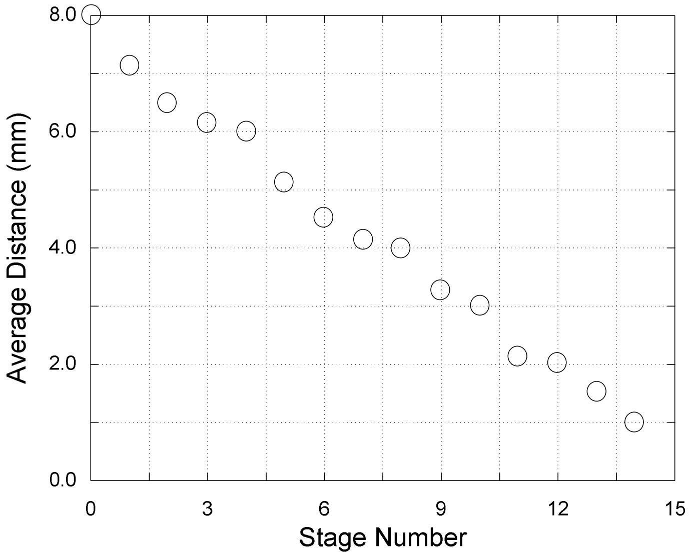


Figure 5.4: Average distance to nearest clock node.

inverters labeled S_1, which are themselves driving the S_3 inverters, and so on. The separation distance between successive stages is 1000 μm. Each additional inversion serves to further distribute and amplify the clock.

It is convenient to think of the distribution process as the spreading or scattering of a *fixed* number of clock reference nodes across the die. The result of this scattering is a

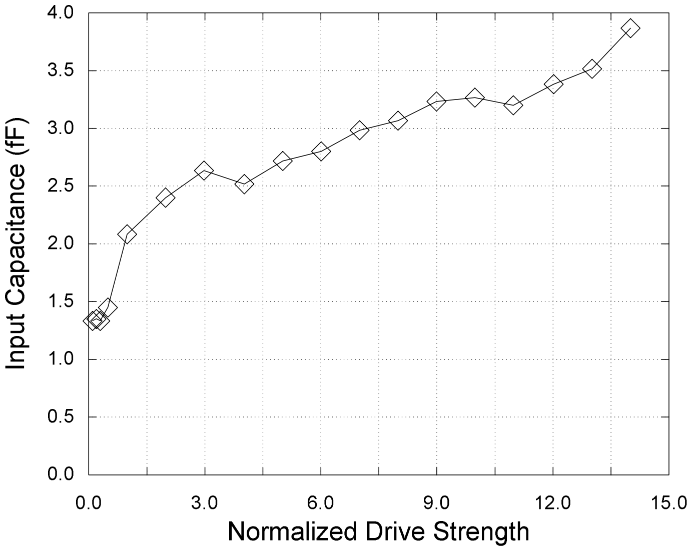


Figure 5.5: Input capacitance versus drive strength for a 3-stage local clock buffer.

reduction, for each load, of the average distance to the nearest clock node. At the same time, the amplification process can be seen as a process that increases the number of clock nodes through replication.

Figure 5.3 shows the simulated power breakdown for each stage of the H-tree. It is clear that stages 0 to 8 require very little power. Together, they account for only 9.5% of the total H-tree power. About two thirds of the H-tree power is consumed by the last two stages: 20.1% for S_{13} and 48.1% for S_{14}. The power of the clock signal is amplified by several orders of magnitude as it travels through the clock distribution network and as the number of clock nodes is multiplied from 1 for S_0 to 64 for S_{14}.

Figure 5.4 shows how the average distance separating each load from the nearest clock node decreases as additional stages are added to the H-tree. The average distance is computed analytically from the topology of the H-tree, assuming that the loads are uniformly distributed. When only S_0 is present, the average distance is 8 mm. It takes 8 additional stages to cut in half the average distance, but very little power.

The overall efficiency of the amplification process can be measured by normalizing the clock power delivered to the sequentials to the total power used for clocking. The

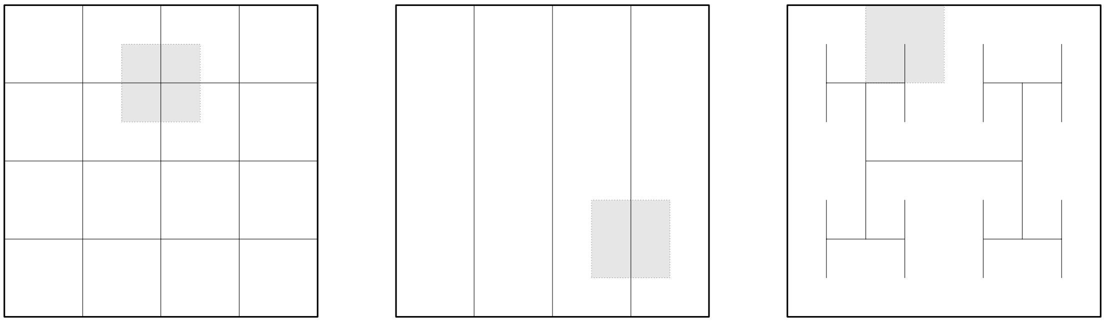


Figure 5.6: Structures for clock distribution.

amplification efficiency corresponding to the microprocessor described by Figure 5.1 is thus 29% / 42% = 69%. The amplification efficiency can never exceed 100%.

5.1.2 Design for Debugability and Design for Testability Requirements

As discussed in [5.4], a few basic features are required to support conventional design for testability (DFT) and design for debugability (DFD) techniques. The first is the ability to bypass the clock normally delivered to the sequential elements. If the PLL does not work properly, the distribution network must be able to deliver a low-frequency clock generated externally.

The clock distribution network must also be able to adjust the timing of the clock delivered to the sequential elements. This includes the ability to stretch or shrink a phase of the clock during a particular cycle. It also includes the ability to move the edges of the clock delivered to certain groups of sequentials. Controlling the clock helps debug and characterize the timing of critical circuit [5.5], [5.6].

5.2 Impact of Load Placement on Power

In general, global clock distribution requires driving a large number of widely scattered loads. This section analyses the impact of the placement of these loads on the interconnect length required for global clock distribution.

Figure 5.5 shows the input capacitance of the 3-stage local clock buffer (LCB) used on a 90-nm microprocessor. The input capacitance changes with the drive strength of the LCB

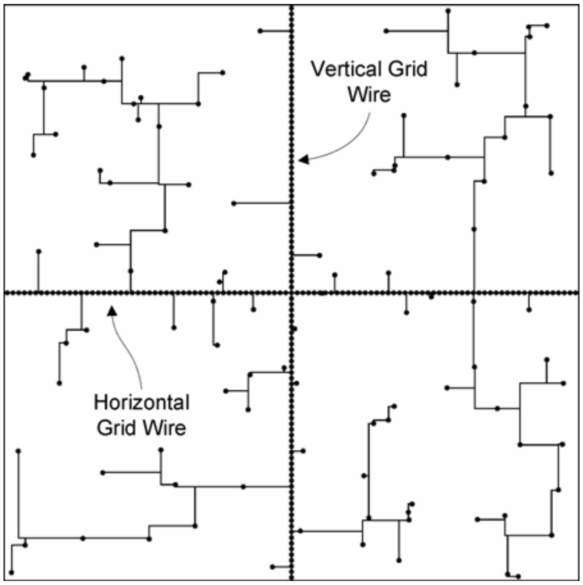


Figure 5.7: Minimum rectilinear Steiner tree for a full grid.

(i.e. with the number of standard loads it can drive). This LCB is constructed with an inverter, a nand gate to stop the clock, and an output inverter. Because of its high gain, its input capacitance is a weak function of the drive strength. When the drive strength is average, the input capacitance is 2.6 fF. Changing the drive strength to the minimum or the maximum perturbs the input capacitance by at most 1.3 fF. In the 90-nm technology used for this microprocessor, 1.3 fF is equivalent to about 6.5 µm of routing. This roughly corresponds the metal-1 length required to connect the devices forming the input stage of the LCB, a negligible amount of interconnect. Clearly, when the gain is high, the drive strength variations of the LCBs have little impact on the load that the global clock distribution network must drive. Then, the global clock power needed to drive the LCBs is primarily determined by the number of LCBs and by the total wire length required to connect them. The size of the LCBs has little impact.

Minimal rectilinear Steiner trees (MRSTs) are useful to analyze the total wire length required to connect a set of placed local clock buffers. An MRST is a tree constructed using Manhattan routing and connecting a given set of terminals with the shortest total interconnect length. The problem of determining an MRST is NP-hard [5.7] and the solution may not be unique. For a given global clock distribution network, an MRST can

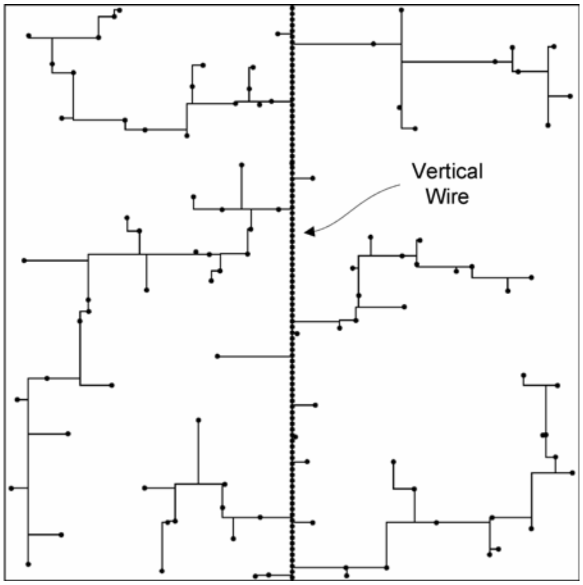


Figure 5.8: Minimum rectilinear Steiner tree for a partial grid.

be constructed to derive a lower bound for the length of interconnect required to drive all the loads. In this chapter, the MRSTs are constructed using an exact algorithm [5.8].

The three global clock distribution structures shown in Figure 5.6 are considered here. The first structure is a complete grid constructed using horizontal and vertical wires separated by the same distance. The second is a *half* grid. It still uses equally-spaced wires, but in only one direction. The third is an H-tree.

5.2.1 Interconnect Length for a Uniform Load Distribution

Figure 5.7 shows an MRST constructed for a full grid structure. The grid itself is modeled by distributing a large number of terminals at regular intervals. These terminals are placed over the grid wires and form a clearly visible cross. The remaining terminals model the loads that must be driven by the grid. These loads are randomly placed and obey a uniform probability distribution.

It is interesting to observe the MRST constructed for the full grid structure includes several short interconnect segments. These segments tend to significantly increase the number of vias required to connect the loads. The reason is that any change in routing direction (from horizontal to vertical or vice versa) requires an additional via. Since the

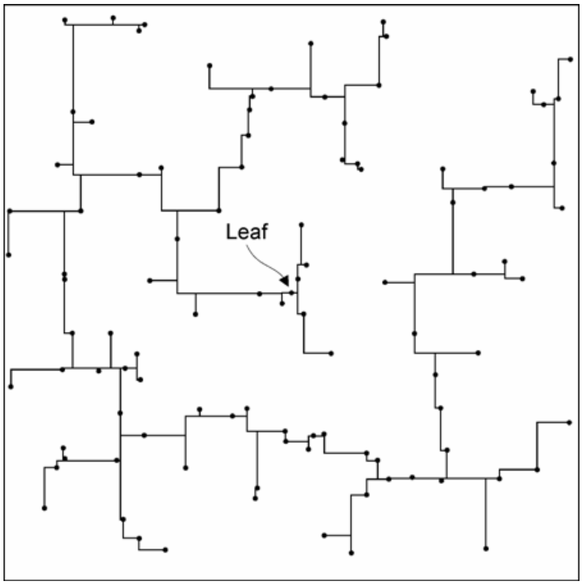


Figure 5.9: Minimum rectilinear Steiner tree for an H-tree.

resistance of these vias can be relatively high compared to the resistance of the short wires, the vias can significantly degrade the signal integrity of the clock at the loads. In practice, the number of vias is often limited which indirectly limits the maximum number of routing direction changes.

The model for the half grid structure is shown in Figure 5.8. This structure has a single vertical grid wire. The wire is modeled as before, with a large number of terminals placed on top of it. The MRST still uses several short wire segments.

With an H-tree, the loads are driven from the leaves, which are modeled by a single point. This is shown in Figure 5.9. The area over which the MRSTs are constructed is arbitrary. The unit square is chosen here.

Each global clock distribution structure has a fixed interconnect length that does not change when the number of loads does. For the full grid, the fixed length is 2. For the half grid, the fixed length is 1. For the H-tree, the fixed length is zero. Of course, each structure is incomplete. The wires and devices that would normally be necessary to drive the full and half grids or the leaves of the H-tree are omitted. It is assumed that the same distribution network would be able to feed all three structures.

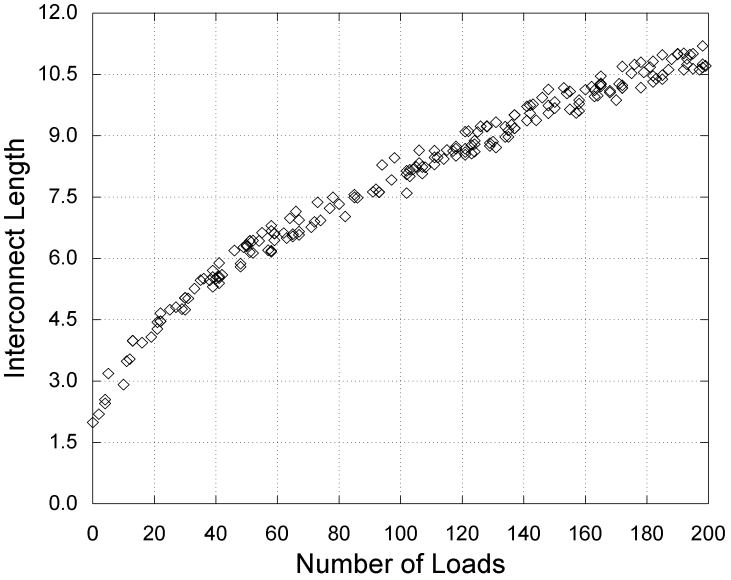


Figure 5.10: Interconnect length versus number of loads for a full grid.

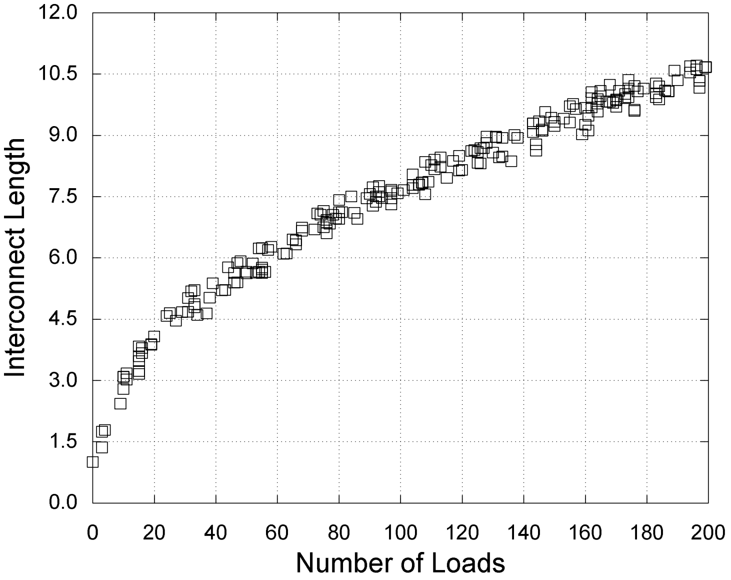


Figure 5.11: Interconnect length versus number of loads for a partial grid.

The relationship between the number of loads and the total wire length required to connect them is shown in Figure 5.10, Figure 5.11, and Figure 5.12 for the three structures. Each point is determined by randomly placing *N* loads and by finding an MRST that connects them. The total interconnect length of the MRST includes the fixed

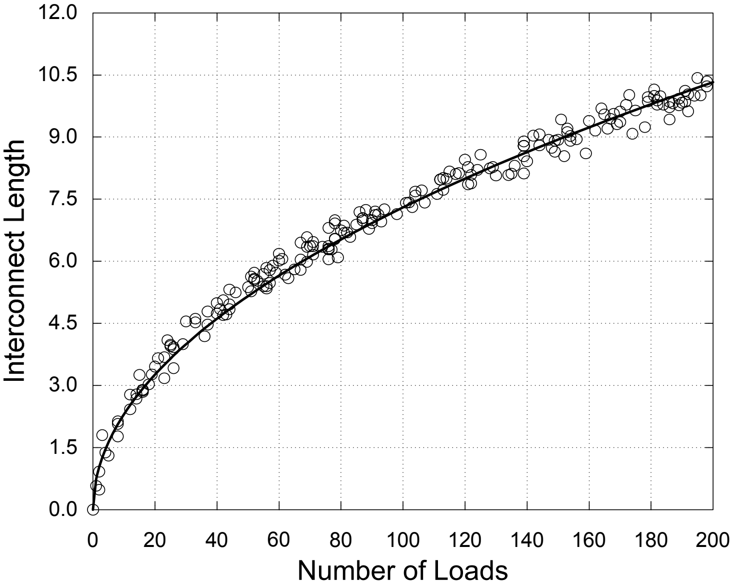


Figure 5.12: Interconnect length versus number of loads for an H-tree.

length of each structure. It never goes below 2 for the full grid and never below 1 for the half grid. When the number of loads to connect is small, adding one more tends to significantly increase the total interconnect length. As the number of loads increases, there are more opportunities for sharing wires and the total interconnect length grows more slowly. When the number of loads is very small, the total length is significantly less with the H-tree structure because its fixed length is null. However, it is interesting to note that when the number of loads gets larger, the three structures produce roughly the same interconnect length. Although the H-tree remains slightly better than the half grid, which remains slightly better than the full grid, the difference in length becomes insignificant with respect to the total length. With 40 loads for instance, the total interconnect length varies between 4.8 and 5.7 for the three structures. The impact on the total length of selecting a particular structure over another is comparable to the length variations caused by the random placement of the loads.

The experiments performed by Bern [5.9] suggest that the expected value for the length of an MRST connecting N uniformly distributed loads is asymptotic to:

$$L = \beta\sqrt{N} \tag{5.1}$$

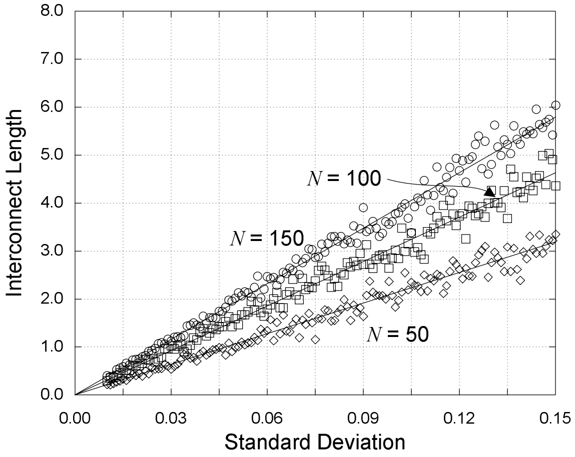


Figure 5.13: Impact of load dispersion on interconnect length.

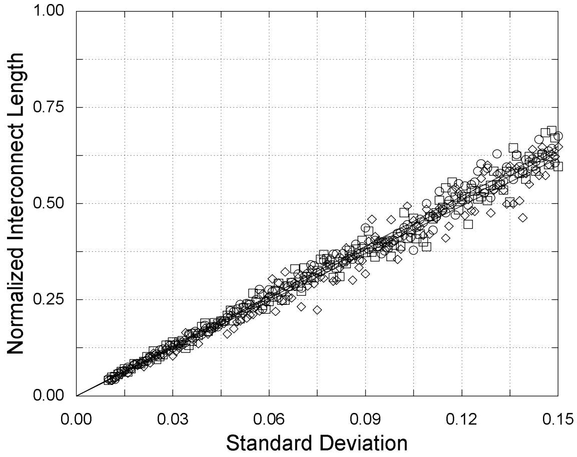


Figure 5.14: Normalized interconnect length versus dispersion.

with β between 0.7 and 0.8. The results for the H-tree provide more accuracy and yield β ≈ 0.73. It is observed in Figure 5.12 that even when N is small, the average length closely follows (5.1). In other words, β is a weak function of N.

5.2.2 Interconnect Length for a Gaussian Load Distribution

The degree of dispersion of the loads has a major impact on the wire length required to connect them. This relationship is analyzed here for an H-tree.

Figure 5.13 shows the length of the MRST required to connect N loads (50, 100, or 150) whose coordinates are independent Gaussian random variables. The loads are centered on the leaf of the H-tree. Their dispersion is controlled by changing their standard deviation σ. The required interconnect length increases as the dispersion increases. When the standard deviation is used as the dispersion metric, the relationship is linear, at least up to $\sigma = 0.15$. Increasing σ beyond that point would start to produce loads with coordinates falling outside the unit square and the resulting load distribution would no longer be Gaussian.

From (5.1), the length required to connect N uniformly distributed loads is asymptotic to $\beta\sqrt{N}$ with $\beta \approx 0.73$. Figure 5.14 shows the length achieved when the distribution is Gaussian normalized to what the length would have been with a uniform distribution. This Gaussian-to-uniform ratio quantifies the advantage of controlling the dispersion of the loads. As the dispersion increases, more wiring is needed. The length ratio then approaches unity, indicating that the full interconnect length is required. In Figure 5.14, the reduction in interconnect length is determined almost exclusively by σ. The number of loads has almost no impact.

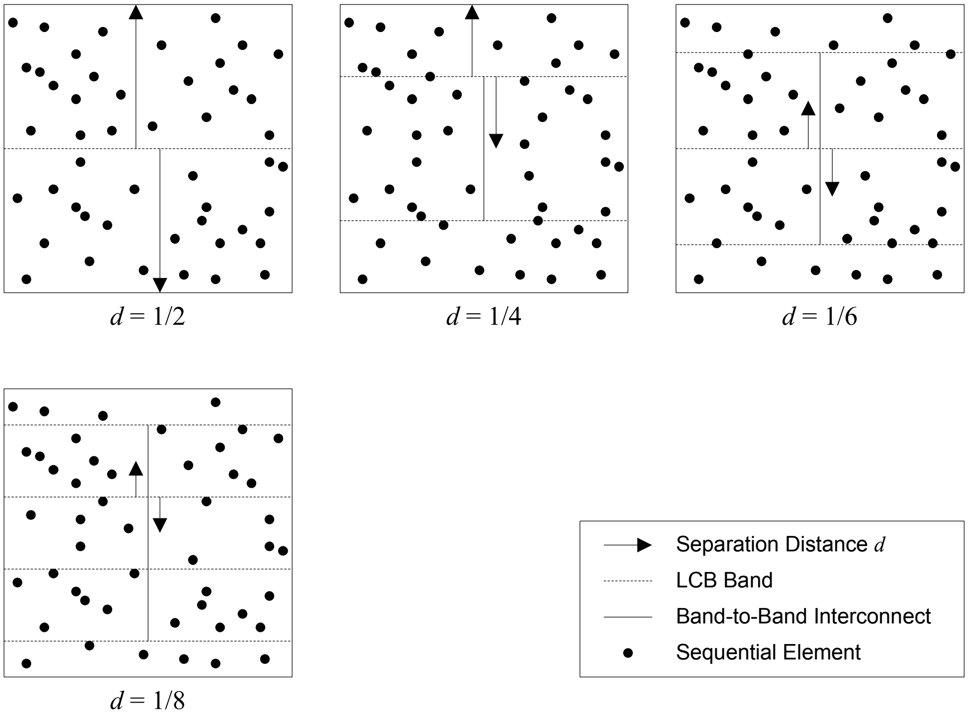
5.2.3 Load Clustering in Bands

As shown in the previous section, the degree of dispersion of the loads has a major impact on the wire length required to connect them. A common design strategy to reduce this length is to cluster the loads. The impact of clustering the loads in horizontal bands is analyzed here.

The minimum number of bands is dictated by the bandwidth of the local clock interconnects. As discussed in Chapter 4, this bandwidth limits the maximum separation distance d that can be allowed between a local clock buffer and a sequential element. A megablock the size of the unit square requires at least B bands where

$$B = \frac{1}{2d} \tag{5.2}$$

As shown in Figure 5.15, clustering the local clock buffers does not necessarily impose constraints on the placement of the sequential elements.



$d = 1/2$ $d = 1/4$ $d = 1/6$

$d = 1/8$

Figure 5.15: Clustering bands versus separation distance.

When the bands are fully populated (i.e. when N is large), the total wire length required to connect all the loads is given by:

$$L = \frac{1}{2d} + (1 - 2d) = B + \left(1 - \frac{1}{B}\right) \tag{5.3}$$

The first term is the number of unit-length bands dictated by the maximum separation distance d while the second term $1 - 2d = 1 - 1 / B$ represents the length of the vertical wire required to connect all the bands together in the center.

Equation (5.3) is valid for multiple bands when N is large. The average wire length ℓ required to connect n loads uniformly distributed over a *single* band can be derived using extreme value theory.

Let x_1, x_2, …, x_n represent the x-coordinates of the loads. The average wire length ℓ can be written as:

$$\ell = E[W - V] \tag{5.4}$$

where *V* and *W* are the minimum and maximum coordinates that must be connected in the band. Both are random variables:

$$\begin{aligned} W &= \max(\tfrac{1}{2}, x_1, x_2, \ldots, x_n) \\ V &= \min(\tfrac{1}{2}, x_1, x_2, \ldots, x_n) \end{aligned} \tag{5.5}$$

where ½ represents the center of each band, which must always be connected. Clearly, *W* cannot be less than ½ or greater than 1. Similarly, *V* is always between 0 and ½.

The probability that x_i is less than *x* is given by:

$$\mathrm{P}[x_i \le x] = x \tag{5.6}$$

for *i* = 1, …, *n*.

The probability that *W* is less than *x* is thus:

$$\mathrm{P}[W \le x] = \begin{cases} 0 & x < \frac{1}{2} \\ \prod_{i=1}^{n} \mathrm{P}[x_i \le x] = x^n & \frac{1}{2} \le x \le 1 \\ 1 & x \ge 1 \end{cases} \tag{5.7}$$

Taking the derivative of the cumulative distribution function yields the probability density function of *W*:

$$f_W(x) = \begin{cases} (\frac{1}{2})^n \, \delta(x - \frac{1}{2}) + n x^{n-1} & \frac{1}{2} \le x \le 1 \\ 0 & \text{elsewhere} \end{cases} \tag{5.8}$$

Since the cumulative distribution function is discontinuous at *x* = ½, the probability density function has an impulse at *x* = ½.

The expected value of *W* is therefore:

$$E[W] = \int_{-\infty}^{\infty} x \, f_W(x) dx = \frac{n + (\frac{1}{2})^{n+1}}{n+1} \tag{5.9}$$

Similarly, it is possible to show that:

$$E[V] = 1 - \frac{n + (\frac{1}{2})^{n+1}}{n+1} \tag{5.10}$$

Substituting into (5.4) yields:

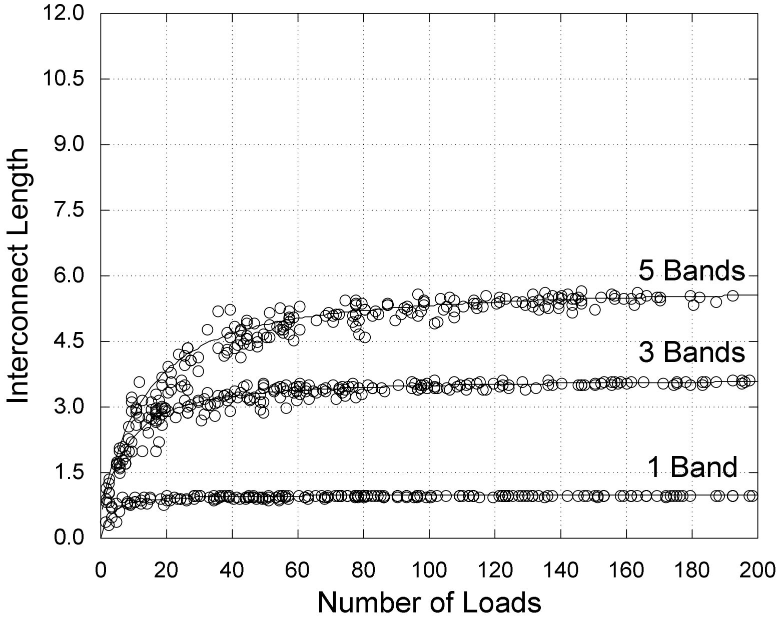


Figure 5.16: Interconnect length for 1, 3, and 5 load clustering bands.

$$\ell = 2\frac{n + (\frac{1}{2})^{n+1}}{n+1} - 1 \tag{5.11}$$

With several bands, the average number of loads per band is:

$$n = \frac{N}{B} \tag{5.12}$$

The total wire length *L* becomes:

$$L = B\ell + (1 - 2d) = B\ell + \left(1 - \frac{1}{B}\right) \tag{5.13}$$

Substituting (5.11) and (5.12) into (5.13) yields:

$$L = B\left(2\frac{\frac{N}{B} + (\frac{1}{2})^{\frac{N}{B}+1}}{\frac{N}{B}+1} - 1\right) + \left(1 - \frac{1}{B}\right) \tag{5.14}$$

As *N* goes to infinity, *L* approaches the value of (5.3):

$$\lim_{N\to\infty} B\left(2\frac{\frac{N}{B} + (\frac{1}{2})^{\frac{N}{B}+1}}{\frac{N}{B}+1} - 1\right) + \left(1 - \frac{1}{B}\right) = B + \left(1 - \frac{1}{B}\right) \tag{5.15}$$

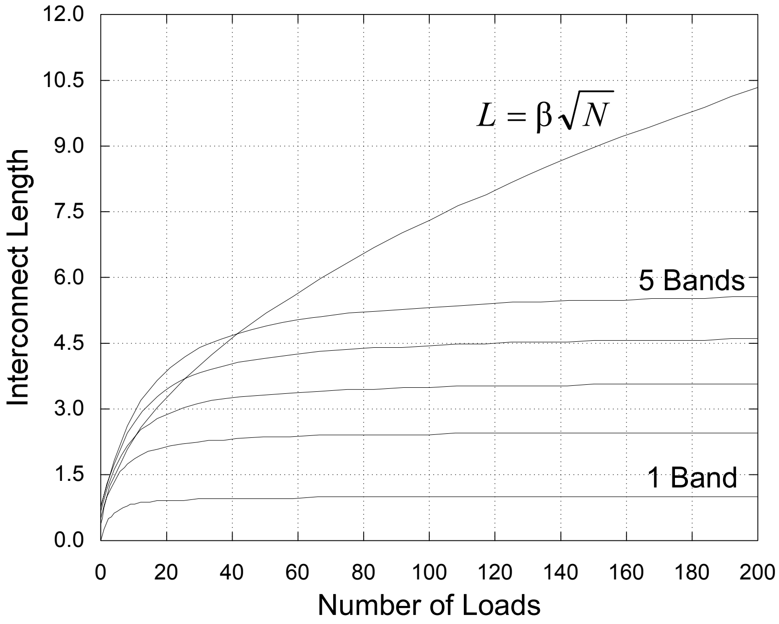


Figure 5.17: Benefit of load clustering.

Equation (5.14) is verified in Figure 5.16 for a megablock with 1, 3, and 5 bands of local clock buffers. The placement of the sequentials within the megablock is unrestricted. The sequentials are uniformly distributed over the unit square. The local clock buffers are clustered however. They are randomly assigned to particular bands and connected horizontally. The local clock buffers within a given band are uniformly distributed. A vertical wire is then added to correct the bands in the center.

When the number of local clock buffers is small, adding a few more significantly increase the total wire length. As the number of loads increases, the bands get more populated and the total wire length required to connect all the local clock buffers saturates. Obviously, making the number of bands as small as possible minimizes the total interconnect length.

Figure 5.17 quantifies the benefit of forcing a band arrangement for the local clock buffers. It compares the total wire length required to connect *N* loads when the loads are randomly placed to the length necessary when the loads are clustered. The total wire length without clustering is given by (5.1). It increases with the square root of *N* and is unbounded. Load clustering almost always results in a shorter wire length. But even when it does not, the total interconnect length remains close to the minimum achievable using a MRST. The physical design simplicity of load clustering is therefore a practical advantage.

5.3 Skew Compensation Strategies

Multiple skew-compensation strategies have been proposed by researchers, but only a few have been used in practice. The different skew-compensation strategies can be classified as centralized or distributed.

5.3.1 Centralized Skew Compensation

Clock distribution networks with centralized skew compensation usually rely on round-trip feedback for clock alignment. They require a centralized controller or skew compensator. Its role is to ensure that the pulses sent to every clock region arrive at the expected time.

The implementation proposed in [5.10] is shown in Figure 5.18. Each region has a local clock tree. The distribution network requires three carefully matched interconnect segments per region: one for the forward path and two for the feedback paths. The forward path delivers clock pulses to the root of every region's local tree. One of the feedback paths is also connected to the root; the other is connected to one of the leaves. The first feedback path allows the central controller to measure the round-trip delay of the clock pulses. The second feedback path is used to measure the delay through the local tree. A simple arithmetic circuit processes the measurements to deduce the phase error of the clock pulses received by the sequential elements. The controller is then able to correct the phase error by speeding up or slowing down the forward path. It is interesting to note that the proposed clock distribution network can intentionally skew the clock delivered to each region in order to increase performance. Unfortunately, the proposed design suffers from two major problems: the relatively coarse resolution of the measurement circuit and the limited precision of the phase correction that can be applied to the forward path.

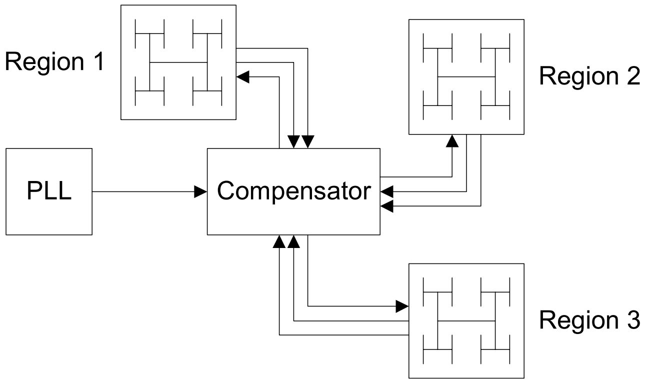


Figure 5.18: Centralized skew compensation strategy proposed in [5.10].

Furthermore, the strategy can be hard to implement in practice because it requires three carefully matched interconnects per region. However, the power dissipated by the skew-compensation circuit is negligible.

An active clock distribution network that uses only one feedback path per receiver (instead of two) is presented in [5.11]. For each clock region, the design uses a phase detector and a variable-delay line. The variable-delay line is in the forward path. The phase detector compares the arrival time of the clock pulses returning to the skew compensator to the arrival time of the reference pulses generated by the PLL. The compensator can adjust the variable-delay line until the arrival of the returning pulses coincides with the arrival of the reference pulses. Then, the returning pulses are synchronized with the reference pulses and the round-trip delay is known. The last step is to cut in half the delay of the variable-delay line. This is required to synchronize the pulses delivered to each region (as opposed to the returning pulses) with the reference clock. The biggest problem of the proposed skew compensation algorithm is that it can only be executed during reset. The reason is that the alignment process significantly perturbs the phase of the distributed clock signals. The algorithm also requires very linear variable-delay lines that have a high resolution and a wide range.

5.3.2 Distributed Skew Compensation

Sending the clock pulses back toward the source is not necessary for skew compensation. Not only does it waste power and routing resources, but it also doubles the clock jitter at the phase detector. A simpler and more practical alternative is distributed skew compensation.

The 650-MHz microprocessor described in [5.12] uses a two-spine clock distribution network. Each spine is driven by its own clock tree. As shown symbolically in Figure 5.19, the clock generated by the PLL gets distributed to the two variable-delay elements before reaching the spines. A phase detector located between the two spines compares the arrival time of the clock pulses. If the pulses arrive too soon at one of the spines, the delay of its variable-delay element is increased. Equivalently, the delay of the other variable-delay element can be decreased. When the active skew-compensation circuit is off, the skew between the two spines is 100 ps. Turning on the compensator reduces the skew to 15 ps. The circuit is practical. It is effectively able to compensate interconnect and device mismatches together with process, voltage, and temperature variations. The circuit does increase clock jitter however.

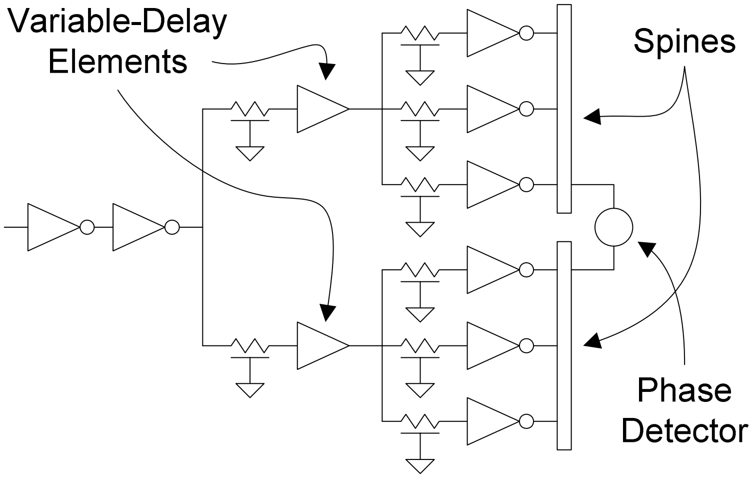


Figure 5.19: Clock distribution network described in [5.12].

The 64-bit microprocessor described in [5.13] runs at 800 MHz. Its clock distribution network is hierarchical. It uses a balanced tree to drive eight clusters of skew compensators. Each compensator drives a regional grid to which multiple local clock buffers are connected. The tree driving the eight clusters is unusual because it distributes two signals: a core clock and a reference clock.

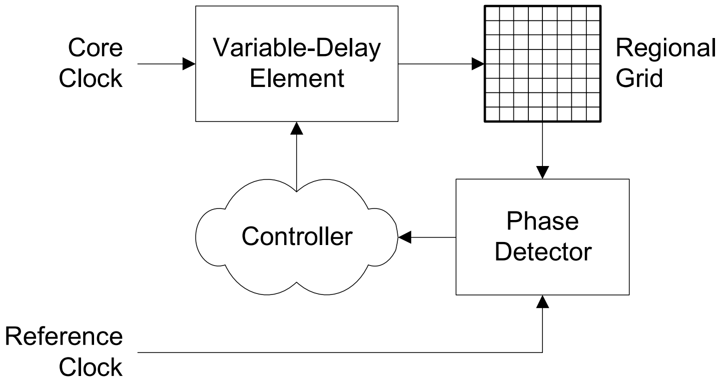


Figure 5.20: Skew compensator used in [5.13].

A block diagram for the skew compensators is shown in Figure 5.20. It is worth noting that the feedback clock coming from the regional grid is aligned with the reference clock, and not the core clock. The reason is that balancing the delay for each load driven by the core clock is relatively difficult, or at least more difficult than routing two signals instead of one. Because the reference clock is distributed with a lightly loaded and very symmetric network, it is easier to balance and it practically eliminates the skew caused by load mismatches. The design of the distribution network facilitates debugging and performance tuning since each compensator can be programmed to introduce intentional skew. Introducing intentional skew is a way of measuring the timing margin of the frequency-limiting circuits. Intentional skew can also be used to increase performance by stealing time from the non-critical paths to help the critical ones. The measured clock inaccuracy across the chip is 28 ps. Without skew compensation, the inaccuracy is estimated at 110 ps.

5.4 Clock Jitter Compensation Strategies

Clock distribution networks with multiple oscillators normally avoid, or at least significantly reduce, the need for routing the clock through long interconnects. They usually have special filters to attenuate jitter. They have the potential to solve the clock inaccuracy accumulation issue associated with conventional distribution strategies. However, distributing the clock using multiple oscillators introduces a number of

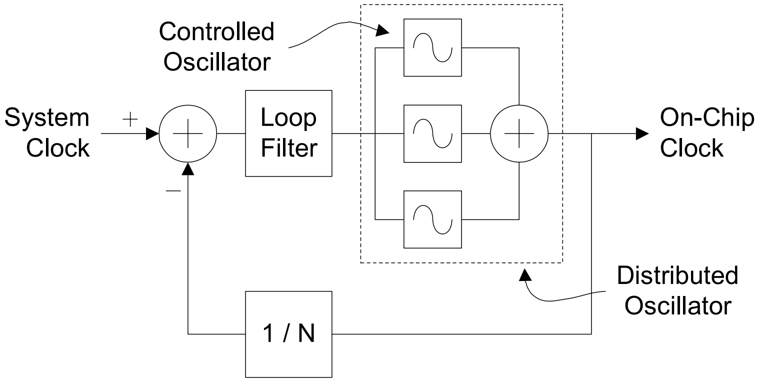


Figure 5.21: Centralized jitter filtering.

synchronization issues not present in conventional distribution networks. The most important are ensuring that the multiple oscillators have the same frequency and making sure that they are in phase.

The clock distribution networks discussed in this section are classified according to their jitter-filtering approach, which can be either centralized or distributed. The centralized approach corresponds to clock networks having a single attenuation filter. Networks with multiple filters correspond to a distributed strategy.

5.4.1 Centralized Jitter Filtering

The clock distribution networks that use centralized jitter filtering look like a conventional PLL having a distributed oscillator. Figure 5.21 shows such a distributed oscillator. The output of the loop filter actually controls multiple individual oscillators that are physically separated. Collectively, they form a single distributed entity. The signals generated by the individual oscillators are averaged to produce the clock delivered to the sequential elements.

Figure 5.22 shows the three-phase distributed oscillator prototype described in [5.14]. The prototype is not a complete clock distribution network. Its loop filter is missing. The circuit is topologically equivalent to a conventional ring oscillator having three inverters. It produces three clock signals phase shifted by 120 degrees. The oscillation frequency can be controlled by changing the delay of the inverters, but this can only be done by adjusting the supply voltage. This is a serious problem. In addition, the oscillator does not necessarily produce a clock having a 50% duty cycle. From a conventional circuit design perspective, a clock with three phases is useless. On the other hand, a balanced duty cycle is strongly needed. To get a duty cycle of 50%, the approach used in conventional PLLs is to make the oscillator run at twice the desired frequency. Then, a frequency divider generates the clock. Here, because the oscillator also distributes the clock, forcing it to run at twice the desired frequency would waste a significant amount of power. The strategy is incompatible with conventional DFT and DFD techniques because the clock cannot be bypassed and because the cycle time cannot be compressed or stretched.

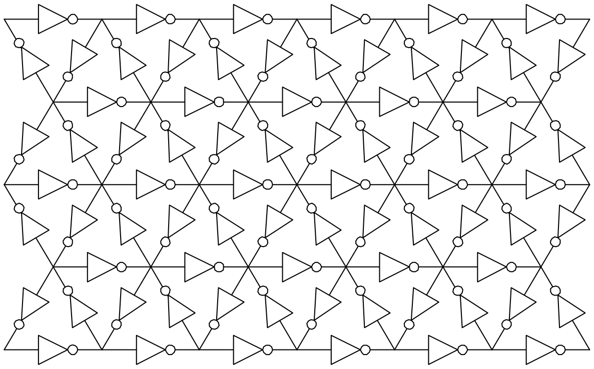


Figure 5.22: Distributed oscillator proposed in [5.14].

A complete clock distribution network using multiple voltage-controlled oscillators (VCOs) is described in [5.15]. The problem of distributing the clock from a single source is replaced by the problem of distributing the analog control voltage produced by the loop filter. Ideally, all the VCOs should receive the same control voltage and have the same frequency. However, if the VCOs receive different control voltages (perhaps because the control voltage is noisy), are not perfectly matched, or do not have the same temperature, then they will have different frequencies. This is very undesirable. To force them to oscillate at the same frequency, the VCO outputs are short circuited. If the phase of a VCO changes, the contention of the others tends to cancel it. Unfortunately, continuous contention wastes power. In addition, the effectiveness of short circuiting the VCOs to align their phase decreases rapidly as the length of the interconnects connecting them increases. Finally, adequately protecting the analog control voltage from noise is a major difficulty in practice. This technique is incompatible with conventional DFT and DFD techniques because the clock cannot be bypassed and because the cycle time cannot be stretched or compressed.

5.4.2 Distributed Jitter Filtering

A clock distribution network with multiple oscillators and phase detectors behaves like a fully distributed PLL. The phase processing is decentralized and multiple filters are required for jitter attenuation.

A clock distribution architecture using distributed jitter filtering is proposed in [5.16]. The jitter accumulation problem associated with conventional clock distribution strategies is avoided because the clock is regenerated just before reaching the sequential elements. The regeneration is performed using an array of PLLs. The issue is to keep them synchronized. For this, it is worth noting that the clock produced by a particular PLL only has to be synchronized to the clocks produced by its neighbors. It is shown in [5.16] that an oscillator cannot always use the average phase of its neighbors for synchronization. The reason is that phase averaging can produce an undesired equilibrium that is stable and in which not all the oscillators have the same phase. This condition is called mode locking. It can occur if two neighbors produce clocks with phases of equal magnitude and opposite sign. Mode locking can be avoided by ensuring that no loop exists in the propagation of the phase information. It is shown in [5.16] that it can also be avoided using special phase detectors. They must have a response that decreases monotonically beyond a phase difference of 90º.

The 16-PLL prototype described in [5.17] uses the second synchronization solution. Each PLL feeds a local clock region. The special phase detectors are inserted between adjacent regions to ensure that the clock produced by each PLL is aligned with the clock of its neighbors. The phase detectors are implemented using two pulse generators and an arbiter. Their output is a differential error current. As required, the gain of each phase detector is negative for large phase differences. Each PLL sums and filters the error signals coming from the adjacent regions. The loop filter is implemented with two differential amplifiers and a fairly small gate capacitor. Its output controls the VCO producing the local clock. The oscillation frequency of the PLL array is controlled by connecting an external system clock to one of the phase detectors. The system clock frequency is not multiplied.

