Clocking Design and Analysis for a 600-MHz Alpha Microprocessor

Daniel W. Bailey and Bradley J. Benschneider

Abstract— **Design, analysis, and verification of the clock hierarchy on a 600-MHz Alpha microprocessor is presented. The clock hierarchy includes a gridded global clock, gridded major clocks, and many local clocks and local conditional clocks, which together improve performance and power at the cost of verification complexity. Performance is increased with a windowpane arrangement of global clock drivers for lowering skew and employing local clocks for time borrowing. Power is reduced by using major clocks and local conditional clocks. Complexity is managed by partitioning the analysis depending on the type of clock. Design and characterization of global and major clocks use both an AWEsim-based computer-aided design (CAD) tool and SPICE. Design verification of local clocks relies on SPICE along with a timing-based methodology CAD tool that includes data-dependent coupling, data-dependent gate loads, and resistance effects.**

Index Terms— **Clocks, delay estimation, electromagnetic coupling, microprocessors, resistance.**

I. Introduction

THE microprocessor discussed in this paper is the third major implementation of the Alpha architecture.[1] It is an out-of-order execution, superscalar microprocessor that performs register renaming, speculative execution, and dynamic scheduling in hardware. It contains four integer execution units and two floating-point execution units, including hardware dedicated to fully pipelined integer multiplication, motion video instructions, and floating-point add, multiply, divide, and square-root operations. There are separate 64-kB, two-way set associative, on-chip instruction and data caches; the data cache is write back and handles two references per cycle. The microprocessor has a four-instruction fetch width but is capable of six-way issue. The initial implementation of the microprocessor is fabricated in a 0.35-μm process, has 15.2 million transistors, is 1.69 × 1.88 cm^2, and at 2.2 V runs at greater than 600 MHz [1].

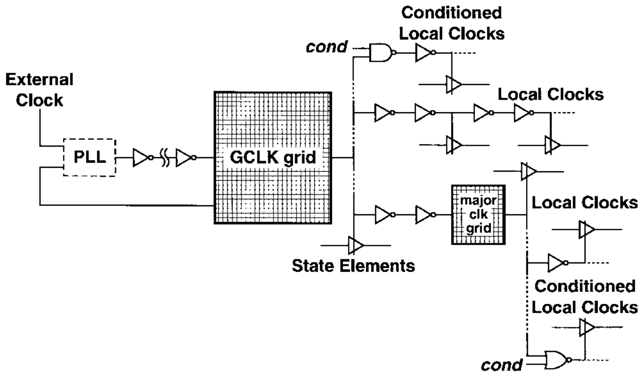

Fig. 1. Clock hierarchy.

The design of the clock distribution network for a high-performance microprocessor like this is necessarily aggressive, involves design tradeoffs, and is therefore challenging. For example, skew directly penalizes cycle time, justifying great efforts to reduce it. Clock power consumption, on the other hand, often conflicts with skew-reduction techniques but can quickly limit packaging options if not anticipated. A large microprocessor at these frequencies will have a large number of critical timing paths, so another consideration is designing a clocking methodology that is flexible enough to solve localized timing problems. The clock design described in this paper addresses these major issues. It is similar to previous Alpha designs [2], [3] in that it uses a single-node, gridded, two-phase global clock, in this case named GCLK, that covers the entire die. It is fundamentally different from previous Alpha microprocessors, however, by including clocks in the distribution network that are several stages past GCLK. This microprocessor is the first Alpha implementation to employ a *hierarchy* of clocks, which is key to how the design challenges mentioned above are met [4].

A diagram of the clock hierarchy of the microprocessor is shown in Fig. 1. Local clocks and local conditional clocks are driven several stages past GCLK. State elements and clocking points exist from zero to eight gates past GCLK. In addition, there are six major regional clocks two gain stages past GCLK with grids juxtaposed with GCLK, but shielded from it. These major clocks also drive local clocks and local conditional clocks. The motivations for implementing this complex clock distribution network are twofold: to improve performance and to save power. How this is accomplished is explained in Sections III and IV. With the increased clocking complexity, however, comes the increased need for rigorous and thorough timing verification.

Previous Alpha microprocessors had a single clock that was tightly controlled. Race-through verification used a simple *gate-count* methodology based on the characterized clock skew and latch hold times. The freedom and complexity of the clock hierarchy in this microprocessor, however, dictates using a *timing-based* methodology. Clocks between driving and receiving latches can be a different number of stages off

Manuscript received April 6, 1998; revised June 4, 1998.

