Statistical Clock Skew Modeling With Data Delay Variations

David Harris and Sam Naffziger

***Abstract*—Accurate clock skew budgets are important for microprocessor designers to avoid hold-time failures and to properly allocate resources when optimizing global and local paths. Many published clock skew budgets neglect voltage jitter and process variation, which are becoming dominant factors in otherwise balanced H-trees. However, worst-case process variation assumptions are severely pessimistic. This paper describes the major sources of clock skew in a microprocessor using a modified H-tree and applies the model to a second-generation Itanium-M processor family microprocessor currently under design. Monte Carlo simulation is used to develop statistical clock skew budgets for setup and hold time constraints in a four-level skew hierarchy. Voltage jitter through the phase locked loop (PLL) and clock buffers accounts for the majority of skew budgets. We show that taking into account the number of nearly critical paths between clocked elements at each level of the skew hierarchy and variations in the data delays of these paths reduces the difference between global and local skew budgets by more than a factor of two. Another insight is that data path delay variability limits the potential cycle-time benefits of active deskew circuits because the paths with the worst skew are unlikely to also be the paths with the longest data delays.**

***Index Terms*—Clock skew, clocks, jitter, noise, process variation, variability.**

I. Introduction

CLOCK skew is a key challenge for high-speed circuit designers because it can degrade performance and cause chip failures. As clock frequency goes up faster than simple process improvement would permit, better clock distribution networks are required to keep skew at a constant fraction of the cycle time. The problem is exacerbated by the growing die size, clock loads, and process variability. Therefore, designers have moved from clock spines and *ad hoc* clock routing to H-trees and grids [1]. Even when the systematic skew is completely designed away, environmental and processing variations lead to significant amounts of skew [2]. Assuming worst-case variations of all parameters leads to skew values that are large enough that design becomes impossible. Thus, the designer needs a statistical model that captures the multiple independent and correlated sources of skew.

This paper develops such a statistical model for clock skew and applies it to the McKinley IA-64 microprocessor. It then describes a generalized skew budget incorporating both clock and data path delay variations. The model reveals several design insights. One is that considering variations in logic delay is essential to avoid pessimistically overbudgeting global skew. Another is that jitter induced by power supply noise on the clock buffers is the dominant source of clock skew in H-trees. A third is that the number of paths between clocked elements has an important influence on the best clock skew budget for the design phase.

We begin in Section II by defining terms and reviewing the literature describing clock skew budgets in high performance microprocessors. In Section III, we present the modified H-tree clock distribution network for the microprocessor being studied with a four-level clock skew hierarchy. We enumerate and quantify in Section IV the major environmental and process variations that lead to skew in both the clock and data paths. Using a Monte Carlo simulation, we develop clock skew budgets in Section V appropriate for setup and hold constraints at various levels of the skew hierarchy. We also examine the sensitivity of these budgets to the key parameters. Finally, we summarize the major insights provided by the model in Section VI.

II. Background

We begin by reviewing definitions and timing constraints from the Sakallah–Mudge–Olukotun timing analysis formulation extended to account for different clock skews between different clock edges and different clock domains [3], [4]. In some situations, the designer is primarily interested in the absolute value of skew, but in other situations, only the difference between skew in different domains. Finally, we survey the sources of clock skew and previous work budgeting skew caused by on-chip variations.

Clock skew is the difference between the nominal and actual interarrival times of a pair of clock edges [5]. We may define a hierarchy of clock domains budgeting skews differently based on the number of shared elements in the clock distribution. For example, we could model clocks sharing a unit-level driver as seeing only “local skew” while other clocks experience “global skew.” Clock skew is smaller for the same edge of a pair of clocks than between different edges because of jitter. Fig. 1 illustrates the impact of clock skew on setup and hold time constraints. All five clocks are nominally identical but are shown with skew that could cause timing violations. Paths within a clock domain budget local skew but paths crossing clock domains budget global skew. Hold time constraints are subject to fewer sources of skew than setup time constraints. We define system parameters, then list the timing constraints in terms of these parameters in the following:

- T_c clock cycle time or period;

Manuscript received October 1, 2000; revised April 3, 2001.

D. Harris is with the Harvey Mudd College, Claremont, CA 91711 USA (e-mail: David_Harris@hmc.edu).

S. Naffziger is with the Hewlett-Packard Corporation, Fort Collins, CO 50525 USA (e-mail: sdn@fc.hp.com).

Publisher Item Identifier S 1063-8210(01)07424-8.

1063–8210/01$10.00 ©2001 IEEE

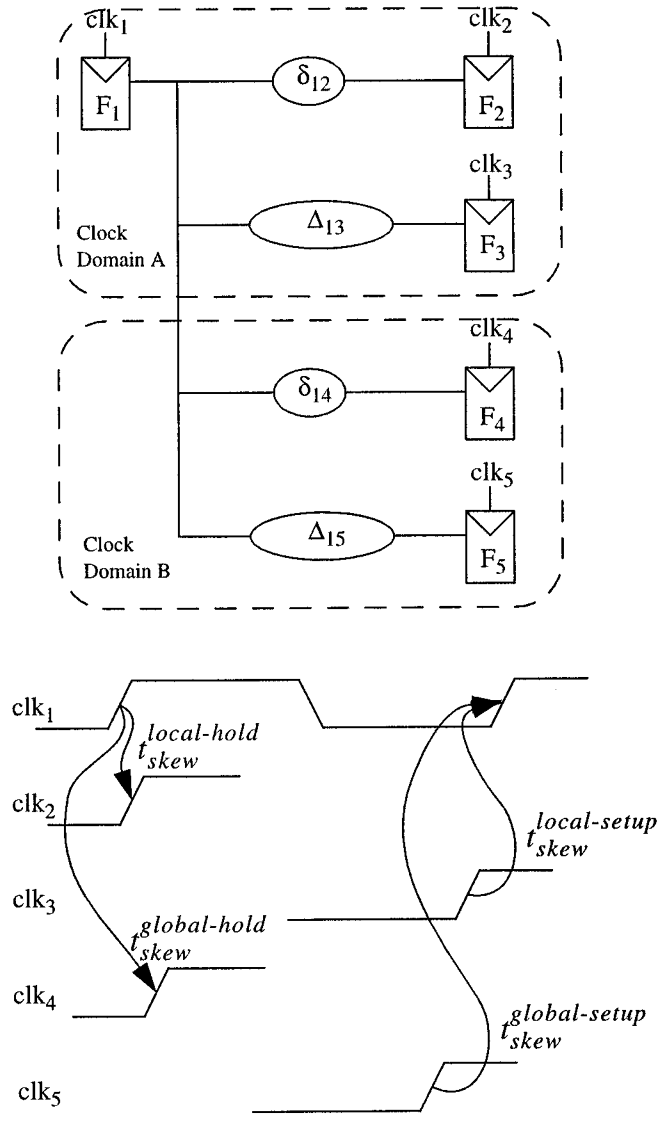


Fig. 1. Supply voltage pinch during peak switching activity.