Although the distributed synchronous clocking strategy proposed in [5.17] promises to significantly reduce the jitter-accumulation problem associated with conventional clock distribution networks, it is incompatible with conventional DFT and DFD techniques. There is no infrastructure to bypass the PLLs and no infrastructure to stretch or compress the cycle time. The prototype also suffers from the complexity of the transfer function of its phase detectors. This complexity makes it hard to predict the clock inaccuracy and to ensure the network stability under different voltage, process, and temperature conditions. That unpredictability tends to force pessimistic assumptions for timing analysis. Finally, the prototype also suffers from its analog feedback mechanism. It requires the transmission of analog signals over relatively long distances to synchronize the PLLs. The problem is that most high-performance chips produce a lot of electrical noise, in particular interlevel coupling noise, that can affect the feedback signals. Noisy feedback signals are an issue because of the limited area available for each loop filter. The filters have to be able to remove enough noise, which is hard without large capacitors. This makes the stability of the clock network even more difficult to analyze and guarantee.

5.5 Capacitive Interlevel Coupling Noise

Interconnect noise has traditionally been, and continues to be, a significant issue for the design of high-frequency clock distribution networks. Typically, interconnect noise is addressed by determining the worst possible capacitive coupling noise that can be induced on a victim line by its parallel neighbors. In [5.18], the worst-case noise is simulated using an RC transmission line model that includes the timing of the attackers. In [5.19], inductance is taken into account and an expression for the peak crosstalk voltage between interconnects on the same layer is derived. The conductors in the layers adjacent to the victim's layer are assumed quiet and are assumed to behave as virtual metal planes. These two assumptions are commonly used during parasitic extraction and for static timing analysis [5.20].

The metal plane approximation yields an accurate interlayer capacitance, provided than the metal density of the adjacent layers exceeds 33% [5.21].

However, the assumption that each plane is quiet is sometimes clearly wrong. For example, it does not apply to the signals orthogonally routed above or below one of the wide dynamic buses of the Itanium 2 microprocessor [5.22]. The bits of such a bus can be sufficiently correlated to switch simultaneously and in the same direction. The bus can therefore inject a considerable amount of noise on the wires located above and below it, especially if the width of the bus is a significant fraction of the length of the wires under attack. Assuming that all interconnects perpendicular to a victim are quiet is therefore unrealistic and optimistic.

In practice, it is often desirable to perform conservative timing analysis, for instance to avoid min-delay races or to ensure sufficient yield at a certain frequency target. However, if the victim is a long wire, it is excessively pessimistic to suppose that the switching of all of its orthogonal neighbors is simultaneous and in the same direction.

The problem is that for long wires, in particular for the ones used for global clock distribution, the error due to the quiet plane assumption is unclear. It has never been rigorously analyzed or quantified. Given that it is an optimistic assumption, it is *a priori* not obvious if it is appropriate when analyzing the timing of clock signals.

This lack of understanding makes jitter estimation much harder and, therefore, creates a timing risk. For the Itanium 2 microprocessor described in [5.23], differential clocking is used to reduce the vulnerability to common-mode coupling noise, partly because of this uncertainty.

This section examines and tests the limits of the quiet metal plane assumption when the individual conductors in the adjacent layers are toggling independently. It rigorously shows that the quiet metal plane assumption is then statistically very good. Clarifying the

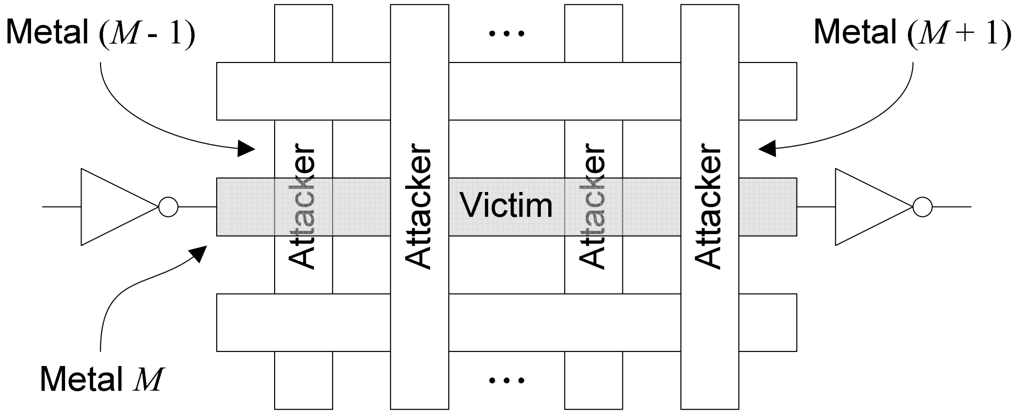


Figure 5.23: Interlevel coupling noise.

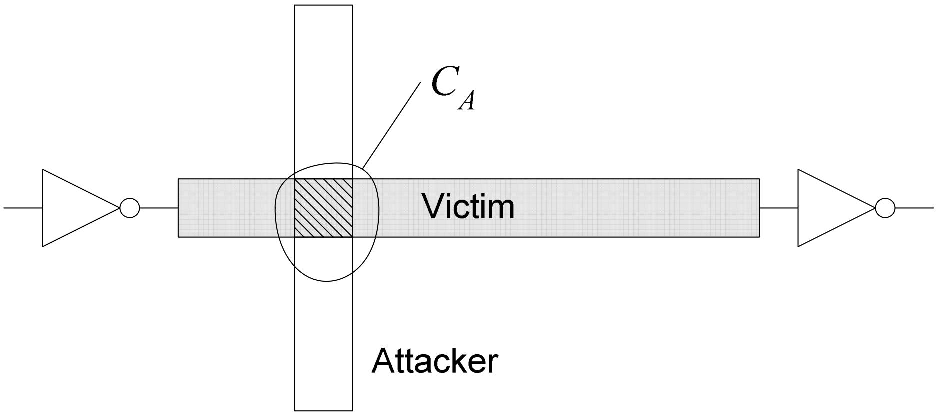


Figure 5.24: Overlap capacitance.

circumstances where the quiet metal plane assumption can be safely used is of tremendous practical interest for high-frequency microprocessor design. It provides the guarantee of conservative timing analysis without excessive pessimism.

The interlevel coupling noise present at the far end of the victim line shown in Figure 5.23 is analyzed when a large number of perpendicular attackers are randomly switching. Because the attackers are perpendicular, the coupling capacitance between any single one of them and the victim corresponds to a very small overlap region, as shown in Figure 5.24. Each attacker is modeled as a Markov chain and the victim is modeled as an RLC transmission line.

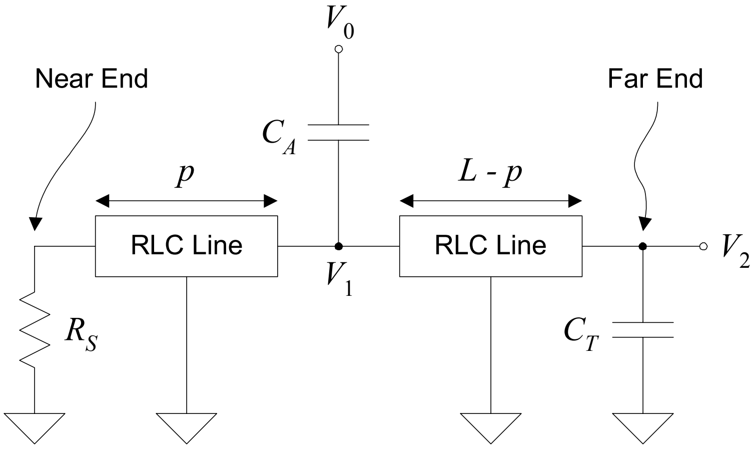


Figure 5.25: Model for the victim line.

The effect of a particular attacker on the noise at the far end of the victim is modeled as shown in Figure 5.25. The capacitance between the attacker and the victim is C_A. The source resistance R_S represents the effective resistance of the circuit driving the victim and C_T is the load capacitance that terminates it. The unit-length resistance, inductance, and capacitance of the victim line are r, l, and c. The length of the victim line is L and p is the position of the attacker. The pitch of each attacker (i.e., its width and space) is λ.

The victim line is modeled as a linear system with multiple inputs (one per attacker) and one output. The power spectral density of the noise at V_2 is obtained in three steps. First, the power spectral density of an individual attacker switching randomly at V_0 is analyzed. Then, its transfer function H between V_0 and V_2 is derived. Since the system has multiple inputs, each attacker has its own transfer function. Given the power spectral density of the attacker at V_0 and the transfer function corresponding to its position, its contribution to the noise at V_2 is easy to compute. Finally, the total power spectral density of the noise at V_2 is determined by superposing the contribution of each individual attacker.

5.5.1 Power Spectral Density for an Attacker Switching Randomly

During any given clock cycle, the probability that a particular attacker switches is given by its activity factor a. Each attacker is modeled by a Markov chain producing the discrete-time sequence $d(n)$, as shown in Figure 5.26. Pulse amplitude modulation (PAM)

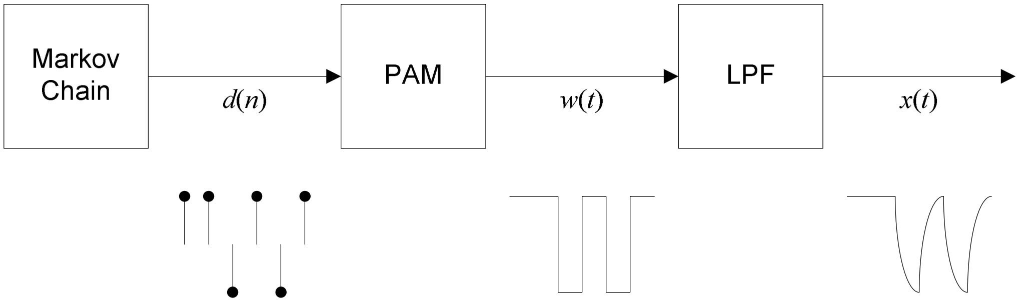


Figure 5.26: Model for the attacker waveforms.

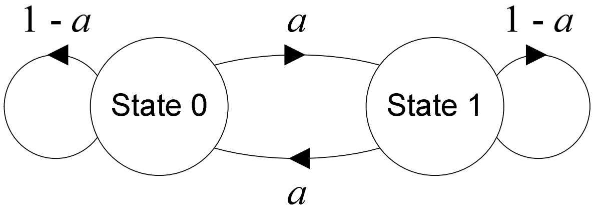


Figure 5.27: Markov chain model for the switching activity of attackers.

is used to convert $d(n)$ to a continuous-time square wave $w(t)$. The square wave is then applied to a low-pass filter (LPF) to produce the attacker waveform $x(t)$. The purpose of the low-pass filter is to give a non-zero transition time to the instantaneous transitions of $w(t)$. This makes $x(t)$ a more realistic integrated circuit signal. This section derives the power spectrum of $x(t)$ using the techniques presented in [5.24] and [5.25].

5.5.1.1 Markov Model

The first step is to determine the power spectral density of the discrete-time data stream $d(n)$. The state transition diagram of the Markov chain is shown in Figure 5.27. The diagram has two states (0 and 1). The binary outputs associated with states 0 and 1 are chosen to be –½ and +½, respectively. This makes $d(n)$ a zero-mean random process with an amplitude of one. The initial state of the Markov chain is randomly selected.

The power spectral density S_D of the data stream is the Z-transform of its autocorrelation function R_D. By definition, the autocorrelation function is the expected value of:

$$R_D(m,n) = E[d(m)d(m+n)] = E[f(\Psi_m)f(\Psi_{m+n})] \tag{5.16}$$

where Ψ_k is the state of the chain at time k. Ψ_k is an element of $\{0, 1\}$. The function f gives the output of the chain when it is in a particular state:

$$f(\Psi) = \Psi - \frac{1}{2} \tag{5.17}$$

Since the Markov chain is stationary, the autocorrelation function depends only on the absolute time difference between samples. With $m = 0$, R_D thus becomes:

$$R_D(n) = E[f(\Psi_0)f(\Psi_{|n|})] \tag{5.18}$$

Because there are two possible states at time 0 and two possible states at time n, there are four possible outcomes to consider:

$$R_D(n) = \sum_{i=0}^{1}\sum_{j=0}^{1} f(i)f(j)p_{0,|n|}(i,j) \tag{5.19}$$

where $p_{0,n}(i, j)$ is the probability of being in state i at time 0 and in state j at time n.

By symmetry, $p_{0,n}(0, 0) = p_{0,n}(1, 1)$ and $p_{0,n}(0, 1) = p_{0,n}(1, 0)$. Therefore,

$$R_D(n) = f(0)f(0)p_{0,n}(0,0) + f(0)f(1)p_{0,n}(0,1) + f(1)f(0)p_{0,n}(1,0) + f(1)f(1)p_{0,n}(1,1) \tag{5.20}$$

can be simplified to

$$\begin{aligned} R_D(n) &= (f(0)f(0) + f(1)f(1))p_{0,n}(0,0) + (f(1)f(0) + f(0)f(1))p_{0,n}(1,0) \\ &= \frac{1}{2}p_{0,n}(0,0) - \frac{1}{2}p_{0,n}(1,0) \end{aligned} \tag{5.21}$$

By Bayes' rule,

$$p_{0,n}(i, j) = p_{n|0}(j \mid i)p_0(i) \tag{5.22}$$

where $p_{n|0}(j \mid i)$ is the probability of being in state j at time n given that the state at time 0 is i. The probability of actually starting in state i at time 0 is $p_0(i)$.

Here, $p_0(i) = ½$ since the two initial states are equally likely. The autocorrelation function therefore becomes:

$$R_D(n) = \frac{1}{2}p_{|n|\,|\,0}(0 \mid 0)p_0(0) - \frac{1}{2}p_{|n|\,|\,0}(0 \mid 1)p_0(1) = \frac{1}{4}p_{|n|\,|\,0}(0 \mid 0) - \frac{1}{4}p_{|n|\,|\,0}(0 \mid 1) \tag{5.23}$$

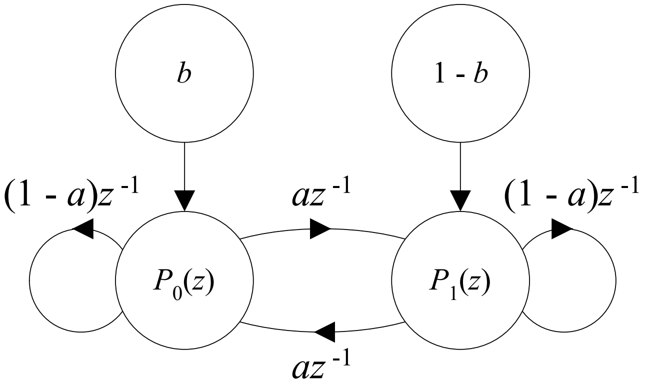


Figure 5.28: Signal flow diagram given arbitrary initial state probabilities.

The Z-transform of $R_D(n)$ is:

$$S_D(z) = \sum_{n=-\infty}^{\infty} R_D(n) z^{-n} = \sum_{n=0}^{\infty} R_D(n) z^{-n} + \sum_{n=-\infty}^{0} R_D(n) z^{-n} - R_D(0) \tag{5.24}$$

Since $R_D(n) = R_D(-n)$,

$$S_D(z) = \sum_{n=0}^{\infty} R_D(n) z^{-n} + \sum_{n=0}^{\infty} R_D(n) z^{n} - R_D(0) \tag{5.25}$$

It is convenient to express $S_D(z)$ as:

$$S_D(z) = S_D^+(z) + S_D^+(z^{-1}) - S_D^+(\infty) \tag{5.26}$$

where

$$S_D^+(z) = \sum_{n=0}^{\infty} R_D(n) z^{-n} \tag{5.27}$$

Substituting (5.23) in (5.27) yields,

$$S_D^+(z) = \frac{1}{4} P_{0|0}(z) - \frac{1}{4} P_{0|1}(z) \tag{5.28}$$

where $P_{0|0}(z)$ is the Z-transform of the sequence describing the probability of being in state 0 at a particular time, given that the initial state was 0. Similarly, $P_{0|1}(z)$ is the Z-transform of the sequence describing the probability of being in state 0 at a particular time, given that the initial state was 1.

The equations describing the state probabilities in the Z-domain are obtained from the signal flow graph of Figure 5.28:

$$\begin{aligned} P_0(z) &= b + (1-a)z^{-1}P_0(z) + az^{-1}P_1(z) \\ P_1(z) &= (1-b) + (1-a)z^{-1}P_1(z) + az^{-1}P_0(z) \end{aligned} \tag{5.29}$$

Solving for $P_0(z)$ yields:

$$P_0(z) = \frac{b + (a-b)z^{-1}}{1 - 2(1-a)z^{-1} + (1-2a)z^{-2}} \tag{5.30}$$

$P_0(z)$ is the Z-transform of the sequence describing the probability of being in state 0 at a particular time, given some initial state probabilities. The initial state probabilities are defined by b.

Setting b to 1 gives:

$$P_{0|0}(z) = P_0(z)\big|_{b=1} = \frac{1 + (a-1)z^{-1}}{1 - 2(1-a)z^{-1} + (1-2a)z^{-2}} \tag{5.31}$$

where $P_{0|0}(z)$ is the Z-transform of the sequence describing the probability of being in state 0 at a particular time, given that the initial state was 0.

Similarly, setting b to 0 yields the Z-transform of the sequence describing the probability of being in state 0 at a particular time, given that the initial state was 1:

$$P_{0|1}(z) = P_0(z)\big|_{b=0} = \frac{az^{-1}}{1 - 2(1-a)z^{-1} + (1-2a)z^{-2}} \tag{5.32}$$

Substituting the results in (5.28) gives:

$$S_D^+(z) = \frac{1}{4}\frac{1 + (a-1)z^{-1} - az^{-1}}{1 - 2(1-a)z^{-1} + (1-2a)z^{-2}} = \frac{1}{4}\frac{1 - z^{-1}}{1 - 2(1-a)z^{-1} + (1-2a)z^{-2}} \tag{5.33}$$

which can be used to compute $S(z)$ using (5.26):

$$\begin{aligned} S_D(z) &= \frac{1}{4}\frac{1 - z^{-1}}{1 - 2(1-a)z^{-1} + (1-2a)z^{-2}} + \frac{1}{4}\frac{1 - z}{1 - 2(1-a)z + (1-2a)z^{2}} - \frac{1}{4} \\ &= \frac{a(1-a)}{(1-(1-2a)z^{-1})(1-(1-2a)z)} \end{aligned} \tag{5.34}$$

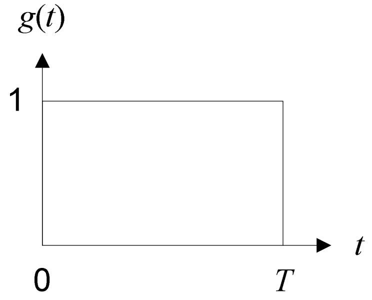


Figure 5.29: Pulse used for modulation.

The autocorrelation function $R_D(n)$ of the data sequence is given by the inverse Z-transform of $S_D(z)$. The result is:

$$R_D(n) = \frac{1}{4}(1-2a)^{|n|} \tag{5.35}$$

Evaluating $S_D(z)$ on the unit circle gives the power spectral density of the discrete time data stream.

$$S_D(e^{j2\pi fT}) = \frac{a(1-a)}{(1-(1-2a)e^{-j2\pi fT})(1-(1-2a)e^{j2\pi fT})} = \frac{a(1-a)}{1+(1-2a)^2-(1-2a)(e^{j2\pi fT}+e^{-j2\pi fT})} \tag{5.36}$$

After further simplification, the desired result is obtained:

$$S_D(e^{j2\pi fT}) = \frac{a(1-a)}{1+(1-2a)^2-2(1-2a)\cos(2\pi fT)} \tag{5.37}$$

The power spectral density of the discrete time data stream $d(n)$ is clearly determined by the switching probability a of the attackers.

5.5.1.2 Pulse Amplitude Modulation

The continuous-time data signal $w(t)$ is constructed by modulating the amplitude of a pulse according to the discrete-time sequence $d(n)$. The pulse is shown in Figure 5.29. It is defined as having a unit amplitude during the time interval [0, T]. Elsewhere, its amplitude is zero. Mathematically,

$$g(t) = u(t)u(T-t) \tag{5.38}$$

T represents the cycle time of the system.

The result of the modulation is:

$$w(t) = \sum_{n=-\infty}^{\infty} d(n) g(t - nT + \Theta) \tag{5.39}$$

where Θ is a random variable uniformly distributed over [0, T] that defines when in the cycle switching can occur.

The power spectral density of *w*(*t*) is given by [5.25]:

$$S_W(f) = \frac{1}{T} \left| G(f) \right|^2 S_D(e^{j2\pi fT}) \tag{5.40}$$

where *G*(*f*) is the Fourier transform of the rectangular pulse:

$$G(f) = \int_{-\infty}^{\infty} g(t) e^{-j2\pi ft} dt = \int_{0}^{T} e^{-j2\pi ft} dt = \frac{\sin(2\pi fT)}{2\pi f} + j \frac{\cos(2\pi fT) - 1}{2\pi f} \tag{5.41}$$

Therefore,

$$\left| G(f) \right|^2 = \left(\frac{\sin(2\pi fT)}{2\pi f} \right)^2 + \left(\frac{\cos(2\pi fT) - 1}{2\pi f} \right)^2 = \frac{2 - 2\cos(2\pi fT)}{(2\pi f)^2} \tag{5.42}$$

Finally,

$$S_W(f) = \frac{1}{T} \frac{2 - 2\cos(2\pi fT)}{(2\pi f)^2} \frac{a(1-a)}{1 + (1-2a)^2 - 2(1-2a)\cos(2\pi fT)} \tag{5.43}$$

It is straightforward to numerically verify that $S_W(f)$ gives the correct power spectral density for *w*(*t*). With *Simulink*, a dynamic system simulator for *MATLAB*, the PAM signal can be constructed using a state element sampling a random variable. Figure 5.30 shows that the power spectral density derived analytically matches the simulated one.

5.5.1.3 Low-Pass Filtering

To produce the attacker waveform actually coupled to the victim, *w*(*t*) is passed through a first-order low-pass filter *C* having a gain *V* and a time constant τ:

$$C(s) = \frac{V}{s\tau + 1} \tag{5.44}$$

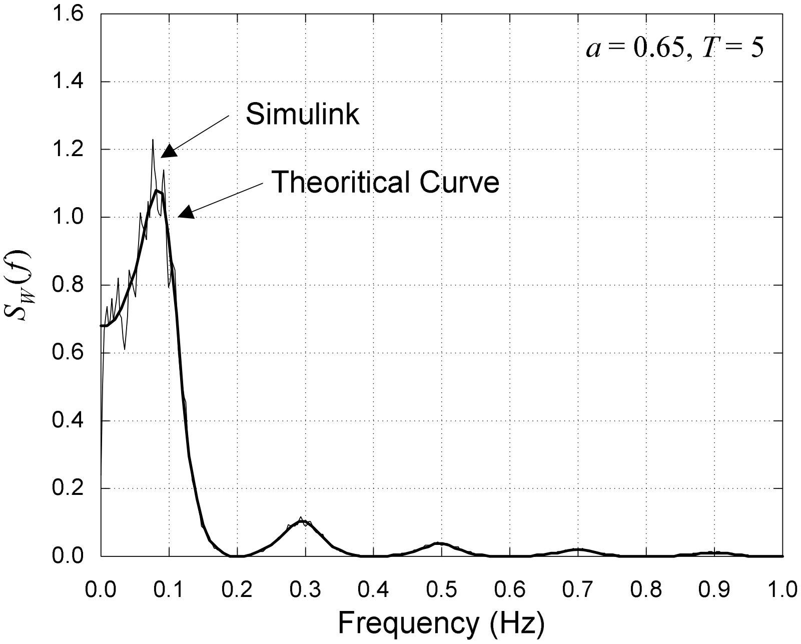


Figure 5.30: Power spectral density for the PAM waveform.

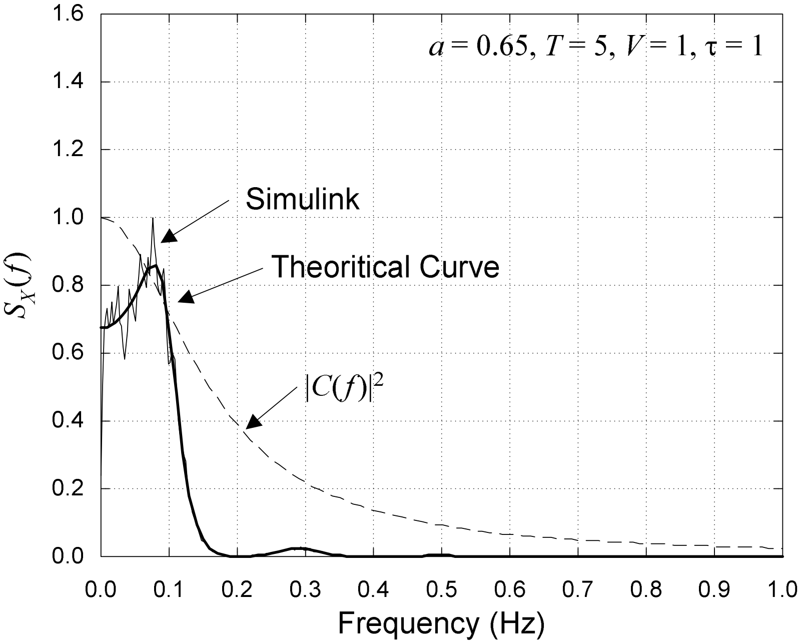


Figure 5.31: Power spectral density for the attacker waveform (after filtering).

The power spectral density of the filter output $x(t)$ is given by [5.24]:

$$S_X(f) = S_W(f)\left|C(j2\pi f)\right|^2 = S_W(f)\frac{V^2}{(2\pi f\tau)^2+1} \tag{5.45}$$

It is worth noting that $x(t)$ is more realistic than a piecewise-linear waveform. Multiple switching events per cycle can be modeled by making T smaller than the actual clock cycle and by adjusting a accordingly.

Substituting the expression for $S_W(f)$ derived earlier yields the power spectral density of the attackers as a function of their cycle time T, switching probability a, voltage V, and time constant τ:

$$S_X(f) = \frac{1}{T}\frac{2-2\cos(2\pi fT)}{(2\pi f)^2}\frac{a(1-a)}{1+(1-2a)^2-2(1-2a)\cos(2\pi fT)}\frac{V^2}{(2\pi f\tau)^2+1} \tag{5.46}$$

The power spectral density for $x(t)$ can also be verified numerically using *Simulink*. The results are shown in Figure 5.31. For convenience, the curve showing the effect of the low-pass filter is also plotted. As before, the theoretical power spectral density matches the simulated one.

The impact of the switching probability a on $S_W(f)$ and $S_X(f)$ is shown in Figure 5.32 and in Figure 5.33. Both spectrums are zero at multiples of 1 / T in frequency. The spectrum of $x(t)$ contains almost no power for $|f| > 1 / T$.

5.5.2 Noise Transfer Function

The victim interconnect is modeled as an ideal transmission line having a unit-length resistance r, inductance l, and capacitance c. The propagation constant γ of the line is frequency-dependant. It is given in the Laplace domain by:

$$\gamma = \sqrt{(r+sl)sc} \tag{5.47}$$

The attenuation constant α and the phase constant β are:

$$\begin{aligned}\alpha &= \mathrm{Re}(\gamma)\\ \beta &= \mathrm{Im}(\gamma)\end{aligned} \tag{5.48}$$

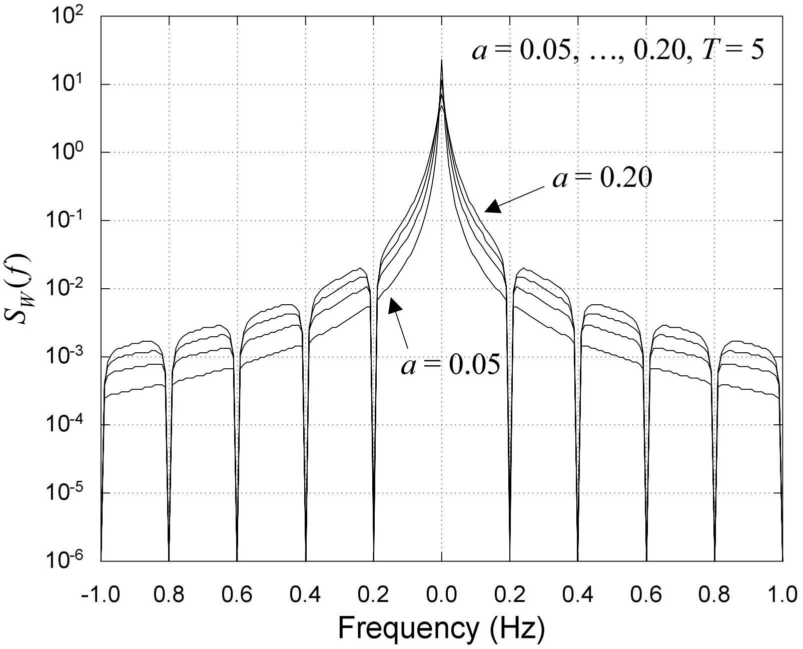


Figure 5.32: Impact of a on the power spectral density of the PAM waveform.

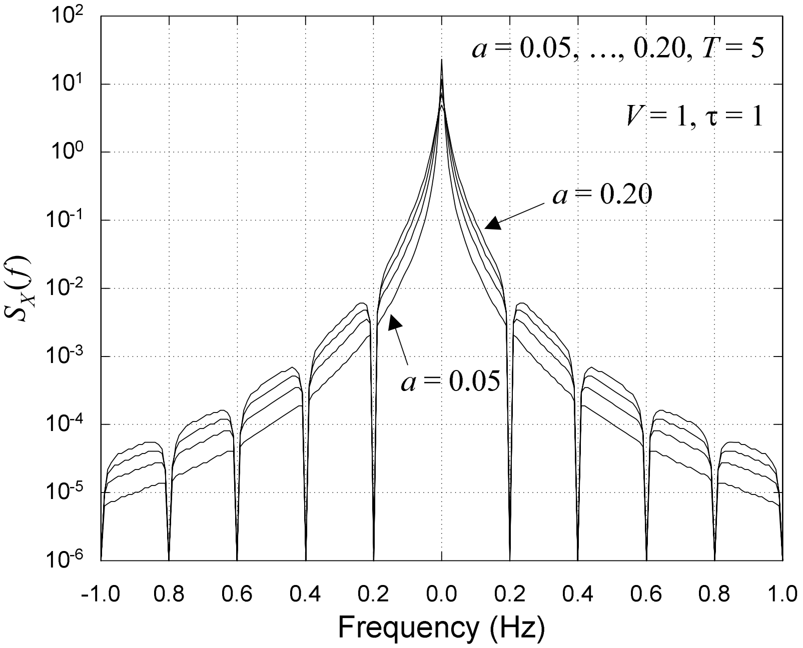


Figure 5.33: Impact of a on the power spectral density of the attacker waveform.

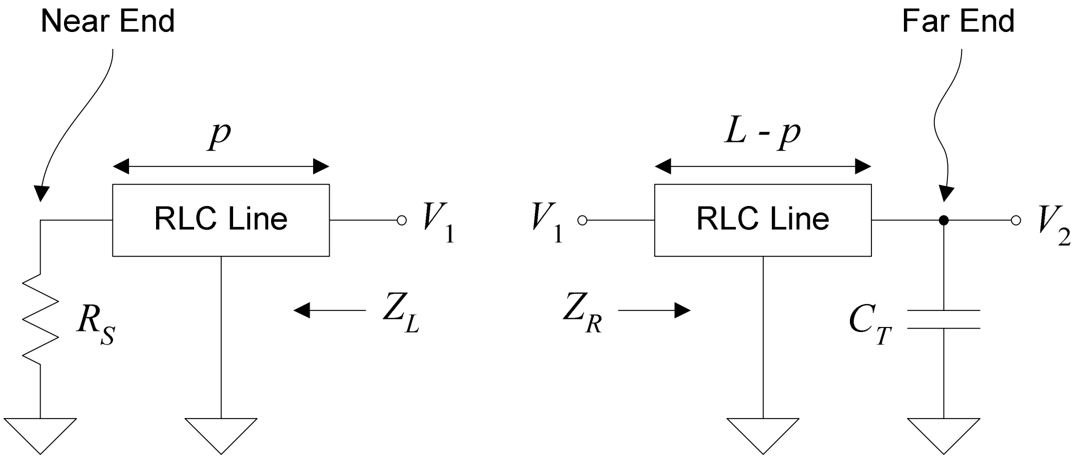


Figure 5.34: Input impedance of left and right transmission lines.

The characteristic impedance of the line is:

$$Z_0 = \sqrt{\frac{r+sl}{sc}} = R_0 + jX_0 \tag{5.49}$$

where $R_0 = \mathrm{Re}(Z_0)$ is the line resistance and $X_0 = \mathrm{Im}(Z_0)$ is the line reactance.

In Figure 5.25, the connection point of the attacker splits the victim in two parallel transmission lines. They both share the same propagation constant, but do not have the same length. They are also terminated differently. This is shown in Figure 5.34.

The input impedance Z_L of the transmission line going left is [5.26]:

$$Z_L = Z_0 \frac{R_S \cosh(\gamma p) + Z_0 \sinh(\gamma p)}{R_S \sinh(\gamma p) + Z_0 \cosh(\gamma p)} \tag{5.50}$$

For the other transmission line, the input impedance is:

$$\begin{aligned} Z_R &= Z_0 \frac{\left(\frac{1}{C_T s}\right)\cosh(\gamma L - \gamma p) + Z_0 \sinh(\gamma L - \gamma p)}{\left(\frac{1}{C_T s}\right)\sinh(\gamma L - \gamma p) + Z_0 \cosh(\gamma L - \gamma p)} \\ &= Z_0 \frac{\cosh(\gamma L - \gamma p) + Z_0 C_T s \sinh(\gamma L - \gamma p)}{\sinh(\gamma L - \gamma p) + Z_0 C_T s \cosh(\gamma L - \gamma p)} \end{aligned} \tag{5.51}$$

Its transfer function is:

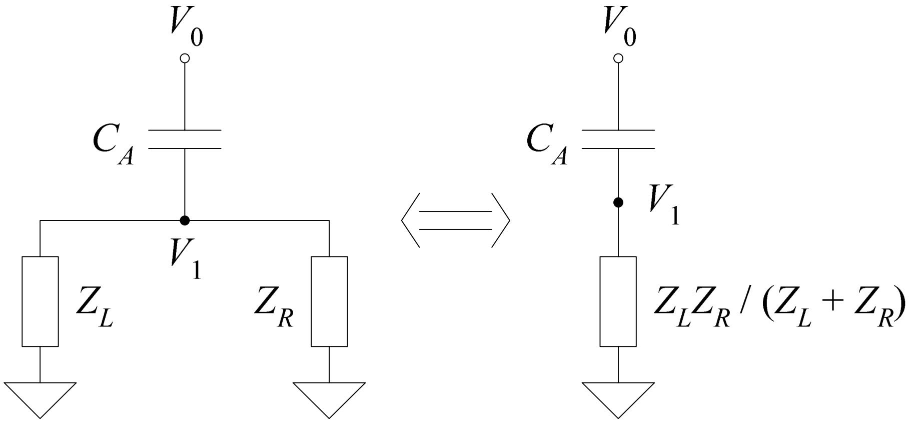


Figure 5.35: Equivalent circuit for the victim line.

$$\frac{V_2}{V_1} = \frac{1}{\cosh(\gamma L - \gamma p) + Z_0 C_T s \sinh(\gamma L - \gamma p)} \tag{5.52}$$

It is convenient to express Z_L and Z_R as:

$$\begin{aligned} Z_L &= Z_0 \frac{A_L}{B_L} \\ Z_R &= Z_0 \frac{A_R}{B_R} \end{aligned} \tag{5.53}$$

where

$$\begin{aligned} A_L &= R_S \cosh(\gamma p) + Z_0 \sinh(\gamma p) \\ B_L &= R_S \sinh(\gamma p) + Z_0 \cosh(\gamma p) \end{aligned} \tag{5.54}$$

and

$$\begin{aligned} A_R &= \cosh(\gamma L - \gamma p) + Z_0 C_T s \sinh(\gamma L - \gamma p) \\ B_R &= \sinh(\gamma L - \gamma p) + Z_0 C_T s \cosh(\gamma L - \gamma p) \end{aligned} \tag{5.55}$$

The transfer function of the transmission line going right becomes:

$$\frac{V_2}{V_1} = \frac{1}{A_R} \tag{5.56}$$

From the perspective of the attacker, the circuit of Figure 5.25 is equivalent to the circuit of Figure 5.35. The transfer function between V_0 and V_1 is the one of a voltage divider:

$$\frac{V_1}{V_0} = \frac{\frac{Z_L Z_R}{Z_L + Z_R}}{\frac{1}{C_A s} + \frac{Z_L Z_R}{Z_L + Z_R}} = \frac{Z_L Z_R C_A s}{Z_L + Z_R + Z_L Z_R C_A s} \tag{5.57}$$

In terms of A_L, B_L, A_R, B_R,

$$\frac{V_1}{V_0} = \frac{A_L A_R Z_0 C_A s}{A_L B_R + A_R B_L + A_L A_R Z_0 C_A s} \tag{5.58}$$

Multiplying by V_2/V_1 yields H, the transfer function between V_0 and V_2:

$$H(s,p,C_A) = \frac{V_2}{V_0} = \frac{V_2}{V_1}\frac{V_1}{V_0} = \frac{1}{A_R}\frac{A_L A_R Z_0 C_A s}{A_L B_R + A_R B_L + A_L A_R Z_0 C_A s} \tag{5.59}$$

Clearly, the transfer function changes with p, the position of the attacker. It is mathematically convenient for the rest of the derivation to factor H as follows:

$$H(s,p,C_A) = F(s,C_A)G(s,p) \tag{5.60}$$

The first factorization step is to show that $A_L\ B_R + A_R\ B_L$ is independent of p. Using the following hyperbolic identities,

$$\begin{aligned} \cosh(\gamma L - \gamma p) &= \cosh(\gamma L)\cosh(\gamma p) - \sinh(\gamma L)\sinh(\gamma p) \\ \sinh(\gamma L - \gamma p) &= \sinh(\gamma L)\cosh(\gamma p) - \cosh(\gamma L)\sinh(\gamma p) \end{aligned} \tag{5.61}$$

it is possible to express A_R and B_R as:

$$\begin{aligned} A_R &= \cosh(\gamma p)(\cosh(\gamma L) + Z_0 C_T s \sinh(\gamma L)) - \sinh(\gamma p)(\sinh(\gamma L) + Z_0 C_T s \cosh(\gamma L)) \\ B_R &= \cosh(\gamma p)(\sinh(\gamma L) + Z_0 C_T s \cosh(\gamma L)) - \sinh(\gamma p)(\cosh(\gamma L) + Z_0 C_T s \sinh(\gamma L)) \end{aligned} \tag{5.62}$$

Then,

$$\begin{aligned} A_L B_R = {} & R_S \cosh^2(\gamma p)(\sinh(\gamma L) + Z_0 C_T s \cosh(\gamma L)) \\ & - R_S \cosh(\gamma p)\sinh(\gamma p)(\cosh(\gamma L) + Z_0 C_T s \sinh(\gamma L)) \\ & + Z_0 \sinh(\gamma p)\cosh(\gamma p)(\sinh(\gamma L) + Z_0 C_T s \cosh(\gamma L)) \\ & - Z_0 \sinh^2(\gamma p)(\cosh(\gamma L) + Z_0 C_T s \sinh(\gamma L)) \end{aligned} \tag{5.63}$$

$$\begin{aligned} A_R B_L = {} & R_S \sinh(\gamma p)\cosh(\gamma p)(\cosh(\gamma L) + Z_0 C_T s \sinh(\gamma L)) \\ & - R_S \sinh^2(\gamma p)(\sinh(\gamma L) + Z_0 C_T s \cosh(\gamma L)) \\ & + Z_0 \cosh^2(\gamma p)(\cosh(\gamma L) + Z_0 C_T s \sinh(\gamma L)) \\ & - Z_0 \cosh(\gamma p)\sinh(\gamma p)(\sinh(\gamma L) + Z_0 C_T s \cosh(\gamma L)) \end{aligned} \tag{5.64}$$

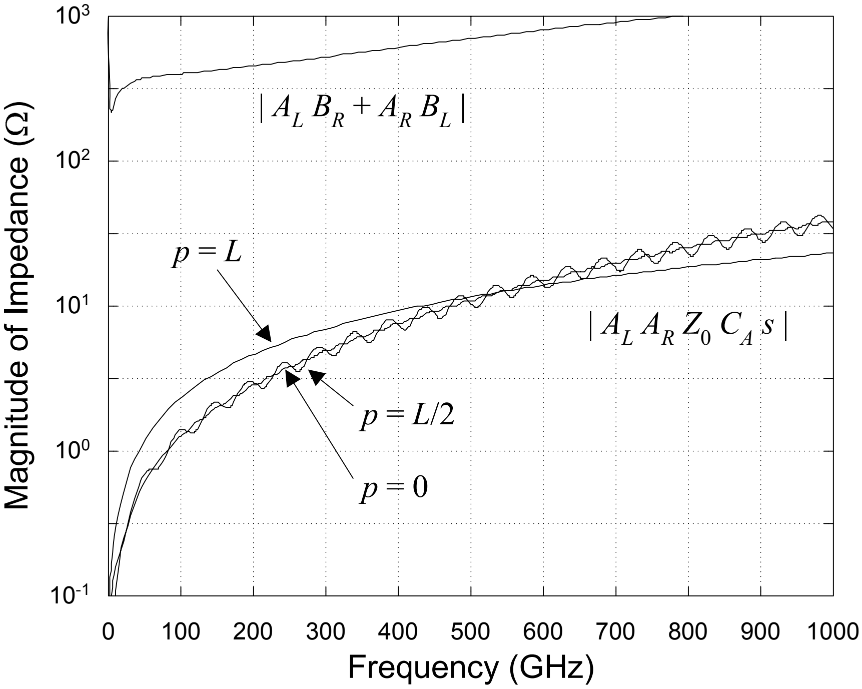


Figure 5.36: Comparison of impedance magnitude responses.

Therefore,

$$\begin{aligned} A_L B_R + A_R B_L = {} & R_S \cosh^2(\gamma p)(\sinh(\gamma L) + Z_0 C_T s \cosh(\gamma L)) \\ & - R_S \sinh^2(\gamma p)(\sinh(\gamma L) + Z_0 C_T s \cosh(\gamma L)) \\ & + Z_0 \cosh^2(\gamma p)(\cosh(\gamma L) + Z_0 C_T s \sinh(\gamma L)) \\ & - Z_0 \sinh^2(\gamma p)(\cosh(\gamma L) + Z_0 C_T s \sinh(\gamma L)) \end{aligned} \tag{5.65}$$

Using the fact that,

$$\cosh^2(\gamma p) - \sinh^2(\gamma p) = 1 \tag{5.66}$$

$A_L\,B_R + A_R\,B_L$ becomes:

$$A_L B_R + A_R B_L = R_S(\sinh(\gamma L) + Z_0 C_T s \cosh(\gamma L)) + Z_0(\cosh(\gamma L) + Z_0 C_T s \sinh(\gamma L)) \tag{5.67}$$

or

$$A_L B_R + A_R B_L = Z_0(R_S C_T s + 1)\cosh(\gamma L) + (Z_0^2 C_T s + R_S)\sinh(\gamma L) \tag{5.68}$$

which is clearly *independent* of p, the position of the attacker.

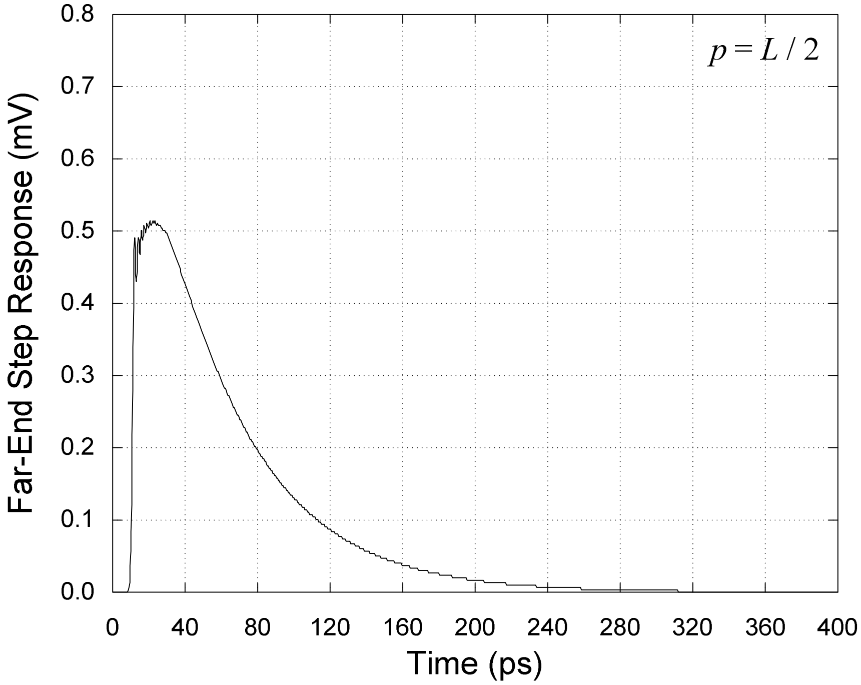


Figure 5.37: Far-end voltage waveform in response to a step input.

The next factorization step is to show numerically that $|A_L\, A_R\, Z_0\, C_A\, s|$ is typically negligible compared to $|A_L\, B_R + A_R\, B_L|$. This is done in Figure 5.36 for a 2000-μm M5 interconnect with r = 0.10 Ω/μm, l = 0.50 pH/μm, and c = 0.20 fF/μm. The resistance of the driver holding the victim is R_S = 50 Ω. The line is terminated by a 10-fF load. The coupling capacitance C_A between the attacker and the victim is taken to be 0.20 fF. Between 0 and 1000 GHz, $|A_L\, A_R\, Z_0\, C_A\, s|$ is at least an order of magnitude smaller than $|A_L\, B_R + A_R\, B_L|$. Below 100 GHz, $|A_L\, A_R\, Z_0\, C_A\, s|$ is 100 times smaller.