The authors are with Compaq Computer Corp., Shrewsbury, MA 01545 USA (e-mail: dan.bailey@compaq.com).

Publisher Item Identifier S 0018-9200(98)07043-7.

[1] Compaq Computer Corp. Alpha 21 264.

0018–9200/98$10.00 © 1998 IEEE

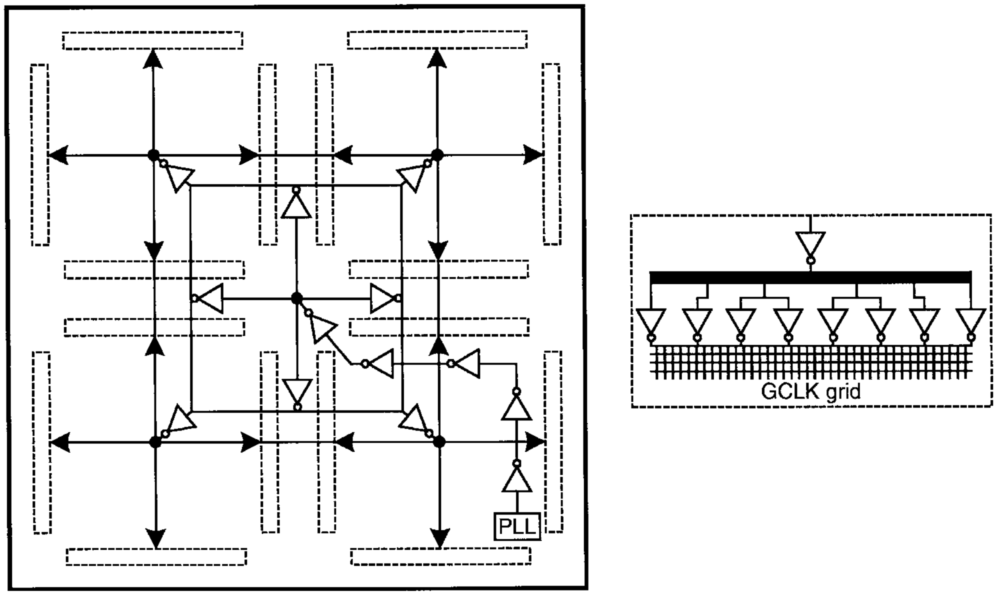

Fig. 2. Global clock distribution network.

GCLK, a factor that must be considered in addition to skew and path-delay variation. In this timing-based methodology, resistance and capacitance are extracted from layout for all the clocks and signals. GCLK and major clocks are characterized both by an AWEsim-based computer-aided design (CAD) tool [5] on a full extracted network and by SPICE based on a simplified extracted network. All local clocks, in comparison, are simulated in SPICE for minimum and maximum delays. These results are used as input to timing-based CAD tools developed in-house that analyze critical paths and race-through paths. In this way, all clock paths and all signal timing paths are rigorously and conservatively checked, while still providing considerable design freedom to solve timing problems.

To describe more thoroughly how the microprocessor clock distribution network was designed and analyzed, the body of this paper is divided into the following sections: global clock, major clocks, local clocks, and verification. First, Section II describes GCLK driver placement, routing and grid layout, and skew simulations and measurements. Section III explains the characterization of major clocks (which herein refers to all large gridded clocks except GCLK). Section IV discusses the design methodology of a clock hierarchy with local clocks and local conditional clocks. Section V, the last section of the body of the paper, describes the clocking verification procedure and effects included in the analysis.

II. Global Clock

As shown in Fig. 1, clock generation is provided by an on-chip, low-jitter phase-locked loop (PLL) [6]. The PLL multiplies a low-frequency (80–200 MHz) external clock and is used for I/O synchronization. It has a separate, regulated 3.3-V power supply and is located in a corner of the chip to minimize noise impact to the PLL. The clock distribution network up to and including GCLK is included in the feedback loop of the PLL to control phase alignment. The PLL is at the root of the global clock distribution tree, which is shown in more detail in Fig. 2.

