A Clock Distribution Network for Microprocessors

Phillip J. Restle, *Member, IEEE*, Timothy G. McNamara, David A. Webber, Peter J. Camporese, Kwok F. Eng, Keith A. Jenkins, *Senior Member, IEEE*, David H. Allen, Michael J. Rohn, Michael P. Quaranta, David W. Boerstler, *Member, IEEE*, Charles J. Alpert, *Member, IEEE*, Craig A. Carter, *Member, IEEE*, Roger N. Bailey, John G. Petrovick, Byron L. Krauter, and Bradley D. McCredie, *Member, IEEE*

***Abstract*—A global clock distribution strategy used on several microprocessor chips is described. The clock network consists of buffered tunable trees or treelike networks, with the final level of trees all driving a single common grid covering most of the chip. This topology combines advantages of both trees and grids. A new tuning method was required to efficiently tune such a large strongly connected interconnect network consisting of up to 6 m of wire and modeled with 50 000 resistors, capacitors, and inductors. Variations are described to handle different floor-planning styles. Global clock skew as low as 22 ps on large microprocessor chips has been measured.**

***Index Terms*—Circuit tuning, clock distribution, inductance, interconnect analysis, transmission lines, visualization.**

I. Introduction

A. Synchronous Clocking

SYNCHRONOUS clocking continues to be the dominant strategy for commercial microprocessors with clock frequencies now exceeding 1 GHz. Large high-performance server microprocessors require global clock distributions with very low uncertainty in clock arrival times. Any uncertainty in the clock arrival times between two points, especially if these points are nearby, can limit clock frequency, or cause functional errors. The difference in clock arrival time between two points is called clock skew. While clock skew can be intentional (as discussed below), unless otherwise specified, clock skew generally refers to undesired or unexpected differences in clock arrival time. When data is launched by a late clock signal, or captured by an early clock, there is insufficient time for the data signal to arrive before the clock, which limits the maximum clock frequency. This type of failure is called a long-path error. Functional errors can also occur when the launching clock is early relative to the capturing clock. In this case, new data arrives before the clock closes the receiving latch for the previous cycle, overwriting the correct data. Such a clock race condition is sometimes called a short-path or early-mode fail, and is an especially costly because it cannot be avoided by slowing down the clock frequency.

Manuscript received August 22, 2000; revised December 1, 2000.

P. J. Restle and K. A. Jenkins are with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: restle@us.ibm.com).

T. G. McNamara, D. A. Webber, P. J. Camporese, and K. F. Eng are with IBM, Poughkeepsie, NY 12601 USA.

D. H. Allen, M. J. Rohn, and M. P. Quaranta are with IBM, Rochester, MN 55901 USA.

D. W. Boerstler, C. J. Alpert, C. A. Carter, J. G. Petrovick, B. L. Krauter, and B. D. McCredie are with IBM, Austin, TX 78758 USA.

R. N. Bailey is with Agere Systems, Austin, TX 78758 USA.

Publisher Item Identifier S 0018-9200(01)03024-4.

B. Local Skew

Since data signals that must travel longer distances are inevitably delayed by buffers and interconnect delays, it is primarily nearby points on the chip that are susceptible to short-path errors caused by clock skew. Thus it is especially important for the clock distribution to achieve low skew between nearby points. The skew between nearby points will be referred to as local skew, and is somewhat arbitrarily defined as the unexpected clock skew within any 1-mm-square area of the chip. Signals traveling more than 1 mm generally have enough interconnect delay that they are less likely to result in a short-path problem.

C. Fast-Path Delay Padding

One typical way to avoid fast-path errors is to add delay to short data paths on the chip. This is usually done by adding buffers to delay such signals, and is often called fast-path delay padding. This method of protecting against the possibility of fast-path errors obviously increases chip area, power, and design time. The larger the local clock skew is expected to be, the more short-path padding must be done. In fact, when a short path is also part of a long path, this padding may in fact degrade performance. To avoid these problems, reducing local skew is highly desirable.

Many global clock distribution strategies seek to reduce local skew within subsets of the design, accepting larger skew between these subsets. When buffered tree networks are used, this can be accomplished by designing a small independent tree or grid for each unit. In this case, the local skew within each unit can be reduced, but there can still be significant skew between points driven by different trees due process and model uncertainties. Since the majority of short paths are likely to occur within a single macro or unit, such a strategy can reduce the need for short-path padding significantly. Unfortunately, any path traversing a boundary will be subject to much larger potential skew.

An ideal global clock distribution achieves low local skew everywhere, not just within certain unit or macro boundaries. This eliminates the timing penalty for signals traveling across macro or unit clock boundaries, reducing the need for short-path padding as well as leaving more time for long paths.

Low skew over larger distances is of course also desirable, but for the largest, highest performance microprocessors, there may be no signals that travel across the whole chip in a single cycle. This means that the skew between opposite edges or corners of the chip will never actually degrade the cycle time by the full amount of the maximum corner-to-corner skew.