Therefore,

$$H(s, p, C_A) = \frac{A_L Z_0 C_A s}{A_L B_R + A_R B_L + A_L A_R Z_0 C_A s} \approx \frac{A_L Z_0 C_A s}{A_L B_R + A_R B_L} \tag{5.69}$$

Substituting expression (5.68) yields:

$$H(s, p, C_A) \approx \frac{A_L Z_0 C_A s}{Z_0 (R_S C_T s + 1)\cosh(\gamma L) + (Z_0^2 C_T s + R_S)\sinh(\gamma L)} \tag{5.70}$$

The final factorization $H(s, p) = F(s, C_A)\, G(s, p)$ is accomplished by setting:

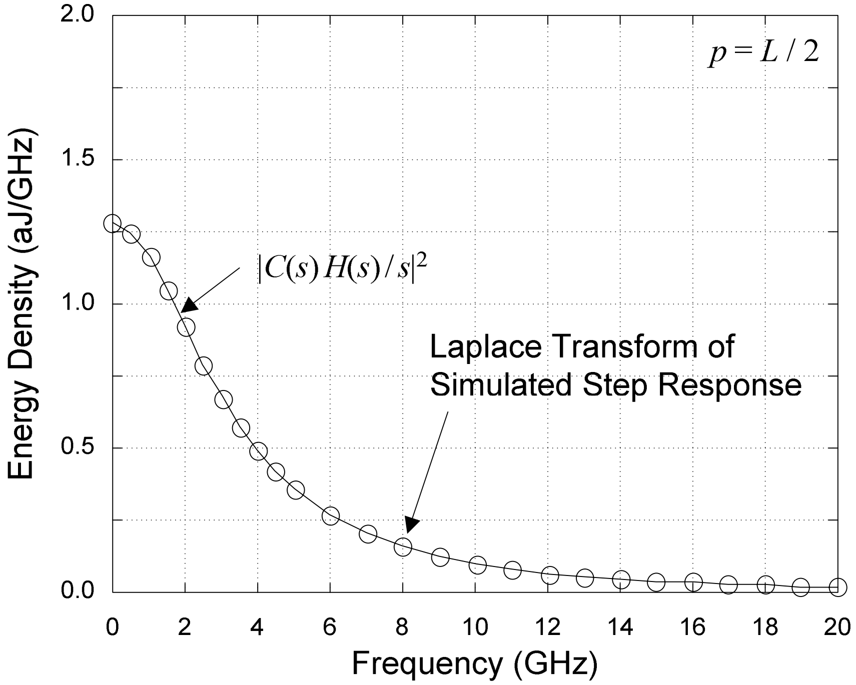


Figure 5.38: Energy spectrum of far-end voltage in response to a step input.

$$F(s, C_A) = \frac{Z_0 C_A s}{Z_0 (R_S C_T s + 1)\cosh(\gamma L) + (Z_0^2 C_T s + R_S)\sinh(\gamma L)} \tag{5.71}$$

and

$$G(s, p) = R_S \cosh(\gamma p) + Z_0 \sinh(\gamma p) \tag{5.72}$$

Figure 5.37 shows the far-end voltage waveform produced when a single attacker switches up. The waveform is simulated using 50 RLC sections to accurately approximate the victim transmission line. The attacker is located at $p = L / 2$. The waveform represents the step response of the system formed by cascading the low-pass filter $C(s)$ with the noise transfer function $H(s, p, C_A)$. Not surprisingly, the amplitude of the far-end noise is very small.

Figure 5.38 shows the energy spectral density of the waveform. The energy density predicted using (5.70) is clearly undistinguishable from the one computed directly from the simulated response. The total energy of the pulse can be computed in the time domain or in the frequency domain. Both computations give 11.5 aJ.

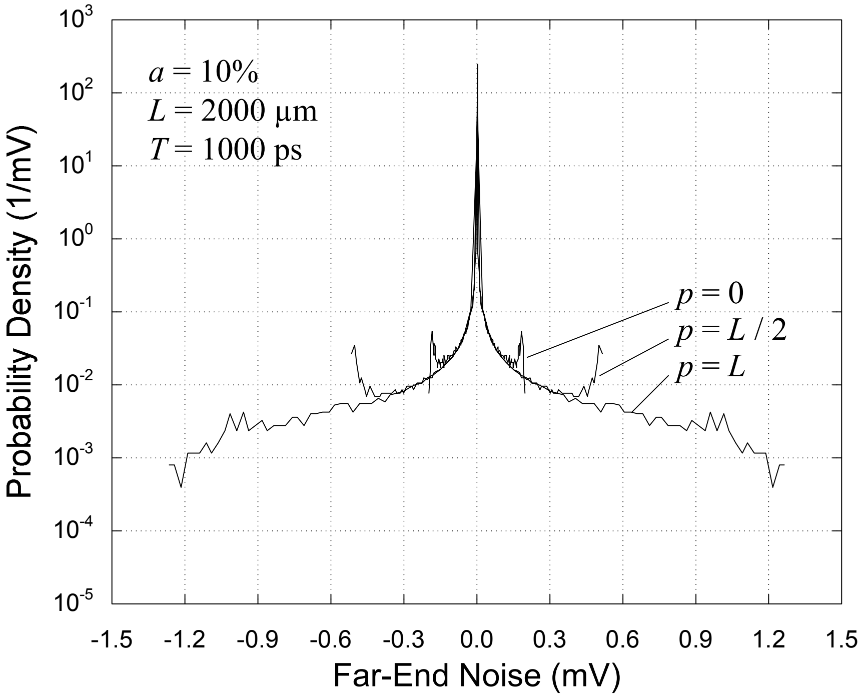


Figure 5.39: Probability density function of far-end noise due to a single attacker.

5.5.3 Power Spectral Density for the Far-End Noise

Let $y(t, p, C_A)$ be the far-end noise waveform due to a single attacker located at position p. Its power spectral density is obtained by filtering the power spectral density of the attacker $S_X(f)$ with the noise transfer function derived earlier:

$$S_Y(f,p,C_A) = S_X(f)\left|H(j2\pi f,p,C_A)\right|^2 = S_X(f)\left|F(j2\pi f,C_A)\right|^2\left|G(j2\pi f,p)\right|^2 \qquad (5.73)$$

The probability density function (PDF) of the far-end noise due to a single attacker switching randomly is the same at any given time. Figure 5.39 shows the PDFs computed numerically for various positions.

When the attacker is located near the driver of the victim line (i.e. when the attacker is located at $p = 0$), the amplitude of the far-end noise is limited to 0.20 mV. This limit corresponds to the magnitude of the step response of the system. When the attacker is moved to the far-end, the magnitude of the step response increases and the limit becomes 1.27 mV.

Table 5.1: Far-end noise power due to a single attacker.

Attacker Position (μm)	Simulated Far-End Noise Power	$\int_{-\infty}^{\infty} S_Y(f,p,C_A)df$
0	0.138 nW	0.139 nW
1000	1.153 nW	1.154 nW
2000	3.571 nW	3.561 nW

It is interesting to note that with a single attacker, the noise produced at any given time cannot be Gaussian since it is bounded. Table 5.1 shows that the power of the noise computed numerically based on the PDFs of Figure 5.39 matches the power computed by integrating the analytical expression for $S_Y(f, p, C_A)$.

Multiple attackers contribute to the *total* far-end noise $n(t)$. The number of orthogonal lines N_{up} and N_{dn} attacking the victim from above and below is determined by the pitch of the interconnect layers over and under it, λ_{up} and λ_{dn}. The two pitches are typically different [5.27]. Since the interlevel dielectric thicknesses are also different, there are two values to consider for the coupling capacitance of the attackers: C_{up} and C_{dn}. Of course, N_{up} and N_{dn} are related to the length of the victim:

$$\begin{aligned} N_{up} &= L/\lambda_{up} \\ N_{dn} &= L/\lambda_{dn} \end{aligned} \tag{5.74}$$

Let P$_{up}$ = $\{0, \lambda_{up}, 2\lambda_{up}, \ldots, N_{up}\lambda_{up}\}$ be the set containing the position of the attackers located above the victim. Similarly, let P$_{dn}$ = $\{0, \lambda_{dn}, 2\lambda_{dn}, \ldots, N_{dn}\lambda_{dn}\}$ be the set containing the position of the attackers located below the victim. The total far-end noise $n(t)$ is obtained by superposing the contributions from all attackers:

$$n(t) = \sum_{p \in P_{up}} y(t,p,C_{up}) + \sum_{p \in P_{dn}} y(t,p,C_{dn}) \tag{5.75}$$

Assuming that the attackers are independent zero-mean random processes makes the derivation of the power spectral density $S_N(f)$ of $n(t)$ easier. Since there are no cross-correlation terms to consider, the power spectral density of the sum is simply the sum of the individual power spectral densities:

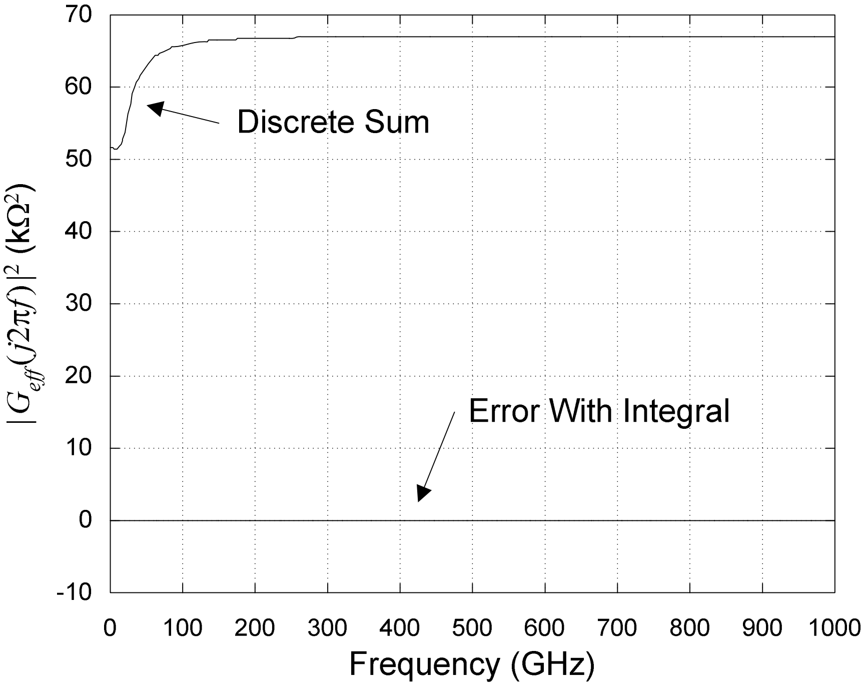


Figure 5.40: Validity of the integral approximation.

$$S_N(f) = \sum_{p \in P_{up}} S_Y(f, p, C_{up}) + \sum_{p \in P_{dn}} S_Y(f, p, C_{dn}) \tag{5.76}$$

Substituting the expression for $S_Y(f, p, C_A)$ yields:

$$S_N(f) = \sum_{p \in P_{up}} S_X(f) \left|F(j2\pi f, C_{up})\right|^2 \left|G(j2\pi f, p)\right|^2 + \sum_{p \in P_{dn}} S_X(f) \left|F(j2\pi f, C_{dn})\right|^2 \left|G(j2\pi f, p)\right|^2 \tag{5.77}$$

or

$$S_N(f) = S_X(f)\left|F(j2\pi f, C_{up})\right|^2 \sum_{p \in P_{up}} \left|G(j2\pi f, p)\right|^2 + S_X(f)\left|F(j2\pi f, C_{dn})\right|^2 \sum_{p \in P_{dn}} \left|G(j2\pi f, p)\right|^2 \tag{5.78}$$

Since the number of attackers is very large, the two discrete sums can be approximated by integrals:

$$\sum_{p \in P} \left|G(j2\pi f, p)\right|^2 \approx \frac{1}{\lambda} \int_0^L \left|G(j2\pi f, p)\right|^2 dp \tag{5.79}$$

With

$$\left|G_{eff}(j2\pi f,\lambda)\right|^2 = \frac{1}{\lambda}\int_0^L \left|G(j2\pi f,p)\right|^2 dp \tag{5.80}$$

the power spectral density of the far-end noise becomes

$$S_N(f) = S_X(f)\left|F(j2\pi f, C_{up})\right|^2 \left|G_{eff}(j2\pi f,\lambda_{up})\right|^2 + S_X(f)\left|F(j2\pi f, C_{dn})\right|^2 \left|G_{eff}(j2\pi f,\lambda_{dn})\right|^2 \tag{5.81}$$

By definition, $G_{eff}(j2\pi f, \lambda)$ can be interpreted as a position-invariant effective transfer function. It produces the same overall effect as all the position-dependant transfer functions combined. In other words, $G_{eff}(j2\pi f, \lambda)$ aggregates the effect of all the attackers into a single effective attacker.

The validity of approximating the discrete sum by an integral to compute $|G_{eff}(j2\pi f, \lambda)|^2$ is confirmed by Figure 5.40. The error introduced by the approximation practically vanishes.

It is possible to derive a closed-form expression for $|G_{eff}(j2\pi f, \lambda)|^2$ by expanding $G(j2\pi f, p)$ in term of exponential functions:

$$G(j2\pi f, p) = \frac{R_S + Z_0}{2} e^{(\alpha + j\beta)p} + \frac{R_S - Z_0}{2} e^{-(\alpha + j\beta)p} \tag{5.82}$$

The conjugate is:

$$G^*(j2\pi f, p) = \frac{R_S + Z_0^*}{2} e^{(\alpha - j\beta)p} + \frac{R_S - Z_0^*}{2} e^{-(\alpha - j\beta)p} \tag{5.83}$$

Taking the product yields:

$$\begin{aligned}\left|G(j2\pi f, p)\right|^2 = {} & \frac{\left|R_S + Z_0\right|^2}{4} e^{2\alpha p} + \frac{\left|R_S - Z_0\right|^2}{4} e^{-2\alpha p} \\ & + \frac{(R_S + Z_0)(R_S - Z_0^*)}{4} e^{j2\beta p} + \frac{(R_S - Z_0)(R_S + Z_0^*)}{4} e^{-j2\beta p}\end{aligned} \tag{5.84}$$

After some algebraic manipulations,

$$\begin{aligned}\left|G(j2\pi f, p)\right|^2 = {} & \frac{R_S^2 + \left|Z_0\right|^2}{2} \cosh(2\,\alpha\,p) + \frac{R_S^2 - \left|Z_0\right|^2}{2} \cos(2\,\beta\,p) \\ & + R_S R_0 \sinh(2\,\alpha\,p) - R_S X_0 \sin(2\,\beta\,p)\end{aligned} \tag{5.85}$$

Integrating from $p = 0$ to L yields the desired closed-form expression:

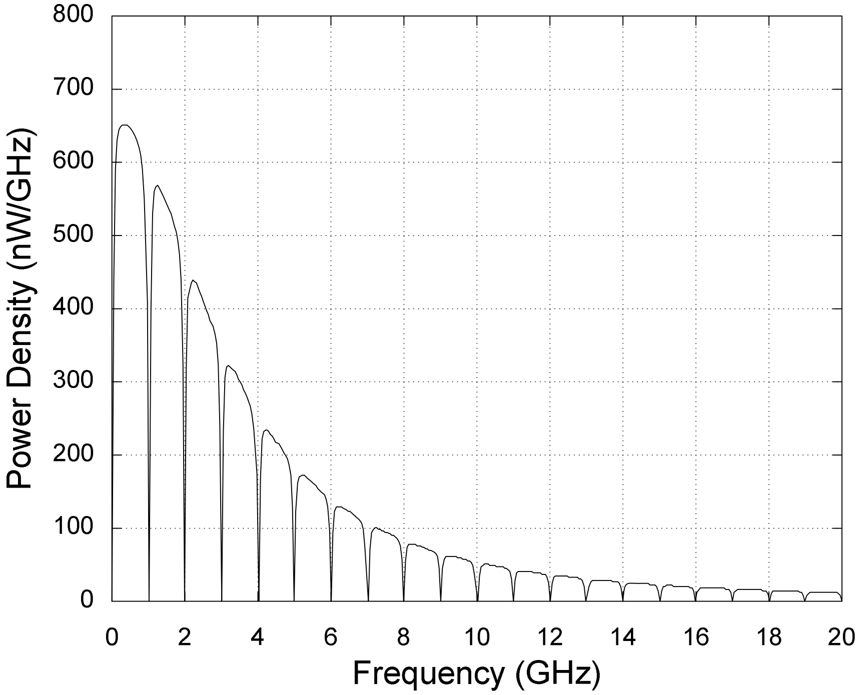


Figure 5.41: Power spectral density for the far-end noise.

$$\begin{aligned}\left|G_{eff}(j2\pi f,\lambda)\right|^2 &= \frac{R_S^2+\left|Z_0\right|^2}{4\lambda\alpha}\sinh(2\alpha L)+\frac{R_S^2-\left|Z_0\right|^2}{4\lambda\beta}\sin(2\beta L)\\ &+\frac{R_S R_0}{\lambda\alpha}\sinh^2(\alpha L)-\frac{R_S X_0}{\lambda\beta}\sin^2(\beta L)\end{aligned} \tag{5.86}$$

The final expression for the power spectral density of far-end noise is obtained by substituting (5.86) into (5.81).

The power spectral density of the far-end noise is shown in Figure 5.41 for a = 0.10, T = 1000 ps, V = 1.2 V, τ = 10 ps, and λ = 1.00 μm. As before, the other parameters represent a 2000-μm M5 interconnect with r = 0.10 Ω/μm, l = 0.50 pH/μm, and c = 0.20 fF/μm. The resistance of the driver holding the victim is R_S = 50 Ω while the line is terminated by a load C_T of 10 fF. There are 2000 orthogonal attackers located above the victim and 2000 below.

The total power P of the far-end noise can be computed analytically by integrating (5.81):

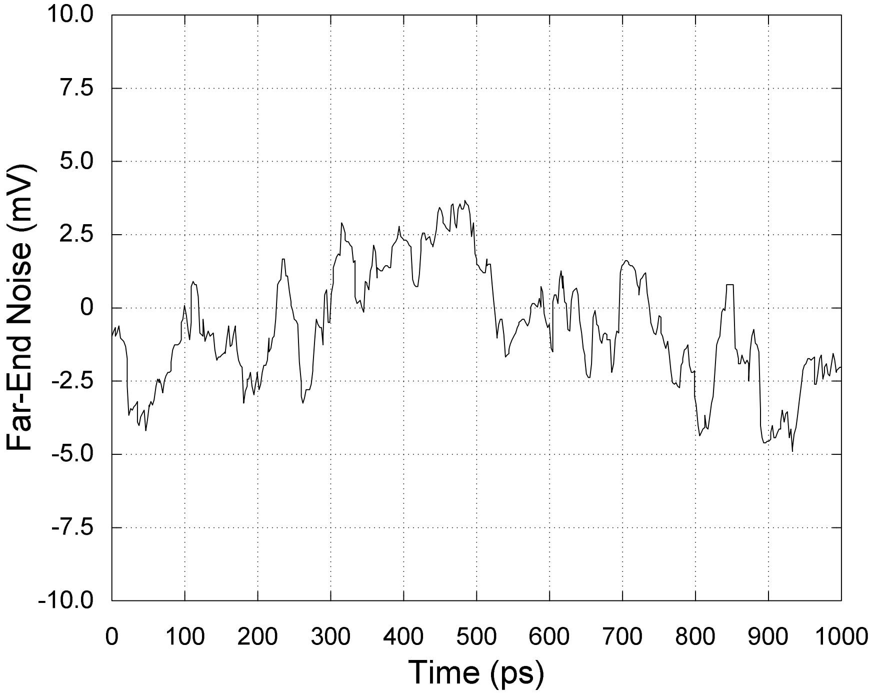


Figure 5.42: Sample far-end noise waveform.

$$P = \int_{-\infty}^{\infty} S_N(f)df \tag{5.87}$$

The result corresponding to the power spectrum of Figure 5.41 is 5.56 μW.

It is interesting to observe that $S_N(f)$ is proportional to $V^2C_A^2 / \lambda$ at all frequencies. From (5.45), it is clear that V^2 measures the power of the attackers. The amount of power transferred to the victim line is also proportional to $|V_1/V_0|^2$ and therefore to C_A^2. Furthermore, the far-end noise is proportional to the number of attackers, which is inversely proportional to λ. Not surprisingly, when the frequency is zero, $|H(j2\pi f)| = 0$. In other words, the far-end noise has no DC component.

Part of a sample far-end voltage waveform produced via simulation is shown in Figure 5.42. The amplitude of the noise is only a few millivolts. Its power, computed numerically in the time-domain, converges to 5.57 μW after several cycles. The power matches the one computed analytically in the frequency domain.

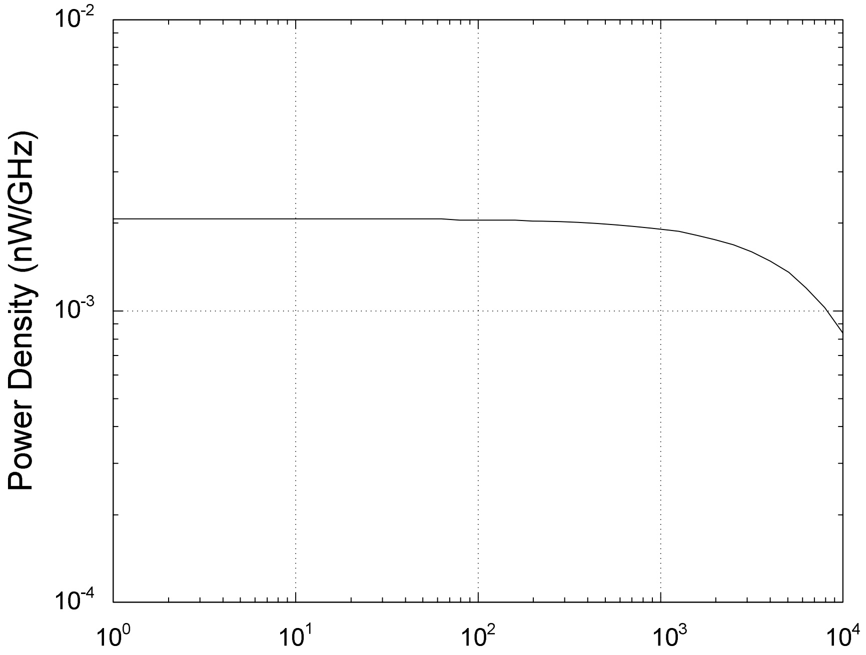


Figure 5.43: Thermal noise power spectral density at room temperature.

For reference, the two-sided power spectral density of thermal noise is given by [5.24]:

$$N_{thermal}(f) = \frac{1}{2}\frac{hf}{e^{hf/kT} - 1} \tag{5.88}$$

where k is Boltzmann's constant (1.381×10^{-23} J/K) and h is Planck's constant (6.626×10^{-34} J/Hz). The spectrum is shown in Figure 5.43 at room temperature (300 K). The power density is 0.002 nW/GHz, up to frequencies of 1000 GHz or so. Compared to the interlevel coupling noise of Figure 5.41, thermal noise is clearly negligible.

5.5.4 Probability Bounds on Interlevel Coupling Noise

This section uses the analytical expression for the power spectral density of the far-end noise $S_N(f)$ given in (5.81) to formally derive a probabilistic bound for the magnitude the noise.

The first step is to note that the far-end noise produced by a large number of independent attackers is approximately Gaussian, even though the noise due to a single attacker is not. The probability density functions shown in Figure 5.39 are even and have a mean of zero.

Their third central moment is therefore zero as well. Lyapunov's central limit theorem states that the sum of N such random variables approaches a Gaussian distribution as N gets large, whether they share the same probability density function or not.

The next step is to analyze the probability that the far-end noise crosses a given level ε. Let M be the mean number of level crossings per unit of time. M is given by Rice's formula [5.28], [5.29]:

$$M = M_0 e^{-\frac{\varepsilon^2}{2P}} \tag{5.89}$$

where P is the power of the far-end noise of (5.87) and M_0 is the mean number of zero crossings per unit of time.

M_0 is given by:

$$M_0 = 2\sqrt{\frac{1}{P}\int_{-\infty}^{\infty} f^2 S_N(f)df} \tag{5.90}$$

Figure 5.44 shows the level crossing rate derived analytically using Rice's formula for the far-end noise of Figure 5.41. It also shows the level crossing rate simulated numerically using 50 RLC sections and a time step of 0.2 ps.

The two curves follow the same trend: they decrease rapidly as ε increases. However, the level crossing rate simulated numerically for ε = 0 is slightly lower than the 50.1 crossings per cycle derived analytically.

The difference is due in part to the time step used for the numerical simulations. With 20 RLC sections, reducing the time step from 1.0 ps to 0.2 ps increases M_0 from 36.9 to 40.2 crossings per cycle. The difference is also due in part to the number of RLC sections used to model the victim line. M_0 is observed to increase from 40.2 to 43.0 crossings per cycle as the number of RLC sections increases from 20 to 50.

Numerically converging on the level crossing rate gets increasingly difficult as ε increases. Getting an accurate average requires the observation of several crossings. As the number of crossings becomes rarer, the simulation times increase rapidly.

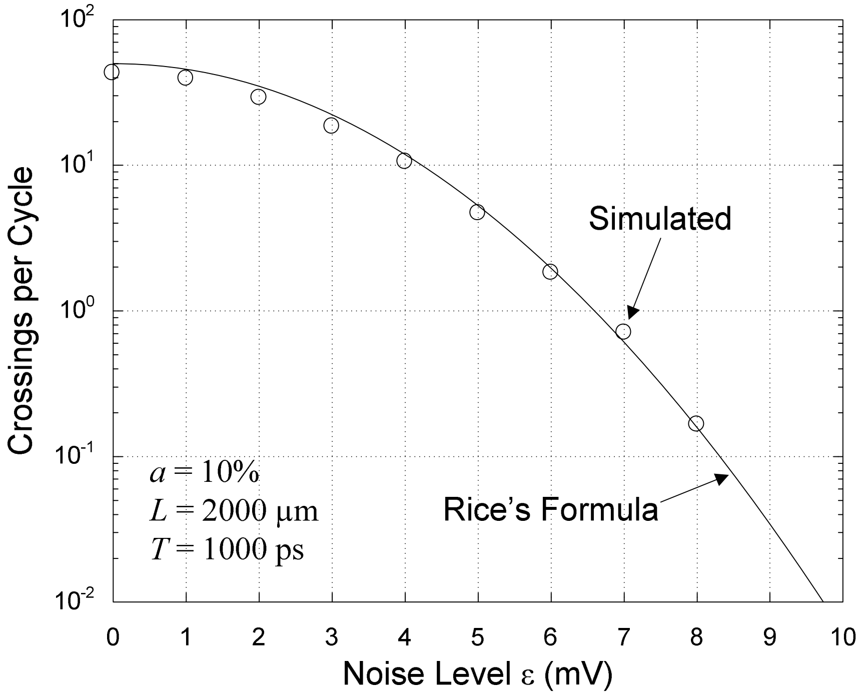


Figure 5.44: Mean number of level crossings per cycle.

For the far-end interlevel coupling noise of Figure 5.41 where P = 5.56 μW and $M_0 = 50.1 \times 10^9$ crossings per second, it takes on average 10 years for the far-end noise to traverse the 22.2-mV level. For practical purposes, it is therefore reasonable to assume that the noise never exceeds that level.

5.6 Summary

The power required for global clock distribution is fairly small, but not negligible. Using minimum rectilinear Steiner trees (MRSTs), it is shown that when the number of loads is very small, the fixed interconnect cost associated with a full or partial clock grid makes H-trees a better choice for minimizing power. However, as the number of loads increases, the difference between the three structures is found to practically vanish. It is also shown that the dispersion of the loads significantly impacts the total length of the wires required to connect them, and therefore, the power required for global clock distribution.

It is observed that most of the skew and jitter compensation strategies require sending signals over long distances and across multilevel interconnect structures, where the capacitive coupling between adjacent layers creates interlevel coupling noise.

The power spectral density $S_N(f)$ for this interlevel coupling noise at the far-end of a victim line is rigorously derived. The result is a novel closed-form expression that accounts for the switching activity of the attackers and the electrical parameters of the victim:

$$\begin{aligned} S_N(f) = S_X(f)\left|F(j2\pi f, C_{up})\right|^2\left|G_{eff}(j2\pi f, \lambda_{up})\right|^2 \\ + S_X(f)\left|F(j2\pi f, C_{dn})\right|^2\left|G_{eff}(j2\pi f, \lambda_{dn})\right|^2 \end{aligned} \tag{5.91}$$

where $S_X(f)$, $F(s, C_A)$, and $|G_{eff}(j2\pi f, \lambda)|^2$ are respectively given by (5.46), (5.71) and (5.86). The first term represents the interlevel coupling noise caused by the attackers located above the victim. The second term is the noise due to the attackers below it.

With parameters typical for 130-nm global interconnects, interlevel coupling noise dominates thermal noise.

A probabilistic bound for the magnitude of the interlevel coupling noise is then derived. It formally shows that for attackers with an uncorrelated switching activity, the assumption that the conductors orthogonally routed above and below the victim behave as a quiet metal plane is statistically very good.

CHAPTER 6

Power-Supply Noise

Power-supply noise tends to produce a significant amount of clock jitter. This chapter examines the common (but typically implicit) assumption that minimizing the worst-case supply voltage drop minimizes the degradation in system performance. First, the impact of power-supply noise on the timing of a device driving a capacitive load is mathematically analyzed using the quasi-linear device model proposed in Chapter 2. Then, more general delay models are derived for device-dominated and interconnect-dominated timing paths. For typical circuits satisfying the locality assumption, Section 6.2 shows that the peak of the noise is largely uncorrelated with its timing impact and that the average supply voltage during switching is more important. Section 6.3 derives a model that accurately predicts the timing impact of the power-supply noise for long interconnects. It is pointed out that for repeated wires, the differential component of a voltage drop can potentially have a greater timing impact than the common-mode component. Circuit arguments are then given to justify why the global interconnects used for clock distribution are particularly vulnerable. In Section 6.4, realistic values for the model parameters are measured on a 2.53-GHz Pentium® 4 microprocessor. The measurements are analyzed in Section 6.5 and strongly suggest that the timing impact of the noise is likely to increase for future technology generations.

6.1 Overview and Impact on Device Timing

In the power distribution literature, it is well understood that power-supply noise is undesirable. This noise can reduce oxide reliability [6.1] and make SRAM cells unstable [6.2]. It can also impact performance.

Most of the literature concerned with the timing impact of the noise concentrates on the *maximum* voltage fluctuation created by the simultaneous switching of a large number of transistors. In [6.3] for instance, Zheng and Tenhunen postulate that it is more useful to know the peak of the noise rather than its time-domain characteristics. Consequently, they propose equations to efficiently compute this peak, under the assumption that all the gates that switch do so at the same time. Other authors have proposed simple analytical models for the time-domain characteristics of the power-supply noise caused by off-chip drivers, again in order to predict its peak. In [6.4], [6.5], equations are derived to describe how this peak is affected by the number of switching and quiet drivers. However, these equations are specific to the context of off-chip drivers and assume that the drivers share an ideal on-chip power supply. Finally, numerous authors have written about the tools and methodologies used in practice to analyze the power-supply noise for an entire chip [6.6]–[6.8]. Once more, these techniques focus on predicting the worst-case voltage drop in order to minimize it.

From a system perspective however, one could argue that the worst-case power-supply noise is only *indirectly* important. The reason is that in practice, the power-supply noise tends to impact timing before causing other failures. When this is the case, the performance impact of the noise (in GHz) is more relevant than the noise itself (in mV).

The idea that the power-supply noise is only indirectly important is supported by recent empirical evidence. This evidence suggests that the relationship between the magnitude of the noise and its impact on performance is far from being simple. For the 130-nm Pentium® 4 microprocessor shown in Figure 6.1, the results presented in [6.9] indicate that removing a large amount of on-chip decoupling capacitance can actually makes some timing paths *faster* instead of slower. The results also show that, sometimes, the on-chip decoupling capacitors have almost no impact on frequency, even when their effect on the power-supply noise is large [6.10].

Despite this *a priori* counter-intuitive relationship, very little work has been done to understand the timing impact of power-supply noise. One reason for this could be the sheer complexity of performing an integrated analysis. For a high-frequency

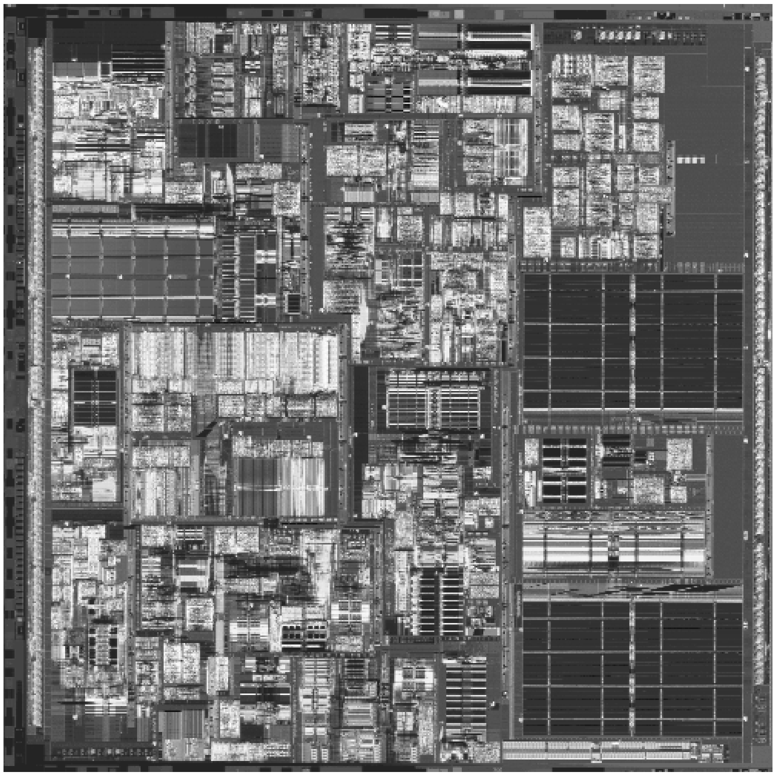


Figure 6.1: Pentium 4 on 130-nm technology.

microprocessor, predicting the timing impact would require a model capable of describing the switching behavior of each transistor as a function of the instructions being processed. This assumes that the instruction sequence maximizing the timing impact of the noise (not just its peak) can somehow be identified or approximated, perhaps by adapting some of the ideas presented in [6.11]. Predicting the timing impact would also require a model for the on-chip power distribution network, package, socket, system board, and voltage regulator. Unfortunately, integrating all these models to perform timing analysis is not yet practical [6.6], [6.12], [6.13]. Currently, the performance impact of the power-supply noise can only be analyzed for simple circuits, despite the use of simplifying statistical assumptions [6.14].

This section derives a simple analytical model to determine the timing impact of the power-supply noise on a device discharging a load capacitance.

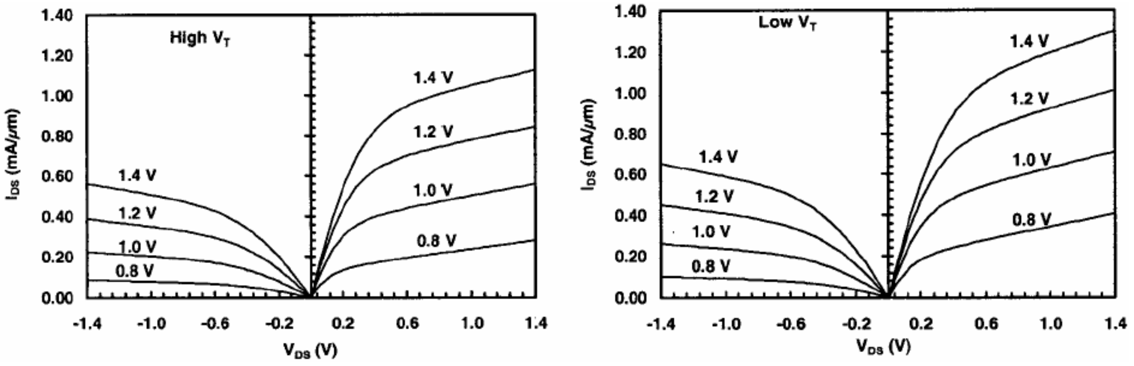


Figure 6.2: I-V curves reported in [6.18].

6.1.1 Output Waveform of a Device Discharging a Capacitive Load

The purpose of this section is to mathematically describe the output waveform of a device discharging a capacitance C_{load} when the device is subjected to power-supply noise. More specifically, the goal is to determine when the output waveform crosses its 50% point and what its transition time is. The derivation is based on the transistor equations proposed in Chapter 2 and is much more general than the one proposed in [6.15].

According to Sakurai and Newton in [6.16] and [6.17], drain current of a transistor in saturation is given by:

$$I_D = I_{D0}\left(\frac{V_{GS}-V_T}{V_0-V_T}\right)^{\alpha}(1+\lambda V_{DS}) \tag{6.1}$$

where V_{GS} is the gate-to-source voltage, V_{DS} is the drain-to-source voltage, V_T is the threshold voltage, λ is the channel-length modulation factor, and I_{D0} $(1 + \lambda V_0)$ is the drain current at $V_{GS} = V_{DS} = V_0$. It is worth emphasizing that V_0 is *not* a free variable because of the link to I_{D0}. V_0 only indicates the voltage at which I_{D0} is characterized. Its value cannot be changed independently. Here, the characterization voltage V_0 is arbitrarily chosen to correspond to the nominal supply voltage $V_{CC} - V_{SS}$ of 1.5 V.

In [6.15], the authors implicitly make two questionable assumptions. First, they assume that channel-length modulation is negligible (i.e. $\lambda = 0$). For the technology described in [6.18], this is clearly incorrect. As shown in Figure 6.2, the drain current is *not* constant

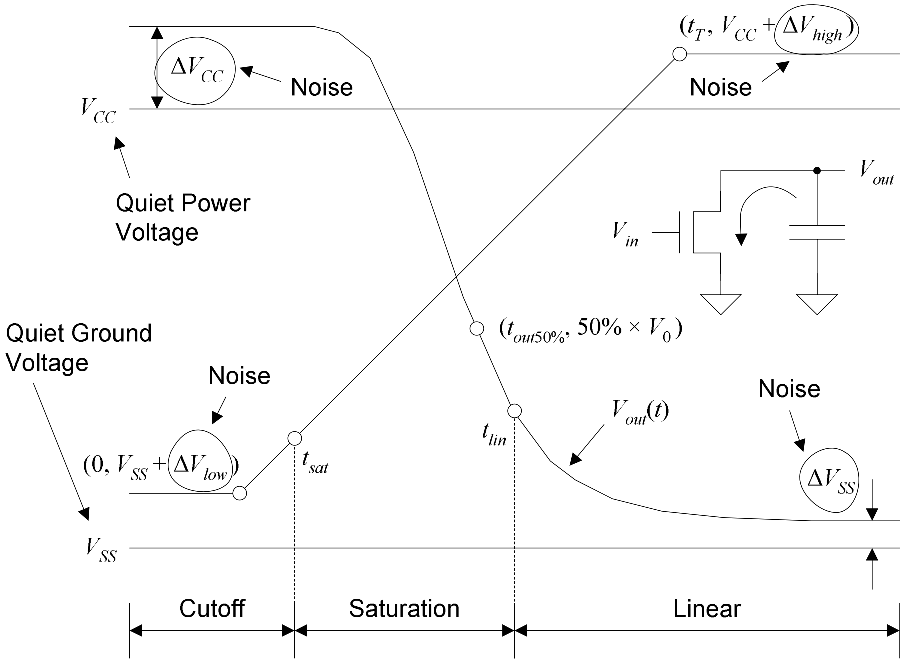


Figure 6.3: Device response.

in saturation. On the contrary, it is a fairly strong function of V_{DS}. They also assume that I_{D0} is *independent* of V_0. This is a problem because if I_{D0} and V_0 are treated as independent variables, then (6.1) becomes non-physical. For a given device, the drain current should *only* be a function of V_{GS} and V_{DS} (as in Figure 6.2). Making I_{D0} and V_0 independent introduces an artificial (and absurd) dependency on the characterization voltage.

The derivation proposed here assumes the four independent perturbations shown in Figure 6.3. The low and high voltages of the input waveform are respectively $(V_{SS} + \Delta V_{low})$ and $(V_{CC} + \Delta V_{high})$ to reflect the power-supply noise affecting the previous stage. The local power and ground levels are $(V_{CC} + \Delta V_{CC})$ and $(V_{SS} + \Delta V_{SS})$. The four independent perturbations make this derivation more general than the one proposed in [6.15], where only two cases are considered: no noise from the previous stage $(\Delta V_{low} = \Delta V_{high} = 0)$ and same noise for the two stages $(\Delta V_{low} = \Delta V_{SS}, \Delta V_{high} = \Delta V_{CC})$.

Based on Figure 6.3, the gate voltage is a ramp. It is given by:

$$V_G(t) = (V_0 + \Delta V_{high} - \Delta V_{low})\frac{t}{t_T} + \Delta V_{low} \tag{6.2}$$

The time at which the device starts conducting (i.e. the time at which the device goes from cutoff to saturation) is when $V_{GS}(t) = V_G(t) - \Delta V_{SS}$ reaches the saturation condition defined by (2.17). This occurs at:

$$t_{sat} = \frac{V_T - \frac{g_D}{g_G}(V_0 + \Delta V_{CC} - \Delta V_{SS} - V_{DS0}) - \Delta V_{low} + \Delta V_{SS}}{V_0 + \Delta V_{high} - \Delta V_{low}} t_T \tag{6.3}$$

The output voltage after time t_{sat} obeys the following differential equation, for as long at the device remains in saturation:

$$-C_{load}V'_{out}(t) = I_D(V_{GS}(t), V_{DS}(t)) \tag{6.4}$$

where $V_{DS}(t) = V_{out}(t) - \Delta V_{SS}$. This definition for $V_{DS}(t)$ assumes that $V_{out}(t)$ is referenced to V_{SS}, the quiet ground voltage. The drain current I_D is given by (2.16):

$$-C_{load}V'_{out}(t) = g_G(V_G(t) - \Delta V_{SS} - V_T) + g_D(V_{out}(t) - \Delta V_{SS} - V_{DS0}) \tag{6.5}$$

Equation (6.4) is of the form:

$$-\frac{C_{load}}{g_D}V'_{out}(t) - V_{out}(t) = mt - b \tag{6.6}$$

where m and b are time-invariant:

$$\begin{aligned} m &= \frac{g_G}{g_D}\frac{V_0 + \Delta V_{high} - \Delta V_{low}}{t_T} \\ b &= \frac{g_G}{g_D}(V_T - \Delta V_{low} + \Delta V_{SS}) + V_{DS0} + \Delta V_{SS} \end{aligned} \tag{6.7}$$

With (6.7), $t_{sat} = (b - V_0 - \Delta V_{CC}) / m$. The particular solution with initial condition $V_{out}(t_{sat}) = V_{init}$ is:

$$V_{out}(t) = \left(m\left(t_{sat} - \frac{C_{load}}{g_D}\right) - b + V_{init}\right)e^{-\frac{g_D}{C_{load}}(t - t_{sat})} - \left(m\left(t - \frac{C_{load}}{g_D}\right) - b\right) \tag{6.8}$$

The initial voltage V_{init} is determined by the noise on V_{CC} and is $V_0 + \Delta V_{CC}$. The device is expected to remain in saturation as the output waveform crosses the 50% point. As shown in Figure 6.2, the drain-to-source voltage must fall well below $V_0 / 2$ for the device to enter its linear region of operation. When the input transition time is slow, the output waveform may cross the 50% point early, while the input is still rising. In this case, the time at which the output crosses the 50% point can be obtained from (6.8) by solving $V_{out}(t) = 50\% \times V_0$:

$$t_{out50\%,\, early} = \frac{b - \frac{1}{2}V_0}{m} + \frac{C_{load}}{g_D}\left[W\left(\left(\frac{g_D}{C_{load}} \frac{mt_{sat} - b + V_{init}}{m} - 1 \right) e^{\frac{g_D}{C_{load}} \frac{mt_{sat} - b + \frac{1}{2}V_0}{m} - 1} \right) + 1 \right] \tag{6.9}$$

where $W(\cdot)$ is the Lambert W function. At $t = t_T$, the input voltage stabilizes to $V_0 + \Delta V_{high}$. The rest of the derivation assumes that the output crosses the 50% point after that time. The output voltage at t_T is given by (6.8):

$$V_{out}(t_T) = \left(m\left(t_{sat} - \frac{C_{load}}{g_D} \right) - b + V_{init} \right) e^{-\frac{g_D}{C_{load}}(t_T - t_{sat})} - \left(m\left(t_T - \frac{C_{load}}{g_D} \right) - b \right) \tag{6.10}$$

For $t > t_T$, the output voltage decreases according to the following differential equation:

$$-C_{load} V'_{out}(t) = I_D(V_0 + \Delta V_{high} - \Delta V_{SS}, V_{out}(t)) \tag{6.11}$$

Equation (6.11) is of the form:

$$-\frac{C_{load}}{g_D} V'_{out}(t) - V_{out}(t) = mt_T - b \tag{6.12}$$

The solution to (6.12) satisfying (6.10) is:

$$V_{out}(t) = (mt_T - b + V_{out}(t_T)) e^{-\frac{g_D}{C_{load}}(t - t_T)} - (mt_T - b) \tag{6.13}$$

Solving (6.13) for 50% × V_0 yields $t_{out50\%,\, late}$:

$$t_{out50\%,\, late} = t_T + \frac{C_{load}}{g_D} \log \frac{mt_T - b + V_{out}(t_T)}{mt_T - b + \frac{1}{2}V_0} \tag{6.14}$$

Using (6.10) for $V_{out}(t_T)$ yields:

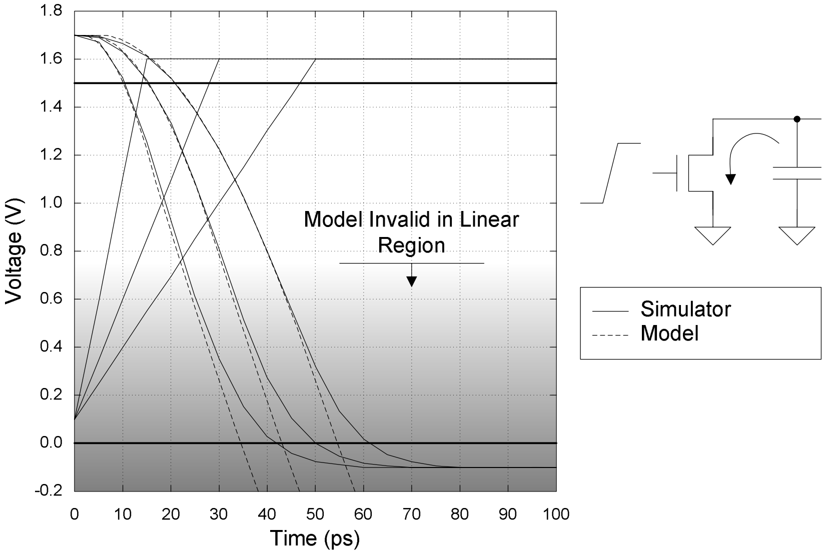


Figure 6.4: Accuracy of output waveform model.

$$t_{out50\%,\,late} = t_T + \frac{C_{load}}{g_D}\log\frac{\left(m\left(t_{sat}-\frac{C_{load}}{g_D}\right)-b+V_{init}\right)e^{-\frac{g_D}{C_{load}}(t_T-t_{sat})}+m\frac{C_{load}}{g_D}}{mt_T-b+\frac{1}{2}V_0} \tag{6.15}$$

Equation (6.15) is valid when $t_{out50\%,\,late} > t_T$.