Fig. 2 also diagrams the locations of clock drivers along the GCLK distribution network. The PLL clock signal is routed along a trunk to the center of the die and is distributed by X trees and H trees [7] to 16 distributed GCLK drivers. The arrangement of GCLK drivers, which resembles four "windowpanes," achieves low skew by dividing the chip into regions, thus reducing the maximum distance from the drivers to the farthest loads. A windowpane arrangement also reduces sensitivity to process variation because each grid pane is redundantly driven from four sides (although only two opposite sides are theoretically needed to attain the same skew). In general, distributing the drivers widely across the chip also has the dual benefits of reducing power-supply collapse and improving heat-dissipation efficiency.

The final two stages of the GCLK distribution network, as shown by the dashed boxes in Fig. 2, use an RC tree [7] to equalize delay from the central predriver to all the individual GCLK drivers. Since the GCLK drivers are large, the Elmore delay model [8], [9] is dominated by the interconnect resistance times the capacitance of the receiving gate. For this application, the interconnect resistance for an RC tree is less than half the resistance of a tapered H tree for the same number of metal tracks. Consequently, in this case, the predriver would have been 50% larger for a tapered H tree than for the RC tree to achieve the same rise and fall times. Since the predriver for the RC tree is smaller, this also allows preceding driver sizes to be reduced, thus compounding the power and area savings.

Power considerations are important because the clocks on microprocessors are usually major consumers of power. This issue is heightened on Alpha microprocessors because of their use of gridded global clocks with aggressive skew targets. The advantages of a gridded clock include 1) skew that is determined largely by grid interconnect density and is insensitive to gate load placement, 2) universal availability

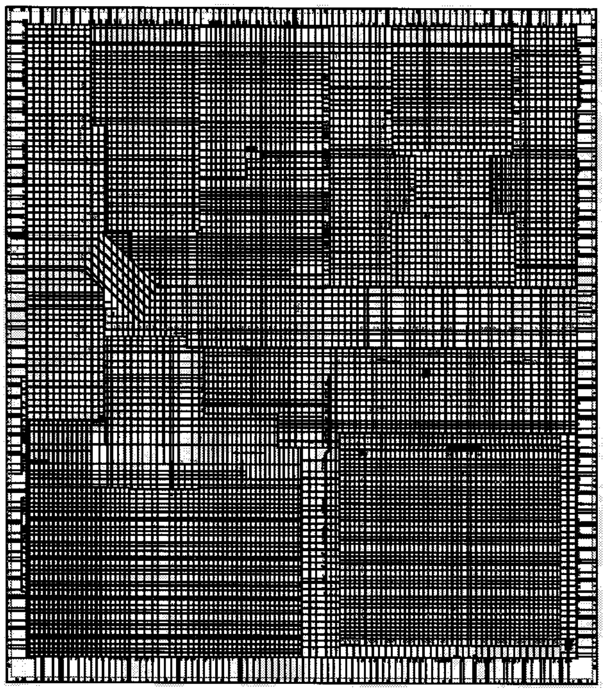

Fig. 3. Global clock grid.

of clock signals, 3) concurrent, independent design, and 4) good process-variation tolerance. The primary disadvantage of a gridded clock is the "extra" capacitance of the grid. At 600 MHz and 2.2 V, typical power usage for the processor is 72 W, of which GCLK uses 10.2 W, and the complete distribution network that eventually drives GCLK uses 5.8 W. With a gridded clock, the power-performance tradeoff is essentially determined by the choice of skew target, which establishes the needed grid density and, therefore, the clock driver size.

The GCLK grid is shown in Fig. 3. It traverses the entire die and uses 3% of the upper level low-impedance interconnect, i.e., of Metal 3 and Metal 4. (This figure is captured from a layout CAD tool. Line widths are misleadingly thick because of screen resolution limitations.) All clock interconnect is laterally shielded with either V_{DD} or V_{SS} interconnect. The microprocessor also has a V_{SS} reference plane sandwiched between Metal 2 and Metal 3 and a V_{DD} reference plane above Metal 4. Except for transverse interconnect, clock grid wires are capacitively and inductively shielded. All clock wires and all lateral shields are manually placed; no place-and-route tools were used.

To simulate GCLK skew, an RC equivalent model is extracted from the layout of the GCLK grid and gate loads on GCLK using an in-house CAD tool. This 420 000-subnodes model is input for an AWEsim-based tool to calculate skew across GCLK. Results of this simulation are shown in Fig. 4. The windowpane placement of GCLK drivers is clearly evident. Fig. 4 also shows 72 ps of total skew at 100°C and 1.8 V assuming worst case conditional loading. Skew on Metal 1 and Metal 2 is less than 10 ps and is included in the simulation results shown.