- Δ_{DC} register setup time;
- Δ_{CD} register hold time;
- Δ_{CQ} maximum register clock-to-Q propagation delay;
- δ_{CQ} minimum register clock-to-Q propagation delay;
- Δ_{ij} maximum logic propagation delay from clocked element i to j;
- δ_{ij} minimum logic propagation delay from clocked element i to j.

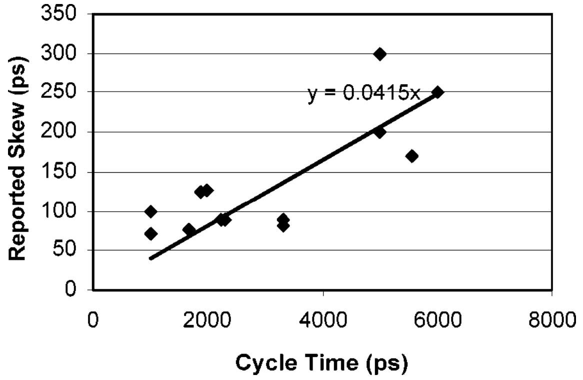


Fig. 2. Die temperature distribution during normal operation.

Setup time violations can be fixed by increasing the clock period so they are only a matter of performance. If all paths used a single global clock skew number the designer could simply seek to minimize clock period without considering skew. The actual silicon would run slower than predicted on account of the skew but skew would not enter design decisions. On the other hand, if some paths budget global skew while others budget a smaller amount of local skew, it is important to accurately define the difference between the skews so that the designer does not overdesign either the local or global paths. Such overdesign costs area, power, and time to market without improving performance. For example, in Fig. 1, we would ideally optimize the local critical path Δ_{13} only until it is slightly longer than the global critical path Δ_{15}

$$\Delta_{13} = \Delta_{15} + \left(t_{\text{skew}}^{\text{global-setup}} - t_{\text{skew}}^{\text{local-setup}}\right). \tag{1}$$

Hold time violations result in nonfunctional silicon so they are much more serious. The cost of overbudgeting skew in hold time checks is extra delay added to short paths. Given the tradeoff between nonfunctional silicon and designing in extra delay, most designers conservatively budget skew for hold time constraints.

In summary, the designer is primarily interested in the difference between skews at different levels of the clock domain hierarchy for setup constraints, but the absolute amount of skew for hold constraints (see Table III). The setup skew budget is subtracted from the time available for logic to propagate from one register to the next. The hold skew budget determines the contamination delay requirement between registers. Analogous constraints exist for systems using latches or domino circuits. Setup time skew budgets may be further refined to describe skew in half-cycle paths, full-cycle paths, and multi-cycle paths, but this paper restricts itself to a single setup time skew budget for each clock domain.

Clock skew sources fall into four major categories. Systematic offsets are the skews determined by a SPICE simulation using nominal component values. Most “zero skew” clock papers only address the systematic offsets. Random offsets are caused by intradie process variations such as channel length variations. Drift, from effects like temperature change, results in low-frequency skew changes. And jitter, especially from voltage noise, leads to high-frequency timing variation. Closed-loop clock generators can adjust for systematic and random offsets and drift but not for jitter. Hence, jitter is the most challenging source of skew to control.

Much work has been done in the area of modeling and characterizing clock skew. Fig. 2 illustrates published clock skews for a number of high-performance microprocessors [6]–[14] as a function of clock period. Notice that the skew budgets have typically been reported as about 4–5% of the clock period.

Most of the published clock skew data are incomplete. Many reflect only simulated systematic offsets [11]–[14] and do not include PLL jitter or any uncertainties in the clock distribution network.

Typical modern processor clock distribution trees have a delay on the order of 1 ns [15], [16]. A simple worst-case analysis budgeting 25% delay variation from environmental factors and 15% delay variation from process mismatch would suggest 400 ps of skew from nonsystematic components. The existence of GHz processors proves that this worst-case analysis is pessimistic. IBM and Compaq have reported measured clock skew in the range of 50–70 ps for clock grids [17]–[19]. These measurements are limited by the challenge of finding the points with greatest skew and measuring jitter. H trees tend to be more susceptible than clock grids to skew from parameter variations because grids short together the drivers and, thus, spatially lowpass filter the variations.

Many researchers have tried to model clock skew caused by on-chip parameter variations. Zarkesh–Ha, Mule, and Meindl developed an analytical model of worst-case skew for an H tree with no repeaters along the tree [20]. The network lacks internal buffering because it drives a very light clock load and worst-case variations are assumed everywhere. Sauter *et al.* [2] simulate clock skew for various distribution networks given actual parameter variations measured on a specific test chip. Nassif considered the impact of random channel length and wire width variation on H tree clock skew [21], [22]. Sylvester *et al.* [23] apply stochastic modeling to interconnect variations and find a 3-σ skew of 20 ps from Monte Carlo analysis, an improvement over a skew-corner prediction of 55 ps. Zanella *et al.* [24] also describe Monte Carlo analysis, but apply it to a Viterbi Decoder with fewer than 1000 flip-flops and do not describe how their variability analysis is derived from physical parameters. Bowman *et al.* [25] have developed a model of the impact of die-to-die and within-die parameter variations applied to the maximum clock frequency of Pentium microprocessors, but focused on critical path delay rather than clock skew.

To our knowledge, this is the first work to apply statistical models of the major components of clock skew to a high-performance microprocessor and to include the effect of variability in data path delay. As a result, we develop a less pessimistic choice of skew budgets for design.

III. Clock Distribution Network

We consider skew in a modified H-tree clock distribution network. An ideal H-tree is perfectly symmetric and has zero skew from systematic mismatches, though it does experience skew from random mismatches, low-frequency environmental drift, and high-frequency voltage jitter. Unfortunately, an ideal H-tree is difficult to place within floorplanning constraints and is unrealistic because loads are not evenly distributed across the die.

Instead, we examine a modified H-tree where clock buffers are positioned where they reasonably fit in the floorplan. The number of buffers at each fork in the tree depends on the clock load served by the fork. Our analysis is general, but for concreteness we apply it to the McKinley microprocessor in a 0.18-μm process. The clock tree serves only the 16 × 14 mm chip core. The remainder of the die, comprising L2 cache arrays and bus drivers with greater tolerance to clock skew, is served by an *ad hoc* clock distribution network offering lower power consumption at the expense of greater skew.