The details of the output waveform beyond the 50% point are ignored. As shown in Figure 6.3, the tangent at that point is reasonably representative of the output transition time.

Figure 6.4 shows that (6.8) and (6.13) can be used to accurately predict the time at which the output crosses the 50% point in the presence of power-supply noise. The simulated waveforms are considered exact. The transition time of the input ramp varies from 15 ps to 50 ps. The output capacitance is 20 fF. The device parameters are g_{G0} = 1.13 mA/µm, g_{D0} = 0.22 mA/µm, V_T = 0.60 V, and V_{DS0} = 0.40 V. They are derived from the I-V curves produced by the simulator and are discussed in Chapter 2. The noise parameters are ΔV_{low} = 0.1 V, ΔV_{high} = 0.1 V, ΔV_{CC} = 0.2 V, and ΔV_{SS} = –0.1V. The output waveforms predicted by the model closely match the simulated ones. As expected, they start to diverge when the output voltage falls below V_{DS0}. There, the device no longer operates in saturation and (6.8) and (6.13) cease to be valid. All output waveforms reach V_0 / 2 well before starting to diverge from the simulated curves.

6.1.2 Impact of Power-Supply Noise on Delay and Transition Time

This section uses the mathematical description of the output waveform derived in 6.1.1 to model the timing impact of power-supply noise.

By definition, the delay d of the device is $t_{out50\%} - t_{in50\%}$. The input ramp described by (6.2) crosses its 50% point at:

$$t_{in50\%} = \frac{\frac{1}{2}V_0 - \Delta V_{low}}{V_0 + \Delta V_{high} - \Delta V_{low}} t_T \tag{6.16}$$

The time at which the output crosses its 50% point is obtained by combining (6.9) and (6.15):

$$t_{out50\%} = t_{out50\%,\ early} + (t_{out50\%,\ late} - t_{out50\%,\ early})u(t_{out50\%,\ early} - t_T) \tag{6.17}$$

The expression for $t_{out50\%}$ assumes that $t_{out50\%,\ early}$ occurs before t_T. If not, the unit step function $u(\cdot)$ becomes 1 and corrects the assumption.

The device delay is therefore:

$$d = t_{out50\%,\ early} - t_{in50\%} + (t_{out50\%,\ late} - t_{out50\%,\ early})u(t_{out50\%,\ early} - t_T) \tag{6.18}$$

Equation (6.19) is exact. It allows the time at which the input ramp stops rising to precede or follow the time at which the output reaches its 50% point.

The output transition time is defined by extrapolating the tangent at the 50% point until it reaches the noisy power supply voltages:

$$t_{Tout} = -\frac{V_0 + \Delta V_{CC} - \Delta V_{SS}}{V'_{out}(t_{out50\%})} \tag{6.19}$$

Since the output is falling, the derivative is negative. The minus sign makes the transition time positive.

From (6.6), when the input ramp is still rising:

$$V'_{out}(t) = -\frac{g_D}{C_{load}}(mt - b + V_{out}(t)) \tag{6.20}$$

At time $t_{out50\%}$ in particular $V_{out} = V_0 / 2$. If $t_{out50\%}$ occurs before t_T,

$$V'_{out}(t_{out50\%,\ early}) = -\frac{g_D}{C_{load}}(mt_{out50\%,\ early} - b + \tfrac{1}{2}V_0) \tag{6.21}$$

After time t_T, from (6.12):

$$V'_{out}(t) = -\frac{g_D}{C_{load}}(mt_T - b + V_{out}(t)) \tag{6.22}$$

Thus, if the output reaches its 50% point after t_T:

$$V'_{out}(t_{out50\%,\ late}) = -\frac{g_D}{C_{load}}(mt_T - b + \tfrac{1}{2}V_0) \tag{6.23}$$

The two expressions for the derivative of V_{out} at $t_{out50\%}$ can be combined as before:

$$V'_{out}(t_{out50\%}) = V'_{out}(t_{out50\%,\ early}) + (V'_{out}(t_{out50\%,\ late}) - V'_{out}(t_{out50\%,\ early}))u(t_{out50\%,\ early} - t_T) \tag{6.24}$$

After substitution, this simplifies to:

$$V'_{out}(t_{out50\%}) = -\frac{g_D}{C_{load}}\left[mt_{out50\%,\ early} - m(t_{out50\%,\ early} - t_T)u(t_{out50\%,\ early} - t_T) - b + \frac{1}{2}V_0\right] \tag{6.25}$$

and the transition time becomes:

$$t_{Tout} = \frac{C_{load}}{g_D}\frac{V_0 + \Delta V_{CC} - \Delta V_{SS}}{mt_{out50\%,\ early} - m(t_{out50\%,\ early} - t_T)u(t_{out50\%,\ early} - t_T) - b + \frac{1}{2}V_0} \tag{6.26}$$

Together, (6.18) and (6.26) describe how the delay and the output transition time of a device discharging a capacitive load are impacted by power-supply noise.

Figure 6.5 shows the accuracy of (6.18) for $C_{load} =$ 40 fF and $t_T = 50$ ps. The four noise parameters (ΔV_{low}, ΔV_{high}, ΔV_{CC}, and ΔV_{SS}) are varied one-by-one. While one of the parameters is swept, the remaining ones remain fixed at zero. Clearly, the model accurately predicts the delay sensitivity to power-supply noise.

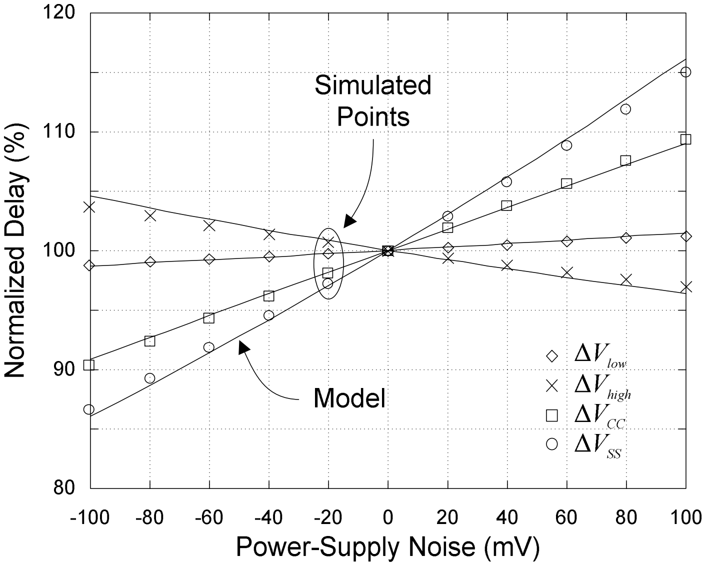


Figure 6.5: Impact of power-supply noise on device delay.

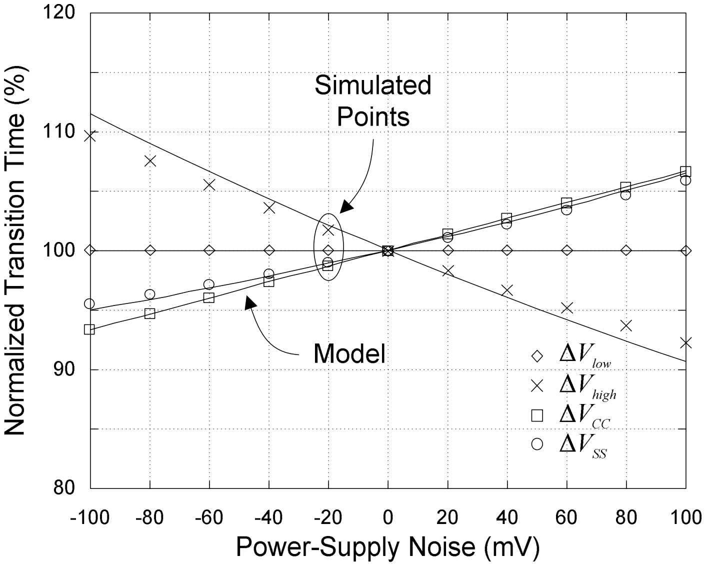


Figure 6.6: Impact of power-supply noise on output transition time.

Figure 6.6 compares the output transition times predicted by (6.26) to the simulated ones. The impact is accurately captured for ΔV_{high}, ΔV_{CC}, and ΔV_{SS}. It interesting, but not surprising, to note that ΔV_{low} has no effect on the output transition time. If $t_{out50\%}$ occurs after t_T then (6.27) becomes:

$$\begin{aligned} t_{Tout} &= \frac{C_{load}}{g_D} \frac{V_0 + \Delta V_{CC} - \Delta V_{SS}}{mt_T - b + \frac{1}{2} V_0} \\ &= \frac{C_{load} (V_0 + \Delta V_{CC} - \Delta V_{SS})}{g_G (V_0 + \Delta V_{high} - \Delta V_{SS} - V_T) + g_D (\frac{1}{2} V_0 - \Delta V_{SS} - V_{DS0})} \end{aligned} \tag{6.27}$$

At $t_{out50\%}$, the gate-to-source voltage is $V_0 + \Delta V_{high} - \Delta V_{SS}$ and the drain-to-source voltage is $V_0 / 2 - \Delta V_{SS}$. These two voltages fix the drain current according to (2.16) and V'_{out} according to (6.4).

6.2 Modeling for Device-Dominated Paths

The timing paths of a typical high-frequency microprocessor can be classified as either device-dominated or interconnect-dominated. Device-dominated paths tend to be localized to the same area of the chip (because they do have short wires). The devices forming such a path share similar power (V_{CC}) and ground (V_{SS}) voltages. On the other hand, interconnect-dominated paths are almost always distributed since they are constructed using long wires. The devices along an interconnect-dominated path generally receive different power and ground voltages.

This section shows that for typical device-dominated paths, the peak of the power-supply noise is largely irrelevant. It demonstrates that the average supply voltage, when taken over the appropriate time window, is useful to accurately predict the delay impact of the noise. Interconnect-dominated are discussed in Section 6.3.

Here, the power and ground voltages are assumed the same everywhere. In other words, the power-supply noise simultaneously affects all devices. Under this locality assumption, $\Delta V_{low} = \Delta V_{SS}$ and $\Delta V_{high} = \Delta V_{CC}$. Generally, ΔV_{CC} and ΔV_{SS} are both non-zero. They define the effective supply voltage V applied to the devices:

$$V = (V_{CC} + \Delta V_{CC}) - (V_{SS} + \Delta V_{SS}) = V_0 + \Delta V_{CC} - \Delta V_{SS} \tag{6.28}$$

When all the devices share the same power and ground voltages, increasing ΔV_{CC} is equivalent to decreasing ΔV_{SS}. It is convenient to write:

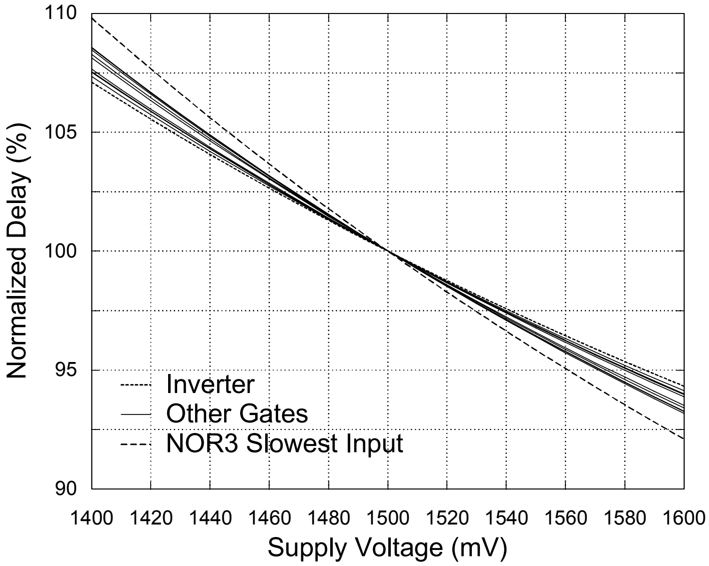


Figure 6.7: Delay versus supply voltage for an inverter and several other gates.

$$\Delta V_{CC} = -\Delta V_{SS} = \tfrac{1}{2}\Delta V \tag{6.29}$$

because it makes the 50% point of V coincide with the 50% point of V_0. This is important because the delay of (6.18) and output transition time of (6.26) are given with respect to 50% × V_0. When the locality assumption holds, ΔV represents *the* power-supply noise.

6.2.1 Delay Versus Supply Voltage

It is well known that the delay of a simple inverter is inversely related to V and that the relationship is generally non-linear [6.16]. It is perhaps less obvious that in a technology like the one described in [6.18], nearly the same delay-versus-voltage relationship holds for a wide range of gates. Figure 6.7 shows how the delay changes for an inverter, two NAND gates, two NOR gates, a full adder, and a multiplexer when the supply voltage changes. For the NAND and NOR gates, two curves are shown: one for the fastest input switching, and the other for the slowest input switching. Although the curves are reasonably linear around V_0 = 1.5 V, they still exhibit a noticeable quadratic component. The gates are characterized in chains, each one driving a copy of itself. The rising and falling delays are averaged. When nearly the same delay-versus-voltage curve holds for a wide range of gates, it is reasonable to assume that the *average* of these curves also holds for all device-dominated paths.

The following derivation uses the results of Section 6.1 to determine the delay-versus-voltage curve of an inverter. Under the locality assumption and from (6.29), the input ramp crosses its 50% point at:

$$t_{in50\%} = \tfrac{1}{2} t_T \tag{6.30}$$

The initial voltage becomes:

$$V_{init} = V_0 + \tfrac{1}{2}\Delta V \tag{6.31}$$

and the device starts conducting at:

$$t_{sat} = \frac{V_T - \frac{g_D}{g_G}(V_0 + \Delta V - V_{DS0})}{V_0 + \Delta V} t_T \tag{6.32}$$

Also,

$$\begin{aligned} m &= \frac{g_G}{g_D}\frac{V_0 + \Delta V}{t_T} \\ b &= \frac{g_G}{g_D} V_T + V_{DS0} - \tfrac{1}{2}\Delta V \end{aligned} \tag{6.33}$$

The delay $d(t_T, C_{load}, \Delta V)$ and the output transition time $t_{Tout}(t_T, C_{load}, \Delta V)$ are still given by (6.18) and (6.26), respectively.

For an inverter part of a device-dominated path, the input transition time is the output transition time of the previous stage. The stage delay is obtained by substituting the output transition time of (6.26) for t_T in (6.18):

$$d_{stage} = d(t_{Tout}, C_{load}, \Delta V) \tag{6.34}$$

Since t_{Tout} is itself a function of t_T, t_T must be chosen such that the following equality holds:

$$t_T = t_{Tout}(t_T, C_{load}, \Delta V) \tag{6.35}$$

An approximate solution to this equation is given by (6.27):

$$t_T \approx \frac{C_{load}(V_0 + \Delta V)}{g_G(V_0 + \Delta V - V_T) + g_D(\tfrac{1}{2}(V_0 + \Delta V) - V_{DS0})} \tag{6.36}$$

The approximation assumes that the gate voltage is close to its maximum when the output crosses its 50% point. When $t_{out50\%}$ occurs after t_T, (6.36) is exact. Under this assumption, the stage delay becomes:

$$d_{stage}(C_{load}, \Delta V) = d\left(\frac{C_{load}(V_0 + \Delta V)}{g_G(V_0 + \Delta V - V_T) + g_D(\tfrac{1}{2}(V_0 + \Delta V) - V_{DS0})}, C_{load}, \Delta V\right) \tag{6.37}$$

The delay-versus-voltage relationship for a chain of inverters implied by (6.37) is obtained by scaling the normalized stage delay. The path delay D is:

$$D = D_0 \frac{d_{stage}(C_{load}, \Delta V)}{d_{stage}(C_{load}, 0)} \tag{6.38}$$

where D_0 is the path delay when there is no noise.

It is remarkable (and non-obvious) that the delay-versus-voltage relationship of (6.38) does not change when C_{load} does. The reason is that $d_{stage}(C_{load}, \Delta V)$ can be factored as follows:

$$d_{stage}(C_{load}, \Delta V) = C_{load}\Phi(\Delta V) \tag{6.39}$$

where $\Phi(\Delta V)$ is independent of C_{load}.

Proof: The stage delay defined in (6.37) can be written as:

$$d_{stage}(C_{load}, \Delta V) = d(C_{load}\ \varphi_1(\Delta V), C_{load}, \Delta V) \tag{6.40}$$

The input transition time, the saturation time, and m become:

$$t_{in50\%} = \tfrac{1}{2} C_{load}\ \varphi_1(\Delta V) = C_{load}\ \varphi_2(\Delta V) \tag{6.41}$$

$$t_{sat} = \frac{V_T - \frac{g_D}{g_G}(V_0 + \Delta V - V_{DS0})}{V_0 + \Delta V} C_{load}\ \varphi_1(\Delta V) = C_{load}\ \varphi_3(\Delta V) \tag{6.42}$$

$$m = \frac{g_G}{g_D} \frac{V_0 + \Delta V}{C_{load}\ \varphi_1(\Delta V)} = \frac{\varphi_4(\Delta V)}{C_{load}} \tag{6.43}$$

Substituting in (6.9) and (6.15) yields:

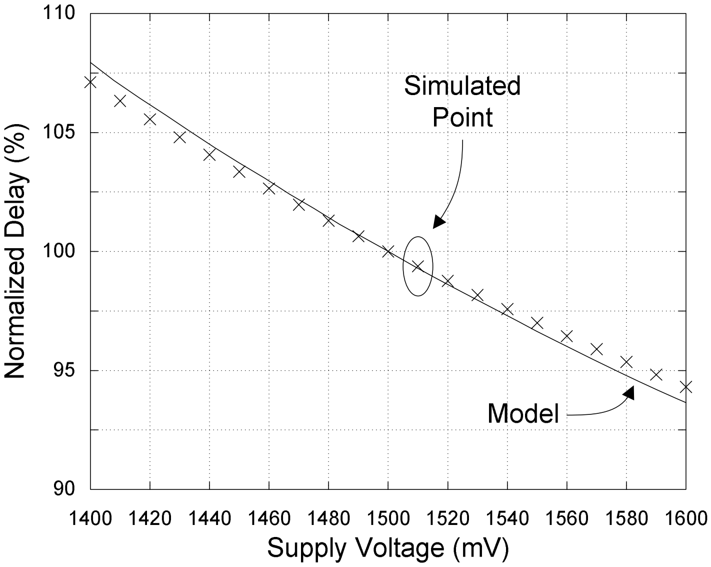


Figure 6.8: Model for inverter delay-versus-voltage relationship.

$$t_{out50\%,\,early} = C_{load}\left[\frac{b-\frac{1}{2}V_0}{\varphi_4}+\frac{1}{g_D}\left[W\left(\left(g_D\frac{\varphi_3\varphi_4-b+V_{init}}{\varphi_4}-1\right)e^{g_D\frac{\varphi_3\varphi_4-b+\frac{1}{2}V_0}{\varphi_4}-1}\right)+1\right]\right] = C_{load}\,\varphi_5(\Delta V) \qquad (6.44)$$

$$t_{out50\%,\,late} = C_{load}\left[\varphi_2+\frac{1}{g_D}\log\frac{\left(\varphi_4\left(\varphi_3-\frac{1}{g_D}\right)-b+V_{init}\right)e^{-g_D(\varphi_2-\varphi_3)}+\frac{\varphi_4}{g_D}}{\varphi_2\varphi_4-b+\frac{1}{2}V_0}\right] = C_{load}\,\varphi_6(\Delta V) \qquad (6.45)$$

Finally, (6.18) becomes:

$$d(t_{Tout}, C_{load}, \Delta V) = C_{load}\left[\varphi_5-\varphi_2+(\varphi_6-\varphi_5)u(\varphi_5-\varphi_1)\right] = C_{load}\Phi(\Delta V) \qquad (6.46)$$

This is the definition for d_{stage} and concludes the proof.

Equation (6.38) implicitly neglects the short-circuit current of the inverters. By approximating each output waveform by its tangent at the 50% point, it also underestimates the transition times. Nevertheless, as shown in Figure 6.8, the delay-versus-voltage relationship predicted by (6.38) closely matches the simulated one. This indicates that the ramp approximation is reasonable.

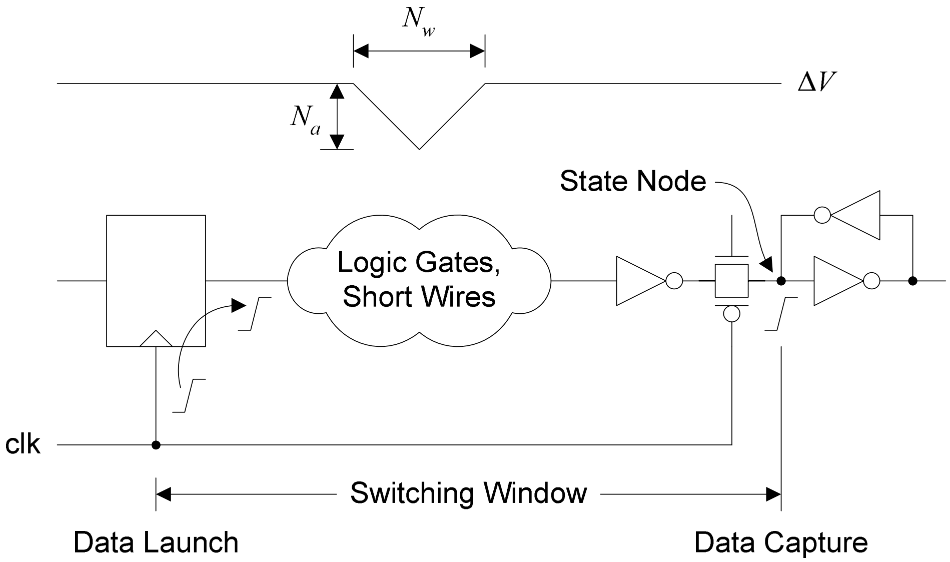


Figure 6.9: Simulation setup.

Because the inverter is faster than the other logic gates of Figure 6.7, its delay-versus-voltage relationship is only an estimate for the average relationship mentioned earlier. When more accuracy is needed, the average delay-versus-voltage relationship can be empirically characterized as follows:

$$D = D_0\left[1 - k_1\frac{\Delta V}{V_0 - V_t} + k_2\left(\frac{\Delta V}{V_0 - V_t}\right)^2\right] \tag{6.47}$$

The technology-dependent coefficients k_1 and k_2 have no units. In Figure 6.7, k_1 is between 0.76 and 1.06 while k_2 is between 1.03 and 1.37.

6.2.2 Impact of Power-Supply Noise on Delay

In (6.38) and (6.47), ΔV is assumed constant (time-invariant). When the supply voltage varies slowly with respect to the clock period, this is reasonable. Assuming a constant ΔV makes the timing impact of the power-supply noise easier to predict: the worst-case delay corresponds to the worst-case noise that can occur when a path switches. In other words, when the supply voltage varies slowly, the delay degradation is proportional to the *peak* of the noise.

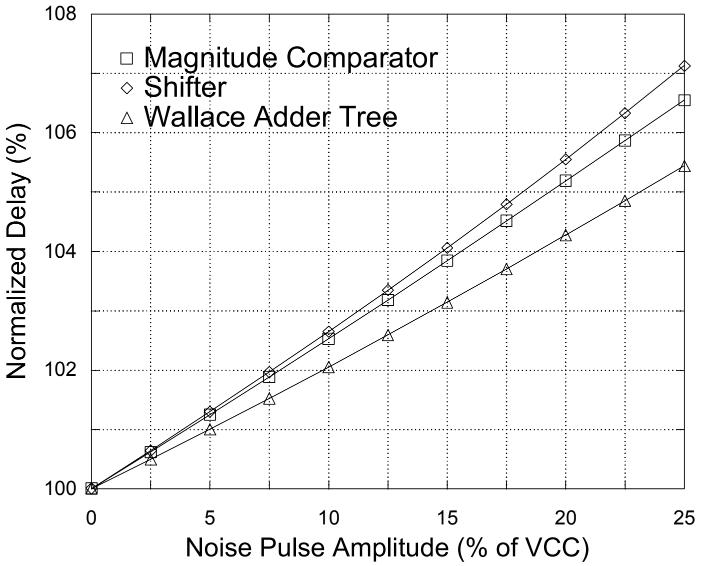


Figure 6.10: Delay versus noise pulse amplitude.

But in reality, ΔV varies with time. To better understand how a rapid supply voltage variation impacts timing, it is useful to run a few simulations. For this purpose, a triangular noise pulse of amplitude N_a and width N_w is applied to three sample circuits: a magnitude comparator, a shifter, and a Wallace adder tree [6.19]. The simulation setup is shown in Figure 6.9. As mentioned earlier, the power and ground voltages are assumed common for all devices because device-dominated paths are typically localized. On the rising edge of the clock, the first flip-flop launches its data. The data then propagates through logic gates and interconnects. It finally stops at the state node of the next sequential element, where it is captured. The time required to reach that state node defines the switching window of each circuit.

When the peak of the noise increases, the delay of each circuit increases. As shown in Figure 6.10, the variation is almost linear. Each curve is expected to have a different slope because each circuit has a different delay. Since the Wallace adder tree is the slowest, the duration of the noise perturbation is a smaller fraction of its delay and the overall impact of the perturbation is smaller. Because the magnitude comparator and the shifter have roughly the same delay, their noise sensitivity is comparable.

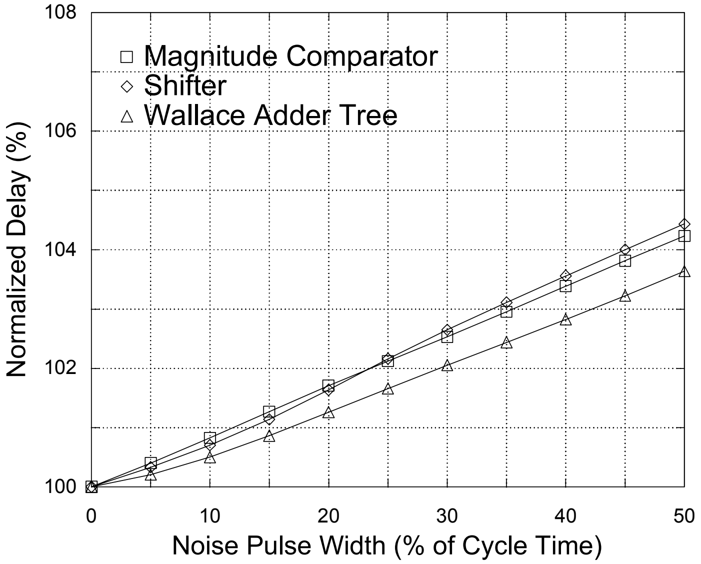


Figure 6.11: Delay versus noise pulse width.

When the width of the pulse increases, the delay increases as shown in Figure 6.11. The pulse width is given as a fraction of the 395-ps cycle time. As before, the variation is almost linear. The curves have different slopes because the delay of each path is different. The Wallace adder tree is still the least sensitive because it is the slowest circuit.

Mathematically, the delay D of each circuit can be written as:

$$D = D_0\left[1 - f(N_w)N_a - g(N_a)N_w - \varphi(N_a, N_w)\right] \tag{6.48}$$

where f and g are the path-dependant noise sensitivity factors and $\varphi(N_a, N_w)$ is a function capturing the higher-order delay dependencies. Of course, f is a function of N_w and g is a function of N_a:

$$\begin{aligned} f(N_w) &= f_0 + f_1 N_w + f_2 N_w^2 + \ldots \\ g(N_a) &= g_0 + g_1 N_a + g_2 N_a^2 + \ldots \end{aligned} \tag{6.49}$$

Since the delay variation is zero when $N_w = 0$ or when $N_a = 0$, $f_0 = g_0 = 0$. The fact that the delay of each circuit is practically linear with respect to both the amplitude and the width of the noise pulse is remarkable. It suggests that $\varphi(N_a, N_w) \approx 0$. It also implies that $f_n \approx g_n \approx 0$ for $n > 1$. Under these circumstances, (6.48) becomes:

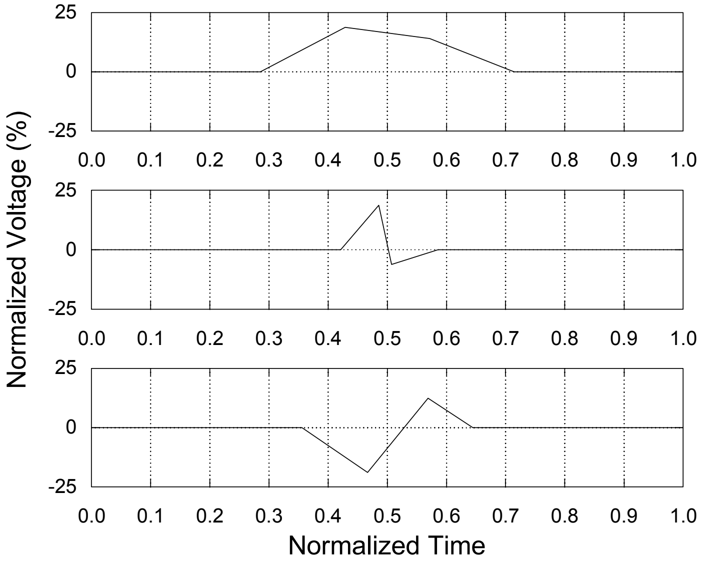


Figure 6.12: Sample noise waveforms.

$$D \approx D_0\left[1-(f_1+g_1)N_w N_a\right] \tag{6.50}$$

Equation (6.50) means that a short pulse with a large amplitude can have the same effect as a wider pulse with a smaller amplitude. From a timing standpoint, it is thus reasonable to postulate that the actual shape of the noise pulse is largely irrelevant and that only its average value is important.

6.2.3 Model Validation

To verify that only the average value of V is important, multiple random noise waveforms like the ones shown in Figure 6.12 are generated and applied to the three sample circuits. The waveforms are constructed using 4 points. Their amplitude is uniformly distributed between –25% and +25% of V_0. The duration of the noise perturbations is swept from 10% to 50% of the cycle time (395 ps) to model rapid supply voltage variations.

First, a random waveform is generated and the response of the three circuits is simulated. The average value of V is computed for each path during its switching interval. This interval starts when the rising edge of the clock reaches 50% × V_0 and ends when the state node of the receiving sequential element reaches its 50% point. When a circuit is done switching, the power-supply noise no longer impacts its timing. Since each circuit is

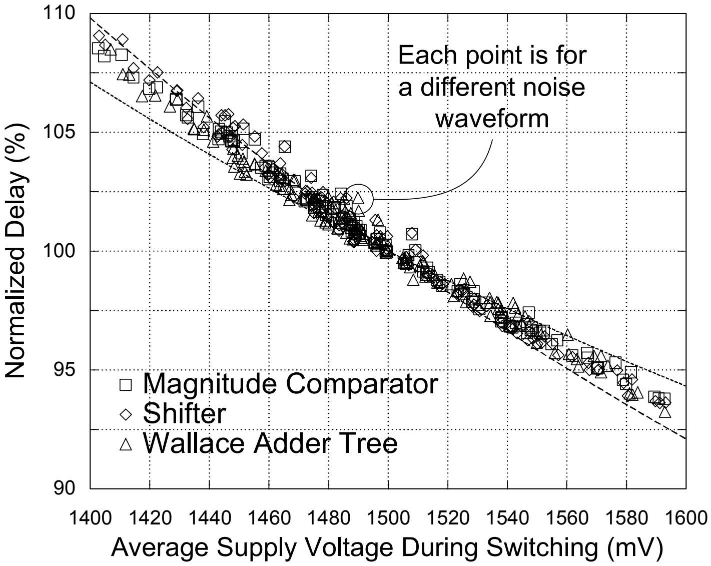


Figure 6.13: Delay versus average supply voltage for three sample circuits.

different, the switching intervals are different. Because the average supply voltage is computed during the switching intervals, the average supply voltages are different as well. The length of the switching interval of each circuit is recorded as its delay together with the average value of V. Next, another random waveform is generated and a new simulation is performed. The process is repeated several times.

The results are shown in Figure 6.13, superimposed on the delay-versus-voltage curves of Figure 6.7. Clearly, the two correlate well. This good correlation is remarkable because it holds even when the *peak* of the power-supply noise reaches 25% of V_0, a value well above the 10% often budgeted in practice.

Because of this good correlation, the average noise during switching N_{avg} can be used to predict the timing impact:

$$D = D_0\left[1 - k_1\frac{N_{avg}}{V_0 - V_t} + k_2\left(\frac{N_{avg}}{V_0 - V_t}\right)^2\right] \approx D_0\left[1 - k_1\frac{N_{avg}}{V_0 - V_t}\right] \tag{6.51}$$

When N_{avg} is small, the quadratic component can be neglected.

6.2.4 Metric for the Performance of Power Distribution Networks

Traditionally, the performance of power distribution networks is measured by the worst-case instantaneous supply voltage drop. However, as shown in the previous section, a large instantaneous supply voltage drop may or may not significantly perturb delay. The duration of the supply voltage drop is as important as its magnitude.

Equation (6.51) suggests a new metric for the performance of power distribution networks: the peak of an *n*-cycle moving average. This metric considers the timing impact of the power-supply noise, not just its magnitude.

In *non-transparent* designs, the critical paths have a length of about (but slightly less than) one clock cycle. The switching window of these critical paths is thus one approximately clock cycle. Taking the average of the power-supply noise during their switching window is therefore equivalent to taking a moving average of the power-supply noise with $n = 1$. Mathematically, the moving average at time x is:

$$N_{avg}(x,n) = \frac{1}{nT_{cycle}} \int_{x-\frac{1}{2}nT_{cycle}}^{x+\frac{1}{2}nT_{cycle}} (\Delta V_{CC}(t) - \Delta V_{SS}(t))dt \qquad (6.52)$$

where T_{cycle} is the cycle time. The peak drop of the moving average measures the timing impact of the noise:

$$\hat{N}_{avg}(n) = \min_{x}\left[N_{avg}(x,n)\right] \qquad (6.53)$$

Some design styles (e.g. latch-based or domino logic) allow critical paths spanning multiple cycles ($n > 1$). Since these long *transparent* paths have a multi-cycle switching window, a multi-cycle moving average is required to quantify how they are impacted by power-supply noise. Critical paths spanning a phase (i.e. half a cycle) also occur in practice. Their moving average requires $n = 0.5$.

For a general design having critical paths spanning different number of cycles $n_1, \ldots, n_k$ (which may or may not include $n = 1$), the effective peak drop (from a timing standpoint) is given by:

$$\hat{N}_{avg} = \min_{i=1,\ldots,k} \left[\hat{N}_{avg}(n_i)\right] \tag{6.54}$$

Equation (6.54) means that the critical paths of an integrated circuit can be analyzed separately, based on their length. The peak drop of the $n = n_i$ moving average measures the performance of the power distribution network for the critical paths having a length n_i. This performance measure is based on how the power-supply noise impacts their timing. For the critical paths having a different length, a different moving average is used. The overall performance of the power distribution network is set by the maximum peak drop (i.e. the minimum supply voltage) observed for the switching windows of interest $n_1, \ldots, n_k$.

Equation (6.54) suggests that the lack of performance of a given power distribution network can be mitigated by increasing the transparency of the critical paths. Making the critical paths longer increases their switching window, which in turn increases the noise averaging effect.

6.3 Modeling for Interconnect-Dominated Paths

Interconnect-dominated paths are constructed by cascading repeaters and long wires. These repeaters are typically inverters or buffers. (Here, a buffer is defined as a non-inverting repeater.) The length, width, and spacing of the wires together with the transistor sizes of the repeaters are optimized during design. The goal is usually to maximize performance under metal usage, power, and noise constraints. This optimization process usually makes the repeater and wire delays comparable [6.20].

As mentioned earlier, interconnect-dominated paths are almost always distributed. The repeaters tend to be physically separated. Because of this separation, the devices along an interconnect-dominated path generally receive different power (V_{CC}) and ground (V_{SS}) voltages.

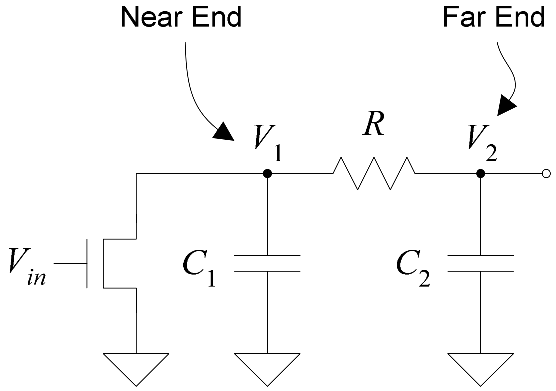


Figure 6.14: Interconnect model.

6.3.1 Differential Equation for the Far-End Voltage

This section derives the differential equation obeyed by the far-end voltage $V_2(t)$ of Figure 6.14. It assumes the input waveform and the power-supply noise of Section 6.1, which are shown again for convenience in Figure 6.15.

The currents flowing through C_1 and C_2 are:

$$\begin{aligned} i_1(t) &= -C_1 V_1'(t) \\ i_2(t) &= -C_2 V_2'(t) \end{aligned} \tag{6.55}$$

The voltage drop through the resistance establishes the relationship between the near-end voltage $V_1(t)$ and $V_2(t)$:

$$V_1(t) = V_2(t) - Ri_2(t) = V_2(t) + RC_2V_2'(t) \tag{6.56}$$

Taking the derivative yields:

$$V_1'(t) = V_2'(t) + RC_2V_2''(t) \tag{6.57}$$

and $i_1(t)$ becomes:

$$i_1(t) = -C_1V_2'(t) - RC_1C_2V_2''(t) \tag{6.58}$$

The drain current through the device is $i(t) = i_1(t) + i_2(t)$:

$$I_D(V_{GS}(t), V_{DS}(t)) = i_1(t) + i_2(t) \tag{6.59}$$

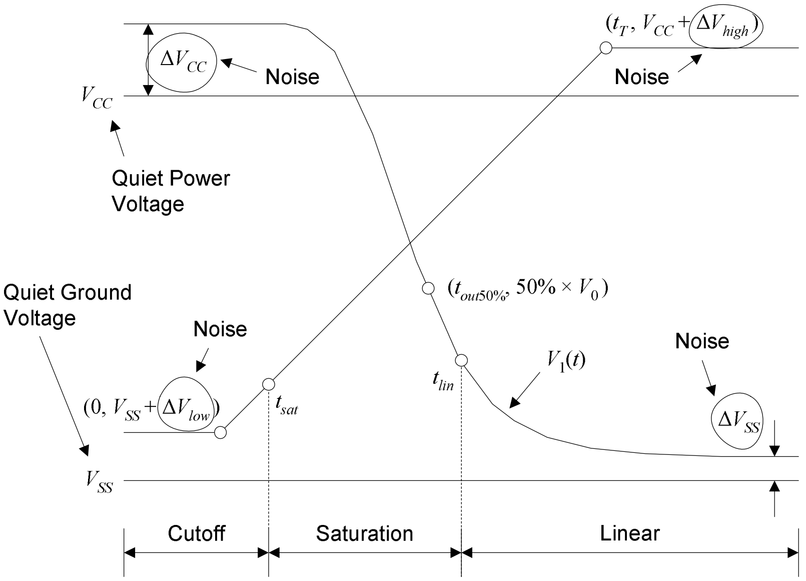


Figure 6.15: Device response.

Substituting the expressions for $i_1(t)$ and $i_2(t)$:

$$I_D(V_{GS}(t),V_{DS}(t)) = -C_1V_2'(t) - RC_1C_2V_2''(t) - C_2V_2'(t) \tag{6.60}$$

The drain current in saturation is given in Chapter 2 by (2.16):

$$I_D(V_{GS}(t),V_{DS}(t)) = g_G(V_{GS}(t) - V_T) + g_D(V_{DS}(t) - V_{DS0}) \tag{6.61}$$

Here, $V_{GS}(t) = V_G(t) - \Delta V_{SS}$ and $V_{DS}(t) = V_1(t) - \Delta V_{SS} = V_2(t) + RC_2V_2'(t) - \Delta V_{SS}$. This definition for $V_{DS}(t)$ assumes that $V_1(t)$ is referenced to V_{SS}, the quiet ground voltage. Thus,

$$I_D(V_{GS}(t),V_{DS}(t)) = g_G(V_G(t) - \Delta V_{SS} - V_T) + g_D(V_2(t) + RC_2V_2'(t) - \Delta V_{SS} - V_{DS0}) \tag{6.62}$$

Combining (6.60) and (6.62) yields the following differential equation:

$$-C_1V_2' - RC_1C_2V_2'' - C_2V_2' = g_G(V_G(t) - \Delta V_{SS} - V_T) + g_D(V_2 + RC_2V_2' - \Delta V_{SS} - V_{DS0}) \tag{6.63}$$

The equation can be rearranged as follows:

$$-\frac{RC_1C_2}{g_D}V_2'' - \frac{C_1 + C_2 + g_D RC_2}{g_D}V_2' - V_2 = \frac{g_G}{g_D}(V_G(t) - \Delta V_{SS} - V_T) - V_{DS0} - \Delta V_{SS} \tag{6.64}$$

and simplified to:

$$k_2 V_2'' + k_1 V_2' - V_2 = \frac{g_G}{g_D}(V_G(t) - \Delta V_{SS} - V_T) - V_{DS0} - \Delta V_{SS} \tag{6.65}$$

where:

$$\begin{aligned} k_1 &= -\frac{C_1 + C_2 + g_D R C_2}{g_D} \\ k_2 &= -\frac{R C_1 C_2}{g_D} \end{aligned} \tag{6.66}$$

Before time t_T, the gate voltage is given by (6.2). Substituting the expression for $V_G(t)$ yields:

$$k_2 V_2'' + k_1 V_2' - V_2 = \frac{g_G}{g_D}\left((V_0 + \Delta V_{high} - \Delta V_{low})\frac{t}{t_T} + \Delta V_{low} - \Delta V_{SS} - V_T\right) - V_{DS0} - \Delta V_{SS} \tag{6.67}$$

With:

$$\begin{aligned} m &= \frac{g_G}{g_D}\frac{V_0 + \Delta V_{high} - \Delta V_{low}}{t_T} \\ b &= \frac{g_G}{g_D}(V_T - \Delta V_{low} + \Delta V_{SS}) + V_{DS0} + \Delta V_{SS} \end{aligned} \tag{6.68}$$

the differential equation finally becomes:

$$k_2 V_2'' + k_1 V_2' - V_2 = mt - b \tag{6.69}$$

The constants m and b have the same definition in Section 6.1. The initial conditions are $V_2(t_{sat}) = V_{init} = V_0 + \Delta V_{CC}$ and $V_2'(t_{sat}) = 0$. As before, t_{sat} is the time at which the device starts conducting in saturation:

$$t_{sat} = \frac{V_T - \frac{g_D}{g_G}(V_0 + \Delta V_{CC} - \Delta V_{SS} - V_{DS0}) - \Delta V_{low} + \Delta V_{SS}}{V_0 + \Delta V_{high} - \Delta V_{low}} t_T = \frac{b - V_{init}}{m} \tag{6.70}$$

At time t_T, the gate voltage saturates. Beyond that point, (6.69) ceases to be valid. The far-end voltage is instead described by:

$$k_2 V_2'' + k_1 V_2' - V_2 = m t_T - b \tag{6.71}$$

At t_T, the far-end voltage and its derivative must be continuous.

Regardless of t_T, (6.69) and (6.71) stop to hold when the device enters its linear region of operation.

6.3.2 Behavior in the Saturation Region

This section derives expressions for the far-end voltage $V_2(t)$ satisfying the differential equations of the previous section. The solution describing $V_2(t)$ before t_T is called the early solution. The one describing $V_2(t)$ after t_T is referred to as the late solution. The early and late solutions are used to compute the near-end voltage $V_1(t)$ and to determine when the device enters its linear region of operation.

Equation (6.69) is a second-order linear differential equation with constant coefficients that is non-homogeneous. The solution is:

$$V_{2,\,early}(t) = \frac{m}{2}\left(k_1 + \frac{2k_2 + k_1^2}{d_1}\right)e^{(t-t_{sat})/\tau_1} + \frac{m}{2}\left(k_1 + \frac{2k_2 + k_1^2}{d_2}\right)e^{(t-t_{sat})/\tau_2} - m(t + k_1) + b \tag{6.72}$$

where:

$$\begin{aligned} \tau_i &= \frac{2k_2}{d_i - k_1} \\ d_i &= (-1)^{i+1}\sqrt{k_1^2 + 4k_2} \end{aligned} \tag{6.73}$$

When it is contextually clear that the derivation refers to the early solution, $V_{2,\,early}(t)$ is simply written $V_2(t)$ for conciseness.

The derivative of $V_2(t)$ is:

$$V_2'(t) = \frac{m}{2\tau_1}\left(k_1 + \frac{2k_2 + k_1^2}{d_1}\right)e^{(t-t_{sat})/\tau_1} + \frac{m}{2\tau_2}\left(k_1 + \frac{2k_2 + k_1^2}{d_2}\right)e^{(t-t_{sat})/\tau_2} - m \tag{6.74}$$

The near-end voltage is obtained by substituting $V_2(t)$ and $V_2'(t)$ in (6.56). The result is:

$$V_1(t) = \sum_{i=1}^{2}\frac{m}{2}\left(k_1 + \frac{2k_2 + k_1^2}{d_i}\right)\left(1 - g_D\frac{d_i - k_1}{2C_1}\right)e^{(t-t_{sat})/\tau_i} - m\left(t - \frac{C_1 + C_2}{g_D}\right) + b \tag{6.75}$$

The time t_{lin} at which the device enters its linear region of operation is the time at which the drain-to-source voltage reaches V_{DS0}:

$$V_1(t_{lin}) - \Delta V_{SS} = V_{DS0} \tag{6.76}$$

Unfortunately, the solution to this equation cannot be expressed in closed form using elementary functions. However, a sequence that converges very rapidly toward the solution can be obtained using Halley's method [6.21].