Measuring GCLK skew is not as straightforward as simulating it. The sample is prepared with a focused ion beam to open the top layer of passivation nitride and to deposit tungsten plugs at 36 sample sites with access to GCLK. An e-beam tester is used to measure clock edges relative to one another; the measured skew is plotted in Fig. 5. Total skew is 65 ps running at a 0°C ambient and 2.2 V, which does not match simulation conditions because of resource restrictions. Direct comparison with Fig. 4 is difficult because of uncertainty and noise from the measurements that are exacerbated by e-beam deterioration of the probe points and the nonplanarity of the tungsten plugs. Because of the coarseness of the measured grid, the windowpane arrangement of GCLK drivers is more difficult to discern. The limited data in Fig. 5, however, show that the RC layout extraction and process correlation are consistent with the simulations.

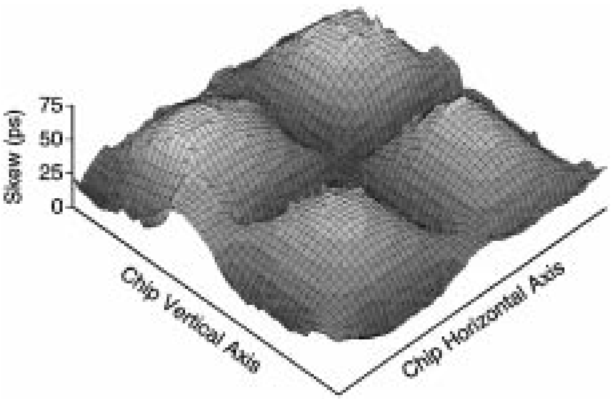

Fig. 4. Simulated global clock skew.

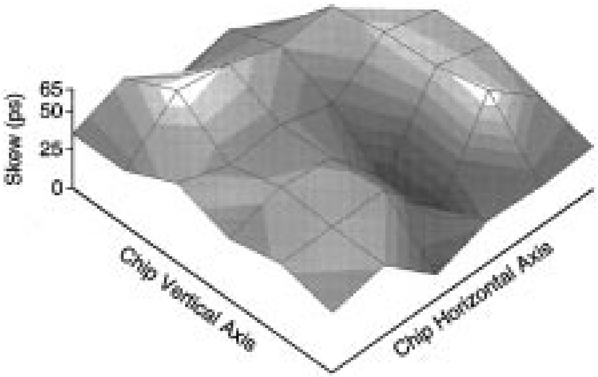

Fig. 5. Measured global clock skew.

III. Major Clocks

In the clock hierarchy diagrammed in Fig. 1, there is a gridded clock two inversions past GCLK called a major clock. There are six major clocks that drive large regional grids over their respective execution units. These major clocks are shown in Fig. 6; the areas they provide clock are listed in Table I. Empty areas in Fig. 6 are the exclusive domains of local clocks, although local clocks are also present in the major clock areas. The grid density varies widely between major clocks, and sometimes even for a single major clock. The densest areas use up to 6% of Metal 3 and Metal 4, twice that of GCLK. The major-clock grid-density variation

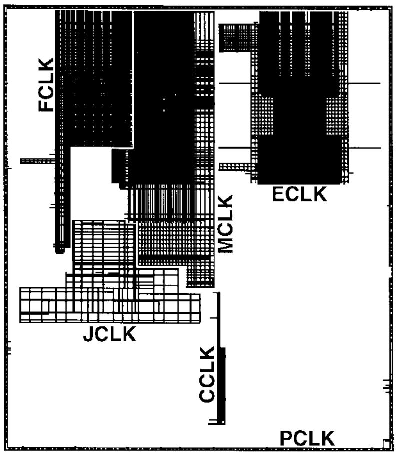

Fig. 6. Major clock grids.

TABLE I

Major Clock	Area
CCLK	bus interface unit
ECLK	integer issue and execution units
FCLK	floating point issue and execution units
JCLK	instruction fetch and branch prediction unit
MCLK	load/store unit
PCLK	pad ring

is a reflection of the wide variation of clock loads. Despite the intermittently heavy grids, however, the dominant reason major clocks are included in the hierarchy is to save power.