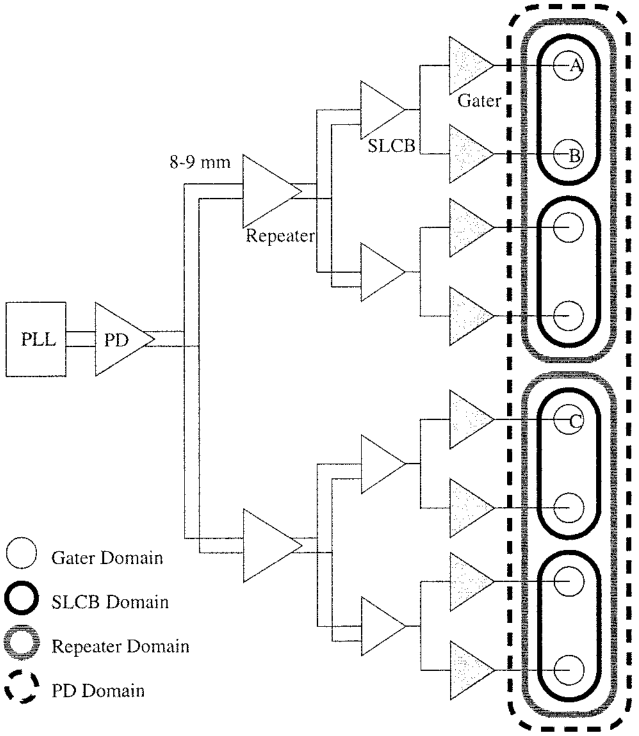


Fig. 3. Sensitivity of hold skew to short path variability.

Fig. 3 shows a simplified representation of the clock network. The root of the modified H tree is a phase-locked loop and primary driver (PD) in the clock unit. It generates a differential clock which is routed on upper metal layers across 8–9 mm of interconnect to five differential repeaters. The differential signalling reduces duty cycle variation and the wide clock lines are interdigitated with supply lines to minimize inductive effects. Each repeater drives three to seven second-level clock buffers (SLCBs), which send single-ended signals to approximately ten banks of gaters each. The gaters provide clock enabling and clock shaping and directly drive banks of latches or dynamic logic along short final clock lines. The modified H-tree is delay matched rather than length matched, meaning that wire widths and lengths are tapered to provide equal delay between clock driver stages under nominal conditions.

The clock distribution network leads to a natural clock domain hierarchy (see Fig. 6). We can define skews for circuits served by the same gater, the same SLCB, the same repeater, or those sharing only the common primary driver [26]. We will refer to these clock domains as Gater, SLCB, repeater, and PD, respectively. For example, all the latches in region A are in a Gater domain. Paths from A to B budget skew in the SLCB domain. Paths from A to C must budget PD domain skew. These skew domains are analogous to local and global clock domains in Section II but provide a finer granularity to avoid unnecessarily budgeting excess skew. We separately define skew from rising edge to rising edge for setup time constraints and at a common edge for hold time constraints. In this study, we do not consider skew between rising and falling edges impacted by duty-cycle variations.

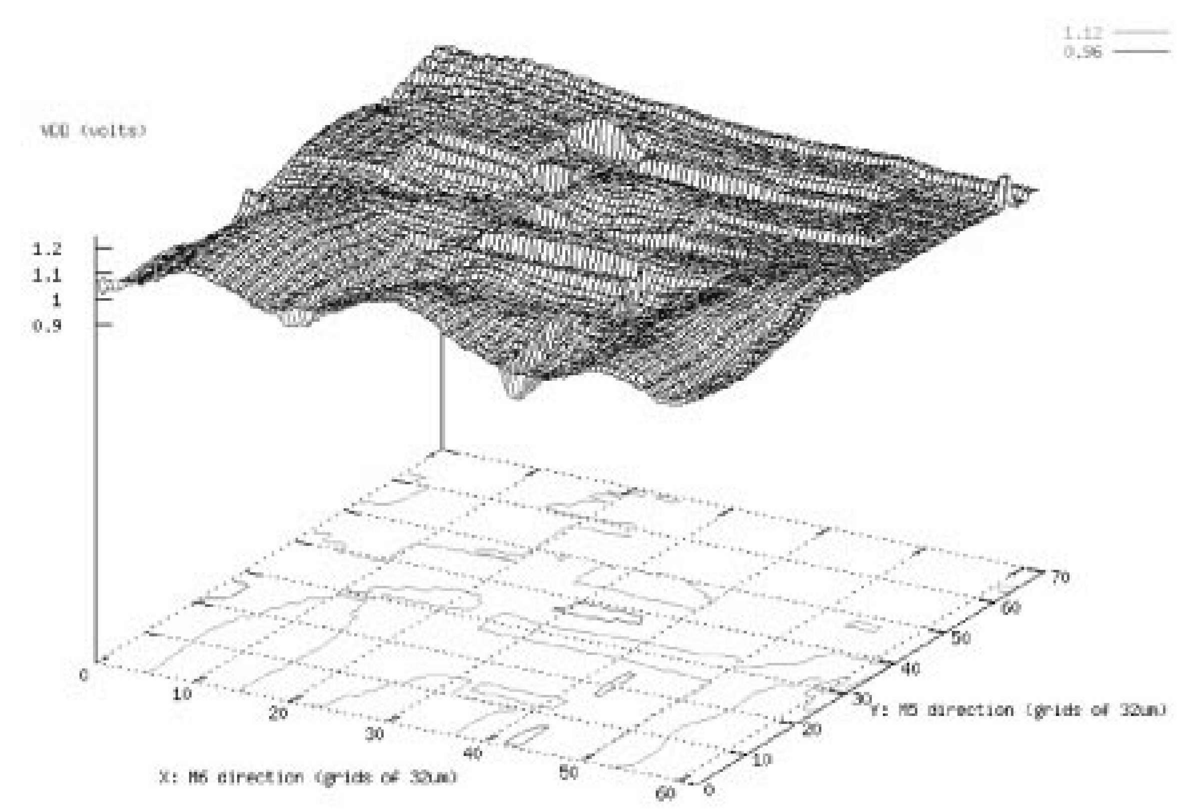


Fig. 4. Sensitivity of setup skew to critical path variability.

This study is limited to modified H-trees. Grids can be used to reduce sensitivity to process variations at the expense of additional power [1], [18].

IV. Statistical Clock Skew Model

A worst-case clock skew model assuming maximum simultaneous variation of each component of clock skew has little correlation with observed skew and is so pessimistic that design becomes nearly impossible. An improved model takes a statistical approach to sum the independent and correlated random variables that impact skew. We will see that even this model is still overly pessimistic and makes design difficult. For greatest realism we must simultaneously consider the variations in delay through logic paths between clocked elements. Ultimately, we are interested in minimizing the expected clock period which is limited by a combination of variations in the data and clock paths. In this section, we examine the primary sources of variation contributing to skew in both the clock and data delays. We then describe a Monte Carlo simulation used to account for these variations simultaneously.

The data presented for the model is based on simulations of a second-generation Itanium processor family microprocessor clock network in the Intel 0.18 μm process [27] at low voltage (1.2 V) and high temperature (110 °C) assuming a 1.3-ns clock period. The process models are old and conservative; actual silicon is substantially faster.

A. Clock Skew Sources

Although the clock distribution network is delay matched to have zero nominal clock skew up to the clock gaters, variations in processing and environment lead to variations in delays between clocks. Each stage of clock buffer is subject to variation in effective channel length L_{e}, threshold voltage V_{t}, operating temperature T, and supply voltage V_{DD}. Clock buffers are tied to the chip power supply but are heavily bypassed. The PLL is isolated from the chip supply and further bypassed to reduce jitter. Interconnect delay is subject to variation caused by wire width and thickness variations, dielectric thickness variation, and mismatches between relative wire and gate capacitance used for delay matching.