Halley's method is easier to apply by introducing the following functional:

$$\mathrm{H}(f(t), t_0) = \lim_{n\to\infty} t_n \tag{6.77}$$

where the first element of the sequence is provided as an argument and the following ones obey this relationship:

$$t_{n+1} = t_n - \frac{2f(t_n)f'(t_n)}{2[f'(t_n)]^2 - f(t_n)f''(t_n)} \tag{6.78}$$

If t_0 is chosen appropriately, the sequence is guaranteed to converge to a root of $f(t)$. In practice, a single iteration is typically sufficient to reach the limit:

$$\mathrm{H}(f(t), t_0) \approx t_0 - \frac{2f(t_0)f'(t_0)}{2[f'(t_0)]^2 - f(t_0)f''(t_0)} \tag{6.79}$$

Assuming that t_T is part of the neighborhood that makes the sequence converge to t_{lin}, the solution to (6.76) is:

$$t_{lin,\,early} = \mathrm{H}(V_{1,\,early}(t) - V_{DS0} - \Delta V_{SS}, t_T) \tag{6.80}$$

where the first and second derivatives of $V_1(t)$ are:

$$\begin{aligned} V_1'(t) &= \sum_{i=1}^{2} \frac{m}{2\tau_i}\left(k_1 + \frac{2k_2 + k_1^2}{d_i}\right)\left(1 - g_D\frac{d_i - k_1}{2C_1}\right)e^{(t-t_{sat})/\tau_i} - m \\ V_1''(t) &= \sum_{i=1}^{2} \frac{m}{2\tau_i^2}\left(k_1 + \frac{2k_2 + k_1^2}{d_i}\right)\left(1 - g_D\frac{d_i - k_1}{2C_1}\right)e^{(t-t_{sat})/\tau_i} \end{aligned} \tag{6.81}$$

The subscript explicitly indicates that $t_{lin,\,early}$ is valid only if it occurs before t_T.

Equation (6.69) ceases to be applicable when the gate voltage stops rising, as discussed earlier. After time t_T, the differential equation describing the far-end voltage is:

$$k_2V_2''+k_1V_2'-V_2 = mt_T - b \tag{6.82}$$

The solution that makes $V_2(t)$ and its derivative continuous at time t_T is:

$$V_{2,\,late}(t) = \sum_{i=1}^{2} \frac{(d_i+k_1)(mt_T - b + V_{2,\,early}(t_T)) + 2k_2V'_{2,\,early}(t_T)}{2d_i} e^{(t-t_T)/\tau_i} - mt_T + b \tag{6.83}$$

where $V_{2,\,early}(t_T)$ and $V'_{2,\,early}(t_T)$ are given by (6.72) and (6.74).

After time t_T, $V_1(t)$ is given by $V_2(t) + RC_2V_2'(t)$:

$$V_1(t) = \sum_{i=1}^{2} \frac{(d_i+k_1)(mt_T - b + V_{2,\,early}(t_T)) + 2k_2V'_{2,\,early}(t_T)}{2d_i} \left(1 - g_D\frac{d_i-k_1}{2C_1}\right) e^{(t-t_T)/\tau_i} - mt_T + b \tag{6.84}$$

The first and second derivatives of $V_1(t)$ are:

$$\begin{aligned} V_1'(t) &= \sum_{i=1}^{2} \frac{(d_i+k_1)(mt_T - b + V_{2,\,early}(t_T)) + 2k_2V'_{2,\,early}(t_T)}{2d_i\,\tau_i} \left(1 - g_D\frac{d_i-k_1}{2C_1}\right) e^{(t-t_T)/\tau_i} \\ V_1''(t) &= \sum_{i=1}^{2} \frac{(d_i+k_1)(mt_T - b + V_{2,\,early}(t_T)) + 2k_2V'_{2,\,early}(t_T)}{2d_i\,\tau_i^2} \left(1 - g_D\frac{d_i-k_1}{2C_1}\right) e^{(t-t_T)/\tau_i} \end{aligned} \tag{6.85}$$

Using Halley's method again yields the sequence for the solution after t_T:

$$t_{lin,\,late} = \mathrm{H}(V_{1,\,late}(t) - V_{DS0} - \Delta V_{SS}, t_T) \tag{6.86}$$

Combining (6.80) and (6.86) using the step function yields an expression for t_{lin} that is valid whether t_{lin} occurs before or after t_T:

$$t_{lin} = t_{lin,\,early} + (t_{lin,\,late} - t_{lin,\,early})\,u(t_{lin,\,early} - t_T) \tag{6.87}$$

6.3.3 Behavior in the Linear Region

After time t_{lin}, the drain current of the device is given by (2.18):

$$I_D = g_G(V_{GS} - V_T)\frac{V_{DS}}{V_{DS0}} = \frac{V_{DS}}{R_{lin}} \tag{6.88}$$

For simplicity, it is assumed that the gate voltage is approximately constant once the device is in the linear region: $V_{GS}(t) \approx V_{GS}(t_{lin})$ for $t > t_{lin}$.

The constant gate voltage makes R_{lin} constant as well:

$$R_{lin} = \frac{1}{g_G} \frac{V_{DS0}}{V_G(t_{lin}) - \Delta V_{SS} - V_T} \tag{6.89}$$

where $V_G(t_{lin})$ is:

$$V_G(t_{lin}) = (V_0 + \Delta V_{high} - \Delta V_{low}) \frac{\min(t_{lin}, t_T)}{t_T} + \Delta V_{low} \tag{6.90}$$

The constant gate voltage assumption introduces an inaccuracy only when t_{lin} occurs before t_T. Otherwise, it is exact. As before,

$$V_{DS}(t) = V_1(t) - \Delta V_{SS} = V_2(t) + RC_2 V_2'(t) - \Delta V_{SS} \tag{6.91}$$

Equating $V_{DS}(t)$ / R_{lin} to (6.60) produces the following differential equation:

$$\frac{V_2 + RC_2 V_2' - \Delta V_{SS}}{R_{lin}} = -RC_1 C_2 V_2'' - (C_1 + C_2) V_2' \tag{6.92}$$

Rearranging yields:

$$k_4 V_2'' + k_3 V_2' + V_2 = \Delta V_{SS} \tag{6.93}$$

where:

$$\begin{aligned} k_4 &= R_{lin} R C_1 C_2 \\ k_3 &= R_{lin}(C_1 + C_2) + RC_2 \end{aligned} \tag{6.94}$$

The solution that makes V_2 and its derivative continuous at t_{lin} is:

$$V_{2,lin}(t) = \sum_{i=1}^{2} \frac{(\delta_i + k_3)(V_2(t_{lin}) - \Delta V_{SS}) + 2k_4 V_2'(t_{lin})}{2\delta_i} e^{\frac{\delta_i - k_3}{2k_4}(t - t_{lin})} + \Delta V_{SS} \tag{6.95}$$

where:

$$\delta_i = (-1)^{i+1} \sqrt{k_3^2 - 4k_4} \tag{6.96}$$

At t_{lin},

$$V_2(t_{lin}) = V_{2,\,early}(t_{lin,\,early}) + (V_{2,\,late}(t_{lin,\,late}) - V_{2,\,early}(t_{lin,\,early}))\, u(t_{lin,\,early} - t_T) \tag{6.97}$$

and

$$V_2'(t_{lin}) = V_{2,\,early}'(t_{lin,\,early}) + (V_{2,\,late}'(t_{lin,\,late}) - V_{2,\,early}'(t_{lin,\,early}))\, u(t_{lin,\,early} - t_T) \tag{6.98}$$

6.3.4 Impact of Power-Supply Noise on Delay and Transition Time

The far-end voltage can cross its 50% point at time $t_{out50\%}$ while the device is in saturation or in the linear region.

In saturation, the 50% point can occur before t_T:

$$t_{out50\%,\ early} = \mathrm{H}(V_{2,\ early}(t) - \tfrac{1}{2}V_0, t_T) \tag{6.99}$$

or after t_T:

$$t_{out50\%,\ late} = \mathrm{H}(V_{2,\ late}(t) - \tfrac{1}{2}V_0, t_T) \tag{6.100}$$

The same is true in the linear region, but the constant gate voltage assumption yields a single equation:

$$t_{out50\%,\ lin} = \mathrm{H}(V_{2,\ lin}(t) - \tfrac{1}{2}V_0, t_{lin}) \tag{6.101}$$

The 50% point occurs at $t_{out50\%,\ early}$ if and only if $t_{out50\%,\ early}$ occurs before t_T and t_{lin}. The 50% point occurs at $t_{out50\%,\ late}$ if and only if $t_{out50\%,\ late}$ occurs after t_T and before t_{lin}. Otherwise, the 50% point occurs at $t_{out50\%,\ lin}$. Mathematically,

$$\begin{aligned} t_{out50\%} = {} & t_{out50\%,\ early}u(t_T - t_{out50\%,\ early})u(t_{lin} - t_{out50\%,\ early}) \\ & + t_{out50\%,\ late}u(t_{out50\%,\ late} - t_T)u(t_{lin} - t_{out50\%,\ late}) \\ & + t_{out50\%,\ lin}u(t_{out50\%,\ lin} - t_{lin}) \end{aligned} \tag{6.102}$$

The delay d_{rise} corresponding to an input rising transition is simply $t_{out50\%} - t_{in50\%}$. As before,

$$t_{in50\%} = \frac{\frac{1}{2}V_0 - \Delta V_{low}}{V_0 + \Delta V_{high} - \Delta V_{low}} t_T \tag{6.103}$$

Thus,

$$\begin{aligned} d = {} & t_{out50\%,\ early}u(t_T - t_{out50\%,\ early})u(t_{lin} - t_{out50\%,\ early}) \\ & + t_{out50\%,\ late}u(t_{out50\%,\ late} - t_T)u(t_{lin} - t_{out50\%,\ late}) \\ & + t_{out50\%,\ lin}u(t_{out50\%,\ lin} - t_{lin}) - t_{in50\%} \end{aligned} \tag{6.104}$$

The transition time at the 50% point is given by:

$$t_{Tout} = -\frac{V_0 + \Delta V_{CC} - \Delta V_{SS}}{V_2'(t_{out50\%})} \tag{6.105}$$

where:

$$\begin{aligned} V_2'(t_{out50\%}) = {} & V_{2,\,early}'(t_{out50\%,\,early})u(t_T - t_{out50\%,\,early})u(t_{lin} - t_{out50\%,\,early}) \\ & + V_{2,\,late}'(t_{out50\%,\,late})u(t_{out50\%,\,late} - t_T)u(t_{lin} - t_{out50\%,\,late}) \\ & + V_{2,\,lin}'(t_{out50\%,\,lin})u(t_{out50\%,\,lin} - t_{lin}) \end{aligned} \tag{6.106}$$

6.3.5 Duality Principle

The delay and transition time expressions of (6.104) and (6.105) assume an n-device switching in response to an input *rising* transition. The corresponding expressions for a p-device in response to an input *falling* transition can be derived based on the following duality principle: increasing V_{high} or V_{CC} has the same effect on an n-device as decreasing V_{low} or V_{SS} on a p-device. Thus, the delay and output transition time when the input is falling are given by changing the device parameters (g_{G0}, g_{D0}, V_T, V_{DS0}) and by applying the following transformations:

$$\begin{aligned} \Delta V_{low} &\to -\Delta V_{high} \\ \Delta V_{high} &\to -\Delta V_{low} \\ \Delta V_{CC} &\to -\Delta V_{SS} \\ \Delta V_{SS} &\to -\Delta V_{CC} \end{aligned} \tag{6.107}$$

In [6.18], g_{G0} and g_{D0} are two times smaller for a p-device, compared to an n-device. V_T and V_{DS0} are similar however.

6.3.6 Model Validation

Figure 6.16 shows the impact of power-supply noise on the delay of a 2400-µm M5 interconnect driven by a 20-µm n-device. The transition time at the input of the n-device is 90 ps. The interconnect parameters are: R = 92.5 Ω, C_1 = 276 fF, and C_2 = 316 fF. These parameters correspond to a width, spacing, and thickness of 0.70 µm, 0.70 µm, and 0.90 µm, respectively. The model is clearly accurate.

The power-supply noise is applied to the four injection points (ΔV_{low}, ΔV_{high}, ΔV_{CC}, ΔV_{SS}). The noise is applied to one point at a time while the remaining ones remain quiet.

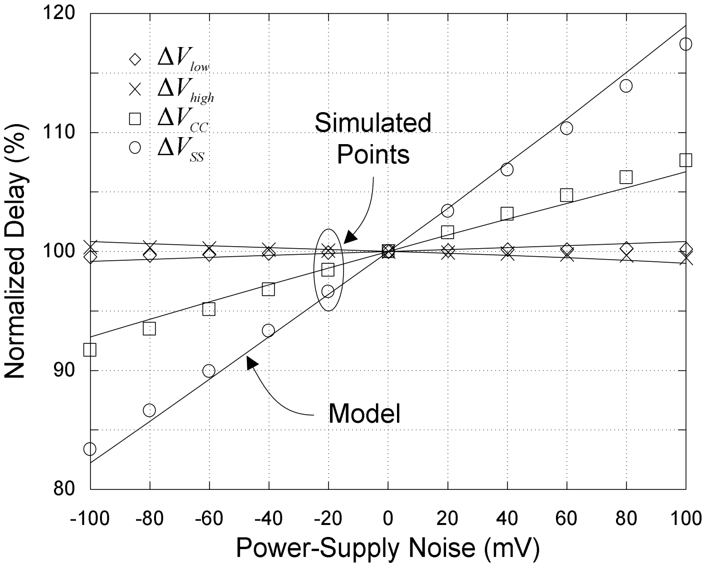


Figure 6.16: Impact of power-supply noise on inverter falling delay.

Figure 6.16 shows that the noise injected remotely (ΔV_{low}, ΔV_{high}) has almost no impact of the delay. It also shows that the opposite is true for the local power-supply noise: when ΔV_{SS} is 100 mV (6.7% of V_0 = 1.500 V), the delay increases by 17.5%. When V_{SS} increases, the gate-to-source voltage of the n-device is reduced. Because of this, its drive current greatly diminishes. The drive current is even further reduced because the drain-to-source voltage also decreases. The impact of the noise on V_{CC} is not as large. The noise on V_{CC} does not change the gate-to-source voltage of the device. However, a V_{CC} increase slows down the output falling transition because it increases the amount of charge stored on C_1 and C_2. It is important to note that the impact of the power-supply noise is asymmetrical.

Figure 6.17 shows the accuracy of the model for several long wire segments that are randomly generated. The input transition time and the driver size are also randomly generated. A point located on the equality line indicates a perfect match. The results show that the model is accurate for a wide range of delays, under quiet conditions.

Figure 6.17 does not include any power-supply noise, but Figure 6.18 does. Like the other model parameters, the power-supply noise is random. The noise is applied

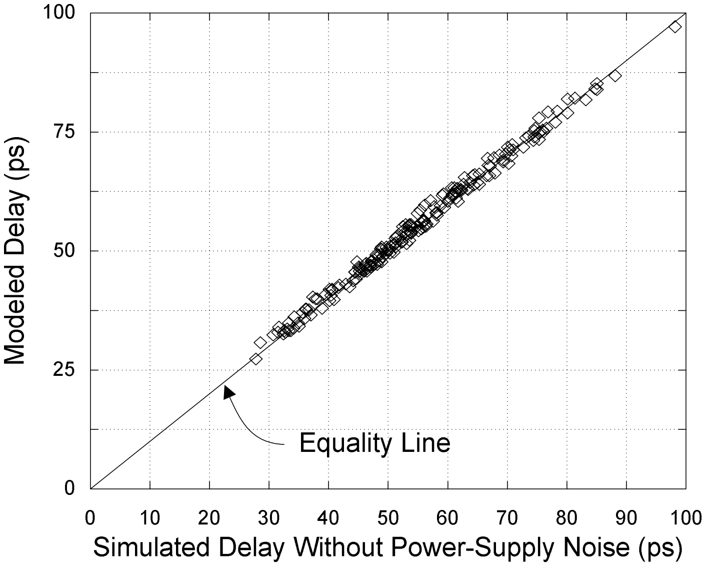


Figure 6.17: Scatter plot for inverter falling delay without power-supply noise.

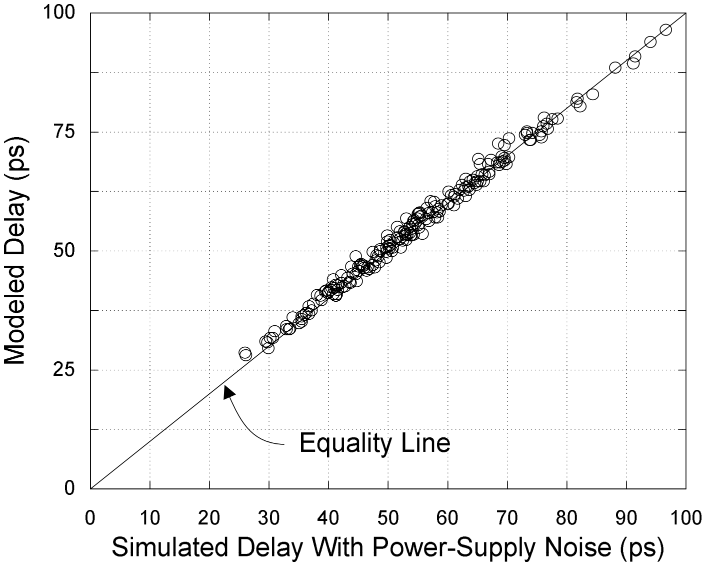


Figure 6.18: Scatter plot for inverter falling delay with power-supply noise.

simultaneously and independently to the four injection points. Clearly, the model is still accurate.

The impact of power-supply noise on the switching of a 40-µm p-device is shown in Figure 6.19. The device is driving the 2400-µm M5 interconnect discussed earlier

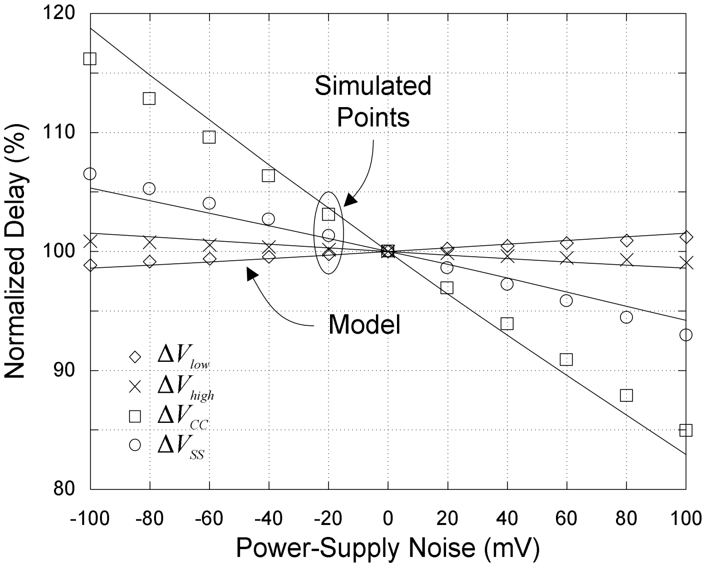


Figure 6.19: Impact of power-supply noise on inverter rising delay.

(R = 92.5 Ω, C_1 = 276 fF, C_2 = 316 fF). The duality principle yields results that are clearly accurate. The impact of the noise injected remotely remains small. The noise injected locally still produces large delay variations. As the local supply voltages increase, it is interesting to observe that the delay of the n-device of Figure 6.16 increases while the delay of the p-device decreases. These opposite responses suggest an opportunity for power-supply noise cancellation.

The scatter plots for the far-end transition times are shown in Figure 6.20 and Figure 6.21. The model predictions are distributed around the equality line, but with more dispersion than in Figure 6.17 and Figure 6.18. The additional dispersion is due to the derivative used in (6.105). Any small error on the far-end voltage waveform at the 50% point is magnified as the slope is extrapolated to produce the transition time.

6.3.7 Convergence of Halley's Method

As shown in Table 6.1, the sequences for $t_{lin,early}$ and $t_{lin,late}$ obtained using Halley's method converge very quickly for typical device and interconnect parameters. With t_T = 90 ps, w_n = 20 μm, R = 92.5 Ω, C_1 = 276 fF, C_2 = 316 fF, and no power-supply noise,

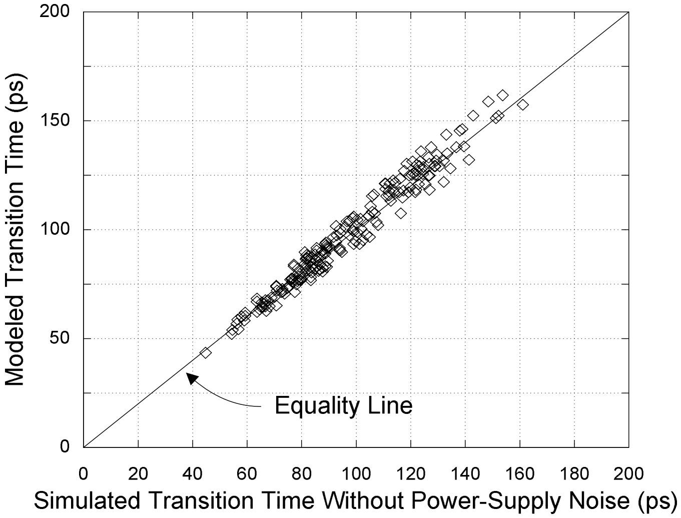


Figure 6.20: Scatter plot for output falling transition time without power-supply noise.

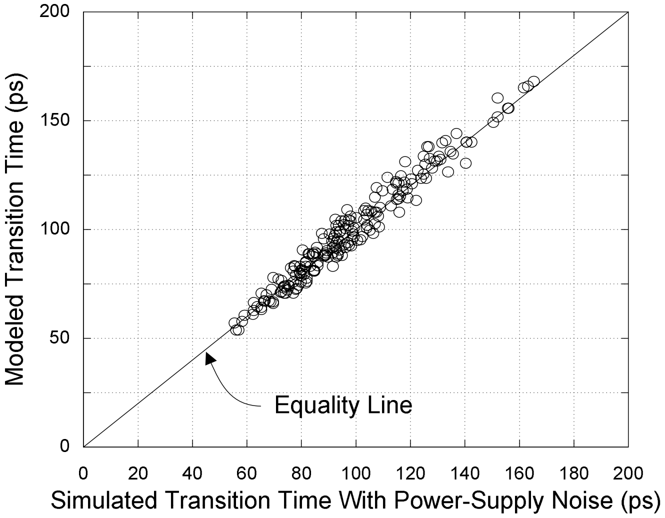


Figure 6.21: Scatter plot for output falling transition time with power-supply noise.

the error almost vanishes after a single iteration. A similar rate of convergence is observed for (6.99), (6.100), and (6.101).

6.3.8 Impact of Physical Design Parameters

The amount of delay variation due to power-supply noise is determined in part by the physical design parameters of the interconnect. To analyze the impact of the physical

Table 6.1: Convergence of Haley's method.

Iteration	$t_{lin,early}$ (ps)	Error (%)	$t_{lin,late}$ (ps)	Error (%)
0	90.0000	15.98	90.0000	12.41
1	77.6579	0.08	79.9345	0.16
2	77.5964	0.00	80.0628	0.00
3	77.5964	0.00	80.0628	0.00
4	77.5964	0.00	80.0628	0.00
5	77.5964	0.00	80.0628	0.00

design parameters, it is useful to associate a jitter coefficient to each noise injection point. Each jitter coefficient is computed as follows. First, a small amount of noise is injected. Then, the resulting delay variation is measured. Finally, the jitter coefficient is produced by taking the ratio of the two. It is expressed in picoseconds per Volt. Each coefficient measures the sensitivity of the delay to the noise at a particular injection point.

Figure 6.22 shows how much delay variation is produced for each noise injection point when the interconnect width varies. The length and spacing are fixed at 2400 µm and 0.70 µm, respectively. For each point, the input transition time is chosen to match the output transition time. This mimics a chain of repeaters driving identical wires, where the output of one stage feeds the next. As the interconnect increases from the minimum of 0.50 µm, the dominant jitter coefficient decreases significantly. At 1.00 µm, the decrease is 29% for V_{SS} and 18% for V_{CC}. The model matches the simulation results fairly well.

A similar graph for the spacing is shown in Figure 6.23. The length and width are fixed at 2400 µm and 0.70 µm, respectively. When the spacing is increased from 0.50 µm to 1.00 µm, the jitter coefficient for V_{CC} decreases by 19%. For V_{SS}, the improvement is 18%.

Increasing the interconnect length significantly increases the jitter coefficient, as shown in Figure 6.24. It also increases the jitter coefficient per millimeter, as shown in Figure 6.25.

When the physical interconnect parameters are varied, the transition times also vary as a result. Figure 6.26 shows that the correlation between the transition times corresponding to the jitter coefficients of Figure 6.22 to Figure 6.25 is very strong. This suggests that the

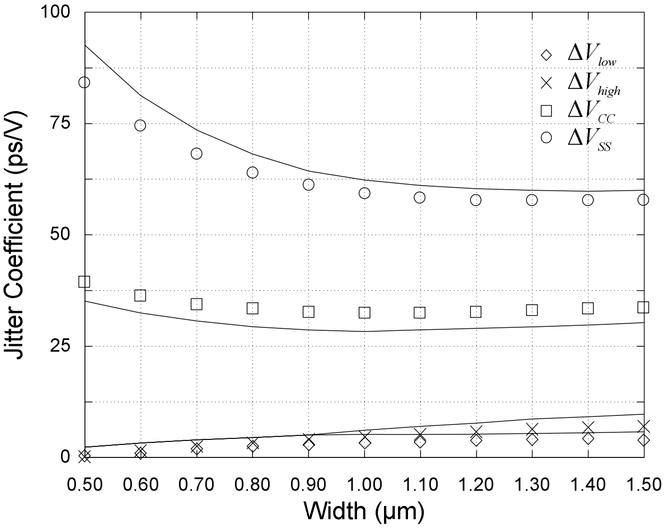


Figure 6.22: Impact of interconnect width on jitter coefficient.

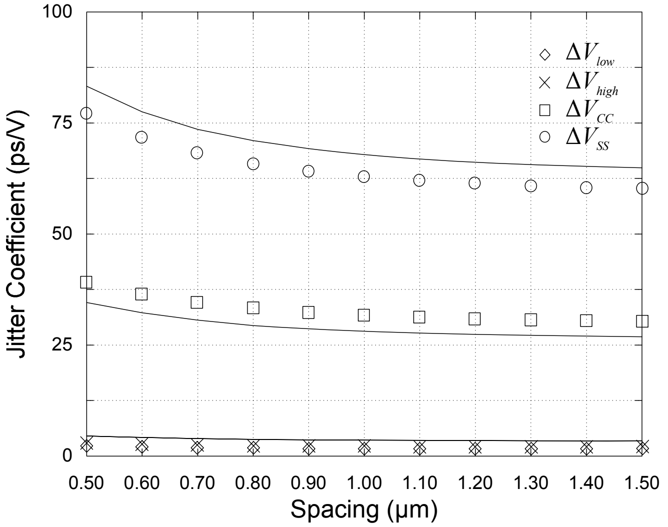


Figure 6.23: Impact of interconnect spacing on jitter coefficient.

physical interconnect parameters impact the jitter coefficient primarily by affecting the transition time.

6.3.9 Impact of Power-Supply Noise on Clock Interconnects

Interconnect-dominated paths tend to be less sensitive than device-dominated paths to chip-scale power-supply noise events. The noise directly affects the delay and the output

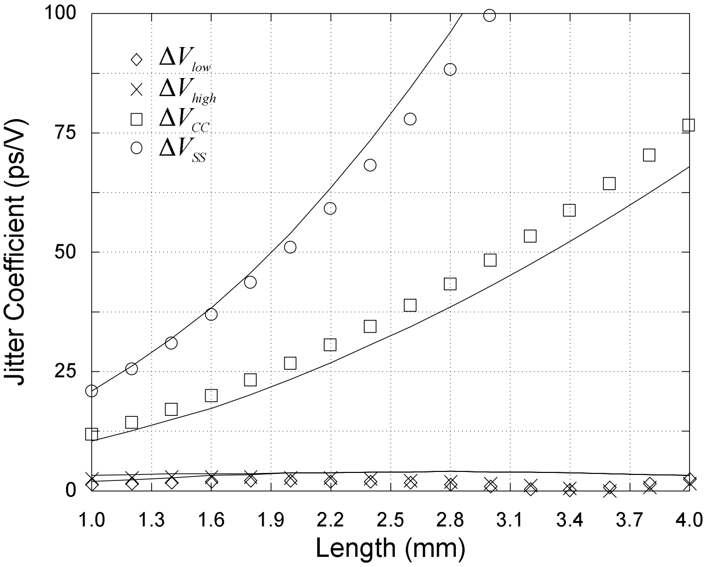


Figure 6.24: Impact of interconnect length on jitter coefficient.

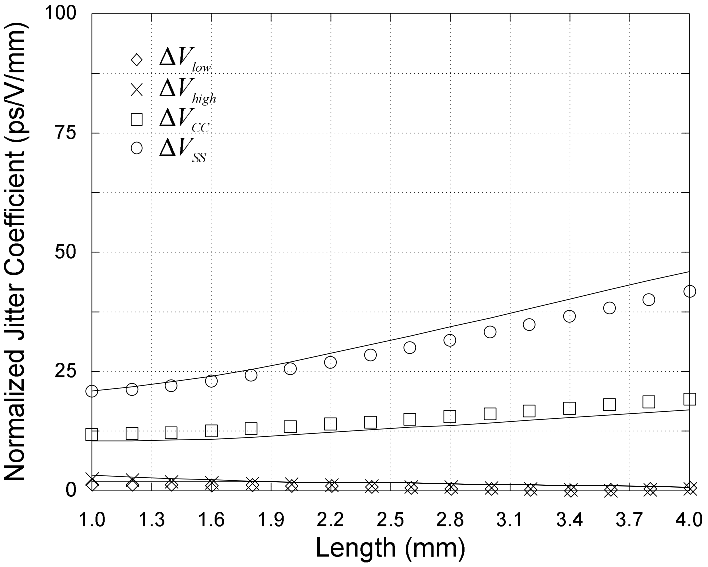


Figure 6.25: Impact of interconnect length on jitter coefficient per millimeter.

edge rate of the repeaters. However, when it modifies the output edge rate of the repeaters, the noise only has an indirect effect on the delay of the wires.

The problem is that some interconnect-dominated paths, in particular the ones used for clock distribution, are getting increasingly long: 400 to 600 ps in [6.22] and 800 ps in

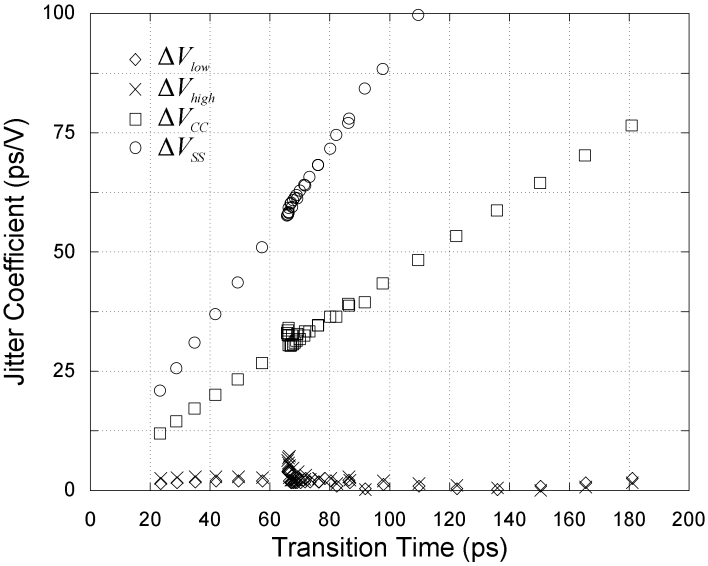


Figure 6.26: Impact of transition time on jitter coefficient.

[6.23]. Since clock frequencies are also increasing, the number of clock cycles spanned by these paths is increasing even *faster*. This is a problem because the number of clock cycles actually *multiplies* the timing impact of the supply voltage variations. For instance, say that a given ΔV variation causes a delay increase of 6% for a one-cycle device-dominated path and a 4% increase for a 2-cycle interconnect-dominated clock path. Then, the delay of the clock path varies by 2 × 4% = 8% of one clock cycle. Thus, for long interconnect-dominated paths, a given amount of noise can cause a delay variation that is a large fraction of the clock cycle, despite being small with respect to the path delay.

When the delay of a clock path varies, clock inaccuracy (i.e. skew or jitter) is created. In a data path through a network of sequential elements, the interval between the data launch and its final capture determines the time available for performing computations. If the clock of the launching sequential element is *late* or if the clock of the capturing element is *early*, the timing margin of the path is directly reduced. If the margin becomes negative, the frequency must also be reduced. Consequently, the performance impact of the power-supply noise on the clock distribution network is highest when it creates a combination of slow and fast (i.e. unbalanced) clock paths.

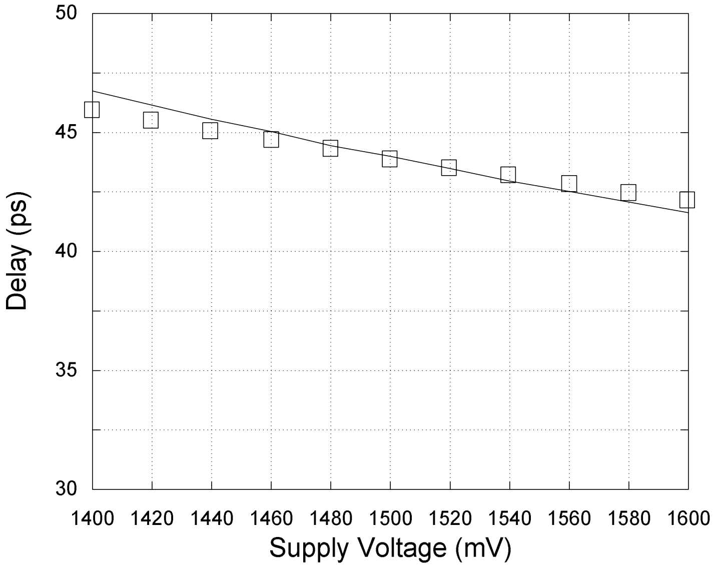


Figure 6.27: Delay versus supply voltage for a 2400-μm M5 interconnect.

In the Pentium® 4 microprocessor, all clock paths are chip-wide. When the supply voltage is not uniform, two paths routed over different regions of the die will experience different voltages and therefore exhibit different delays. This makes the clock distribution network particularly sensitive to globally *differential* (or non-uniform) voltage drops. If the clock paths become long enough, the differential component of a voltage drop can create enough clock inaccuracy to make its frequency impact greater than the frequency impact of the common-mode component on device-dominated paths.

6.3.10 Delay Versus Supply Voltage for Chip-Scale Noise Events

For a long interconnect with several repeaters, a chip-scale power-supply noise event can resemble a momentary nominal supply voltage variation. An event with sufficient spatial uniformity makes the power and ground voltages the same for all repeaters. If the duration of the event is longer that propagation delay of the interconnect, the event can be modeled as a change in the nominal supply voltage instead of a supply voltage change at each repeater.

Figure 6.27 shows the impact of the nominal supply voltage V_0 on the timing of an inverter driving a 2400-μm M5 interconnect. The width, spacing, and thickness are

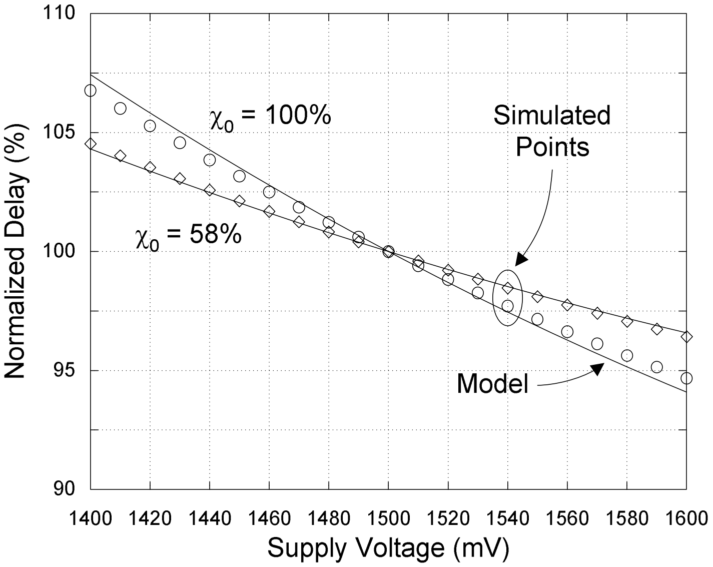


Figure 6.28: Alternative delay model versus simulation results.

respectively 0.70 µm, 0.70 µm, and 0.90 µm (as before). This corresponds to 1.25 times the minimum pitch. As V_0 increases, the device delay decreases. The delay curve predicted by the model closely matches the simulation results. The maximum error is 0.8 ps or 1.8%.

Alternatively, the delay D of an interconnect-dominated path can be modeled by modifying (6.47) where k_1 and k_2 are only valid for lumped loads. For distributed RC or RLC loads, δ_1 and δ_2 are used instead. In general, the response of a wire to a voltage ramp is a complicated function of the voltage transition time [6.24], [6.25]. With these coefficients, that complexity does not have to be analytically modeled. The nominal delay D_0 can be written as the length L of the path divided by the signal propagation velocity v_0 (including the repeater delays) when the supply voltage is quiet. The result is:

$$D = \frac{L}{v_0}\left(1 - \delta_1 \frac{\Delta V}{V_0 - V_t} + \delta_2 \left(\frac{\Delta V}{V_0 - V_t}\right)^2\right) \tag{6.108}$$

The magnitude ΔV of the chip-scale noise event is assumed relatively uniform over the region spanned by the path. For the technology described in [6.18], the coefficients δ_1 and

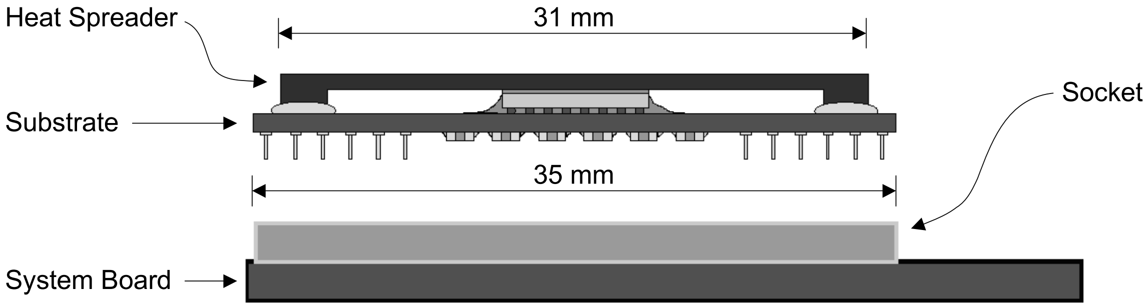


Figure 6.29: Microprocessor package.

δ_2 track χ_0, the fraction of the path delay taken by the repeaters when $\Delta V = 0$. Empirically, it is found that δ_1 and δ_2 closely obey the following relationships:

$$\delta_1 = (1 - (1 - k_1^{inv})\,\chi_0)\,\chi_0 \tag{6.109}$$

$$\delta_2 = k_2^{inv}\,\chi_0 \tag{6.110}$$

where k_1^{inv} and k_2^{inv} are coefficients similar, but not identical, to the ones defined in (6.47). The difference is that they characterize the delay-versus-voltage curve corresponding to an inverter sized for equal rise and fall transition times instead of the average delay-versus-voltage curve used in (6.47). For the technology described in [6.18], k_1^{inv} and k_2^{inv} are respectively 0.72 and 1.01. For typical wire resistance, capacitance, and inductance values, $\delta_1 = (0.80 \pm 0.10)\,\chi_0$ and $\delta_2 = (1.10 \pm 0.25)\,\chi_0$.

In Figure 6.28, the delay versus voltage relationship of a long interconnect-dominated path is compared to the one predicted by (6.108). The repeaters are inverters inserted every 1260 μm. The wires are on the M5 layer. Their width, spacing, and thickness are still 0.70 μm, 0.70 μm, and 0.90 μm. The wire inductance is taken to be 500 fH/μm and is assumed independent of frequency. The wire resistance is also assumed frequency-independent. The parameters χ_0, δ_1, and δ_2 are respectively 58%, 0.80 χ_0, and 1.10 χ_0. The delay versus voltage relationship predicted when 100% of the path delay is taken by the repeaters (i.e. when the distance separating each inverter is zero) is also shown. Equation (6.108) is reasonably accurate in both cases.

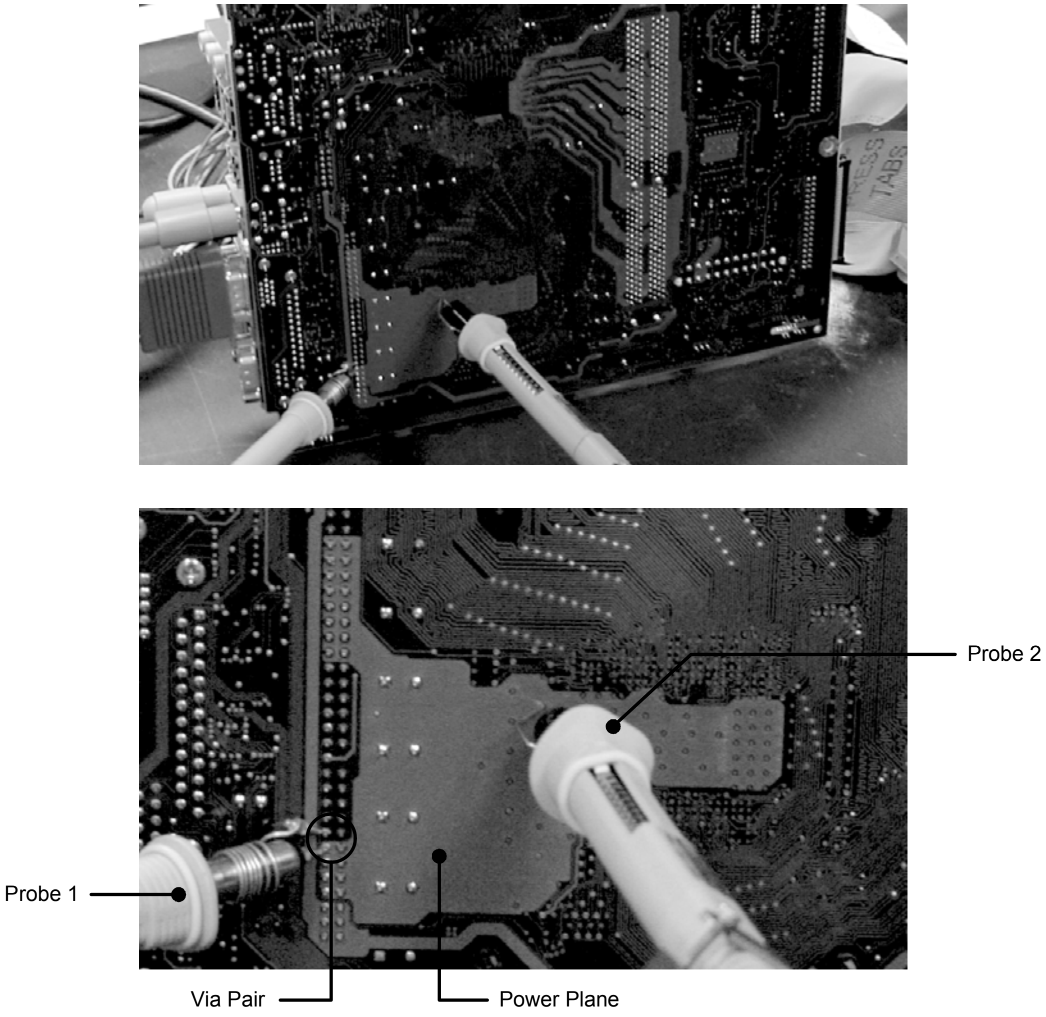
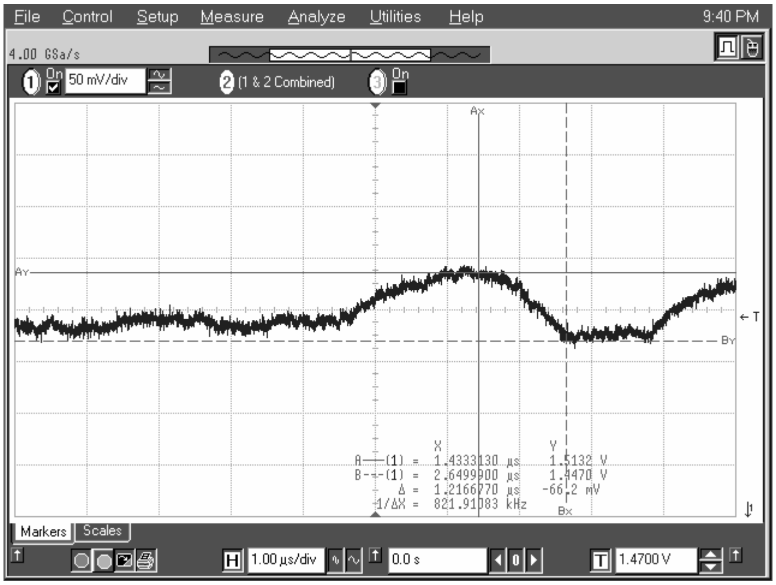


Figure 6.30: Setup for power-supply noise measurements.

6.4 Measurements

Sections 6.2 and 6.3 introduced analytical models predicting the timing impact of power-supply noise. These models assume that the properties of the noise are known *a priori*. The properties of the noise could theoretically be established by performing simulations. However, as discussed in Section 6.1, the simulation approach is not yet practical for complex systems.

In this section, the properties of the noise are directly measured instead. The measurements are taken on a 2.53-GHz microprocessor. These measurements, together



```
4.00 GSa/s
1 On 50 mV/div

               X                    Y
A——(1) =  1.4333130 µs     1.5132 V
B--(1) =  2.6499900 µs     1.4470 V
     Δ =  1.2166770 µs   -66.2 mV
 1/ΔX =  821.91083 kHz

H 1.00 µs/div   0.0 s   T 1.4700 V
```

Figure 6.31: Measured power-supply noise waveform.