Major clocks driven by a gridded global clock substantially reduce power in two ways. First, major clock drivers, which are two gain stages past GCLK, are localized to the clock loads. A gridded global clock without major clocks would require larger drivers and a denser grid to deliver the same clock skew and edges. Second, major clock grids are locally sized to meet the skew targets. At 600 MHz and 2.2 V, the major clocks use 14.0 W. If the same loads had been placed on the GCLK grid and the 75-ps skew target maintained, an estimated 40 W, at least, would have been needed instead of the 24 W for GCLK plus major clocks.

While clock power is important, clock performance on Alpha microprocessors is paramount. Major clocks are designed so that delay from GCLK is centered at 300 ps. The target specifications for skew are ±50 ps. The target specifications for 10–90% rise and fall times are <320 ps. All major clocks easily meet both sets of objectives. Early SPICE simulations are based on both extracted (clock load) layout and best estimate grid models. These simulations verify that target specifications are achieved under a variety of scenarios, including constant temperature, worst case intradie temperature variation, constant voltage, worst case intradie voltage variation, zero GCLK skew, and GCLK skew positioned according to GCLK simulations. The lattermost simulation ensures that major clock skew is conservatively set to 100 ps and is not additive to GCLK skew because it already includes it. As with GCLK, major clocks are characterized with an AWEsim-based tool on full-extracted layout to make sure skew and edge rates match early SPICE simulations and beat the target specifications.

Extracted grids and loads on major clocks are also reduced and simulated a second, independent time with SPICE. This serves as a redundant check to ensure the integrity of the major clock drivers and grids. Accurate performance characterization of the major clocks is important because they are the default clocks for state elements in their execution units. There are six times the number of loads on the major clocks, in aggregate, than there are on GCLK.

IV. Local Clocks

As illustrated in Fig. 1, local clocks are generated as needed from any clock, including other local clocks. In contrast to major clocks, local clocks are generally neither gridded nor shielded. There are no strict limits on the number, size, or logic function of local-clock buffers, and there is no duty-cycle requirement, although timing path constraints must always be met. Local clocks have permitted ranges for clock rise and fall times, but with only this restriction there is considerable design freedom.

This framework of flexible guidelines presents opportunities to reduce power. Logic gates can be used as clock buffers to create local conditional clocks, i.e., gated clocks [10]. When data are conditioned but clocks are unconditional, signal power is saved but clock power is wasted. With conditional clocking, clocks and signals are kept dormant when not in use. At 2.2 V and 600 MHz, local (unconditional) clocks use 7.6 W, and local conditional clocks use a maximum of 15.6 W, assuming they switch every cycle. There are issues to using conditioned local clocks, nonetheless, that must be accounted for, including increased verification complexity, a potential increase in clock path delay variation, and additional clock load compared to an inverter. Peak power reduction is an unqualified benefit, however, when conditional clocks are placed on exclusive execution units.

A clock hierarchy with local clocks can translate into a significant performance advantage, too, by locally managing clock path delay variations. Performance can be improved by adjusting the number of clock buffers on driving and receiving state elements or, in other words, “time borrowing” to solve timing-path problems [11], [12]. For example, to fix a critical path, inverters are added to the clock path of the receiving latch. Time borrowing represents a degree of freedom to solve local timing issues that is unavailable when strictly using a single clock methodology.

While the advantages of including local clocks are significant, each clock requires thorough and detailed analysis. Consider capacitive coupling. Parasitic coupling on major clocks and GCLK is negligible because of the large grid and load capacitance, and because coupling effects are statistically averaged across the network. Local clocks, on the other hand, are typically small and potentially experience a much wider range of delays due to data-dependent variations than large

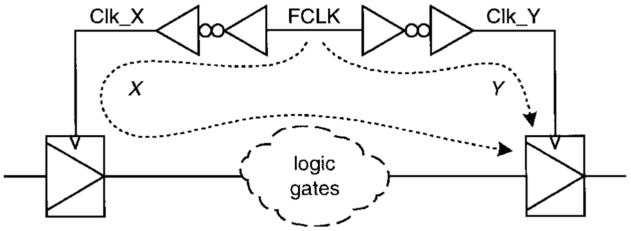

Fig. 7. Example circuit illustrating race-through and critical path constraints.

grids. Worst case assumptions for coupling, gate loads, and resistance effects cannot be applied across all local clocks uniformly because many race- and speed-critical paths have tight margins. A more feasible approach is to model each local clock individually using assumptions specific to each node to provide accurate timing analysis.