The variation in delay appears as a fraction of the total delay of each stage, so it is important to minimize the clock buffer delays. The PD and repeaters each have a delay of 150 ps. The SLCB has a delay of 280 ps. Simple gaters have a delay of 180 ps. These delays exclude wire RC flight times. The variations include the following.

- The power network was designed to see no more than ±100 mV supply variation. However, this full variation may be seen from cycle to cycle or between any two points on the die at a given instant. Fig. 4 shows the processor voltage distribution from a full-chip voltage simulation during peak switching activity. The deepest pinches correspond to the integer execution unit and the memory input/output (I/O) pads. Time-dependent power grid collapse is discussed further in Section V.
- Full-chip power simulations in Fig. 5 show a variation of 20 °C across the core during normal operation. The thermal map shows smooth variations in temperature with a maximum gradient of 10 °C between gaters served by

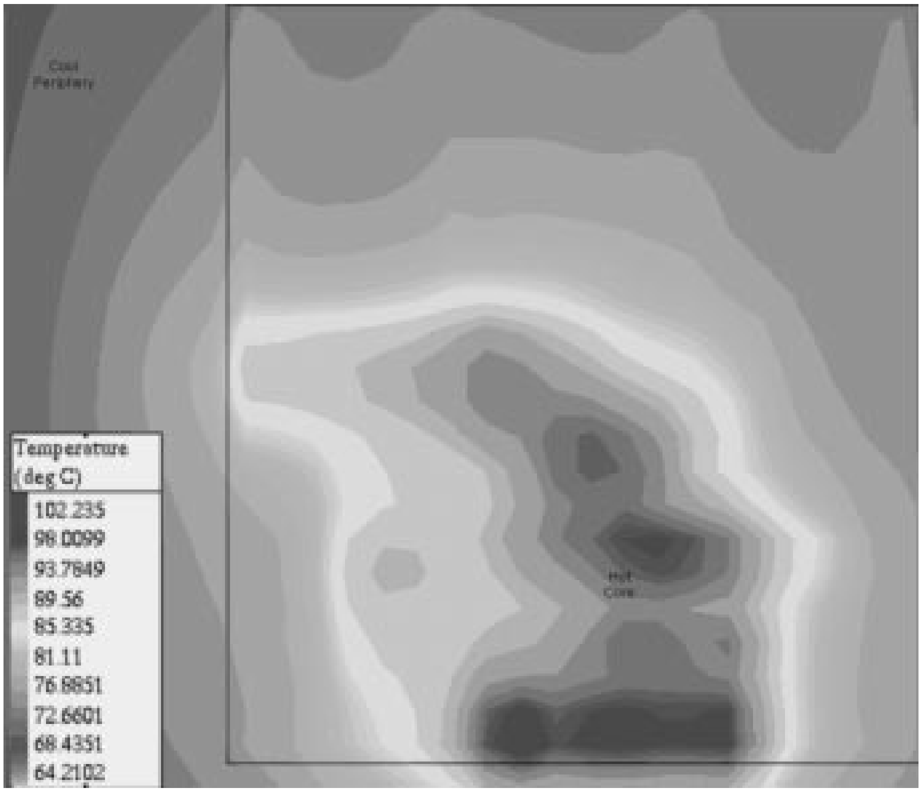


Fig. 5. Sensitivity to number of nearly critical/short paths.

a common SLCB. The cache arrays along the periphery run cooler but do not contain clock buffers. This map is consistent with thermal images of other high-performance processors such as the Alpha [6].

- Intradie L_{e} variations come from two sources: systematic components slowly varying across the die and random components that apparently are spatially uncorrelated. The systematic components have a half range of 12.5 nm for transistors separated by 4 mm or more. Transistors in a local area see smaller variations. The random components have a standard deviation of 3.3 nm.
- V_{t} variations display an inverse area dependence [28]. The standard deviation of threshold voltage is 16.8 mV for small NMOS transistors, 14.6 mV for small PMOS transistors, 7.9 mV for wide NMOS transistors, and 6.5 mV for wide PMOS transistors. Wide transistors are defined as those with a width exceeding 12.5 μ. No data was available concerning about spatial correlation of threshold voltages, so we assumed the thresholds to be spatially uncorrelated.
- Oxide thickness variations also impact transistor performance. However, it is difficult to distinguish their effects from threshold or channel length variation. Therefore, delay variations caused by oxide thickness are lumped into the variations from the other two process parameters.

Sensitivity to variations was determined from Spice simulation of clock buffers using BSIM3 models. Voltage sensitivity is 13% delay change per 100 mV voltage change. Temperature sensitivity is 1.5%/20°. Channel length sensitivity is 8%/10 nm from systematic variations. Systematic channel length variation leads to a uniform delay distribution with a half range of 10% of the gate delay. The cumulative effect of random channel length variation is negligible on account of the large number of transistors varying independently. Monte Carlo Spice simulation shows a standard deviation of 2% in the delay of clock buffers caused by threshold variations.

There are several other sources of skew beyond the clock buffers. Test chip measurements show that the PLL may experience 15 ps of cycle-to-cycle jitter. This represents improvements in power supply filtering and process technology compared to some recently published processor PLLs [29], [30]. Mismatches in the shielded delay-matched wires are budgeted at up to 2 ps between the PD and repeaters, 3 ps between the repeaters and SLCBs, and 8 ps between the SLCBs and gaters. Wire process variation beyond these mismatches was considered small enough that it was not modeled. In addition, the local clock wire after the gater may contribute up to 20 ps of wire RC and nonuniform gater loads may contribute up to 15 ps of delay variation. These budgets are consistent with the measured data from [17] that indicates buffer delay variations are the dominant source of clock skew.

Table I summarizes which components of clock skew impact specific skew budgets. The components are categorized as systematic errors, random process variation, low-frequency drift, or high-frequency jitter. The impact of each component is listed for hold and setup checks. It is further categorized based on the clock domains it effects, i.e., paths that share the same gater, same SLCB, same repeater, or nothing but the primary driver.