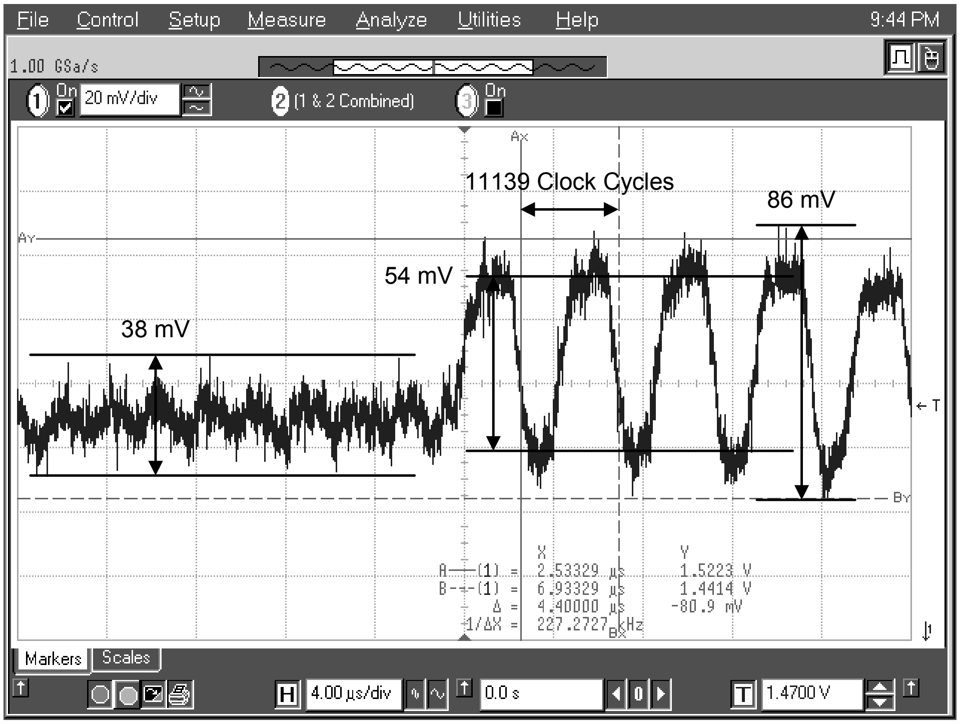
with (6.47) and (6.108), are then used to quantitatively estimate the timing impact of the power-supply noise on the performance of the system.

The microprocessor has a nominal supply voltage of 1.500 V. It uses the 130-nm logic technology described in [6.18]. The 533-MHz system bus is a quad-pumped bus and runs off a 133-MHz system clock.

The microprocessor utilizes a flip-chip pin grid array package technology and plugs into a 478-pin socket [6.26], as shown in Figure 6.29. The system board has 4 layers of copper. The power and ground planes are 35-µm thick. For on-chip power distribution, the microprocessor has 85 V_{CC} (power) and 181 V_{SS} (ground) pins. A two-phase voltage regulator is located on the system board. It supplies power to the microprocessor from one side of the package.

The supply voltage is probed under the system board between two pairs of V_{CC} and V_{SS} vias. The first pair is located near the top of the package; the other is near the bottom. The measurement setup is shown in Figure 6.30.

Figure 6.31 shows a typical power-supply noise waveform measured over 10 µs. The time scale for the measurement represents several thousand clock cycles. Any event



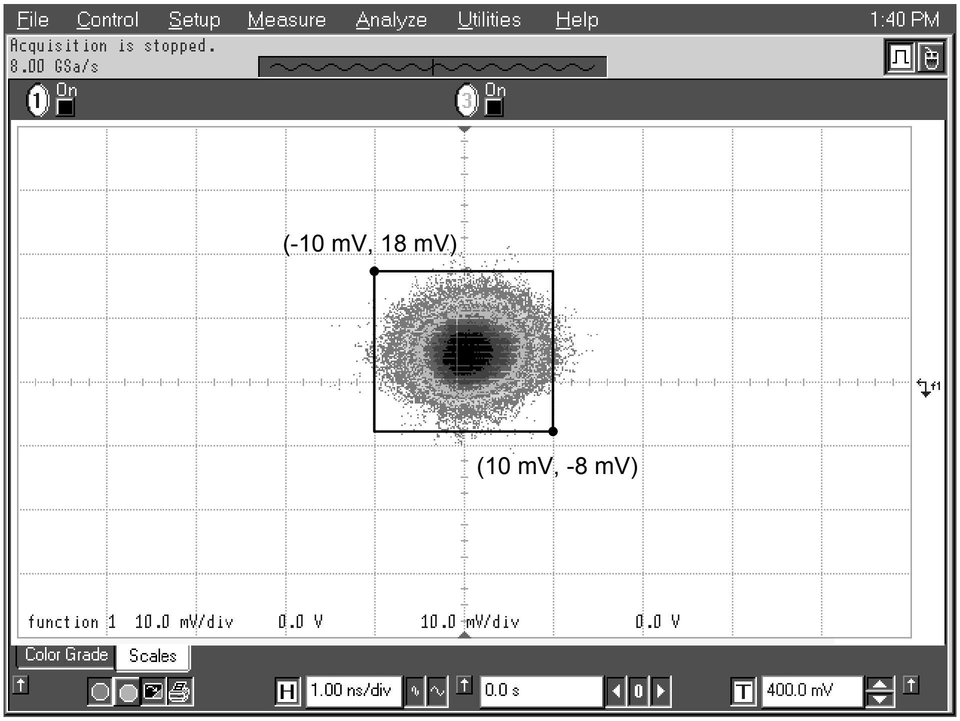
```
               X                  Y
A——(1) =  2.53329 µs         1.5223 V
B--(1) =  6.93329 µs         1.4414 V
     Δ =  4.40000 µs        -80.9 mV
  1/ΔX =  227.2727 kHz
```

Figure 6.32: Another measured power-supply noise waveform.

whose duration is much less than that cannot be reliably captured. This is particularly true for the power-supply noise over one clock cycle. Because the measurements do not reveal the high-frequency components of the noise, the timing impact of these components cannot be analyzed. In this section, the power-supply voltage waveforms are assumed quasi-static over one clock cycle. In other words, during any given clock cycle, the supply voltage is assumed nearly constant.

Figure 6.32 shows another noise waveform. For this measurement, the operating system is in a relatively steady state: a fixed number of high-activity user processes are running together with some low-activity system processes. The captured waveform shows that the supply voltage is initially close to its minimum for 20 µs. During this time, the peak-to-peak noise on the signal is 38 mV. Then, the voltage starts going up and down at a frequency of 227 kHz. The amplitude of the quasi-sinusoidal component is 54 mV.

This behavior suggests that the power-supply noise is a strong function of the activity of the microprocessor. Since the peak-to-peak noise when the supply voltage is oscillating is



Acquisition is stopped.
8.00 GSa/s

function 1 10.0 mV/div 0.0 V 10.0 mV/div 0.0 V

1.00 ns/div 0.0 s 400.0 mV

Figure 6.33: Background noise.

86 mV, 63% of the noise appears driven by the operating system. It is worth noting that the supply voltage can stay close to its minimum or maximum for extended periods of time (here, roughly 5600 clock cycles).

The measurements (taken over a few minutes) when the two via pairs are probed simultaneously are shown in Figure 6.33 and Figure 6.34. One voltage is plotted versus the other.

When no power is supplied to the package of the microprocessor, the two voltage measurements should ideally be zero. Figure 6.33 shows that they practically are. It also indicates that the measurements from the *x*-axis channel are contaminated by ±10 mV of background noise. The *y*-axis channel has a 5-mV offset and ±13 mV of background noise. The 5-mV offset is due to a slight calibration error of the oscilloscope. Most of the background noise is probably due to the ground wire loop of each probe. As visible in Figure 6.30, these loops have an area that is small, but not completely negligible. The area of each loop is largely set by the physical separation of the V_{CC} and V_{SS} vias being

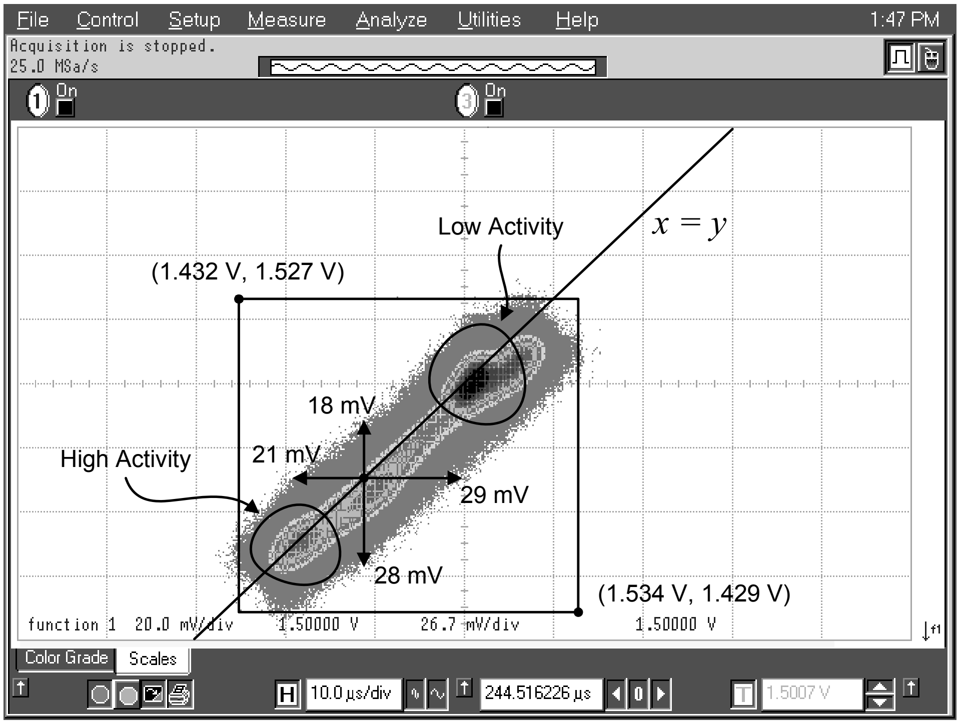


Figure 6.34: Power-supply noise when the microprocessor is active.

probed. Therefore, the loops can still respond to the parasitic magnetic fields randomly generated in the laboratory and contaminate the real signals to measure. The shape of the cluster of points is compatible with white Gaussian noise.

Figure 6.34 shows (on a different scale) the power-supply noise when the microprocessor is active. Again, one voltage measurement is plotted against the other.

Nominally, the two voltages should both be close to 1.500 V. The (1.500 V, 1.500 V) point is located in the center of Figure 6.34. The data shows that the two measurements are very near that center point when the activity of the microprocessor is low. The darker portion of the low-activity region indicates where the two measured voltages are most often found.

However, when the microprocessor is very active, the two measured voltages can drop significantly. The diagonal line of Figure 6.34 indicates where the two voltages are equal. When the activity increases, both voltages tend to drop along that line. The cluster of

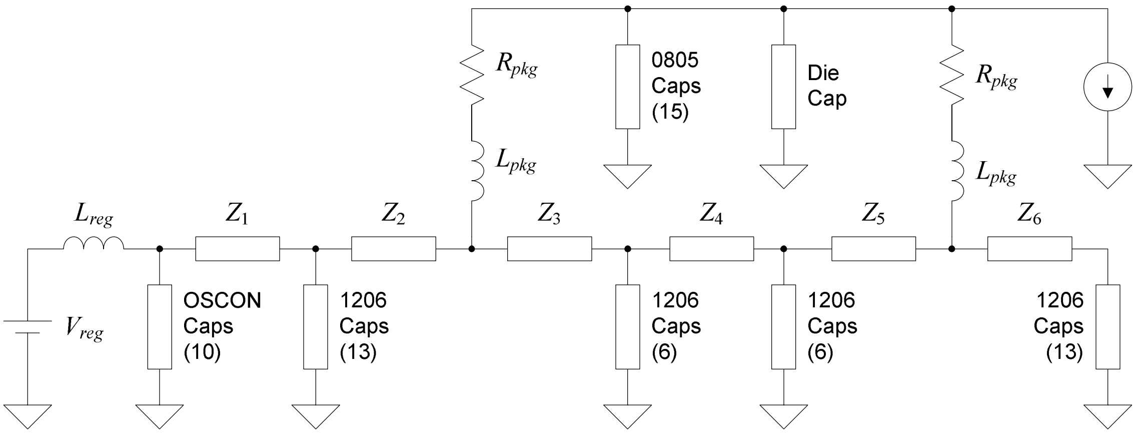


Figure 6.35: Model of power distribution network.

points shows a peak-to-peak voltage variation of (1.534 V – 1.432 V) = 102 mV on the *x*-axis. Since this channel includes ±10 mV of background noise, the real peak-to-peak variation is (102 mV – 20 mV) = 82 mV. For the *y*-axis, the maximum peak-to-peak variation is (1.527 V – 1.429 V) = 98 mV. Removing the ±13-mV background noise yields 72 mV of real peak-to-peak voltage variation.

The maximum supply voltage droop experienced by the microprocessor is defined with respect to nominal supply voltage value of 1.500 V. In Figure 6.34, the maximum droop is found on the *y*-axis channel. There, the cluster of points shows a droop of (1.500 V – 1.429 V) = 71 mV. With the background noise removed and the 5-mV offset corrected, the droop is (71 mV – 13 mV + 5 mV) = 63 mV or 4.2% of V_{CC}.

It is interesting to note that the data of Figure 6.34 also indicates some differential power-supply noise. The cluster density shows that the voltage on the *x*-axis is often closer to its nominal value of 1.500 V than the one on the *y*-axis. This is especially true in the transition region located between the low and high activity regions. This suggests that the microprocessor core can draw more current from one side of the package or that some lateral voltage drop occurs on the system board. The maximum differential noise is (29 mV – 10 mV) = 19 mV or 1.3% of the nominal supply voltage, after removing

Table 6.2: Characteristics of decoupling capacitors.

Capacitor	Capacitance (μF)	ESR (mΩ)	ESL (nH)
OSCON	560	9.0	6.000
IDC 1206	10	3.0	1.000
IDC 0805	1	3.0	0.045

Table 6.3: Resistance and inductance of system board.

Segment	Resistance (mΩ)	Inductance (pH)
1	0.27	80
2	0.33	11
3	0.39	104
4	0.20	52
5	0.39	104
6	0.64	200

10 mV to account for the background noise contaminating the *x*-axis channel. The maximum differential noise can occur when the common-mode voltage is minimal.

6.5 On-Chip Power-Supply Noise

The non-zero impedance of the package and on-chip power distribution network of the 2.53-GHz microprocessor of the previous section increases the noise seen at the devices. The on-chip multilayer power grid is connected to the package power planes by an array of controlled collapse chip connections, also known as C4 bumps [6.27]. Compared to wire-bond technology, where the package delivers power only to the periphery of the chip, these bumps provide a large number of nearly ideal voltage sources at uniformly distributed points [6.28]. These points are scattered over the entire surface of the chip and are located directly above the power grid.

The on-chip noise can be estimated by combining the board-level measurements discussed earlier with the power distribution network model of Figure 6.35. The model represents the voltage regulator, the system board, the package, and the on-chip power distribution network. It is known to be reasonably accurate for the first, second, and third on-chip voltage droops [6.9], [6.10].

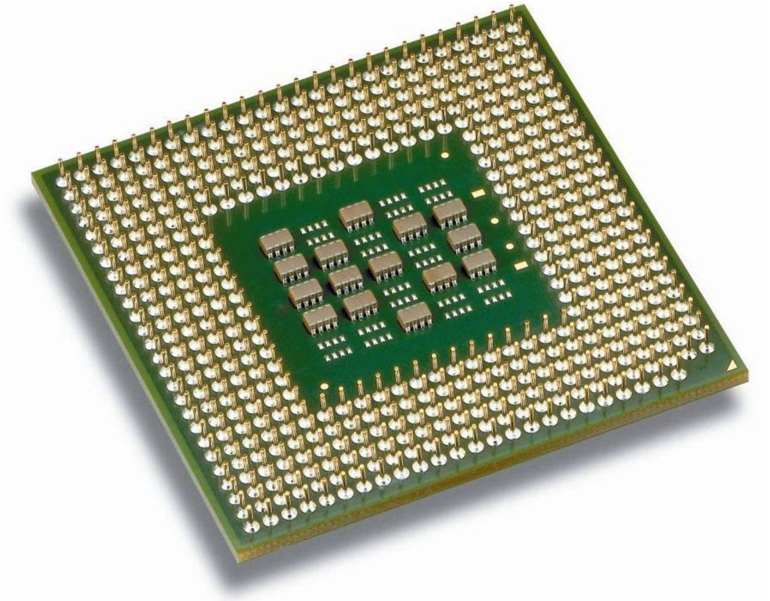


Figure 6.36: Land side capacitors.

The regulator supplies 1.5 V and has an inductance L_{reg}. It drives 10 low-frequency organic semiconductor (OSCON) decoupling capacitors that are placed on the system board, relatively far from the package. A total of 38 mid-frequency 1206 interdigitated (IDC) capacitors are placed next to the package, including 12 in the package cavity. The value and the effective series resistance (ESR) and inductance (ESL) of the various decoupling capacitors is given in Table 6.2.

The resistance and inductance of the system board segments corresponding to the placement of the OSCON and 1206 IDC capacitors suggested in [6.29] are shown in Table 6.3. For the system on which the measurements were taken, these placement guidelines were closely followed.

The model also includes 15 high-frequency 0805 IDC capacitors that are mounted on the land side of the package, as shown in Figure 6.36. The resistance and inductance of the package are 25 mΩ and 3.3 nH per pair of power and ground pins [6.29]. Since there are 85 pairs and since the model uses two electrical branches to model the connection between the package and the system board, R_{pgk} = 25 mΩ / (85 / 2) = 0.6 mΩ. Similarly, L_{pkg} = 3.3 nH / (85 / 2) = 78 pH.

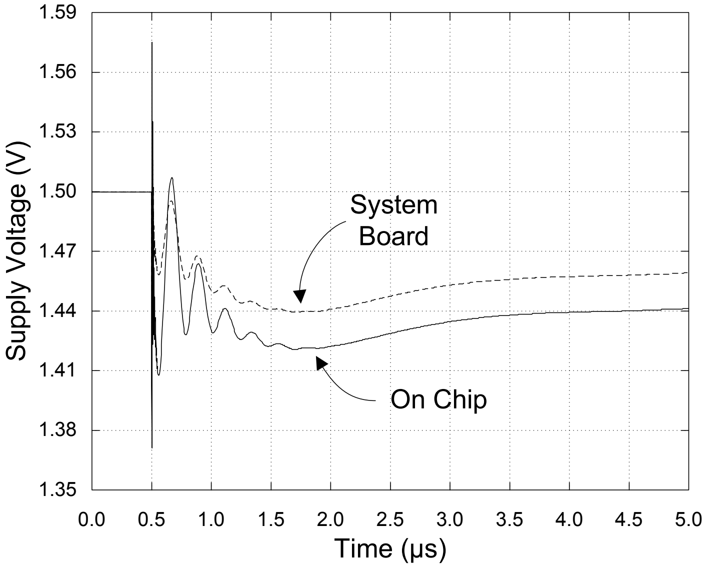


Figure 6.37: Supply voltage on system board and on chip.

The on-chip decoupling capacitance is estimated at 150 nF with an ESR of 0.5 mΩ.

Figure 6.37 shows the supply voltage on the system board and on-chip when the model is excited by a current ramp going from 0 A to 41 A in 3 clock cycles (1185 ps). The magnitude of the first droop of the on-chip supply voltage is 131 mV. The first droop is a few nanoseconds long and is primarily determined by the package inductance and the on-chip decoupling capacitance. The second droop is determined by the IDC capacitors placed on the package and on the system board. It lasts about 1.5 μs and has an amplitude of 79 mV. The model's prediction for the maximum droop on the system board is 60 mV. It is remarkably close to the 63 mV measurement discussed in the previous section. It is therefore reasonable to assume that the model produces a reasonable estimate for the noise at the interface between the C4 bumps and the top on-chip metal layer.

In addition to the noise of Figure 6.37, the noise seen at the devices includes the ohmic or *IR* voltage drop due to the on-chip power distribution network of the microprocessor. Figure 6.38 shows the *IR* drop map corresponding to a multilayer power grid typical for such chips. The pitch of the power grid is assumed sufficiently small to make its inductance negligible compared to the package inductance. The *IR* drop map is for V_{CC}. It

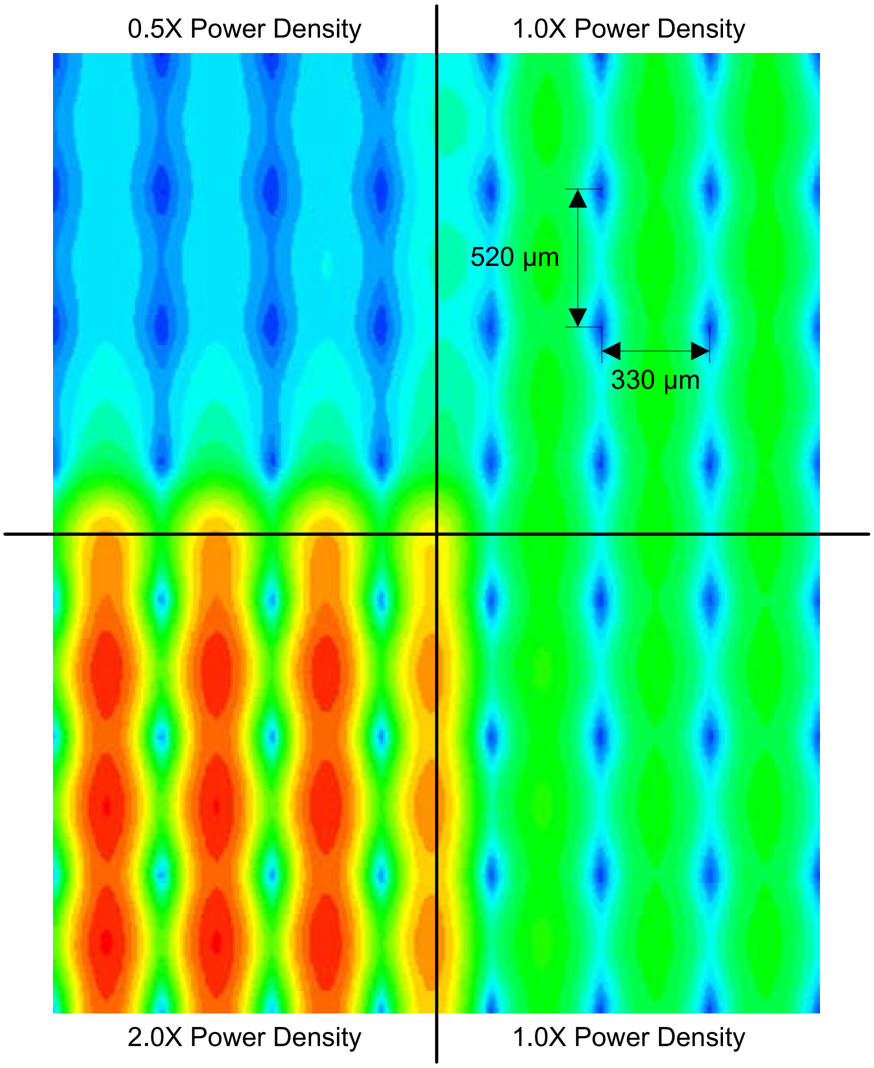


Figure 6.38: Ohmic power-supply noise on chip.

spans an area containing 64 C4 bumps. The vertical pitch of 520 μm and the horizontal pitch of 330 μm between V_{CC} bumps are consistent with the ones used on the Itanium 2 microprocessor discussed in [6.30], which uses the same 130-nm technology. The position of the C4 bumps corresponds to the points of minimum droop (shown in dark blue) and is clearly visible.

The upper-right and lower-right quadrants of Figure 6.38 show the voltage droop for a typical power density, which is defined here as the ratio of the thermal design power of the chip (61.5 W) to its area (131 mm^2) [6.26]. The result is 470 mW/mm^2 at 1.5 V. The

Table 6.4: Clock wire delay versus technology generation.

Technology Generation (nm)	Frequency (GHz)	Die Edge (mm)	Signal Velocity (µm/ps)	Clock Wire Delay (ps)	Clock Wire Delay (Clock Cycles)
130	1.684	15.0	20.0	750.0	1.26
115	2.317	15.0	20.0	750.0	1.74
100	3.088	15.0	20.0	750.0	2.32
90	3.990	15.0	20.0	750.0	2.99
80	5.173	15.0	20.0	750.0	3.88
70	5.631	15.0	20.0	750.0	4.22
65	6.739	15.0	20.0	750.0	5.05

switching activity of the devices is modeled using triangular current sources that are uniformly distributed.

The upper-left quadrant corresponds to a region where the devices are less active and where the power density is halved. The voltage droop is much smaller there. The lower-left quadrant is very active. It consumes twice the typical power. The result is a 1.4% V_{CC} droop.

6.6 Projections Using the Power-Supply Noise Timing Impact Model

The delay models of Sections 6.2 and 6.3 and the measurements of Section 6.4 can be used to analyze and extrapolate the impact of power-supply noise on timing for microprocessors. According to the 2002 Update of the International Technology Roadmap for Semiconductors, the clock frequency will scale as shown in Table 6.4. Die size however is no longer expected to grow. It is projected to remain between 140 mm^2 (for cost-performance products) and 310 mm^2 (for high-performance products). Here, a 225-mm^2 die is assumed. The clock signal propagation velocity is taken to be 20 µm/ps. This value is not expected to improve [6.20] and is in line with the ones published in [6.22]. Together, these assumptions imply a 750-ps clock distribution delay. This fixed delay will reach about 3 clock cycles for the 90-nm technology generation.

A lower bound on the frequency penalty associated with the power-supply noise can be estimated based on the board-level measurements described in Section 6.4. When the supply voltage suffers a common-mode drop of 63 mV, all device-dominated paths slow down. According to (6.47), with $k_1 = 0.91$ and $k_2 = 1.20$, the delay increase is:

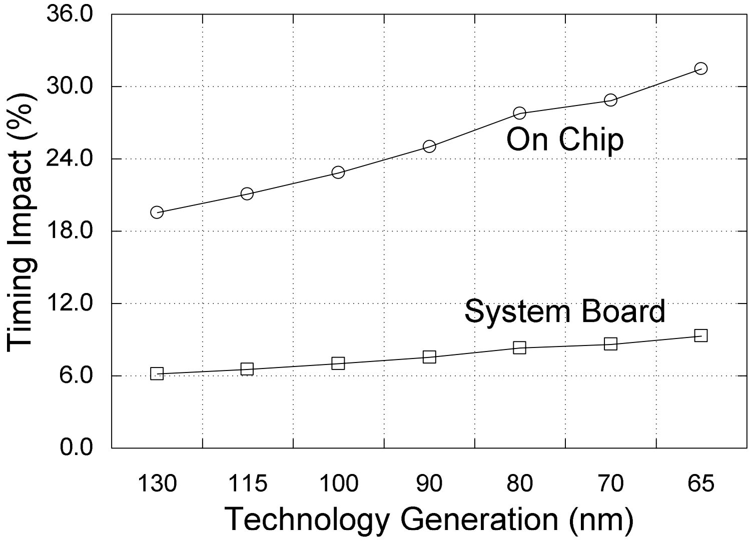


Figure 6.39: Timing impact of power-supply noise.

$$\frac{D}{D_0} - 1 = -k_1 \frac{-63\,\mathrm{mV}}{V_0 - V_t} + k_2 \left(\frac{-63\,\mathrm{mV}}{V_0 - V_t} \right)^2 = 5.1\%$$

At the same time, the supply voltage can exhibit 19 mV of differential noise. Thus, while some clock paths may suffer a 63-mV voltage drop, others may only experience a drop of (63 mV – 19 mV) = 44 mV. From (6.108), with $\chi_0 = 58\%$, $\delta_1 = 0.80\,\chi_0$, and $\delta_2 = 1.10\,\chi_0$, the delay difference corresponding to this voltage difference is:

$$\frac{\Delta D}{D_0} = \left[1 - \delta_1 \frac{-63\,\mathrm{mV}}{V_0 - V_t} + \delta_2 \left(\frac{-63\,\mathrm{mV}}{V_0 - V_t} \right)^2 \right] - \left[1 - \delta_1 \frac{-44\,\mathrm{mV}}{V_0 - V_t} + \delta_2 \left(\frac{-44\,\mathrm{mV}}{V_0 - V_t} \right)^2 \right] = 0.8\%$$

According to Table 6.4, at 2.53 GHz, the clock wires have a latency of about 2 clock cycles. The clock inaccuracy introduced by these paths when they become unbalanced is thus equal to 2 × 0.8% = 1.6% of the cycle time. This increase in clock inaccuracy is assumed to affect the device-dominated paths experiencing the 63-mV voltage drop. Under this scenario, the total frequency penalty due to the power-supply noise is estimated at 5.1% + 1.6% = 6.7%.

Assuming that the worst-case common-mode and differential noise on the system board will respectively remain at 4.2% and 1.3% of the nominal supply voltage (as measured in

Section 6.4), the frequency penalty is expected to steadily increase as shown in Figure 6.39. The timing impact of the noise present on the system board could reach 7.6% for the 90-nm technology generation.

Based on the analysis of Section 6.5, it is reasonable to assume that the noise at the interface between the C4 bumps and the top on-chip metal layer can reach 131 mV and last several clock cycles. It also reasonable to assume an additional ohmic drop of 1.4% (i.e. 21 mV) for V_{CC} and V_{SS}. This increases the maximum common-mode droop from 63 mV on the system board to 131 mV + 2 × 21 mV = 173 mV on chip. The corresponding timing degradation is:

$$\frac{D}{D_0} - 1 = -k_1 \frac{-173\,\text{mV}}{V_0 - V_t} + k_2 \left(\frac{-173\,\text{mV}}{V_0 - V_t} \right)^2 = 15.6\%$$

A long clock path routed over a region of the die where the power density is low would not suffer from a significant ohmic droop. Its supply voltage would only be reduced by the common-mode droop of 131 mV minus the differential noise of 19 mV. Its droop would thus be 112 mV. The delay difference between a long clock path experiencing a droop of 173 mV and a path experiencing a droop of only 112 mV is:

$$\frac{\Delta D}{D_0} = \left[1 - \delta_1 \frac{-173\,\text{mV}}{V_0 - V_t} + \delta_2 \left(\frac{-173\,\text{mV}}{V_0 - V_t} \right)^2 \right] - \left[1 - \delta_1 \frac{-112\,\text{mV}}{V_0 - V_t} + \delta_2 \left(\frac{-112\,\text{mV}}{V_0 - V_t} \right)^2 \right] = 3.1\%$$

The total frequency penalty including the on-chip power-supply noise is therefore estimated at 15.6% + 2 × 3.1% = 21.8%. As shown in Figure 6.39, the total frequency penalty is significantly higher when the on-chip noise is taken into account.

6.7 Summary

The power-supply noise injection models derived for a device driving a lumped load or a long interconnect are accurate. The mathematical complexity introduced by making the load distributed instead of lumped is significant, but the simplicity of the quasi-linear device model of Chapter 2 enables an analytical solution.

When analyzing the impact of power-supply noise on timing, the commonly used assumption that minimizing the *peak* of the supply voltage drop maximizes performance is wrong for the circuits of Section 6.2. It is more general to consider the average supply voltage while a circuit is switching. When the switching window is short with respect to the duration of the noise perturbation, minimizing the average voltage drop is equivalent to minimizing its peak. However, when the switching window is comparable to the duration of the noise perturbation, the peak is no longer sufficient to predict the performance of the circuit. The time-domain characteristics of the noise waveform become important. Fortunately, for typical device-dominated paths, these characteristics can be summarized reasonably well by averaging the noise waveform.

The board-level measurements presented in Section 6.4 show that the supply voltage can drop by 4.2% for the 2.53-GHz Pentium 4 microprocessor. They also indicate the presence of differential noise (1.3% of the supply voltage). A model representing the voltage regulator, the system board, the package, and the on-chip power distribution network is used to extrapolate the on-chip noise based on these board-level measurements.

The timing impact of the board-level power-supply noise is a reduction in frequency estimated at 6.7%. The total frequency penalty including the on-chip power-supply noise is estimated at 21.8%.

CHAPTER 7

Clock Distribution Using Multiple PLLs and Digital Alignment

This chapter presents a novel architecture for clock distribution that uses multiple phase-locked loops (PLLs) for clock generation and digital feedback for skew compensation. The architecture is naturally suited for multi-core microprocessors.

Unlike the clock distribution network architectures discussed in Chapter 5 [5.2], [5.6], [5.10]-[5.17], the one proposed here is compatible with conventional design-for-debugability (DFD) and design-for-testability (DFT) techniques [5.4], significantly reduces the jitter accumulation problem [5.17], and supports multiple independent frequency and voltage scaling regions for better energy efficiency.

The feedback mechanism used for skew compensation requires variable-delay elements that are digitally adjustable. To implement them, this chapter proposes a new circuit topology that is based on a novel digitally adjustable resistor. The new circuit is compact and can be used to control the timing of the clocks in small linear steps.

Finally, two new skew compensation algorithms are proposed and analyzed. Compared to the simpler algorithms used in the past to align a limited number of clock domains [7.1], [7.2], [7.3], the new algorithms are significantly more scalable. They are also much more efficient at reducing skew when working with variable-delay elements having a limited range.

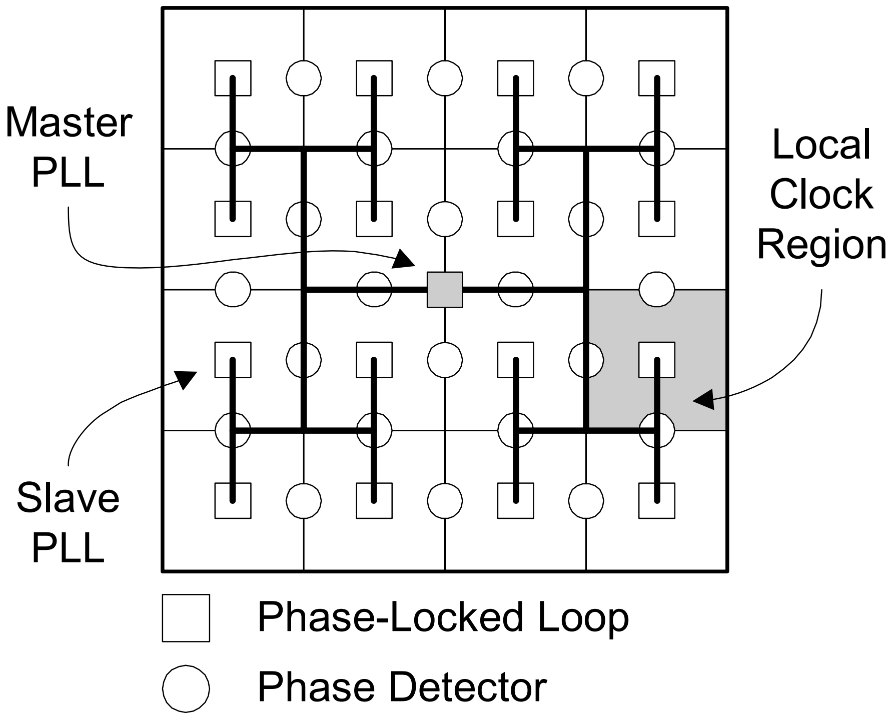


Figure 7.1: Novel multi-PLL clock distribution architecture.

7.1 Architecture for Clock Distribution Using Multiple PLLs

The clock distribution architecture proposed relies on distributed jitter filtering. The phase processing is decentralized and multiple filters are required for jitter attenuation.

As shown in Figure 7.1, the proposed architecture consists of a global distribution network feeding an array of PLLs. The PLLs are kept synchronized using variable-delay elements that are digitally controllable. To prevent the PLLs from getting caught in an undesired stable equilibrium [7.4], the synchronization information is propagated in a single direction.

7.1.1 Global Clock Distribution

The multi-PLL distribution network described in [7.5] suffers from a basic problem, as discussed earlier in Chapter 5: the PLLs cannot be bypassed and the timing of the clock cannot be controlled cycle by cycle. This makes it incompatible with conventional DFT and DFD techniques. In addition, it does not allow independent frequency and voltage scaling over different local clock regions.

The proposed solution is to build a conventional clock distribution network driven by a master PLL to feed the array of slave PLLs. The role of the master PLL is to filter the jitter contaminating the low-frequency system clock. The master PLL also synchronizes the phase of the system clock with the phase of the regional core clocks produced by the slave PLLs. This synchronization compensates for the delay of the conventional global clock distribution network. A variable-delay element is inserted before each slave PLL to terminate the global clock network. The variable-delay elements are used for skew compensation.

Since the master PLL and the slave PLLs can be bypassed, the sequential elements can be controlled step by step. Furthermore, the duty cycle of the clock can be adjusted at the output of each slave PLLs and intentional skew is easier to introduce since the variable-delay elements are digitally controllable. Therefore, this solution is compatible with conventional DFT and DFD techniques.

7.1.2 Synchronization of the Phase-Locked Loops

The role of the slave PLL array is to attenuate the clock jitter introduced by the global distribution network. Because the phase correction loop of each slave PLL is local (as opposed to global, like in [7.5]), it is quieter. Compared to long loops, these tight loops are less sensitive to supply voltage noise. Moreover, the slave PLLs can be designed and characterized independently from the rest of the clock distribution network since their feedback is entirely internal. This makes them easier to replicate.

Figure 7.2 schematically shows how the variable-delay elements are used to compensate skew. When at least one phase detector determines that a particular local clock region is too fast or too slow, a digital signal is returned to the variable-delay element controlling this region. By performing some basic digital signal processing on the phase error signal, the variable-delay element can decide how to adjust its delay.

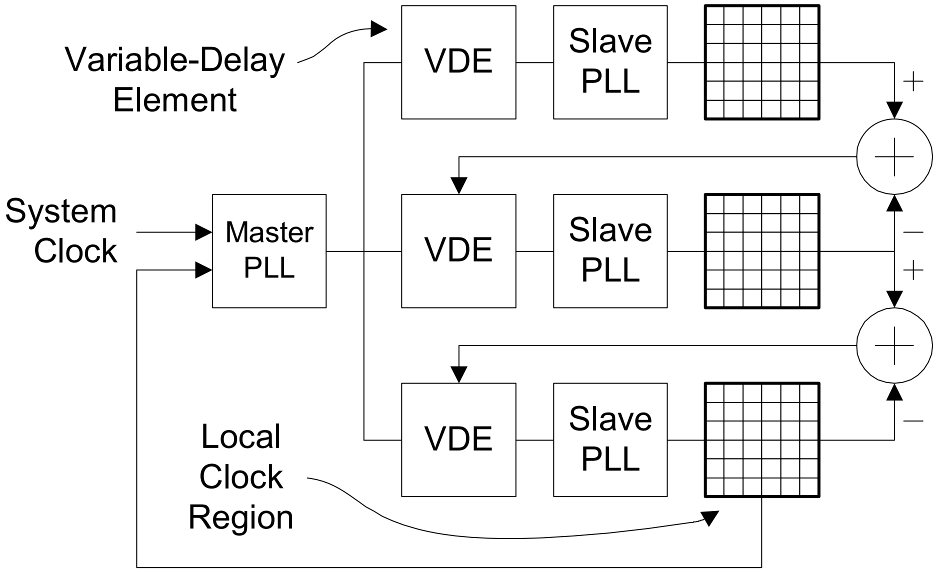


Figure 7.2: Skew compensation mechanism.

Digital synchronization has a number of advantages over analog synchronization. Digital feedback signals are significantly less sensitive to noise, in particular to interlevel coupling noise, and, unlike analog signals, they can be routed without taking special precautions. The phase detectors producing the feedback signals can be much simpler than the analog phase detectors designed to avoid mode-locking in [7.5].

The clock distribution network proposed here prevents mode-locking by construction. The phase averaging mechanism is designed to avoid loops in the propagation of the phase information, as shown in Figure 7.3. The drawback of not directly using the slave PLLs for phase alignment is that the range of the variable-delay elements must be large enough to compensate the global clock skew.

A dedicated finite-state machine controls the initial acquisition of the locked state, like in [7.6]. To do this, the slave PLL array is initially bypassed. This makes the multi-PLL clock distribution network behave like a conventional network. It is worth noting that even when the array is bypassed, the slave PLLs can still try to lock to the master PLL. Once the master PLL and the slave PLLs achieve phase lock, the PLL array can be enabled. Enabling the PLL array can introduce small phase perturbations, but the master

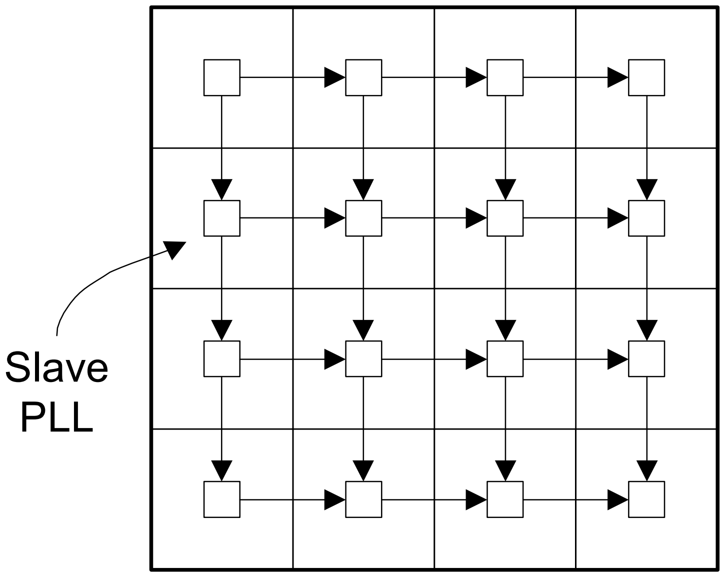


Figure 7.3: Phase information propagation.

PLL can quickly recover. When every PLL is phase locked again, skew compensation (i.e. phase alignment) can begin.

Detailed algorithms for phase alignment are given later in this chapter.

7.1.3 Independent Frequency Scaling

For microprocessors, dynamic frequency and voltage scaling is a very important power reduction technique. The technique is traditionally applied to an entire chip by adjusting the voltage and frequency according to the characteristics of its work load. When a chip is idle, high performance is not required. The voltage and frequency can then be reduced to cut dynamic and leakage power.

The proposed clock distribution architecture has the distinct advantage of begin capable of generating a different frequency for each local clock region. This is essential to enable a finer granularity of frequency and voltage scaling.

Having multiple frequency domains complicates skew compensation however. The solution proposed here is to allow the phase detectors controlling the variable-delay elements to operate only at certain times, when all the clocks are supposed to be aligned. This is shown in Figure 7.4 with four local frequency domains. The local clocks are 3, 4,

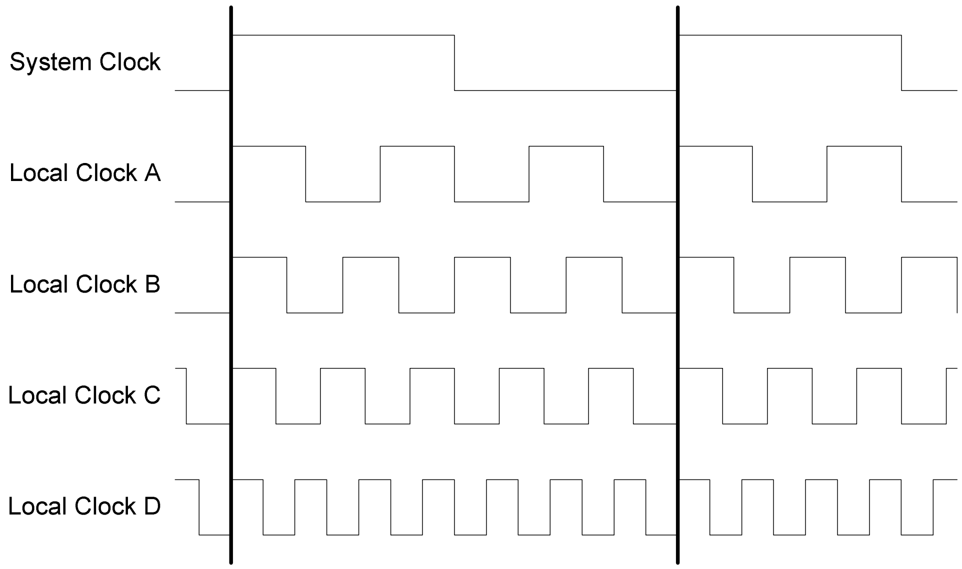


Figure 7.4: Periodic phase alignment.

5, and 7 times faster than the system clock. When the frequency of each local clock is restricted to be an integer multiple of the frequency of the system clock, all the clocks should be aligned when the system clock rises. If not, skew compensation is required. Therefore, the solution to the multiple frequency domain problem is simply to use the system clock to sample the phase detectors and update the settings of the variable-delay elements.

Another problem is that dynamic voltage scaling can significantly change the latency of the clock through the local clock regions. However, any large latency variation in a particular region can be corrected by the slave PLL feeding it. This requires making the feedback loop of the slave PLL longer by moving the feedback point after the regional clock distribution network. Since a longer feedback loop could potentially increase jitter, another solution would be to make sure that the variable-delay elements have enough range to compensate the latency variation.

7.1.4 Local Clock Distribution

Any conventional strategy can be used to distribute the local clock. The clock of each local region is compared to the clock of the neighboring regions using binary phase detectors. Since the feedback is digital, the clock of a local region can be easily gated to save power without compromising the stability of the entire network. The only action required is to instruct the neighboring regions to ignore the phase information coming from the sleeping region.

7.2 Digital Feedback Mechanism to Maintain PLL Synchronization

The clock produced by a particular PLL only has to be synchronized to the clocks produced by its neighbors. It is shown in [7.4] that an oscillator cannot always use the average phase of its neighbors for synchronization. The reason is that phase averaging can produce an undesired equilibrium that is stable and in which not all the oscillators have the same phase. This condition is called mode locking. It can occur if two neighbors produce clocks with phases of equal magnitude and opposite sign.

Mode locking can be avoided by ensuring that no loop exists in the propagation of the phase information. It is shown in [7.4] that mode locking can also be avoided by ensuring that the magnitude of the phase error between any pair of neighbors is below 90º. It is also shown in [7.4] that if this phase error guarantee cannot be provided, mode locking can still be avoided by using special phase detectors. These phase detectors must have a response that decreases monotonically beyond a phase difference of 90º.