V. Verification

To analyze the clock hierarchy accurately, each local clock is simulated in the context of how it affects critical paths and races. These timing path constraints are illustrated by Fig. 7. In this example, delay X is measured from a major clock (FCLK) through a local clock named Clk_X, through a driving latch (namely, "clock to Q"), through some arbitrary logic, and to a receiving latch. Delay Y is measured from FCLK to a local clock named Clk_Y. To meet speed requirements, the following constraint must be met:

$$\max(X) + t_{\text{setup}} + t_{\text{skew}} - \min(Y) \leq T \qquad (1)$$

where t_{setup} is the setup time of the receiving latch, t_{skew} is the specified skew across GCLK or a major clock (in this case, FCLK), and T is the cycle time goal. For race-through verification, the following constraint must be met:

$$\frac{\min(X)}{\max(Y) + t_{\text{hold}} + t_{\text{skew}}} \geq M \qquad (2)$$

where t_{hold} is the hold time of the receiving latch and M is a unitless number greater than 1.0 that determines the sliding margin. By constraining races by a ratio M, instead of an absolute margin as in (1), the margin increases for longer paths. A sliding margin is more conservative and thus more desirable than an absolute margin because race-through is a functionality issue and because risk of delay variation increases with path length. As seen from (1) and (2), both the minimum and the maximum delays are needed for X and Y to test for race and critical path failures.

Major factors that can affect minimum and maximum path delays are process variations, power-supply variations, temperature, data-dependent interconnect capacitance, data-dependent gate capacitance, and interconnect resistance. Two techniques are used to account for these effects. First, process and environmental variations are bounded by the choice of M in (2). That is, M has the dual role of providing margin and accounting for process, voltage, and temperature variations. Second, as described in the following paragraphs, worst case but accurate coupling, gate loads, and interconnect resistance are explicitly modeled for each node when calculating minimum and maximum path delays.

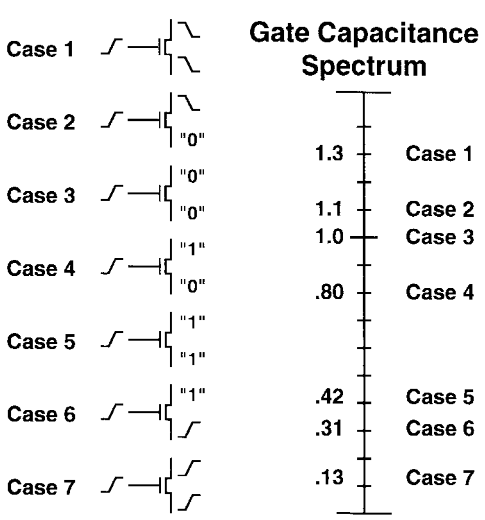

Fig. 8. Data-dependent gate loading.

Capacitive coupling is simulated in the standard way. All coupling capacitances are extracted from layout. If nothing is known about aggressor signals, worst case transitions are assumed for calculating effective capacitance. For example, if a victim clock is rising, then for maximum delay, an aggressor signal is assumed to be falling at the same time, whereas for minimum delay, it is assumed to be simultaneously rising. If aggressor signals are known to be exclusive or complementary, relief is conservatively applied.

Gate loading is a second source of effective capacitance that can strongly influence minimum and maximum path delays. The effective capacitance can vary dramatically depending on how the load device is biased. Fig. 8 illustrates how the maximum effective gate capacitance changes on an NMOS device for different terminal biases [13]. Data-dependent gate capacitance modeled this way includes transient channel charge and Miller effects. As can be seen in Fig. 8, worst case biasing gives ten times the effective gate capacitance as best case biasing. In all, there are 44 configurations for minimum and maximum NMOS and PMOS transistor loads. Instead of always assuming worst case biasing, the most conservative yet appropriate configuration is chosen depending on the context of the device. Thus, worst case gate loads are still assumed in the timing analysis, but accuracy is improved by considering many configurations and disregarding those that do not apply.