Cycle-to-cycle skew budgets must include jitter from the entire clock distribution network. All skew budgets include variations from mismatched buffer and interconnect delay. For ex-

TABLE I
TIMING CONSTRAINTS FOR FIG. 8

		Same Clock Edge (hold)				Cycle-to-Cycle (setup)			
Component	**Type**	**Gater**	**SLCB**	**Repeater**	**PD**	**Gater**	**SLCB**	**Repeater**	**PD**
Primary Clock Driver Sources									
PLL Jitter	Jitter					x	x	x	x
PD V	Jitter					x	x	x	x
PD T	Drift								
PD Le	Random								
PD Vt	Random								
Repeater Sources									
PD to Repeater Wire Mismatch	Syst.				x				x
Repeater V	Jitter				x	x	x	x	x
Repeater T	Drift				x				x
Repeater Le	Random				x				x
Repeater Vt	Random				x				x
Repeater Load	Syst.				x				x
SLCB Sources									
Rep. to SLCB Wire Mismatch	Syst.			x	x			x	x
SLCB V	Jitter			x	x	x	x	x	x
SLCB T	Drift			x	x			x	x
SLCB Le	Random			x	x			x	x
SLCB Vt	Random			x	x			x	x
SLCB Load	Syst.			x	x			x	x
Gater Sources									
SLCB to Gater Wire Mismatch	Syst.		x	x	x		x	x	x
Gater V	Jitter		x	x	x	x	x	x	x
Gater T	Drift		x	x	x		x	x	x
Gater Le (global)[a]	Random			x	x			x	x
Gater Le (local)	Random		x				x		
Gater Vt	Random		x	x	x		x	x	x
Gater Load	Syst.		x	x	x		x	x	x
Final Route Sources									
Local wire RC	Syst.	x	x	x	x	x	x	x	x

[a] L_e variations are spatially correlated. Therefore, we budget less skew from gater channel length variations between two gaters that share the same SLCB and must be nearby than from gaters using different SLCBs.

ample, paths in the Gater domain only see mismatches in the local wire RC delay after the gater. Paths in the PD domain see mismatches in repeater, SLCB, gater, and local wire delays. Primary driver random process variations and drift impact all paths equally and, therefore, do not contribute to skew budgets.

Table II summarizes the magnitude of each skew source in picoseconds. Most variations are specified as the half range X of a uniform distribution ($\pm X$). Threshold voltage variations are specified as a standard deviation σ of a normally distributed random variable. The channel length variation between two nearby gates is approximately half the variation seen across the die.

A conservative global setup skew budget assuming worst-case/3-σ variations of each parameter sums to 508 ps, about 40% of the cycle time. Clearly, we cannot design to such a conservative budget.

B. Data Skew Sources

Data paths are usually designed assuming worst-case environmental conditions but typical processing. Designers optimize paths until they meet a frequency target under these assumptions. As a result, a chip nominally has many paths forming a "wall" just above the target frequency. However, the data delays are subject to variations just like clock delays. This

TABLE II
IMPACT OF SKEW SOURCES ON SKEW BUDGETS

Component	Half-Range	Sigma
Primary Clock Driver Sources (150 ps PD delay)		
PLL Jitter	7.5	
PD V	19.5	
Repeater Sources (150 ps repeater delay)		
PD to Repeater Wire Mismatch	1	
Repeater V	19.5	
Repeater T	1.1	
Repeater Le	15	
Repeater Vt		3
Repeater Load	5	
SLCB Sources (280 ps SLCB delay)		
Rep. to SLCB Wire Mismatch	1.5	
SLCB V	36.4	
SLCB T	2.1	
SLCB Le	28	
SLCB Vt		5.6
SLCB Load	10	
Gater Sources (180 ps gater delay)		
SLCB to Gater Wire Mismatch	4	
Gater V	23.4	
Gater T	1.4	
Gater Le (global)	18	
Gater Le (local)	9	
Gater Vt		3.6
Gater Load	7.5	
Final Route Sources		
Local wire RC	10	

TABLE III
MAGNITUDE OF SKEW SOURCES

	Gater	SLCB	Repeater	PD
t_{skew}^{setup}	232	267	302	312
t_{skew}^{hold}	20	106	229	280

results in data skew and a distribution of path delays around the frequency target.

Not all data paths are designed to be exactly at the frequency target; some are close but have positive margin. Intradie process variation results in path delay variations. Not all paths encounter worst-case environment. Tool and model inaccuracies result in further data skew.

We modeled data skew as the sum of three components: a uniformly distributed delay reflecting positive margin from design and better than worst-case environment, a uniform delay from L_e variations impacting all the gates in the path and a normally distributed delay from V_t variations.

Based on preliminary timing analysis results, we considered nearly critical max-delay paths to be uniformly distributed with 0 to 50 ps of slack from the 1300 ps target cycle time. Because paths are usually more distributed than are single clock buffers, we budget only a ±5% path delay variation on account of channel length variations. Simulations show the path delay variation from threshold voltage variations has a standard deviation of 0.67% of the cycle time. This is a smaller percentage variation than that of a single clock buffer because there are many more stages of gates varying independently.

Similarly, we assume min-delay paths have a uniformly distributed slack of 0–30 ps plus random variations with a 30-ps standard deviation.

C. Monte Carlo Simulation

Simple root sum square (RSS) sums of the skew sources are not adequate to predict clock skew because many of the sources are not Gaussian. Moreover, the clock skew sources are only significant for paths that are nearly critical and this set of paths depends on the distribution of data delays. Therefore, a Monte Carlo simulation is used to determine skew budgets.

The skew sources were provided to a Monte Carlo simulation. The simulation models the clock and data paths of N chips. It randomly selects values for the systematic and random variations in the delay through each clock buffer and logic path. It adds worst-case variations from drift and jitter because these components of skew vary in time; skew budgets must be adequate for the system to operate correctly at all times. The setup skew calculated for each level of the clock hierarchy is the difference between the cycle time predicted based on paths in that level of the hierarchy and the nominal 1.3-ns cycle time achieved if there were no skew or data delay variations. The hold skew calculated for each level of the clock hierarchy is the amount of padding necessary to add to each path in that level of the hierarchy to ensure all paths satisfy hold time.

Using worst-case temperature drift is pessimistic because the die temperature is spatially dependent. However, the temperature variations are small enough that this pessimism is insignificant. Using worst-case voltage jitter is a more serious limitation of the model; this is addressed in Section V-C

The Monte Carlo simulation slightly simplifies the actual clock distribution network, assuming a single PD drives five repeaters, each of which drive six SLCBs, each of which drive ten gaters. For setup budgets it assumes there are 500 nearly critical paths sharing only the primary driver, 100 sharing each repeater, 1000 sharing each SLCB, and ten sharing each gater. These numbers are estimated from preliminary static timing analysis reports. For hold budgets it assumes the same number of paths are short and require min-delay padding. These path counts were estimated from the chip timing database which enumerates all of the paths of concern late in the design phase. The database showed that the largest number of nearly critical paths are within individual functional units and thus share a common SLCB.