Two fundamental PLL synchronization strategies are analyzed here: cyclic and acyclic phase averaging. It is argued that both theoretically and practically, acyclic phase averaging is preferable. Then, an implementation for the variable-delay elements required for PLL synchronization is presented.

7.2.1 Cyclic Versus Acyclic Phase Averaging

It is argued in [7.5] that from a clock jitter perspective, cyclic phase averaging is better than acyclic phase averaging. However, from a practical perspective, the opposite may be true. The argument is based on the comparison of two multi-PLL clock distribution networks modeled as linear systems. One of the systems propagates the phase alignment information in a single direction. The other uses cyclic phase averaging. Both systems are shown in Figure 7.5. The open-loop transfer function of each oscillator *G*(*s*) is given by:

$$G(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2} \tag{7.1}$$

where $\zeta = (1/2)^{1/2} \approx 0.707$ is its damping factor and $\omega_n = 2\pi \times (25\text{ MHz})$ is its natural frequency. A 1-GHz core clock is assumed. Jitter is modeled by white noise with $\sigma = 10$ ps.

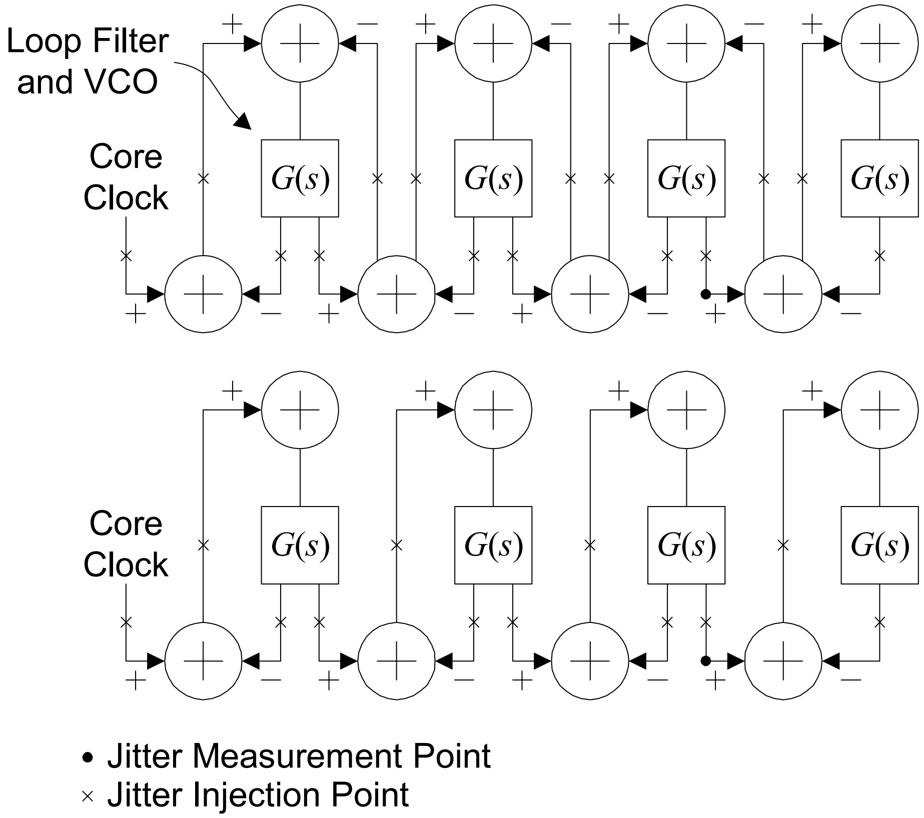


Figure 7.5: Cyclic (top) and acyclic (bottom) phase averaging.

The residual jitter of the two systems is compared for low, intermediate, and high frequencies. The assumption is that a single interconnect segment is noisy in each system. The conclusion reached in [7.5] is that the two systems have the same performance at low and high frequencies. For intermediate frequencies however, the conclusion is that cyclic phase averaging is better because the additional error signals produced by the phase detectors significantly improve noise attenuation.

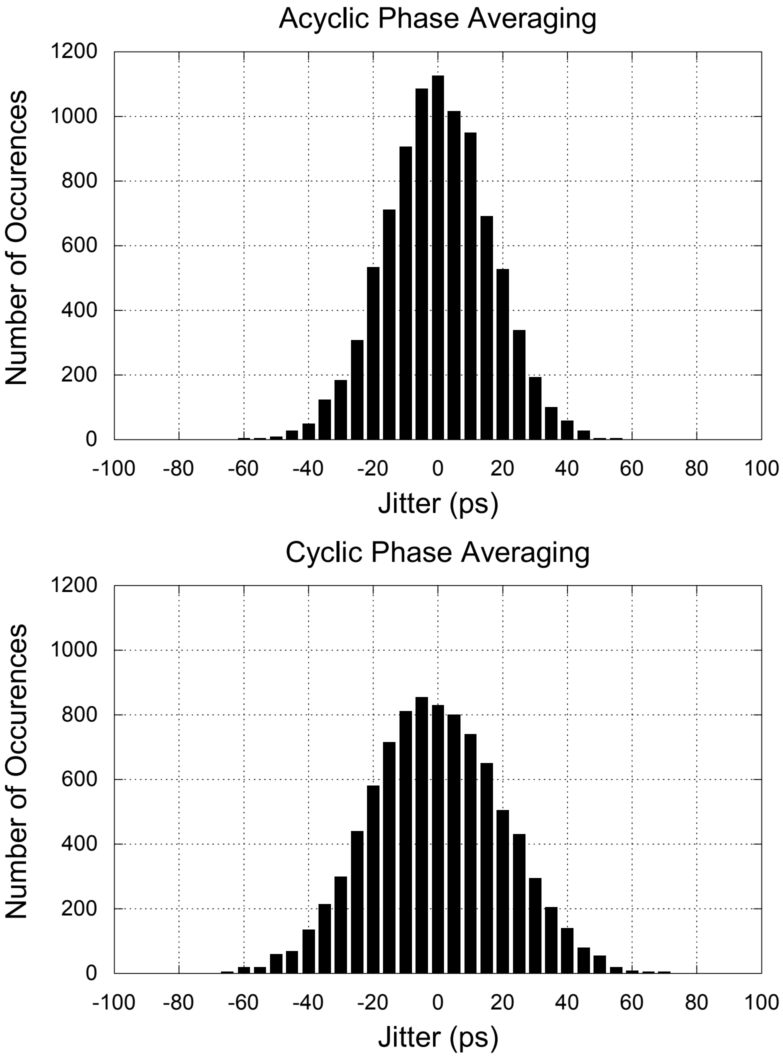


Figure 7.6: Jitter histograms.

The assumption that each clock distribution network has a single noise injection point is incorrect and is worth revisiting. It is much more realistic to suppose that each long interconnect introduces jitter. This makes a big difference because with respect to acyclic phase averaging, cyclic phase averaging uses twice as many feedback interconnects. The jitter histogram when each long interconnect is noisy is shown in Figure 7.6 for both systems. Jitter is measured at the output of the third oscillator, just before reaching the fourth phase detector. It is interesting to observe that cyclic phase averaging introduces slightly more jitter. Under the more realistic noise assumption, it appears that acyclic phase averaging is better then cyclic phase averaging.

Combining acyclic phase averaging with digital feedback also enables certain optimizations that are not possible with the cyclic phase averaging strategy presented in [7.5]. For instance, the slave PLLs described in the clock distribution network proposed here can be designed to maximize jitter attenuation, as opposed to minimizing the probability of mode locking. The reason is simply that mode locking is impossible by construction. With acyclic phase averaging and digital feedback, certain clock gating optimizations that would make a clock distribution network using analog cyclic phase averaging unstable also become possible.

7.2.2 Implementation of the Variable-Delay Element

The multi-PLL clock distribution architecture described in this chapter requires variable-delay elements for skew compensation. The variable-delay element proposed here is logically equivalent to an inverter. It is constructed using a digitally adjustable resistor.

In [7.7], a digitally adjustable resistor is used to control the impedance of a line driver. The resistor is built using a row of transistors having binary-weighted widths. The conductance of the row can be varied linearly, but its resistance cannot. In [7.1], a skew compensation circuit uses a variable-delay element constructed using inverters and transmission gates. The transmission gates are used to digitally connect or disconnect capacitors to the output of the inverters. Unfortunately, the delay steps are non-linear and remain relatively coarse. Another variable-delay element is proposed in [7.8]. It achieves a 26-ps resolution in a 350-nm fabrication technology. However, it requires an output multiplexer and occupies a rather large area.

The new circuit proposed here is a significant generalization of the ideas presented in [7.7] because multiple transistor rows are allowed and because the binary-weighted transistor width constraint is removed. The result is a digitally adjustable resistor that can be controlled with a considerably higher resolution and over a wider range.

The general circuit topology of the variable-delay inverter is shown in Figure 7.7. An *n*-bit control signal called *b* sets the delay between the rising input transition and the falling output transition. The pull-down stack uses an array or n-devices in which multiple rows are allowed. The transistor array actually forms the digitally adjustable resistor. Each bit of the control signal is connected to the gate of one transistor in the array.

The rising output transition cannot be controlled by the proposed circuit. However, two variable-delay elements in series would permit the manipulation of both edges. Alternatively, another digitally adjustable resistor build using p-devices could be added to the pull-up stack. Using p-devices instead of n-devices would perhaps reduce the substrate noise sensitivity of the circuit.

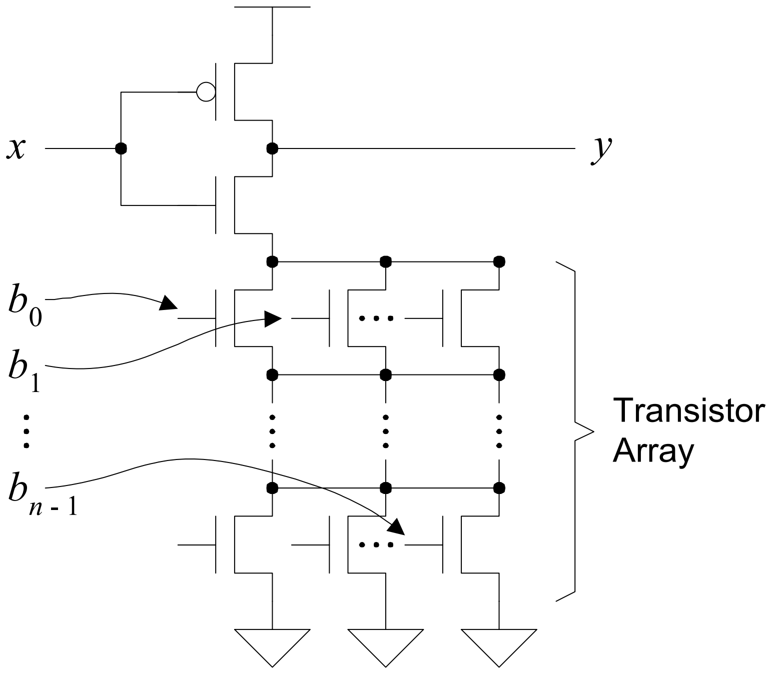


Figure 7.7: General circuit topology.

In Figure 7.7, it is reasonable to assume that all the control bits for a particular row cannot be simultaneously zero to guarantee that the output can switch. Every control bit combination that blocks the pull-down stack is considered illegal. The resistance of the pull-down stack is minimal when all the n transistors of the array are conducting. The maximum resistance is achieved when, for each row, only the smallest device conducts.

In general, different control bits produce different pull-down resistances. It is worth noting that a relatively small number of bits can produce a large number of resistance values (roughly 2^n). The problem is deciding which control bit combinations to use and which to avoid. For this, it is convenient to assign a resistance label L to each combination of control bits $b[n-1{:}0]$ that can be legally applied to the array. The resistance label is just an integer. The problem of choosing which combinations to apply to the digitally adjustable resistor then becomes a coding problem.

It is possible to carefully choose the transistors widths such that the resistance values are fairly evenly distributed between their minimum and maximum, like in Figure 7.8. The horizontal axis represents the control bits that must be applied to produce a particular resistance value. Each set of control bits determines which transistors are on and which are off. In other words, each set of control bits defines a particular combination of transistors connected in parallel and in series in the array.

With a large number of points densely scattered between R_{min} and R_{max}, the resistance of the pull-down stack can be varied in small steps. For any desired resistance value between R_{min} and R_{max}, the existence of a transistor combination producing nearly the same value is guaranteed if the scattering density is high enough. Mathematically, any desired pull-down resistance $R(L)$ lying between R_{min} and R_{max} can be very closely approximated by $\hat{R}(L)$ if the control bits associated with the resistance label L are chosen such that

$$\left|\hat{R}(L) - R(L)\right| \tag{7.2}$$

is minimized. Because the desired pull-down resistances are arbitrary, they can follow the points of any linear or non-linear function of L. The only restriction is that the function must be bounded by R_{min} and R_{max}.

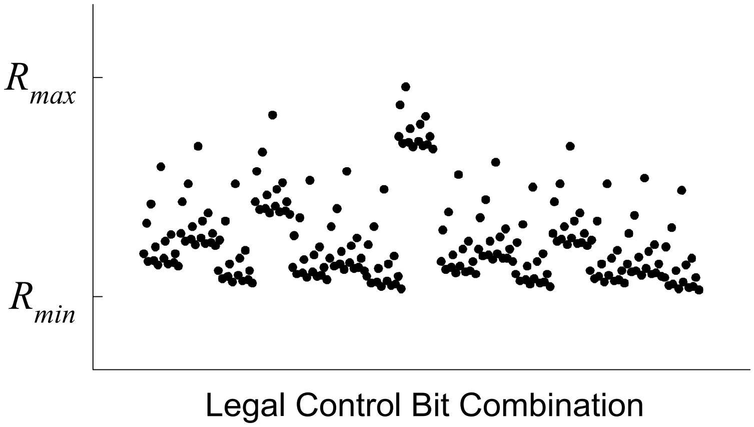


Figure 7.8: Resistance distribution.

When a digitally adjustable resistor is subjected to random process variations, the resistance corresponding to a particular control bit combination varies randomly from its nominal value. However, some control bit combinations can be more robust than others. For instance, if the channel length of every transistor is varied randomly, the resistance corresponding to the control bit combination labeled L_0 may deviate from its nominal value by 2%. The resistance corresponding to L_1, another control bit combination, may change by 5%. In other words, the resistance associated with L_0 is more stable than the resistance associated with L_1.

The control bits to apply to the digitally adjustable resistor can be chosen to take advantage of the fact that some combinations are more stable than others. Let L be the label of a particular set of control bits. Also, let the range of resistance values corresponding to L be defined by $\hat{R}_{min}(L)$ and $\hat{R}_{max}(L)$ when the digitally adjustable resistor is random perturbed. Mathematically, the best control bit combination to approximate a desired pull-down resistance $R(L)$ lying between R_{min} and R_{max} minimizes:

$$\max(\left|\hat{R}_{min}(L) - R(L)\right|, \left|\hat{R}_{max}(L) - R(L)\right|) \qquad (7.3)$$

Equivalently, the optimal control bit combination minimizes the worst-case resistance deviation from the desired value.

Figure 7.9 shows a variable-delay inverter having a 2-by-4 transistor array followed by a conventional inverter. The input is *x*. The output of the variable-delay inverter is *y*. A dummy load is connected to *z*, the output of the conventional inverter. The conventional inverter helps to protect the output of the variable-delay inverter against load variations. The purpose of the circuit is to adjust the timing of the rising edge of *z* in small linear steps. Varying the digitally adjustable resistor linearly is not the primary goal. In fact, by purposely choosing non-linear steps for the digitally adjustable resistor, some second-order delay variations due to the second inverter can be compensated. In particular, the digitally adjustable resistor is used here to compensate the variable shape and transition time of *y*.

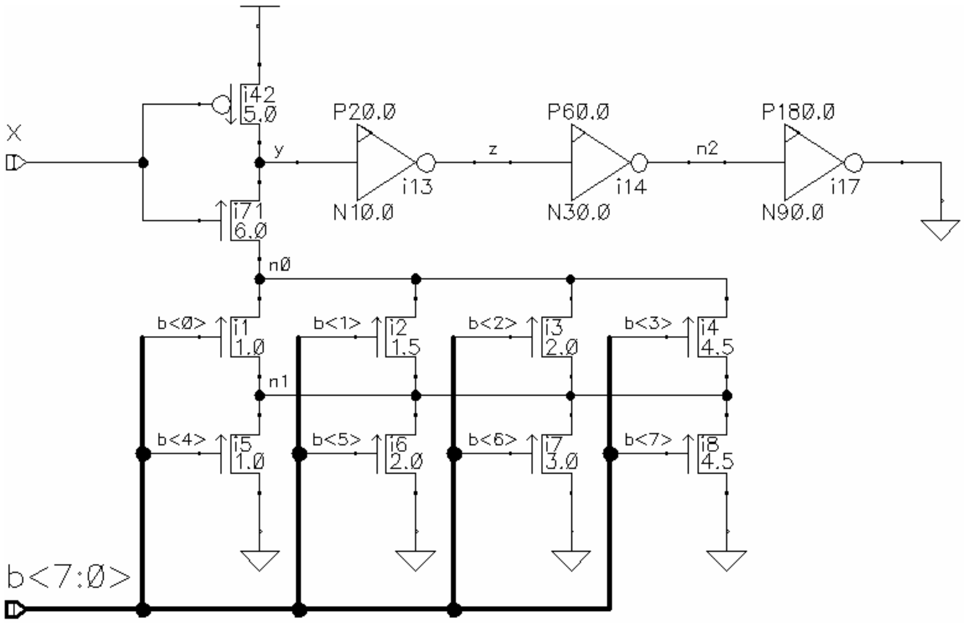


Figure 7.9: Simulation schematic.

Since the falling edge of *y* is adjustable, so is the rising edge of *z*. There are 256 possible control bit combinations: 225 are legal and 31 are illegal. By choosing 51 of the legal combinations as described earlier, the rising edge of *z* can be varied in 1-ps steps over a 50-ps range.

Figure 7.10 shows the simulated waveform associated with each combination. The delay steps are too small to make the 51 rising edges distinctly visible. However, the achievable delay points are clearly shown in Figure 7.11. The transition time of the rising edge is nearly constant. Obviously from the graph, the falling edge is practically not disturbed by the variable resistance of the pull-down stack. When the rising edge moves, the position of the falling edge remains within 0.5 ps of its average value. The reason for this stability is the n-device connected to node *y*. When *y* rises, *z* falls. But because that n-device is off when *y* rises, only the capacitance of *y* gets charged. Since the control bits do not affect this capacitance, they do not affect the rising delay of *y* either.

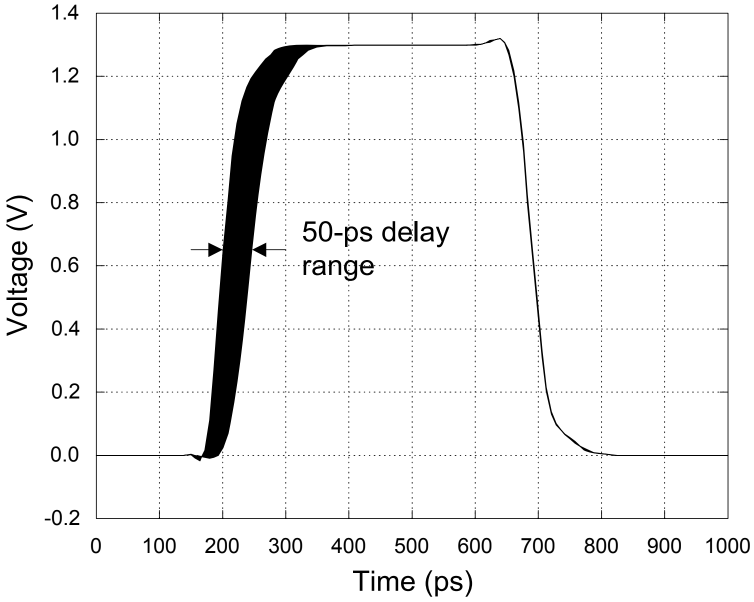


Figure 7.10: Simulated waveforms.

Figure 7.11 also shows the robustness of the digitally adjustable resistor against uncorrelated channel length variations. The points represent the delays nominally achievable. Each line represents a particular channel length variation experiment. For each experiment, the channel length of each transistor in the simulation schematic is varied randomly. Each experiment is run under nominal process, voltage, and temperature conditions. It is interesting to observe that the random variations tend to increase or decrease all the achievable delays relatively uniformly. The delay steps are not significantly affected in any of the random experiments and remain at 1 ps.

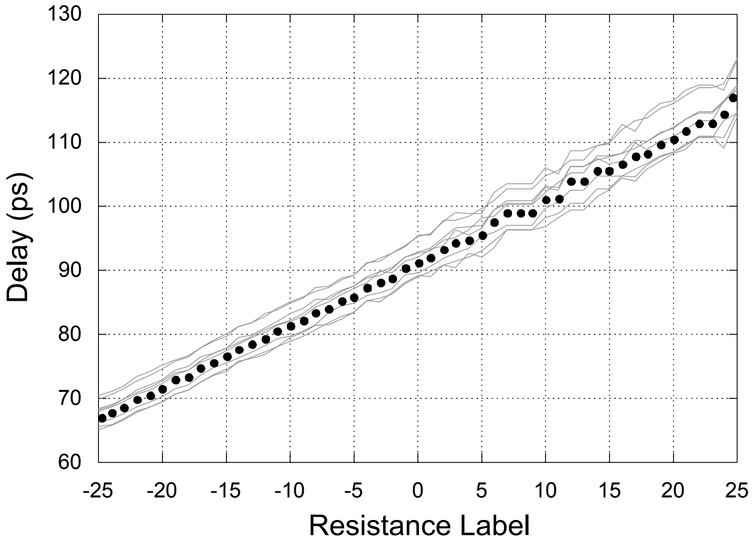


Figure 7.11: Effect of channel length variations on delay.

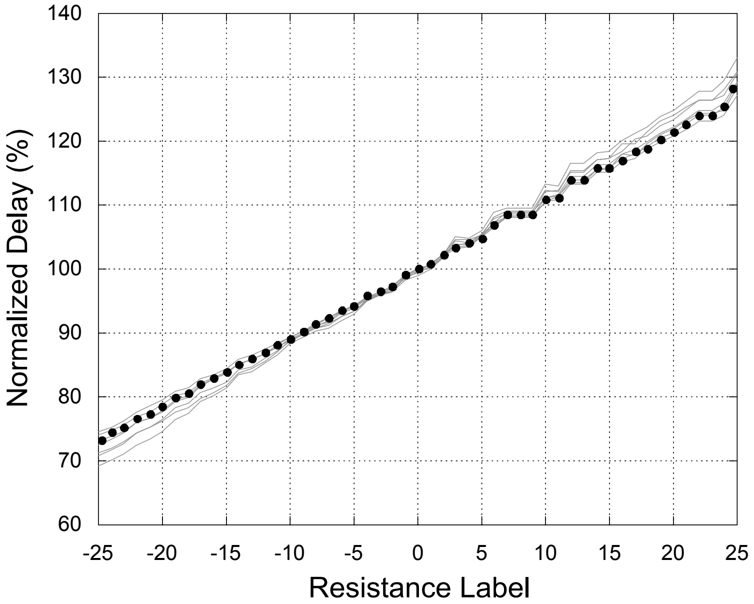


Figure 7.12: Effect of process, voltage, and temperature.

Figure 7.12 shows how the variable-delay inverter is affected when the design corner changes. More precisely, the supply voltage and the temperature are respectively varied by 25% and 50 degrees at various process corners. Like most circuits, the absolute speed of the variable-delay element varies considerably. However, for the purpose of comparing its behavior under the different design corners, it is convenient to normalize its speed. Here, all the curves are normalized to their average. Regardless of the design corner, the control bit combination labeled $L = -25$ is always about 30% faster than the combination labeled $L = 0$. The remarkable similarity of all the curves indicates that changing the process, voltage, and temperature conditions uniformly moves the delays points. In other words, although the design corner has a considerable influence on the absolute range of the variable-delay inverter, it does not significantly affect its linearity.

Here, the control bits are only optimized for the nominal design corner. However, a different set of control bit combinations could be used for each design corner to further improve the linearity of the circuit.

7.3 Clock Alignment Algorithms

There are many ways of using the digital feedback mechanism proposed earlier to keep the PLLs synchronized. This section describes the two clock alignment algorithms that were first proposed in [7.9]. Both algorithms are more general than the one later proposed by Dike *et al.* in [7.10] since they do not assume that mode locking is practically unlikely.

The two algorithms described here are designed for continuous operation. As in [7.10] and [7.11], they suppose that being able to compensate slow variations in temperature and supply voltage is important. But unlike the hierarchical skew compensation strategy used on the Pentium 4 [7.12], they do not suffer from phase error accumulation.

The first algorithm is the simpler of the two. It is called the spinning wheel algorithm. It imposes more difficult specifications for the design of the variable-delay elements. Dike's algorithm reuses many of the same ideas and is very similar to it. The second is the range sharing algorithm. It is more complex, but it also makes the variable-delay elements simpler to design.

7.3.1 Spinning Wheel Algorithm

The spinning wheel algorithm starts with the clock produced by the slave PLL located in the upper-left corner of the chip. This PLL is called the reference PLL. The idea is to propagate its phase along the main diagonal of the die until all the other slave PLLs have the same phase. The algorithm produces the same PLL alignment order as the spinning wheel shown in Figure 7.13.

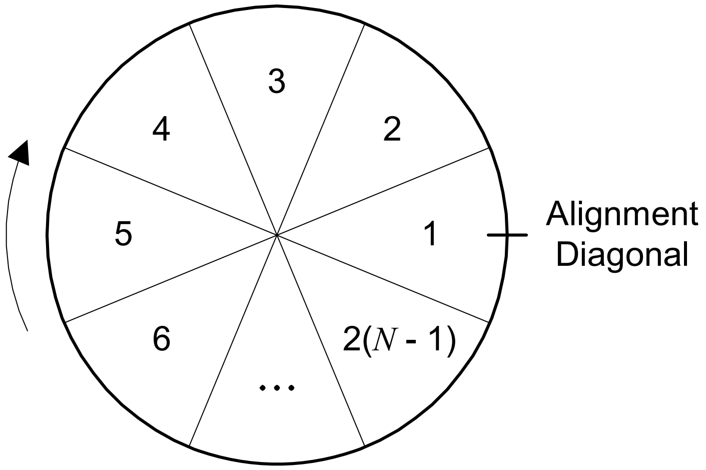


Figure 7.13: Spinning wheel phase alignment.

Generally, for an N-by-N array, the algorithm needs 2 × (N – 1) alignment steps. For the 4-by-4 array shown in Figure 7.14, six steps are required. The duration of each step is fixed. During step 1, the PLLs located on the first alignment diagonal (i.e. $PLL_{1,2}$ and $PLL_{2,1}$) adjust their phase to the phase of the corner PLL, $PLL_{1,1}$. During step 2, the PLLs located on the second diagonal align themselves to the PLLs located on the first diagonal. The other intermediate steps are similar. During step 6, the PLL located on the last diagonal finally aligns its phase to the phase of the PLLs located on the fifth diagonal. After the completion of the last step, the alignment process can occasionally be restarted for another pass to compensate slow temperature and supply voltage variations. It is worth noting that before the first pass, the clocks can be significantly misaligned. The reset controller of the chip should wait for its completion before signaling the end of the reset sequence.

7.3.1.1 Details

The spinning wheel algorithm can be implemented using two simple controllers: a global alignment controller and a local alignment controller. A single instance of the global controller is needed. One instance of the local controller is required for each variable-delay element.

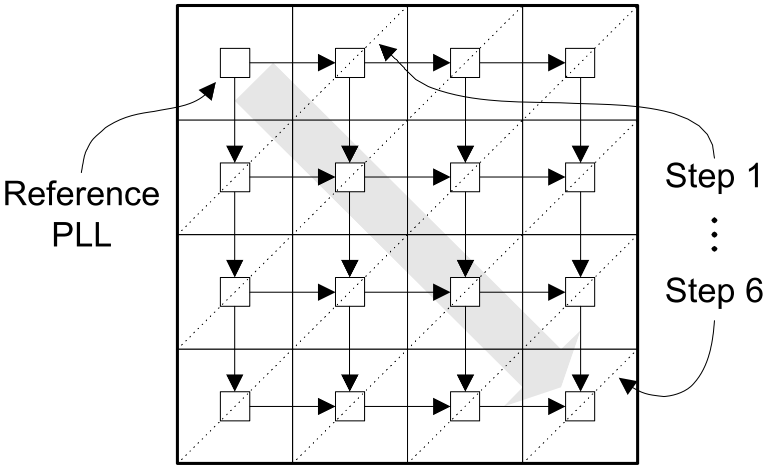


Figure 7.14: Unidirectional phase alignment.

The role of the global alignment controller is to manage the local controllers. It ensures that the alignment process correctly proceeds along the main diagonal. Specifically, the global controller needs to send a signal to enable the appropriate local controllers during each alignment step. The simplest strategy for this is to limit the duration of each alignment step. The global controller also decides when to restart the alignment process.

Each local alignment controller manages a variable-delay element. Every controller either receives one or two digital phase error signals. Because the phase error signals can be fairly noisy, they are filtered using a simple digital low-pass filter, as in [7.11]. The filtered signals are then used to decide if the delay of the variable-delay element should be increased or decreased. This digital filter can be implemented using shift registers and a majority decoder. Even though the local controllers can receive one or two phase error signals, a single circuit is sufficient. A PLL receiving just one phase error signal can be considered as receiving two signals that are always identical.

Multi-PLL clock distribution is justified in part because the clock produced by the master PLL can be significantly corrupted by skew and jitter when it reaches the variable-delay elements. This is a problem when it comes to clocking the alignment controllers. The solution is to use a frequency divider to produce a slower clock for them. The frequency of the slow clock has to be low enough (say 4 or 8 times slower) to make the inaccuracy of the fast clock a small fraction of the slow cycle. With this trick, all the alignment controllers can be clocked by the master PLL.

7.3.1.2 Analysis

There are two main problems with the spinning wheel algorithm. The first problem is related to the limited range of the variable-delay elements; the second is caused by their limited resolution.

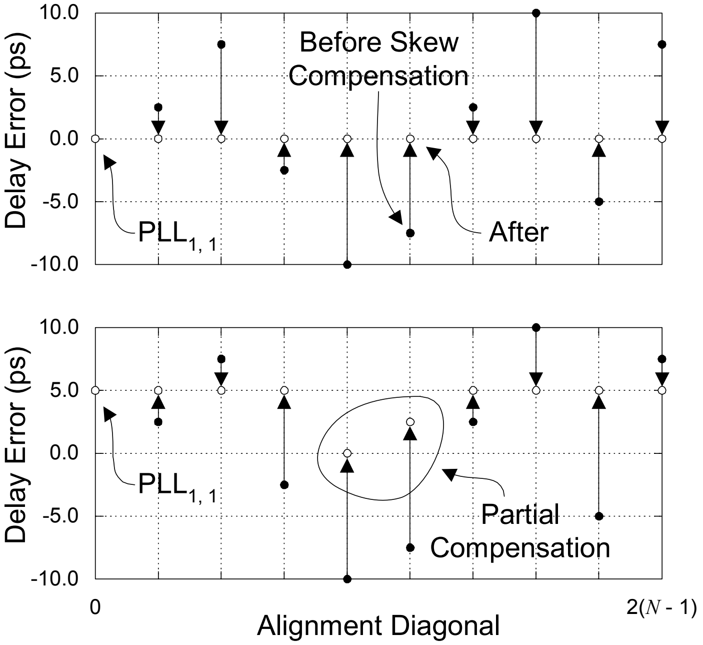


Figure 7.15: Complete (top) and incomplete (bottom) skew compensation.

The range of the variable-delay elements must be relatively wide to guarantee the compensation of the skew introduced by the clock distribution network. Figure 7.15 assumes that the distribution network introduces 20 ps of skew for a 10-GHz clock. The range of the variable-delay elements is also supposed 20 ps. In other words, they can increase or decrease their delay by a maximum of 10 ps. If the delay error for $\mathrm{PLL}_{1,1}$ is zero, the range of the variable-delay elements is sufficient to completely compensate the

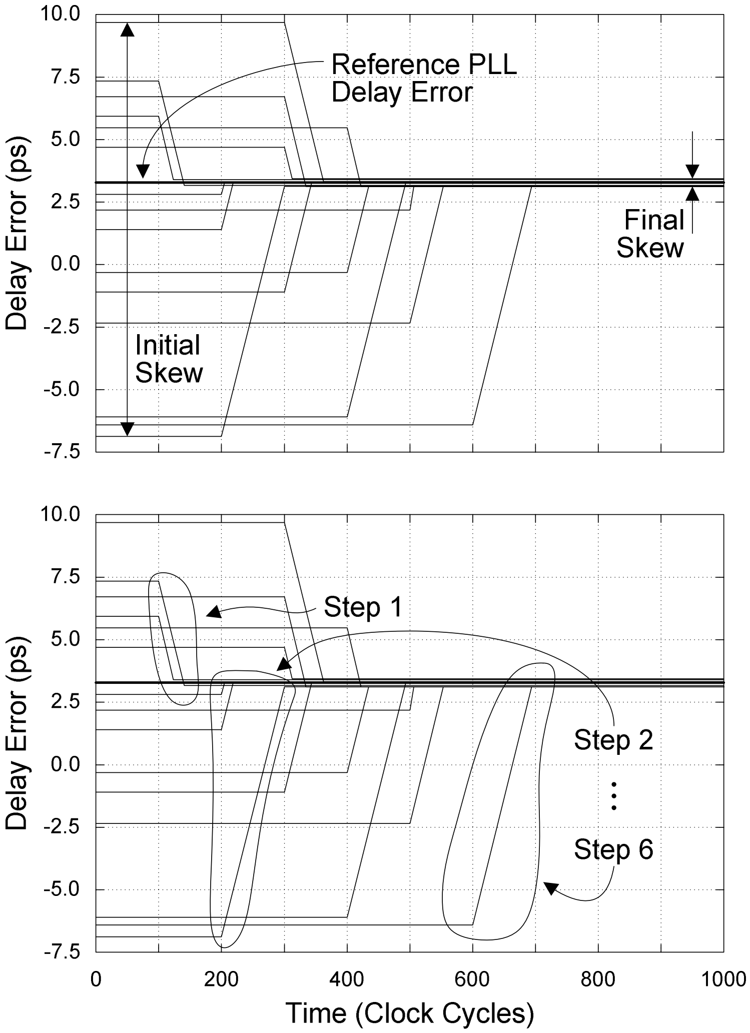


Figure 7.16: Spinning wheel skew reduction (top) and alignment steps (bottom).

skew. However, if the delay error for the reference PLL is 5 ps, the variable-delay elements may only be able to partially compensate the skew. In general, the spinning wheel algorithm requires the range of the variable-delay elements to be twice the worst-case skew to guarantee its compensation.

The second problem is the limited resolution of the variable-delay elements. When the phase of a PLL becomes aligned to the phase of its upstream neighbors, its local controller keeps updating the variable-delay element. This continuous update makes the phase oscillate above and below its ideal value. When the global controller instructs that

PLLs to stop updating its variable-delay element, the phase is within plus or minus a delay step of the desired value. Therefore, two converged clocks can be two delay steps apart. This is an issue if the variable-delay elements have a relatively coarse resolution.

7.3.1.3 Simulation Results for a 10-GHz Clock

The phase alignment process for a clock distribution network with 16 PLLs is shown in Figure 7.16 for a hypothetical 10-GHz clock. Initially, the clock inaccuracy is 16 ps. One slave PLL produces a clock reaching the sequential elements 9 ps too late. Another produces a clock reaching the sequential elements 7 ps too soon. The clock produced by the reference PLL is late by 3 ps with respect to its nominal arrival time.

The clocks are aligned one diagonal at a time. Each diagonal is enabled for 100 clock cycles by the global controller. This 100-cycle duration is determined *a priori* by the step size of the variable-delay elements, the bandwidth of the slave PLLs, and the amount of skew to compensate. If it is too short, the skew compensation process may not complete. If it is too long, the time required to synchronize all the PLLs may become prohibitive. The alignment process is clearly visible. At time 100, step 1 begins. The variable-delay elements of the two slave PLLs located of the first alignment diagonal are activated. Their clocks start to drift toward the reference clock and stop when they reach it. Step 2 begins at time 200 and is similar. After the completion of the 6 alignment steps, the global clock skew is reduced below 1 ps. The clock skew between adjacent PLLs is actually smaller.

Figure 7.17 shows what happens when the range of the variable-delay elements is insufficient. The problem is the rather large delay error of the reference PLL. The two slave PLLs located on the first alignment diagonal have enough range to reach the reference clock. However, two of the PLLs on the second alignment diagonal saturate before reaching it. Since the worst saturation point is 5 ps below the reference PLL, the residual skew is 5 ps.

The wires connecting the master PLL to the slave PLLs introduce a significant amount of jitter. However, this jitter is strongly attenuated by the slaves. Even more jitter is

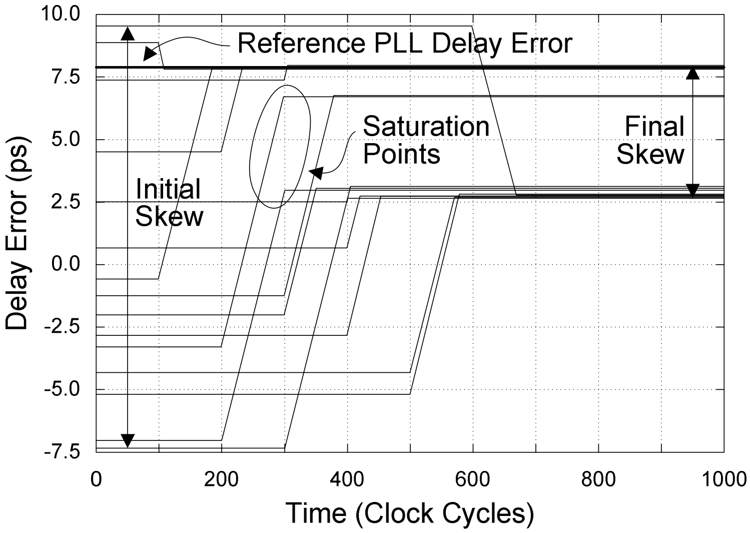


Figure 7.17: Out-of-range phase alignment condition.

introduced by the local clock distribution networks and by the phase detectors. This jitter is not filtered and actually dominates the clock inaccuracy at the end of the alignment process. For clarity, the jitter is not shown in Figure 7.16 and Figure 7.17.

7.3.2 Range Sharing Algorithm

The range sharing phase alignment algorithm is a solution to the inefficient range management of the spinning wheel algorithm. It is motivated by the general difficulty of constructing a circuit having both a high precision and a wide range. The higher complexity of the range sharing algorithm reduces the range required for the variable-delay elements. Reducing this range helps to make their delay step small. The step size that the variable-delay elements can achieve is important because it limits the amount of skew that can be compensated.

Just like the spinning wheel algorithm, the range sharing algorithm starts with the clock produced by the reference PLL located in the upper-left corner of the chip. Its phase is still propagated along the main diagonal of the die. However, if a variable-delay element reaches one of its delay limits, a signal is sent back to the reference PLL. The reference PLL can then try to share the range of its variable-delay element with the out-of-range PLL.

7.3.2.1 Details

The range sharing phase averaging algorithm requires three controllers: a global alignment controller, a local alignment controller, and a reference alignment controller. As its name implies, the reference alignment controller is needed for the reference PLL.

The global controller manages the alignment process along the main diagonal. Unlike the controller used by the spinning wheel algorithm, this controller must also manage the out-of-range conditions that the local controllers may signal. This is done by interrupting the alignment sequence along the main diagonal and forwarding a signal to the controller of the reference PLL.

The controller of the reference PLL decides how to manage the requests for phase adjustment coming from the downstream PLLs. A delay increase alleviates insufficient minimum range problems. A decrease alleviates insufficient maximum range problems. For example, the partial skew compensation problem shown in Figure 7.15 is a maximum range problem. It can be solved by decreasing the delay of the variable-delay element of the reference PLL by 5 ps. Of course, the variable-delay element of the reference PLL can also reach the limits of its compensation range. Then, range sharing is no longer possible.

7.3.2.2 Analysis

Range sharing can significantly increase the number of clock cycles required to complete the first skew compensation pass. The worst case is when the PLL located on the last alignment diagonal signals an out-of-range condition. Then, the variable-delay element of the reference PLL is updated and the alignment process restarts almost from scratch.

A simple solution to limit the number of iterations required to complete the first compensation pass is to use large delay steps for the reference PLL. When a local controller signals an out-of-range condition, the variable-delay element of the reference PLL can be adjusted by one step in the appropriate direction. The larger the step, the faster the reference PLL can reach an acceptable position. The drawback of using large steps is that if the window of acceptable positions is too small, it may be missed. Another solution is to perform a binary search to find an acceptable setting for the variable-delay element of the reference PLL. The variable-delay element of the reference PLL starts with a large step for fast convergence. Then, the step size is cut in half whenever an out-of-range condition occurs. This ensures that if the reference controller has a setting that prevents saturation, it is not missed.

7.3.2.3 Simulations Results for a 10-GHz Clock

The phase alignment process for a clock distribution network with 16 PLLs is shown in Figure 7.18. Like before, the variable-delay elements have a 20-ps range. They can increase or decrease their delay by up to 10 ps. Initially, the reference PLL has a delay error of -6 ps and one of the slave PLLs has an error of almost 10 ps. Since each error can only be moved 10 ps, the slave PLL can only reach 0 ps. This means that the limited range of its variable-delay element would cost 6 ps of skew with the spinning wheel algorithm.

Here, when the variable-delay element of that slave PLL saturates, a message is sent to the reference controller. This controller then increases the delay of its variable-delay element by 5 ps to help the variable-delay element that is out-of-range. After this, the alignment process continues. Of course, a 5-ps delay adjustment cannot completely bridge the original 6-ps gap, but the adjustment reduces it to 1 ps.

The variable-delay element of the slave PLL still saturates with a 1-ps gap. This eventually produces another message for the reference controller. The step size is cut in half and the delay is increased again. The new adjustment makes the phase of the reference PLL go too high because it makes some variable-delay elements saturate. The overshoot occurs at time 400 and is corrected at time 800 by the binary search process. After the determination of a correct setting for the variable-delay element of the reference PLL, the range sharing algorithm behaves like the spinning wheel algorithm and rapidly converges. At the end of the alignment process, the skew between adjacent PLLs is well below 1 ps.

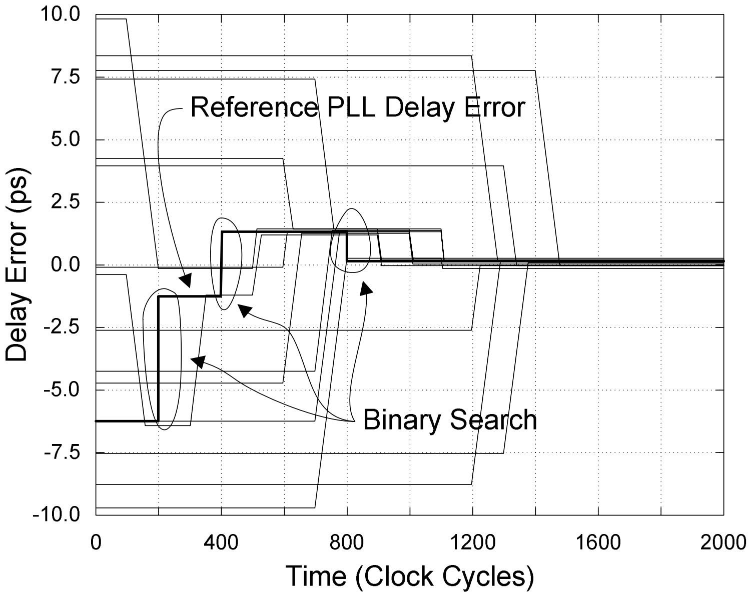


Figure 7.18: Range sharing phase alignment.

7.3.3 Comparison to Hierarchical Skew Compensation

Unlike the hierarchical skew compensation algorithm used on the Pentium 4 microprocessor [7.12], the spinning wheel and range sharing algorithms do not suffer from phase error accumulation.

The problem with the hierarchical skew compensation strategy used on the Pentium 4 is that the phase of some physically adjacent clock regions is processed through several phase detectors in series. Because these phase detectors are subject to within-die variations, they are always biased. This bias results in incorrect feedback signals for the variable-delay elements and introduces clock skew. For instance, with the binary phase comparison network feeding the 4-by-4 array shown in Figure 7.19, some adjacent regions are separated by five phase detectors while some non-adjacent regions are separated by seven detectors. The minimum step size of the variable-delay elements also introduces skew. As shown in Figure 7.20, the skew introduced depends if the phase comparison begins with an early clock or a late clock. When the adjustable clock is early, its phase is increased one step size at a time, until it passes the trip point of the phase detector. Then, the phase of the adjustable clock is guaranteed to be to the right of the trip

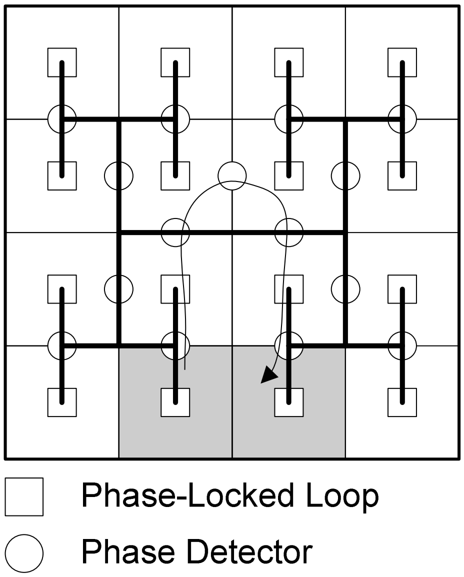


Figure 7.19: Hierarchical skew compensation.

point. Conversely, starting with a late clock yields a final phase that is always left of the trip point.

As opposed to the hierarchical skew compensation strategy of the Pentium 4, the spinning wheel and range sharing algorithms ensure that adjacent clock regions are always separated by a single phase detector.