Interconnect resistance is another effect that is explicitly modeled for both maximum and minimum delay calculations. A simple approach for calculating RC delay is to omit interconnect resistance for minimum-delay paths and to include resistance for maximum-delay paths. This is not always the worst case approximation, however, because of resistance isolation effects. Fig. 9 shows a clock distribution network without and with resistance [Fig. 9(a) and (b), respectively]. The drivers at the bottom of each circuit are positioned physically close to one another, while the upper pair of drivers is distant. The waveforms in Fig. 9(c) show that the delay to the distant driver (V'_{RC1}) is increased by the interconnect resistance, but that the delay to the near driver (V_{RC1}) is

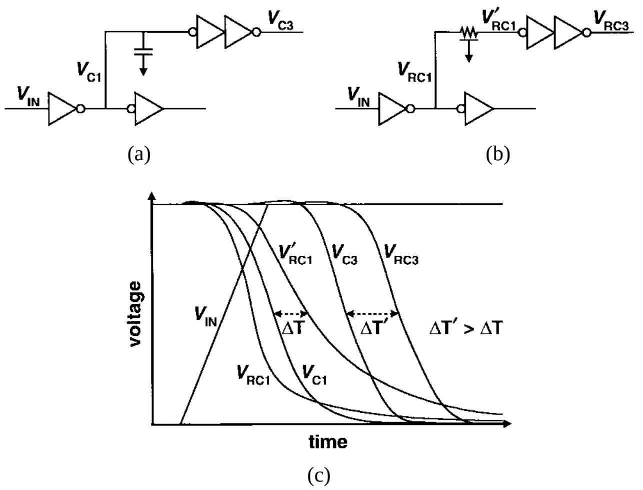

Fig. 9. Resistance isolation and downstream effects. (a) Example clock distribution network without resistance. (b) Example clock distribution network including resistance. (c) Waveforms.

actually *decreased* vis-à-vis the case without resistance (V_{C1}). That is, the short clock path is faster when simulated with resistance because the resistance isolates the gate capacitance of the distant driver.

In addition, interconnect resistance effects can have a lingering influence on later device delays. In Fig. 9(c), ΔT is the additional delay directly caused by including interconnect resistance. This delay increases to $\Delta T'$ after progressing through two more drivers. This increase is caused by the slower edge-rate of V'_{RC1} due to the resistance. A simple model that only accounts for the RC delay to the input of the second stage does not account for the accumulated delay of the entire path.

To include the effects described above in the timing-analysis CAD tools, comprehensive resistance and capacitance information is extracted from layout, which includes the physical position of all source, drain, and gate connections. A detailed RC network is built for each local clock tree starting from either a major clock or GCLK. Gate capacitance models are placed throughout the network using the appropriate terminal biasing assumptions. Two SPICE simulations are run for each node, one using minimum effective capacitance assumptions and the other using maximum effective capacitance assumptions. Minimum and maximum delays are specified individually per device load and applied to the timing-based methodology CAD tools for timing path analysis. In all, more than 60 000 local clock nodes were simulated with SPICE in this manner using an aggregate 19 million simulation elements and 14 million simulation nodes.

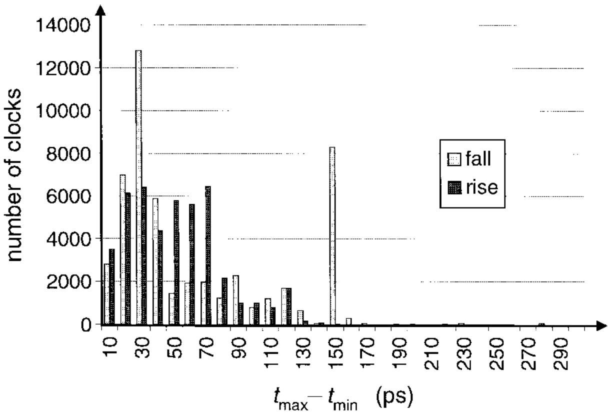

Fig. 10. Differences between minimum and maximum clock node delays.

Fig. 10 is a histogram from all these simulations of the variation between minimum and maximum delays per local clock. Some variations are as high as 280 ps. This illustrates two points. First, some local clocks can tolerate very high variations, as high as 280 ps in some cases, in addition to major clock skew and still pass all timing restrictions. Second, having each clock individually and thoroughly analyzed is necessary, since applying the worst case 280 ps of minimum-to-maximum variation to all local clocks would have crippled the design. Performing detailed analysis on each clock independently gives significantly more freedom for the majority of local clocks than would have been possible by applying a worst case skew as used on the major clocks and GCLK.