For setup constraints, we are interested in the median skew budgets over the N chips because we bin parts and set frequency targets for typical processing and typical skew. For hold constraints, we select the 95th percentile skew budget so that the

TABLE IV
CLOCK SKEW BUDGETS WITHOUT DATA DELAY VARIATIONS (PS)

	Gater	SLCB	Repeater	PD
t_{skew}^{setup}	19	54	89	99
t_{skew}^{hold}	20	59	109	122

TABLE V
CLOCK SKEW BUDGETS WITHOUT DATA DELAY VARIATIONS OR JITTER (PS)

	Gater	SLCB	Repeater	PD
t_{skew}^{setup}	285	309	313	322
t_{skew}^{hold}	44	117	226	275

TABLE VI
SKEW BUDGETS WITH DATA DELAY VARIATIONS (PS)

	Gater	SLCB	Repeater	PD
t_{skew}^{setup}	72	97	100	109
t_{skew}^{hold}	44	70	106	117

TABLE VII
SKEW BUDGETS WITH DATA DELAY VARIATIONS BUT WITHOUT JITTER (PS)

	Gater	SLCB	Repeater	PD
t_{jitter}^{setup}	64	76	117	120
t_{skew}^{setup}	151	188	232	244
t_{jitter}^{hold}	0	31	93	96
t_{skew}^{hold}	44	101	199	213

yield loss to min-delay failures is low. We found $N = 400$ was a sufficiently large number of simulations to give errors of less than 1% while consuming only a few minutes of CPU time.

V. RESULTS

This section reports the results of the Monte Carlo skew simulations. First results are shown without considering data delay variations. Jitter dominates the skew budgets, so numbers are also reported with and without jitter to quantify how improved jitter control would benefit skew. Then results are shown taking into account variations in the data delay. This reduces the difference in skew budgets between levels of the clock skew hierarchy. Jitter from power supply variation is still a dominant component of skew, so a full-chip power grid simulation is used to better understand the temporal and spatial distribution of power supply pinch and thus avoid pessimistic jitter budgets. The modeling involves a number of assumptions so a sensitivity analysis is performed to determine which assumptions are most important.

A. Skew Budget Without Data Delay Variations

Clock skew was first determined considering only variations in the clock paths, not the data path. This is the conventional method of budgeting clock skew.

Table IV lists the clock skews determined by the Monte Carlo simulation. As expected, the skew increases at higher levels of the hierarchy. The skew in the setup paths is dominated by 213 ps of cycle-to-cycle jitter that impacts even the most local paths. Clearly, this jitter caused by voltage noise in the clock buffers and PLL is the dominant source of skew.

Table V lists the clock skews neglecting jitter. It shows the potential benefits of reducing voltage noise. The setup skew budgets are uniformly improved by the 213 ps of cycle-to-cycle jitter. Local (gater) hold skew budgets had no jitter, so do not improve. The hold skew budgets are improved for the more global paths that were subject to jitter between different clock buffers. In all cases, the hold skew is larger than the setup skew because a the budget is selected at the 95th percentile of chips rather than the median.

B. Skew Budget With Data Delay Variations

Skew budgets were then determined considering variations in both the clock and data paths. It is unlikely that the longest path on the chip exists between the two clocks with the worst skew. Given random data path delay variations, it is also unlikely that the worst-case path is global because there are far more local paths. We determine a generalized skew budget that describes the impact of both clock and data delay variations.

Table VI lists the clock skews determined by the Monte Carlo simulation accounting for variations in the data delay as well as the clock delay. The variability in data path delay is significant for setup time calculations and on the median chip adds 75 ps to the longest data path delay. This leads to the greatest increase in skew budgets for paths sharing a common gater which saw little clock delay variation. Similarly, it raises the setup skew budget for paths in the SLCB domain to almost match the repeater domain because there are far more paths sharing SLCBs than sharing only repeaters. Considering data delay variations, the difference between global (PD) and local (gater) setup skew budgets $t_{\mathrm{skew}}^{\mathrm{PD\text{-}setup}} - t_{\mathrm{skew}}^{\mathrm{gater\text{-}setup}}$ is expected to be only 37 ps rather than 80 ps indicated in Table IV. This factor of two change is important because the designer is primarily interested in the difference between global and local skew budgets. The change is attributed to the fact that there are more paths involving gaters. A similar changeis seen with hold time paths, but is smaller in magnitude because the short paths see less data delay variation.

Table VII lists the clock skews neglecting jitter. Even with the addition of random skew from data delay variations, a comparison between Tables VI and VII shows that jitter accounts for

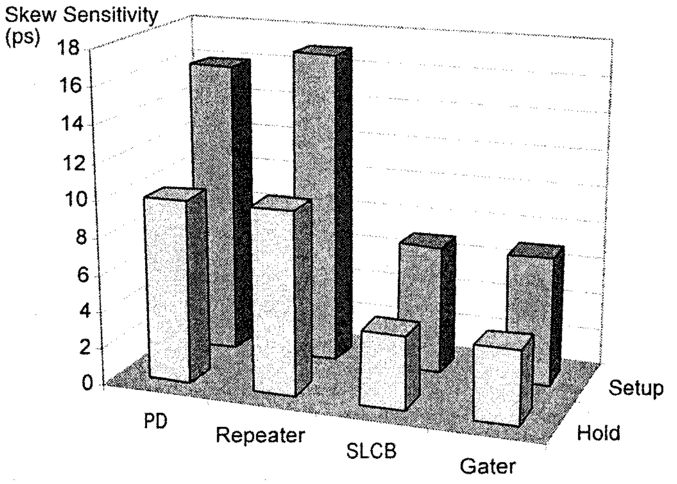


Fig. 6. Clock distribution network and clock domains.

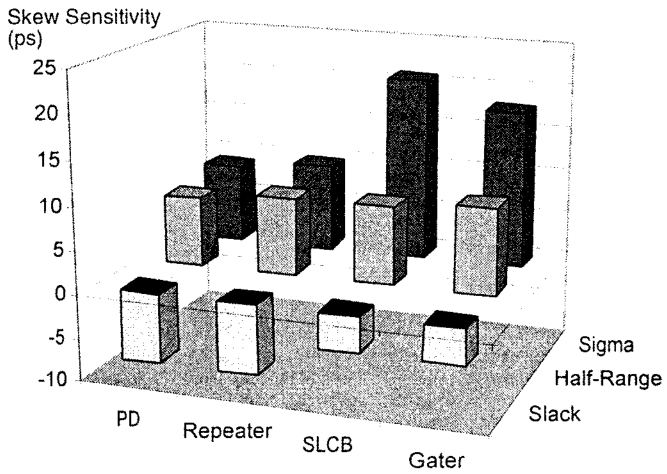


Fig. 7. Clock skew versus cycle time for high-performance microprocessors.

66% of the setup skew budget for circuits sharing only the PD and 75% for circuits sharing a gater.