The advantage of having a single phase detector between adjacent clock regions can be quantified by modeling the bias B of each phase detector as a zero-mean Gaussian random variable:

$$B \sim N(0, \sigma_B) \tag{7.4}$$

The skew M due to the minimum step size Δ of the variable-delay elements can be modeled as a random variable uniformly distributed over $[0, \Delta]$:

$$M \sim U(0, \Delta) \tag{7.5}$$

The total skew S for a phase detector where the phase comparison starts with an early clock is:

$$S = B + M \tag{7.6}$$

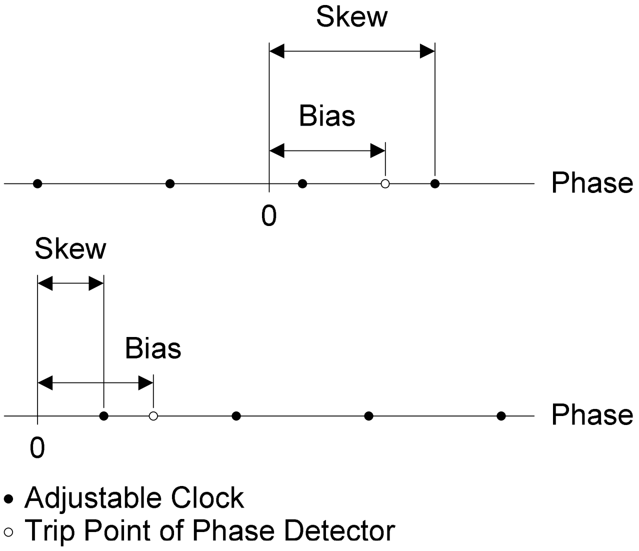


Figure 7.20: Phase detector skew.

If the phase comparison starts with a late clock, *S* is given by:

$$S = B - M \tag{7.7}$$

With hierarchical skew compensation, it is advantageous to alternate the direction of the phase comparisons across the levels of the hierarchy. Since the mean of *M* is half a step size, alternating the direction helps the variable-delay elements avoid saturation. Alternating the direction of the phase comparisons is also possible with the spinning wheel and range sharing algorithms.

Figure 7.21 shows the results of a numerical simulation of the skew accumulated through the five phase detectors of Figure 7.19. The bias of the phase detectors has a standard deviation σ_B of 1.0 ps. The minimum step size Δ of the variable-delay elements is also 1.0 ps. With hierarchical skew compensation, the skew is 2.2 times higher than with either the spinning wheel or the range sharing algorithm.

7.4 Summary

The multi-PLL clock distribution proposed here is compatible with conventional design-for-debugability (DFD) and design-for-testability (DFT) techniques. The slave PLLs

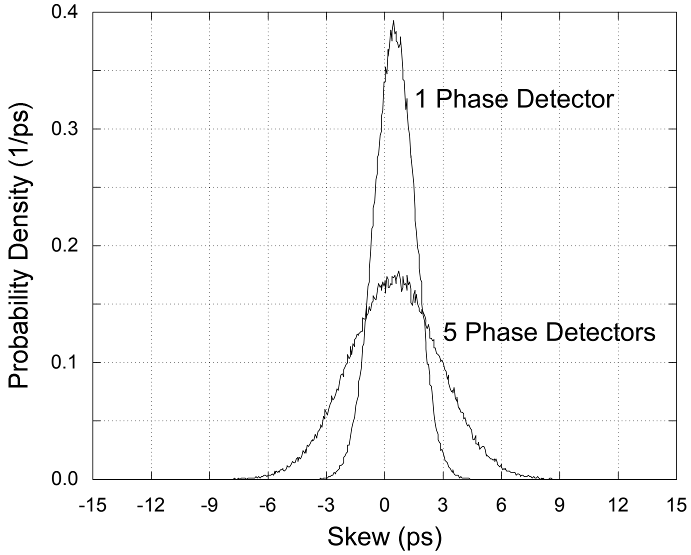


Figure 7.21: Skew accumulation due to phase detectors.

significantly reduces the jitter accumulation problem associated with conventional clock distribution techniques. They also enable independent frequency and voltage scaling for the regional core clocks for better power reduction.

The proposed network of variable-delay elements can perform skew compensation even when different frequencies are generated for the regional core clocks. The spinning wheel phase alignment algorithm has a lesser micro-architectural impact and is faster than the range sharing algorithm. However, it is not as efficient at compensating skew. Nevertheless, its performance can still be acceptable if the range of the variable-delay elements used for skew compensation is wide enough. Generally, the range sharing algorithm is preferable only if the skew to compensate can exceed half the range of the variable-delay elements. Both algorithms outperform the hierarchical skew compensation algorithm used on the Pentium 4 microprocessor [7.12] by significantly reducing the skew accumulated through the phase detectors. For an array of 4-by-4 local clock regions, the skew is 2.2 times higher with hierarchical compensation.

CHAPTER 8

Conclusions

The objective of this dissertation is to better understand clock distribution in order to identify opportunities and strategies for improvement by analyzing the conditions under which the optimal tradeoff between power and performance can be achieved, by modeling the constraints associated with local and global clocking, by evaluating the impact of noise, and by investigating promising new design strategies for future integrated systems.

This chapter summarizes the important conclusions of the dissertation and the main contributions of this research. It also proposes possible areas for additional research.

8.1 New Quasi-Linear MOSFET Model

The MOSFET model described in Chapter 2 is mathematically simpler than the one proposed by Sakurai and Newton about 15 years ago [2.1]. When the device is in saturation, the new model is more accurate because it does not neglect channel-length modulation. There, the drain current is modeled as a *linear* function of the gate-to-source and drain-to-source voltages. The fixed drain-to-source voltage separating the linear and saturation regions of operation also contributes to the model's mathematical simplicity.

8.2 Optimal Tradeoff Between Power and Sequencing Overhead

The analysis of the tradeoff between power and sequencing overhead is based on two generic power and frequency models. The first model describes how the dynamic power, the short-circuit power, and the subthreshold leakage power of a synchronous digital system vary with the supply voltage. Its validity is confirmed by ring oscillator simulations and by measurements taken on a 200-nm G4 microprocessor [3.11]. The frequency scaling model only captures the first-order supply voltage dependence, but can still describe accurately the measurements taken on a 150-nm SPARC V9 microprocessor [3.9] and on a 130-nm Itanium microprocessor [3.17]. The notions of the power-performance cost and benefit are then formally defined. It is mathematically shown that the tradeoff between power and performance is locally optimal if and only if the cost of any design change equals its benefit.

8.3 Local Clocking

For flip-flops, the setup time required for capturing a zero is generally different from the setup time for capturing a one. So are the times required for launching a zero and a one. For paths whose polarity cannot be determined *a priori*, the sequencing overhead is the worst-case setup time plus the worst-case launch time. The optimal setup time is traditionally defined as the setup time minimizing that sequencing overhead. The generalized definition proposed in Section 4.2 minimizes the sequencing overhead under all possible clock arrival times. Unlike the traditional definition, the generalized one accounts for flip-flop transparency and for clock inaccuracy. A model for the data-dependant clock jitter resulting from the switching activity of sequentials with naked clocks is proposed. It is shown that the impact of the clock transition time on the sequencing overhead is significant.

Section 4.3 derived a model to analyze the properties of the local clock buffers used in [4.4]. The model is applied to better understand how the delay tracks the supply voltage. It shows that the impact of device sizing on tracking is relatively small and that sizing the devices for equal rise and fall delays is not mandatory. The model is also used to analyze the gain of the local clock buffer. It quantifies the relationship between gain and delay. It shows that local clock buffers driving small loads can suffer a significant gain loss due to the parasitic wire capacitance of their internal nodes.

Finally, a non-linear model is proposed for the bandwidth of a local clock buffer driving a local interconnect. The non-linearity is required to adequately capture the behavior of the devices. The model is used to analyze crosstalk jitter and to examine the conditions where clock shielding is advantageous from a power standpoint. A typical 130-nm local clock interconnect is analyzed, with and without shielding. The results show that for a hypothetical 1.2-V chip running at 5.0 GHz and with parameters similar to the microprocessor described in [4.4], removing the shields can save 2.14 W of power while only increasing the crosstalk jitter by 1.7 ps.

8.4 Global Clocking

The power required for global clock distribution is fairly small, but not negligible. Using minimum rectilinear Steiner trees (MRSTs), it is shown that when the number of loads is very small, the fixed interconnect cost associated with a full or partial clock grid makes H-trees a better choice for minimizing power. However, as the number of loads increases, the difference between the three structures is found to practically vanish. It is also shown that the dispersion of the loads significantly impacts the total length of the wires required to connect them, and therefore, the power required for global clock distribution.

It is observed that several skew and jitter compensation strategies require sending signals over long distances and across multilevel interconnect structures, where the capacitive coupling between adjacent layers creates interlevel coupling noise. The power spectral density for this interlevel coupling noise at the far-end of a victim line is rigorously derived. The result is a closed-form expression that accounts for the switching activity of the attackers and the electrical parameters of the victim.

A probabilistic bound for the magnitude of the interlevel coupling noise is then derived. It formally shows that for attackers with an uncorrelated switching activity, the assumption that the conductors orthogonally routed above and below the victim behave as a quiet metal plane is statistically very good.

8.5 Timing Impact of Power-Supply Noise

The power-supply noise injection models derived in Chapter 6 for a device driving a lumped load or a long interconnect are accurate. The mathematical complexity introduced by making the load distributed instead of lumped is significant, but the simplicity of the quasi-linear device model of Chapter 2 enables an analytical solution.

When analyzing the impact of power-supply noise on timing, the commonly used assumption that minimizing the *peak* of the supply voltage drop maximizes performance is wrong for the circuits of Section 6.2. It is more general to consider the average supply voltage while a circuit is switching. When the switching window is short with respect to the duration of the noise perturbation, minimizing the average voltage drop is equivalent to minimizing its peak. However, when the switching window is comparable to the duration of the noise perturbation, the peak is no longer sufficient to predict the performance of the circuit. The time-domain characteristics of the noise waveform become important. Fortunately, for typical device-dominated paths, these characteristics can be summarized reasonably well by averaging the noise waveform.

The board-level measurements presented in Section 6.4 show that the supply voltage can drop by 4.2% for the 2.53-GHz Pentium 4 microprocessor. They also indicate the presence of differential noise (1.3% of the supply voltage). A model representing the voltage regulator, the system board, the package, and the on-chip power distribution network is used to extrapolate the on-chip noise based on these board-level measurements. The timing impact of the board-level power-supply noise is an estimated 6.7% frequency reduction. The total frequency penalty including the on-chip power-supply noise is estimated at 21.8%.

8.6 Clock Distribution Using Multiple PLLs

The multi-PLL clock distribution proposed in Chapter 7 is naturally suited for multi-core microprocessors. It is compatible with conventional design-for-debugability (DFD) and design-for-testability (DFT) techniques. The slave PLLs significantly reduce the jitter accumulation problem associated with conventional clock distribution techniques. They also enable independent frequency and voltage scaling for the regional core clocks for better energy efficiency.

8.7 Digital Skew Compensation

Chapter 7 also proposed a network of variable-delay elements to perform skew compensation, even when different frequencies are generated for the regional core clocks. The spinning wheel phase alignment algorithm has a lesser micro-architectural impact than the range sharing algorithm. It is also simpler. However, the spinning wheel algorithm is not as efficient at skew compensation. Nevertheless, its performance can still be acceptable if the range of the variable-delay elements used for skew compensation is wide enough. Generally, the range sharing algorithm is preferable only if the skew to compensate can exceed half the range of the variable-delay elements. Both algorithms outperform the hierarchical skew compensation algorithm used on the Pentium 4 microprocessor [7.12] by significantly reducing the skew accumulated through the phase detectors.

8.8 Future Work

One of the possible extensions of this dissertation includes research to better understand the impact of random local clock buffer (LCB) placement on global clock power. High-gain LCBs tend to have a relatively small input capacitance. In practice, the global clock power required to charge and discharge the input capacitance of such LCBs can often be neglected compared to the power consumed by the global clock interconnects. When this is the case, the power required for global clocking is determined to a large extent by the number and position of the LCBs as well as by the routing strategy used to connect them. The simple scenarios analyzed in Chapter 5 could be supplemented by including realistic electrical limits on maximum interconnect delay and on maximum LCB load per interconnect.

Another research avenue is half-frequency clocking using dual-edge triggered (DET) flip-flops for lower power. DET flip-flops are flip-flops that capture data on both the rising and falling edges of the clock. They are more complex than conventional flip-flops. DET flip-flops typically contain more circuitry and thus occupy more area. They also tend to make design for testability more difficult or expensive. However, half-frequency clocking promises to potentially cut in half the power used for global and local clocking. To achieve this potential, low-power DET flip-flops are needed. In addition, an efficient strategy is required to disable the clock for idle DET flip-flops. The reason is that dual-edge triggering can complicate clock gating. If a particular clock edge supplied to a DET flip-flop is suppressed to save power, the direction of the next clock edge must be flipped compared to what it would have been without gating. Furthermore, if the power consumed using DET flip-flops in response to a single clock transition is much higher than the one that would be consumed using regular flip-flops, the power advantage resulting from half-frequency clocking may be lost completely.

To better understand the impact of power-supply noise, it would also be interesting to investigate measurement techniques to better probe the supply voltage waveforms actually reaching the devices under realistic conditions of operation. Accurate probing at the device level is very challenging for high-frequency microprocessors. First, physically reaching the device to measure through the package can be difficult. In addition, the small amplitude and large bandwidth of the waveform of interest, i.e. the power-supply noise waveform, make it difficult to distinguish from the measurement noise. Care must also be taken to avoid perturbing the signal of interest with the impedance of the probe. These measurements would help clarify the statistical properties of the power-supply noise on a cycle-by-cycle basis with arbitrary input vectors. They would also help validate the argument that when predicting the timing impact of the noise on a circuit, the average supply voltage during switching is more important that the peak of the noise.

The multi-PLL clock distribution architecture of Chapter 7 allows several clock domains, each getting a different supply voltage and running at a different frequency. The ability to simultaneously perform dynamic voltage and frequency scaling on multiple clock domains promises important power savings. It also makes exchanging data between these domains more difficult. This creates opportunities for research into synchronization strategies optimized for that purpose.

8.9 Publications

This research has resulted in the following publications:

1. M. Saint-Laurent and M. Swaminathan, "Impact of Interconnects on the Optimal Power-Performance Tradeoff for Clock Distribution in Microprocessors", IEEE Electrical Performance of Electronic Packaging, pp. 311-314, 2000.

2. M. Saint-Laurent and M. Swaminathan, "A Multi-PLL Clock Distribution Architecture for Gigascale Integration", IEEE Computer Society Workshop on VLSI, pp. 30-35, 2001.

3. M. Saint-Laurent, M. Swaminathan, and J. D. Meindl, "On the Micro-Architectural Impact of Clock Distribution Using Multiple PLLs", IEEE International Conference on Computer Design, pp. 214-220, 2001.

4. M. Saint-Laurent, P. Zarkesh-Ha, M. Swaminathan, and J. D. Meindl, "Optimal Clock Distribution with an Array of Phase-Locked Loops for Multiprocessor Chips", IEEE Midwest Symposium on Circuits and Systems, pp. 454-457, 2001.

5. M. Saint-Laurent and M. Swaminathan, "A Digitally Adjustable Resistor for Path Delay Characterization in High-Frequency Microprocessors", IEEE Southwest Symposium on Mixed-Signal Design, pp. 61-64, 2001.

6. M. Saint-Laurent and G. P. Muyshondt, "A Digitally Controlled Oscillator Constructed Using Adjustable Resistors", IEEE Southwest Symposium on Mixed-Signal Design, pp. 80-82, 2001.

7. M. Saint-Laurent, Z. Ajmal, M. Swaminathan, and J. D. Meindl, "A Model for Interlevel Coupling Noise in Multilevel Interconnect Structures", IEEE International Interconnect Technology Conference, pp. 110-112, 2001.

8. M. Saint-Laurent, V. G. Oklobdzija, S. S. Singh, and M. Swaminathan, "Optimal Sequencing Energy Allocation for CMOS Integrated Systems", IEEE International Symposium on Quality Electronic Design, pp. 194-199, 2002.

9. M. Saint-Laurent and M. Swaminathan, "Impact of Power-Supply Noise on Timing in High-Frequency Microprocessors", IEEE Electrical Performance of Electrical Packaging, pp. 261-264, 2002.

10. M. Saint-Laurent and M. Swaminathan, “Impact of Power-Supply Noise on Timing in High-Frequency Microprocessors”, IEEE Transactions on Advanced Packaging, Volume 27, pp. 135-144, February 2004.

11. M. Saint-Laurent and M. Swaminathan, “A Model for Power-Supply Noise Injection in Long Interconnects”, IEEE International Interconnect Technology Conference, pp. 113-115, June 2004.

12. M. Saint-Laurent, “A Model for Interlevel Coupling Noise in Multilevel Interconnect Structures”, Submitted to IEEE Transactions on Electron Devices for Publication, August 2004.

8.10 Summary

Integrated systems with billions of transistors on a single chip are a now reality. These systems include multi-core microprocessors and are built today using deca-nanometer devices organized into synchronous digital circuits. Appropriate clocking strategies are imperative to keep the devices properly coordinated. These strategies have a significant impact on the frequency of operation and, consequently, on the performance of the systems. The clocks are also responsible for a large fraction of the power consumed by these systems.

The conditions under which the optimal tradeoff between power and performance can be achieved have been derived. Models for the constraints associated with local and global clocking have been developed and validated. The impact on timing of power-supply noise and of interlevel coupling noise has been modeled and analyzed. Finally, promising new design strategies for future integrated systems have been proposed.

Possible extensions of this dissertation include research to better understand the impact of random local clock buffer placement on global clock power, half-frequency clocking using dual-edge triggered flip-flops, the properties of the supply voltage waveforms actually reaching the devices under realistic conditions of operation, and the synchronization strategies that should be used for efficiently crossing clock domains having different supply voltages and running at different frequencies.

References

[1.1] S. D. Naffziger *et al.*, "The Implementation of Itanium 2 Microprocessor", IEEE Journal of Solid-State Circuits, November 2002.

[1.2] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-Power Dissipation in a Microprocessor", ACM International Workshop on System Level Interconnect Prediction, February 2004.

[1.3] P. J. Restle *et al.*, "A Clock Distribution Network for Microprocessors", IEEE Journal of Solid-State Circuits, May 2001.

[1.4] A. Asenov, S. Kaya, and J. H. Davies, "Intrinsic Threshold Voltage Fluctuations in Decananometer MOSFETs Due to Local Oxide Thickness Variations", IEEE Transactions on Electron Devices, January 2002.

[1.5] E. Essini, "On the Modeling of Surface Roughness Limited Mobility in SOI MOSFETs and Its Correlation to the Transistor Effective Field", IEEE Transaction on Electron Devices, March 2004.

[1.6] R. W. Keyes, "Effect of Randomness in the Distribution of Impurity Ions on FET Thresholds in Integrated Electronics", IEEE Journal of Solid-State Circuits, August 1975.

[1.7] X. Tang, V. K. De, and J. D. Meindl, "Intrinsic MOSFET Parameter Fluctuations Due to Random Dopant Placement", IEEE Transactions on VLSI Systems, December 1997.

[1.8] A. Asenov *et al.*, "Simulation of Intrinsic Parameter Fluctuations in Decananometer and Nanometer-Scale MOSFETs", IEEE Transactions on Electron Devices, September 2003.

[1.9] S. Xiong et al., "Is Gate Line Edge Roughness a First-Order Issue in Affecting the Performance of Deep Sub-Micro Bulk MOSFET Devices?", IEEE Transactions on Semiconductor Manufacturing, August 2004.

[1.10] A. Asenov, S. Kaya, and A. R. Brown, "Intrinsic Parameter Fluctuations in Decananometer MOSFETs Introduced by Gate Line Edge Roughness", IEEE Transactions on Electron Devices, May 2003.

[1.11] C. Ouyang *et al.*, "An Analytical Model of Multiple ILD Thickness Variation Induced by Interaction of Layout Pattern and CMP Process", IEEE Transactions on Semiconductor Manufacturing, August 2000.

[1.12] B. E. Stine *et al*., “The Physical and Electrical Effects of Metal-Fill Patterning Practices for Oxide Chemical-Mechanical Polishing Processes”, IEEE Transactions on Electron Devices, March 1998.

[1.13] R. Chang, Y. Cao, C. J. Spanos, “Modeling the Electrical Effects of Metal Dishing Due to CMP for On-Chip Interconnect Optimization”, IEEE Transactions on Electron Devices, 2004.

[1.14] F. Chen and D. Gardner, “Influence of Line Dimensions on the Resistance of Cu Interconnections”, IEEE Electron Device Letters, December 1998.

[1.15] A. K. Wong, “Microlithography: Trends, Challenges, Solutions, and Their Impact on Design”, IEEE Micro, April 2003.

[1.16] L. Chen, L. S. Milor, C. H. Ouyang, W. Maly, and Y.-K. Peng, “Analysis of the Impact of Proximity Correction Algorithms on Circuit Performance”, IEEE Transactions on Semiconductor Manufacturing, August 1999.

[1.17] M. Orshansky, L. Milor, and C. Hu, “Characterization of Spatial Intrafield Gate CD Variability, Its Impact on Circuit Performance, and Spatial Mask-Level Correction”, IEEE Transactions on Semiconductor Manufacturing, February 2004.

[1.18] D. W. Boerstler, “A Low-Jitter PLL Clock Generator for Microprocessors with Lock Range of 340-612 MHz”, IEEE Journal of Solid-State Circuits, April 1999.

[1.19] H.-T. Ahn, and D. J. Allstot, “A Low-Jitter 1.9-V CMOS PLL for UltraSPARC Microprocessor Applications”, IEEE Journal of Solid-State Circuits, March 2000.

[1.20] A. Asenov, R. Balasubramaniam, A. R. Brown, and J. H. Davies, “RTS Amplitudes in Decananometer MOSFETs: 3-D Simulation Study”, IEEE Transactions on Electron Devices, March 2003.

[1.21] Z. Çelik-Butler, “Low-Frequency Noise in Deep-Submicron Metal-Oxide-Semiconductor Field-Effect Transistors”, IEE Proceedings Circuits, Devices, and Systems, February 2002.

[1.22] Y.-M. Jiang and K.-T. Cheng, “Vector Generation for Power Supply Noise Estimation and Verification of Deep Submicron Designs”, IEEE Transactions on VLSI Systems, April 2001.

[1.23] E. G. Friedman, “Clock Distribution Networks in Synchronous Digital Integrated Circuits”, Proceedings of the IEEE, May 2001.

[1.24] D. Harris, M. Horowitz, and D. Liu, "Timing Analysis Including Skew", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, November 1999.

[1.25] D. W. Bailey and B. J. Benschneider, "Clocking Design and Analysis for a 600-MHz Alpha Microprocessor", IEEE Journal of Solid-State Circuits, November 1998.

[1.26] N. A. Kurd *et al.*, "Multi-GHz Clocking Scheme for Intel Pentium 4 Microprocessor", IEEE International Solid-State Circuits Conference, 2001.

[1.27] S. Tam *et al.*, "Clock Generation and Distribution for the First IA-64 Microprocessor", IEEE Journal of Solid-State Circuits, November 2000.

[1.28] Q. Zhu and S. Tam, "Package Clock Distribution Design Optimization for High-Speed and Low-Power VLSI's", IEEE Transactions on Components, Packaging, and Manufacturing Technology—Part B, February 1997.

[1.29] K. B. Hardin, J. T. Fessler, and D. R. Bush, "Spread Spectrum Clock Generation for the Reduction of Radiated Emissions", IEEE International Symposium on Electromagnetic Compatibility, 1994.

[1.30] Y. Lee and R. Mittra, "Electromagnetic Interference Mitigation by Using a Spread-Spectrum Approach", IEEE Transactions on Electromagnetic Compatibility, May 2002.

[1.31] H.-H. Chang, I-H. Hua, and S.-I. Liu, "A Spread-Spectrum Clock Generator With Triangular Modulation", IEEE Journal of Solid-State Circuits, April 2003.

[2.1] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas", IEEE Journal of Solid-State Circuits, April 1990.

[2.2] M. Miura-Mattausch, U. Feldmann, A. Rahm, M. Bollu, and D. Savignac, "Unified Complete MOSFET Model for Analysis of Digital and Analog Circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, January 1996.

[2.3] B. Austin, K. Bowman, X. Tang, and J. D. Meindl, "A Low Power Transregional MOSFET Model for Complete Power-Delay Analysis of CMOS Gigascale Integration (GSI)", IEEE International ASIC Conference, September 1998.

[2.4] K. A. Bowman, B. L. Austin, J. C. Eble, X. Tang, and J. D. Meindl, "A Physical Alpha-Power Law MOSFET Model", IEEE Journal of Solid-State Circuits, October 1999.

[2.5] A. S. Roy, J. M. Vasi, and M. B. Patil, “A New Approach to Model Nonquasi-Static (NQS) Effects for MOSFETs—Part I: Large-Signal Analysis”, IEEE Transactions on Electron Devices, December 2003.

[2.6] S. Yang *et al.*, “A High-Performance 180-nm Generation Logic Technology”, IEEE International Electron Devices Meeting, 1998.

[2.7] S. Thompson *et al.*, “An Enhanced 130 nm Generation Logic Technology Featuring 60 nm Transistors Optimized for High Performance and Low Power at 0.7-1.4V”, IEEE International Electron Devices Meeting, 2001.

[2.8] Y. W. Kim *et al.*, “50-nm Gate Length Logic Technology with 9-Layer Cu Interconnects for 90-nm Node SoC Applications”, IEEE International Electron Devices Meeting, 2002.

[2.9] M. Khare *et al.*, “A High Performance 90-nm SOI Technology with 0.992 μm^2 6T-SRAM Cells”, IEEE International Electron Devices Meeting, 2002.

[2.10] F. L. Yang *et al.*, “A 65-nm Node Strained SOI Technology with Slim Spacer”, IEEE International Electron Devices Meeting, 2003.

[2.11] T. Sakurai and A. R. Newton, “A Simple MOSFET Model for Circuit Analysis”, IEEE Journal of Solid-State Circuits, April 1991.

[2.12] K. A. Bowman, L. Wang, X. Tang, and J. D. Meindl, “A Circuit-Level Perspective of the Optimum Gate Oxide Thickness”, IEEE Transactions on Electron Devices, August 2001.

[2.13] R. M. Corless *et al.*, “On the Lambert W Function”, Advances in Computational Mathematics, 1996.

[2.14] S. E. Thompson *et al.*, “A 90-nm Logic Technology Featuring Strained-Silicon”, IEEE Transactions on Electron Devices, November 2004.

[3.1] R. Gonzalez and M. Horowitz, “Energy Dissipation in General Purpose Microprocessors”, IEEE Journal of Solid-State Circuits, September 1996.

[3.2] V. Zyuban and P. Kogge, “Optimization of High-Performance Superscalar Architectures for Energy Efficiency”, International Symposium on Low-Power Electronics and Design, 2000.

[3.3] E. G. Friedman, “Clock Distribution Networks in Synchronous Digital Integrated Circuits”, Proceedings of the IEEE, May 2001, pp. 665-692.

[3.4] T. Xanthopoulos *et al.*, “The Design and Analysis of the Clock Distribution Network for a 1.2-GHz Alpha Microprocessor”, IEEE International Conference on Solid-State Circuits, 2001, pp. 402-403.

[3.5] P. J. Restle *et al.*, “A Clock Distribution Network for Microprocessors”, IEEE Journal of Solid-State Circuits, May 2001.

[3.6] V. Stojanovic and V. G. Oklobdzija, “Comparative Analysis of Master-Slave Latches and Flip-Flops for High-Performance and Low-Power Systems”, IEEE Journal of Solid-State Circuits, April 1999, pp. 536-548.

[3.7] M. Saint-Laurent and M. Swaminathan, “Impact of Interconnects on the Optimal Power-Performance Tradeoff for Clock Distribution in Microprocessors”, IEEE Topical Meeting on Electrical Performance of Electronic Packaging, 2000.

[3.8] V. Zyuban and P. Strenski, “Unified Methodology for Resolving Power-Performance Tradeoffs at the Microarchitectural and Circuit Levels”, International Symposium on Low-Power Electronics and Design, 2002.

[3.9] R. Heald *et al.*, “A 3rd-Generation SPARC V9 64-b Microprocessor”, IEEE Journal Solid-State Circuits, November 2000, pp. 1526-1538.

[3.10] K. Nose and T. Sakurai, “Analysis and Future Trend of Short-Circuit Power”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, September 2000, pp. 1023-1030.

[3.11] C. Nicoleta *et al.*, “A 450-MHz RISC Microprocessor with Enhanced Instruction Set and Copper Interconnect”, IEEE Journal of Solid-State Circuits, November 1999, pp.1478-1491.

[3.12] H. Sánchez *et al.*, “A Versatile 3.3/2.5/1.8-V CMOS I/O Driver Built in a 0.2-µm, 3.5-nm Tox, 1.8-V CMOS Technology”, IEEE Journal of Solid-State Circuits, November 1999, pp.1501-1511.

[3.13] A. Keshavarzi, K. Roy, and C. Hawkins, “Intrinsic Leakage in Low Power Deep Submicron CMOS ICs”, IEEE International Test Conference, 1997, pp. 146-155.

[3.14] K. A. Bowman *et al.*, “A Physical Alpha-Power Law MOSFET Model”, IEEE Journal of Solid-State Circuits, October 1999, pp. 1410-1414.

[3.15] D. Harris and M. Horowitz, “Skew-Tolerant Domino Circuits”, IEEE Journal of Solid-State Circuits, November 1997, pp. 1702-1711.

[3.16] T. Sakurai and A. R. Newton, “Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas”, IEEE Journal of Solid-State Circuits, April 1990.

[3.17] J. Stinson and S. Rusu, “A 1.5GHz Third Generation Itanium Processor”, IEEE International Solid-State Circuits Conference, 2003.

[4.1] P. J. Restle *et al.*, “A Clock Distribution Network for Microprocessors”, IEEE Journal of Solid-State Circuits, May 2001.

[4.2] D. Harris and S. Naffziger, “Statistical Clock Skew Modeling With Data Delay Variations”, IEEE Transactions on VLSI Systems, December 2001.

[4.3] E. G. Friedman, “Clock Distribution Networks in Synchronous Digital Integrated Circuits”, Proceedings of the IEEE, May 2001, pp. 665-692.

[4.4] N. Bindal *et al.*, “Scalable Sub-10ps Skew Global Clock Distribution for a 90nm Multi-GHz IA Microprocessor”, IEEE International Solid-State Circuits Conference, 2003.

[4.5] N. Nedovic and V. G. Oklobdzija, “Dynamic Flip-Flop with Improved Power”, Proceedings of the 26th European Solid-State Circuits Conference, Stockholm, Sweden, 2000.

[4.6] H. Partovi *et al.*, “Flow-Through Latch and Edge-Triggered Flip-Flop Hybrid Elements”, IEEE International Solid-State Circuits Conference, 1996, pp. 138-139.

[4.7] V. Stojanovic and V. G. Oklobdzija, “Comparative Analysis of Master-Slave Latches and Flip-Flops for High-Performance and Low-Power Systems”, IEEE Journal of Solid-State Circuits, April 1999.

[4.8] C. F. Webb *et al.*, “A 400-MHz S/390 Microprocessor”, IEEE Journal of Solid-State Circuits, November 1997.

[4.9] S. D. Naffziger *et al.*, “The Implementation of Itanium 2 Microprocessor”, IEEE Journal of Solid-State Circuits, November 2002.

[4.10] N. A. Kurd *et al.*, “Multi-GHz Clocking Scheme for Intel Pentium 4 Microprocessor”, IEEE International Solid-State Circuits Conference, 2001.

[4.11] S. Thompson *et al.*, “An Enhanced 130 nm Generation Logic Technology Featuring 60 nm Transistors Optimized for High Performance and Low Power at 0.7-1.4V”, IEEE International Electron Devices Meeting, 2001.

[4.12] A. Deutsch *et al.*, “On-Chip Wiring Design Challenges for Gigahertz Operation”, Proceedings of the IEEE, April 2001.

[4.13] T. Sakurai, “Closed-Form Expressions for Interconnection Delay, Coupling, and Crosstalk in VLSI’s”, IEEE Transaction on Electron Devices, January 1993.

[4.14] T. Sakurai, “Approximation of Wiring Delay in MOSFET LSI”, IEEE Journal of Solid-State Circuits, August 1983.

[4.15] K. Nabors *et al.*, “Lumped Interconnect Models Via Gaussian Quadrature”, Design Automation Conference, 1997.

[4.16] A. B. Kahng *et al.*, “On Switch Factor Based Analysis of Coupled RC Interconnects”, Proceedings of the IEEE/ACM Design Automation Conference, 2000.

[4.17] D. Sylvester and C. Hu, “Analytical Modeling and Characterization of Deep-Submicrometer Interconnect”, Proceedings of the IEEE, May 2001.

[4.18] S.-H. Wong, G.-Y. Lee, and D.-J. Ma, “Modeling of Interconnect Capacitance, Delay, and Crosstalk in VLSI”, IEEE Transactions on Semiconductor Manufacturing, February 2000.

[5.1] P. Zarkesh-Ha *et al.*, “Prediction of Net-Length Distribution for Global Interconnects in a Heterogeneous System-on-a-Chip”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, December 2000.

[5.2] P. J. Restle *et al.*, “A Clock Distribution Network for Microprocessors”, IEEE Journal of Solid-State Circuits, May 2001, pp. 792 -799.

[5.3] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, “Interconnect-Power Dissipation in a Microprocessor”, ACM International Workshop on System Level Interconnect Prediction, February 2004.

[5.4] G. J. van Rootselaar, and B. Vermeulen, “Silicon Debug: When Scan Chains Alone are not Enough”, IEEE International Test Conference, 1999, pp. 892-902.

[5.5] M. Saint-Laurent, and M. Swaminathan, “A Digitally Adjustable Resistor for Path Delay Characterization in High-Frequency Microprocessors”, IEEE Southwest Symposium on Mixed-Signal Design, 2001, pp. 80-82.

[5.6] N. A. Kurd, J. S. Barkatullah, R. O. Dizon, T. D. Fletcher, and P. O. Madland, “Multi-GHz Clocking Scheme for Intel® Pentium® 4 Microprocessor”, IEEE International Solid-State Circuits Conference, 2001, pp. 404-405.

[5.7] M. R. Garey, R. L. Graham, and D. S. Johnson, "The Complexity of Computing Steiner Minimal Trees", SIAM Journal of Applied Mathematics, 1977, pp. 835-859.

[5.8] D. W. Warme, P. Winter, and M. Zachariasen, "GeoSteiner 3.1", http://www.diku.dk/geosteiner.

[5.9] M. W. Bern, "Two Probabilistic Results on Rectilinear Steiner Trees", ACM Symposium on Theory of Computing, 1986, pp. 433-441.

[5.10] H.-Y. Hsieh, W. Liu, M. Clements, and P. Franzon, "Self-Calibrating Clock Distribution With Scheduled Skews", IEEE International Symposium on Circuits and Systems, 1998, pp. 470-473.

[5.11] H. Sutoh, and K. Yamakoshi, "A Clock Distribution Technique with an Automatic Skew Compensation Circuit", IEICE Transactions on Electronics, February 1998, pp. 277-283.

[5.12] R. Senthinathan, S. Fischer, H. Rangchi, and H. Yazdanmehr, "A 650-MHz, IA-32 Microprocessor with Enhanced Data Streaming for Graphics and Video", IEEE Journal of Solid-State Circuits, November 1999, pp. 1454-1465.

[5.13] S. Tam *et al.*, "Clock Generation and Distribution for the First IA-64 Microprocessor", IEEE Journal of Solid-State Circuits, November 2000, pp. 1545-1552.

[5.14] L. Hall, M. Clements, W. Liu, and G. Bilbro, "Clock Distribution Using Cooperative Ring Oscillators", Conference on Advanced Research in VLSI, 1997, pp. 62-75.

[5.15] H. Mizuno, and K. Ishibashi, "A Noise-Immune GHz-Clock Distribution Scheme Using Synchronous Distributed Oscillators", IEEE International Solid-State Circuits Conference, 1998, pp. 404-405.

[5.16] G. A. Pratt, and J. Nguyen, "Distributed Synchronous Clocking", IEEE Transactions on Parallel and Distributed Systems, March 1995, pp. 314-328.

[5.17] V. Gutnik, and A. P. Chandrakasan, "Active GHz Clock Network Using Distributed PLLs", IEEE Journal of Solid-State Circuits, November 2000, pp. 1553-1560.

[5.18] K. L. Shepard, V. Narayanan, and R. Rose, "Harmony: Static Noise Analysis of Deep Submicron Digital Integrated Circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, August 1999, pp. 1132-1150.

[5.19] J. A. Davis, and J. D. Meindl, "Compact Distributed RLC Interconnect Models—Part II: Coupled Line Transient Expressions and Peak Crosstalk in Multilevel Networks", IEEE Transactions on Electron Devices, November 2000, pp. 2078-2087.

[5.20] E. You *et al.*, “Parasitic Extraction for Multimillion-Transistor Integrated Circuits: Methodology and Design Experience”, IEEE Custom Integrated Circuit Conference, 2000, pp. 491-494.

[5.21] D. Sylvester and C. Hu, “Analytical Modeling and Characterization of Deep-Submicrometer Interconnect”, Proceedings of the IEEE, May 2001, pp. 634-664.

[5.22] E. S. Fetzer *et al.*, “A Fully Bypassed Six-Issue Integer Datapath and Register File on the Itanium-2 Microprocessor”, IEEE Journal of Solid-State Circuits, November 2002, pp. 1433-1440.

[5.23] S. Tam, R. D. Limaye, and U. N. Desai, “Clock Generation and Distribution for the 130-nm Itanium® 2 Processor With 6-MB On-Die L3 Cache”, IEEE Journal of Solid-State Circuits, April 2004, pp. 636-642.

[5.24] E. A. Lee, and D. G. Messerschmitt, Digital Communication, 2nd ed. Norwell, MA: Kluwer, 1994.

[5.25] J. G. Proakis, Digital Communications, 3rd ed. New York, NY: McGraw-Hill, 1995.

[5.26] D. K. Cheng, Fundamentals of Engineering Electromagnetics. Reading, MA: Addison-Wesley, 1993.

[5.27] S. Thompson *et al.*, “An Enhanced 130 nm Generation Logic Technology Featuring 60 nm Transistors Optimized for High Performance and Low Power at 0.7 – 1.4 V”, Proc. IEEE Int. Electron Device Meeting, 2001, pp. 257-260.

[5.28] S. O. Rice, “Mathematical Analysis of Random Noise”, Bell Systems Technical Journal, 1944, pp. 282-332.

[5.29] E. V. Bulinskaya, “On the Mean Number of Crossings of a Level by a Stationary Gaussian Process”, Theory of Probability and its Applications, 1961, pp. 435-438.

[6.1] J. H. Stathis, “Physical and Predictive Models of Ultrathin Oxide Reliability in CMOS Devices and Circuits”, IEEE Transactions on Device and Materials Reliability, March 2001.

[6.2] T. Ichikawa and M. Sasaki, “A New Analytical Model of SRAM Cell Stability in Low-Voltage Operation”, IEEE Transactions on Electron Devices, January 1996.

[6.3] L.-R. Zheng and H. Tenhunen, “Fast Modeling of Core Switching Noise on Distributed LRC Power Grid in ULSI Circuits”, IEEE Transactions on Advanced Packaging, August 2001.

[6.4] A. Kabbani and A. J. Al-Khalili, “Estimation of Ground Bounce Effects on CMOS Circuits”, IEEE Transactions on Components and Packaging Technology, June 1999.

[6.5] H.-R. Cha and O.-K. Kwon, “An Analytical Model of Simultaneous Switching Noise in CMOS Systems”, IEEE Transactions on Advanced Packaging, February 2000.

[6.6] A. Dharchoudhury, R. Panda, D. Blaauw, and R. Vaidyanathan, “Design and Analysis of Power Distribution Networks in PowerPC Microprocessors”, Design Automation Conference, 1998.

[6.7] M. Zhao, R. V. Panda, S. S. Sapatnekar, and D. Blaauw, “Hierarchical Analysis of Power Distribution Networks”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, February 2002.

[6.8] H. H. Chen and J. S. Neely, “Interconnect and Circuit Modeling Techniques for Full-Chip Power Supply Noise Analysis”, IEEE Transactions on Components, Packaging, and Manufacturing Technology—Part B, August 1998.

[6.9] T. Rahal-Arabi, G. Taylor, M. Ma, and C. Webb, “Design and Validation of the Pentium III and Pentium 4 Processors Power Delivery”, Symposium on VLSI Circuits, 2002.

[6.10] T. Rahal-Arabi, G. Taylor, M. Ma, and C. Webb, “Design and Validation of the Core and IOs Decoupling of the Pentium® III and Pentium® 4 Processors”, Electrical Performance of Electronic Packaging, 2002.

[6.11] Y.-M. Jiang and K.-T. Cheng, “Vector Generation for Power Supply Noise Estimation and Verification of Deep Submicron Designs”, IEEE Transactions on Very Large Scale Integration Systems, April 2001.

[6.12] S. R. Nassif and J. N. Kozhaya, “Fast Power Grid Simulation”, ACM/IEEE Design Automation Conference, 2000.

[6.13] R. Panda, D. Blaauw, R. Chaudhry, V. Zolotov, B. Young, and R. Ramaraju, “Model and Analysis for Combined Package and On-Chip Power Grid Simulation”, International Symposium on Low-Power Electronics and Design, 2000.

[6.14] Y.-M. Jiang and K.-T. Cheng, “Analysis of Performance Impact Caused by Power Supply Noise in Deep Submicron Devices”, Design Automation Conference, 1999.

[6.15] L. H. Chen, M. Marek-Sadowska, and F. Brewer, “Buffer Delay Change in the Presence of Power and Ground Noise”, IEEE Transactions on Very Large Scale Integration Systems, June 2003.

[6.16] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas", IEEE Journal of Solid-State Circuits, April 1990.

[6.17] T. Sakurai and A. R. Newton, "A Simple MOSFET Model for Circuit Analysis", IEEE Journal of Solid-State Circuits, April 1991.

[6.18] S. Thompson *et al.*, "An Enhanced 130 nm Generation Logic Technology Featuring 60 nm Transistors Optimized for High Performance and Low Power at 0.7-1.4V", IEEE International Electron Devices Meeting, 2001.

[6.19] N. H. E. West and K. Eshraghian, Principles of CMOS VLSI Design: A System Perspective, Second Edition. Reading, MA: Addison-Wesley, 1993, Chapter 8.

[6.20] R. Ho, K. W. Mai, and M. A. Horowitz, "The Future of Wires", Proceedings of the IEEE, April 2001.

[6.21] T. R. Scavo and J. B. Thoo, "On the Geometry of Halley's Method", American Mathematical Monthly, 1995.

[6.22] P. J. Restle *et al.*, "A Clock Distribution Network for Microprocessors", IEEE Journal of Solid-State Circuits, May 2001.

[6.23] P. J. Restle et al., "The Clock Distribution of the Power4 Microprocessor", IEEE International Solid-State Circuits Conference, 2002.

[6.24] J. A. Davis, A Hierarchy of Interconnect Limits and Opportunities or Gigascale Integration, Chapter 3, Ph.D. Dissertation, Georgia Institute of Technology, 1999.

[6.25] D. Sylvester and C. Yu, "Analytical Modeling and Characterization of Deep-Submicrometer Interconnect", Proceedings of the IEEE, May 2001.

[6.26] "Intel® Pentium® 4 Processor with 512-kB L2 Cache on 0.13 Micron Process at 2 GHz, 2.20 GHz, 2.26 GHz, 2.40 GHz, and 2.53 GHz", http://developer.intel.com, May 2002.

[6.27] R. Mahajan, K. Brown, and V. Atluri, "The Evolution of Microprocessor Packaging", Intel Technology Journal, Q3 2000.

[6.28] E. Chiprout, "Fast Flip-Chip Power Grid Analysis Via Locality and Grid Shells", International Conference on Computer-Aided Design, 2004.

[6.29] "Intel® Pentium® 4 Processor in the 478-pin Package / Intel® 850 Chipset Family Platform Design Guide", http://developer.intel.com, January 2003.

[6.30] S. Rusu *et al.*, “A 1.5-GHz 130-nm Itanium 2 Processor With 6-MB On-Die L3 Cache”, IEEE Journal of Solid-State Circuits, November 2003.

[7.1] G. Geannopoulos and X. Dai, “An Adaptive Digital Deskewing Circuit for Clock Distribution Networks”, IEEE International Solid-State Circuits Conference, 1998.

[7.2] S. Tam *et al.*, “Clock Generation and Distribution for the First IA-64 Microprocessor”, IEEE Journal of Solid-State Circuits, November 2000.

[7.3] N. A. Kurd, J. S. Barkatullah, R. O. Dizon, T. D. Fletcher, and P. O. Madland, “Multi-GHz Clocking Scheme for Intel® Pentium® 4 Microprocessor”, IEEE International Solid-State Circuits Conference, 2001.

[7.4] G. A. Pratt, and J. Nguyen, “Distributed Synchronous Clocking”, IEEE Transactions on Parallel and Distributed Systems, March 1995.

[7.5] V. Gutnik, and A. P. Chandrakasan, “Active GHz Clock Network Using Distributed PLLs”, IEEE Journal of Solid-State Circuits, November 2000.

[7.6] M. Saint-Laurent, and M. Swaminathan, “A Multi-PLL Clock Distribution Architecture for Gigascale Integration”, IEEE Computer Society Workshop on VLSI, 2001.

[7.7] T. J. Gabara and S. C. Knauer, “Digitally Adjustable Resistors in CMOS for High-Performance Applications”, IEEE Journal of Solid-State Circuits, August 1992.

[7.8] H. Noda *et al.*, “An On-Chip Clock-Adjusting Circuit with Sub-100-ps Resolution for a High-Speed DRAM Interface”, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, August 2000.

[7.9] M. Saint-Laurent, M. Swaminathan, and J. D. Meindl, “On the Micro-Architectural Impact of Clock Distribution Using Multiple PLLs”, International Conference on Computer Design, 2001.

[7.10] C. E. Dike, N. A. Kurd, P. Patra, and J. Barkatullah, “A Design for Digital, Dynamic Clock Deskew”, Symposium on VLSI Circuits, 2003.

[7.11] R. Senthinathan *et al.*, “A 650-MHz, IA-32 Microprocessor with Enhanced Data Streaming for Graphics and Video”, IEEE Journal of Solid-State Circuits, November 1999.

[7.12] N. A. Kurd, J. S. Barkatullah, R. O. Dizon, T. D. Fletcher, and P. O. Madland, “A Multigigahertz Clocking Scheme for the Pentium® 4 Microprocessor”, IEEE Journal of Solid-State Circuits, November 2001.