VI. Summary

Many aggressive design techniques are implemented in the clocking of the latest Alpha microprocessor. A gridded global clock with a windowpane arrangement of final distributed drivers is used to lower the skew. The final stage is distributed with an RC tree to reduce power. Major clocks driven by the gridded global clock provide more power savings. The rest of the hierarchy is comprised of local clocks and local conditional clocks, which provide the design freedom for both power savings and performance improvements.

Design verification relies on SPICE, an AWEsim-based CAD tool, and a timing-based methodology CAD tool. Data-dependent coupling, data-dependent gate loads, and resistance are included in the local clock simulations. Minimum and maximum clock-path delays at each device load are input to the timing analysis tools. Although the clock hierarchy allows performance and power design goals to be met, it also requires both the precise characterization of global and major clocks and the rigorous analysis of all local clocks.

Acknowledgment

The authors wish to acknowledge the following people who made instrumental contributions to the designs of GCLK, major clocks, or the verification tools: R. Allmon, S. Bell, R. Dupcak, H. Fair, J. Farrell, B. Gieseke, M. Lamere, M. Matson, B. McGee, J. Mylius, and M. Smith. Principal contributors to the timing CAD tools were B. Grundmann, N. Nassif, N. Rethman, and E. Shriver.

References

[1] B. Gieseke *et al.*, "A 600 MHz superscalar RISC microprocessor with out-of-order execution," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 1997, pp. 176–177.

[2] D. Dobberpuhl *et al.*, "A 200-MHz 64-bit dual-issue CMOS microprocessor," *Digital Tech. J.*, vol. 4, pp. 35–50, 1992.

[3] W. Bowhill *et al.*, "Circuit implementation of a 300 MHz 64-bit second-generation CMOS alpha CPU," *Digital Tech. J.*, vol. 7, pp. 100–118, 1995.

[4] H. Fair and D. Bailey, "Clocking design and analysis for a 600 MHz alpha microprocessor," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers,* Feb. 1998, pp. 398–399.

[5] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, 1990.

[6] V. von Kaenel, D. Aebischer, R. van Dongen, and C. Piguet, "A 600 MHz CMOS PLL microprocessor clock generator with a 1.2 GHz VCO," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers,* Feb. 1998, pp. 396–397.

[7] E. Friedman, *Clock Distribution Networks in VLSI Circuits and Systems.* New York: IEEE Press, 1995.

[8] W. Elmore, "The transient response of damped linear network with particular regard to wideband amplifiers," *J. Appl. Phys.,* vol. 19, pp. 55–63, 1948.

[9] T. Sakurai, "Approximation of wiring delay in MOSFET LSI," *IEEE J. Solid-State Circuits,* vol. 18, pp. 418–426, 1983.

[10] J. Montanaro *et al.*, "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1703–1714, 1996.

[11] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. Comput.,* vol. 39, pp. 945–951, 1990.

[12] T.-H. Chao, Y.-C. Hsu, J.-M. Ho, K. D. Boese, and A. B. Kahng, "Zero skew clock routing with minimum wirelength," *IEEE Trans. Circuits Syst. II,* vol. 39, pp. 799–814, 1992.

[13] W. J. Grundmann, D. Dobberpuhl, R. Allmon, and N. Rethman, "Designing high performance CMOS microprocessors using full custom techniques," in *Proc. 34th Design Automation Conf.,* Aug. 1997, pp. 722–727.

Daniel W. Bailey received the B.S.E.E. degree from the University of Cincinnati, Cincinnati, OH, in 1984 and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign in 1986 and 1990, respectively.

He was a Postdoctoral Research Associate at the University of Florida before joining the Faculty at the University of South Carolina as an Assistant Professor in 1991. He joined the Alpha Development Group, Compaq Computer Corp., Shrewsbury, MA, in 1995, where he has since contributed to the design and characterization of the latches and clocks on the Alpha 21264.

Bradley J. Benschneider received the B.S.E.E. degree *(magna cum laude)* from the University of Cincinnati, Cincinnati, OH, in 1987.

He is a Principal Semiconductor Design Engineer in the Alpha Development Group, Compaq Computer Corp., Shrewsbury, MA. He has contributed to the development of two generations of Alpha microprocessors and multiple generations of VAX processors. He led the implementation effort of the memory management and LD/ST units on the Alpha 21264 and drove the verification effort of the local clock network for the 21264. He is the author or coauthor of six technical papers. He has received one patent and has another pending for his work on the Alpha 21264.