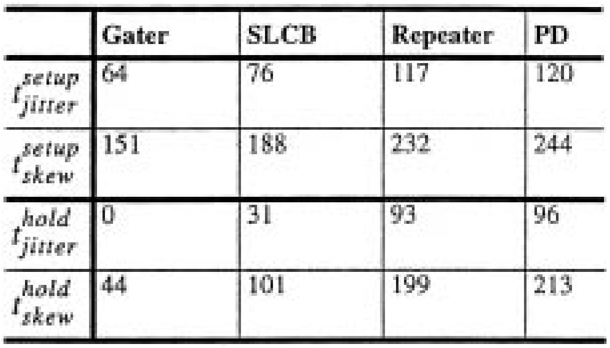
This data leads to an interesting conclusion about the importance of clock skew and active clock deskew networks [26]. Imagine that all critical paths are designed to meet some target cycle time T_c without considering skew or data delay variations. Assume jitter is zero; it is not affected by active deskew. A straightforward calculation of clock skew given in Table V would suggest the PD setup skew is 99 ps, so the clock period would be limited to $T_c + 99$ ps. Therefore, one might suppose an active deskew system could improve the period by 99 ps. According to Table VII, the expected clock period would be limited to $T_c + 109$ ps on account of PD setup skew when data delay variations are also considered. Now suppose the systematic and random clock skew and drift were driven to zero through some ideal active deskew network. The clock period would still be limited to $T_c + 75$ ps on account of the data delay variations in our model. In other words, the maximum cycle time improvement of the active deskew system is 34 ps, not 99 ps predicted without considering data delay variation. This demonstrates that the interaction between data skew and clock skew must be considered when evaluating the benefits of clock skew reduction on cycle time. Jitter reductions would be more beneficial, but active feedback deskew methods do not have the bandwidth to compensate for cycle-to-cycle jitter (see Fig. 7).

C. Jitter Reduction

The cycle-to-cycle jitter in the skew budgets is 213 ps because we assume that power supply noise of up to the ±100 mV design target may occur between any two clock buffers. However, power supply noise exhibits both spatial and temporal locality so such worst-case variations are unlikely to impact all clocks. Moreover, the chip tends to be most quiet just before the clock edge when the SLCBs and gaters are firing. A full-chip power grid simulation provides data to make better jitter estimates.

The full-chip power grid simulation includes models of static and dynamic logic with appropriate power densities, the chip and package power distribution networks, and the chip and package bypass capacitance. The simulation includes a 20 A step load applied to the core of the chip. Power and ground waveforms are extracted at the locations of the clock buffers and provided to a clock network simulation that determines the cycle-to-cycle and buffer-to-buffer jitter. The power grid simulation shows both spatial and temporal correlations in the supply noise. In particular, the cycle-to-cycle supply voltage variation at a particular clock buffer is generally less than the cycle-to-cycle variation between two buffers.

TABLE VIII
SKEW BUDGETS WITH DATA DELAY VARIATIONS AND SIMULATED JITTER (PS)



	Gater	SLCB	Repeater	PD
t^{setup}_{jitter}	64	76	117	120
t^{setup}_{skew}	151	188	232	244
t^{hold}_{jitter}	0	31	93	96
t^{hold}_{skew}	44	101	199	213

Table VIII lists the simulated jitter figures (including a fixed 15 ps PLL setup jitter) and the resulting skew budgets including jitter and other sources of clock and data delay variation.

We see the setup skew numbers in Table VIII represent an 80–130 ps improvement over those of Table VI. Gater skew domains reduce jitter from the 213 ps budget to only 64 ps because the power supply noise impacting the clock buffers is correlated from cycle to cycle. PD skew domains see less jitter improvement because the clock buffers scattered across the chip see less correlation. Hold skew also benefits, but less so because jitter was a smaller portion of the hold skew budgets.

These jitter numbers may be optimistic because it is unlikely the simulation captured the worst possible supply noise. This would be a fruitful area for further research.

D. Sensitivity Analysis

The skew budgets are based on estimates of the number of paths with low slack and the variability seen in the data delay.

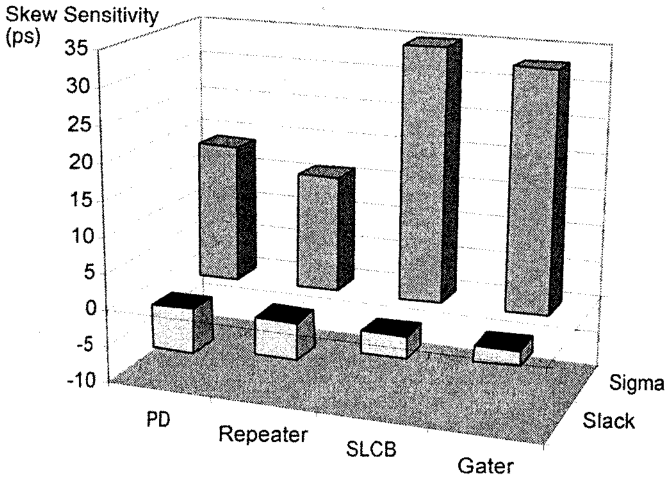


Fig. 8. Examples of setup and hold time constraints and clock domains.

This section describes the sensitivity of clock skew in each level of the skew hierarchy to the estimates. In each figure, the X axis represents the clock domain and the Z axis represents the change in skew, measured in picoseconds.

Fig. 3 shows the sensitivity to the number of short and long paths. The bars indicate the increase in skew from systems with half the estimated number of paths to those with twice the estimated number of paths. $t_{\mathrm{skew}}^{\mathrm{repeater-setup}}$ shows the greatest sensitivity, varying by 17 ps with changes in the assumed number of nearly critical paths. In general, the sensitivity is relatively low. This is important to the designer because the actual number of nearly critical paths is unknown until very late in the design cycle.

Fig. 4 shows the sensitivity of $t_{\mathrm{skew}}^{\mathrm{setup}}$ to the variability in critical path length. The first, light row of bars indicates change in skew when the nearly critical paths are assume to have up to 100 ps slack, rather than up to 50 ps slack. This would account for some paths having better than worst-case supply voltage. Because there are a large number of paths this decreases the expected skews only slightly. The second, darker row of bars indicates change in skew when the nearly critical paths increase the half range of the uniformly distributed portion of delay variation by 10 ps. Again, with a large number of paths, it is highly likely that one will take on nearly worst-case data skew so the overall skew increases by 10 ps. The third, darkest row of bars indicates sensitivity to increasing the standard deviation of the normally distributed portion of delay variations by 10 ps. This is the most important source of variability and has the greatest impact on paths in the SLCB domain because it represents the greatest number of paths, one of which might see many standard deviations of variation.

Fig. 3 shows the sensitivity of $t_{\mathrm{hold}}^{\mathrm{skew}}$ to the variability in short path length. The first light row of bars indicates change in skew when the nearly short paths are assumed to have up to 60 ps of positive margin rather than 30 ps. Again, with a large number of paths, this only decreases the skew budget slightly. The second dark row of bars indicates change in skew when the short paths increase the standard deviation in delay by 10 ps. As with long paths, this is the largest source of variability.

In all cases, the mean and median skew budgets from the Monte Carlo simulation are equal to within the accuracy of the simulation. The standard deviation of worst-case skew from chip to chip for the Gater and SLCB domains was less than 10 ps and for the repeater and PD domains was 10–16 ps.

VI. Conclusion

This paper has presented a set of setup and hold skew budgets in a four-level clock domain hierarchy for the McKinley microprocessor. For setup constraints, the designer is primarily concerned about the difference between skews at different levels of the hierarchy, determining how much more margin must be provided for global paths than for local paths. Poor skew budgets result in overdesign of either local or global paths. For hold time constraints, the designer is concerned about the absolute skew seen by the path. Inadequate skew budgets result in nonfunctional silicon.

The budgets were derived from a Monte Carlo simulation of the major skew sources. Such a statistical approach is important to avoid gross pessimism of summing worst-case clock skew components. Simulation was necessary to model the different number of paths between elements seeing different amounts of skew and because many components of skew exhibit a uniform rather than normal distribution.

The simulations show that jitter caused by voltage noise in the clock distribution buffers is the largest source of skew in H trees and must be controlled as well as possible. They also show that modeling variations in data delay as well as clock network delay is important in generating realistic skew budgets. Considering such variations reduces the difference between global and local skew by a factor of two. Variations in data delay also reduce the potential cycle time benefits of active deskew circuits because the paths experiencing the greatest clock skew are unlikely to be the ones with the longest data delay.

Acknowledgment

The authors would like to thank S. Wells, T. Chen, T. Michalka, and R. McGowen for providing advice and data. The processor design is due to the enormous efforts of a dedicated design team. They would also like to thank the anonymous reviewers who provided numerous helpful suggestions.

References

[1] P. Restle and A. Deutsch, “Designing the best clock distribution network,” in *Proc. VLSI Symp.*, Honolulu, HI, June 1998, pp. 2–5.

[2] S. Sauter, D. Cousinard, R. Thewes, D. Schmitt-Landsiedel, and W. Weber, “Clock skew determination from parametric variations and chip and wafer level,” in *Proc. 4th Int. Workshop Statistical Metrology*, Kyoto , Japan, June 1999, pp. 7–9.

[3] K. Sakallah, T. Mudge, and O. Olukotun, “Analysis and design of latch-controlled synchronous digital circuits,” *IEEE Trans. Comput.-Aided Design*, vol. 11, pp. 322–333, Mar. 1992.

[4] D. Harris, M. Horowitz, and D. Liu, “Timing analysis including clock skew,” *IEEE Trans. Comput.-Aided Design*, vol. 18, no. 11, Nov. 1999.

[5] D. Harris, *Skew-Tolerant Domino Circuits*. San Francisco, CA: Morgan Kaufmann, 2001.

[6] P. Gronowski *et al.*, “High performance microprocessor design,” *IEEE J. Solid-State Circuits*, vol. 33, pp. 676–686, May 1998.

[7] K. Nowka and T. Galambos, “Circuit design techniques for a GHz integer microprocessor,” in *Proc. Int. Conf. Computer Design*, Austin, TX, Oct. 1998, pp. 11–16.

[8] S. Healey *et al.*, "A 7th-Generation x86 microprocessor," *ISSCC Dig. Tech. Papers*, pp. 92–93, Feb. 1999.

[9] J. Alvarez *et al.*, "450 MHz PowerPC microprocessor with enhanced instruction set and copper interconnect," *ISSCC Dig. Tech. Papers*, pp. 96–97, Feb. 1999.

[10] R. Colwell and R. Steck, "A 0.6 μm BiCMOS processor with dynamic execution," *ISSCC Dig. Tech. Papers*, pp. 176–177, Feb. 1995.

[11] S. Santhanam *et al.*, "A low-cost, 300-MHz, RISC CPU with attached media processor," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1829–1839, Nov. 1998.

[12] N. Vasseghi, K. Yeager, E. Sarto, and M. Seddighnezhad, "200 MHz superscalar RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1675–1686, Nov. 1996.

[13] N. Gaddis and J. Lotz, "A 64-b quad-issue CMOS RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1697–1702, Nov. 1996.

[14] C. Maier *et al.*, "A 533-MHz BiCMOS superscalar RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1625–1634, Nov. 1997.

[15] P. Restle, K. Jenkins, A. Deutsch, and P. Cook, "Measurement and modeling of on-chip transmission line effects in a 400 MHz microprocessor," *IEEE J. Solid-State Circuits*, vol. 33, pp. 662–665, Apr. 1998.

[16] C. Nicoletta *et al.*, "A 450-MHz RISC microprocessor with enhanced instruction set and copper interconnect," *IEEE J. Solid-State Circuits*, vol. 34, pp. 1478–1491, Nov. 1999.

[17] P. Restle *et al.*, "A clock distribution network for microprocessors," *IEEE J. Solid-State Circuits*, vol. 36, pp. 792–799, May 2001.

[18] D. Bailey and B. Benschneider, "Clocking design and analysis for a 600-MHz alpha microprocessor," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1627–1633, Nov. 1998.

[19] P. Sanda, D. Knebel, J. Kash, H. Casal, J. Tsang, E. Seewann, and M. Papermaster, "Picosecond imaging circuit analysis of the POWER3 clock distribution," *ISSCC Dig. Tech. Papers*, pp. 372–373, Feb. 1999.

[20] P. Zarkesh-Ha, T. Mule, and J. Meindl, "Characterization and modeling of clock skew with process variations," in *Proc. Custom Integrated Circuits Conf.*, San Diego, CA, May 1999, pp. 441–444.

[21] S. Nassif, "Within-chip variability analysis," in *Proc. Int. Electron Device Meet.*, San Francisco, CA, Dec. 1998, pp. 283–286.

[22] A. Chrakasan, W. Bowhill, and F. Fox, Eds., *Design of High-Performance Microprocessor Circuits*. New York: IEEE Press, 2001, ch. 6, pp. 98–115.

[23] D. Sylvester, O. Nakagawa, and C. Hu, "Modeling the impact of back-end process variations on circuit performance," *VLSI Symp. Dig. Tech. Papers*, pp. 58–61, June 1999.

[24] S. Zanella, A. Nardi, M. Quarantelli, A. Neviani, and C. Guardiani, "Analysis of the impact of intra-die variance on clock skew," *IEEE Trans. Semicond. Manuf.*, vol. 13, pp. 401–407, Nov. 1999.

[25] K. Bowman, S. Duvall, and J. Meindl, "Impact of Die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution," in *Proc. ISSCC Dig. Tech. Papers*, San Francisco, CA, Feb. 2001, p. 17.6.

[26] S. Tam, S. Rusu, U. Desai, R. Kim, J. Zhang, and I. Young, "Clock generation and distribution for the first IA-64 microprocessor," *IEEE J. Solid-State Circuits*, vol. 35, pp. 1545–1552, Nov. 2000.

[27] S. Yang *et al.*, "A high performance 180 nm generation logic technology," in *Proc. Int. Electron Device Meet.*, San Francisco, CA, Dec. 1998, pp. 8.1.1–8.1.4.

[28] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, Oct. 1989.

[29] R. Heald *et al.*, "Implementation of a 3rd-generation SPARC V9 64b microprocessor," *Proc. ISSCC Dig. Tech. Papers*, pp. 412–413, Feb. 2000.

[30] J. Silberman *et al.*, "A 1.0-GHz single-issue 64-Bit PowerPC integer processor," *IEEE J. Solid-State Circuits*, vol. 33, Nov. 1998.

David Harris, photograph and biography not available at the time of publication.

Sam Naffziger, photograph and biography not available at the time of publication.