D. Global and Local Clock Distribution

It is important to make a distinction between the global clock distribution and local clock generation and distribution. For all but one of the chips described here, a single global clock signal, critical to chip timing, was distributed over the whole chip. Then local clock blocks were used to buffer the global clock, usually generating two local clock phases. These local clock buffers also often controlled scan and test modes. The details and complexity of these local clock blocks varied widely between chips, but in all cases hundreds or thousands of local clock blocks were used to locally buffer the global clock. The power dissipated in these local clock buffers was much larger than the power dissipated in the global clock distribution buffers and wiring.

A wide variety of technologies have been proposed and pursued to reduce the significant power used in global clock distribution. There have been designs where a significant fraction of the microprocessor chip power (40%) was dissipated in the global clock distribution [1]. However, since interconnects have not scaled as well as devices, it has become necessary in more recent high-performance microprocessors, including the designs described in this paper [2]–[6] to locally buffer and amplify the global clock signal. Thus it is important to realize that unless large amounts of local clock gating is done (turning off large portions of the local clock distribution) the local clock distribution in modern processors consumes at least an order of magnitude more power than the global clock distribution. This is simply due to the large total capacitance of all the latches and clocked dynamic gates on the chip. It is a misconception that the total chip power could be reduced significantly by reducing the power used in the global clock distribution. Much of the power used in the global clock distribution is used simply to amplify the clock signal enough to drive the total input capacitance of the local clock buffers. While research into radically new methods of clock distribution is justified, a successful method must include a very low skew and low jitter method of amplifying the clock signal to drive the large capacitive load consisting of hundreds or thousands of local clock buffers distributed very nonuniformly over the chip.

E. Jitter

Undesired variations in the cycle period are called jitter. Jitter reduces performance because some cycles are short, and the chip frequency must be slowed to avoid long-path errors. Generally, a phase-locked loop (PLL) is used to generate a low-jitter clock on chip, often at an integer multiple of an external reference clock. In the past, much of clock jitter came from the PLL and its sensitivity to power-supply noise. However, PLL designs have continued to improve such that PLL jitter has scaled well with chip cycle time. On the other hand, the total delay of the global clock distribution has not scaled with chip cycle time. In fact, the total delay, which is now interconnect dominated, has tended to increase with increasing chip size and the number of clocked circuits. Since the global clock is buffered at many places on the chip, it uses the regular relatively noisy V_{dd} and Gnd, so that much of the total clock jitter is from buffer delay variations in the global clock distribution due to power-supply noise [5]. In addition to the requirement of a low-jitter PLL, the jitter from the clock distribution must be controlled by controlling and minimizing the delay of the global clock distribution. This can be done through optimal buffer and transmission line design, and by minimizing nonperiodic power-supply noise.

F. Known Skew

It is important to note the difference between known clock skew and unexpected clock skew. Skew from modeling errors or process variations is unexpected clock skew, and is always undesired. However, known differences in clock signal arrival times can be tolerated. In fact, skew can even be beneficial if cleverly designed and accurately modeled. For example, by carefully introducing clock skew, it may be possible to allow a specific logic and circuit implementation to run faster, and have improved timing margins on many paths. However, if adding skew can improve performance and margins significantly, this probably means that the logic between latches is not well balanced. In most cases, restructuring the logic or circuit tuning can achieve timing improvements similar to adding skew.

G. Designing for Minimum Global Clock Uncertainty

For all of the microprocessors described here, the primary global clock distribution network was designed to have minimum skew. This decision was made not only to simplify the design and reduce timing complexity, but because it was found that skew in the global clock distribution was highly correlated with clock uncertainty due to modeling or process errors. In the relatively few cases where adding skew was considered an attractive method for improved timing, the skew was created locally by selecting among different local clock generating circuits from a library. In addition, in some cases, programmable local clock generators were used to create different amounts of skew during testing [7]. Although these programmable local clock generators consume somewhat more chip area, they are useful for debugging timing and noise issues, and sometimes even allowed an increase chip performance by programming skew into certain local clock generators judiciously. The choice of minimum skew as the goal for the global clock distribution enables the global clock distribution to have a simple goal, but more importantly, we believe this choice allows the minimization of clock uncertainty from model errors and process variations when using the strategy described in the next section.

II. Many Trees Driving a Single Grid

With the goal of developing a single global clock distribution method to be used for a variety of high-performance server microprocessors, a strategy was developed consisting of many trees driving a single clock grid. Six chips using this clock distribution network will be described, and will be referred to as chips A to F. Chip A was a test chip leading to microprocessor B. Chips B–F correspond to [2]–[6], respectively. In all these chips, the clock network consists of a series of buffered treelike networks driving a final set of 16–64 sector buffers. While the number and placement of these sector buffers vary greatly between designs, in all cases each buffer drives a final tunable tree network. Finally, all of these tunable trees drive the same single clock grid covering most of the chip (Fig. 1), except for

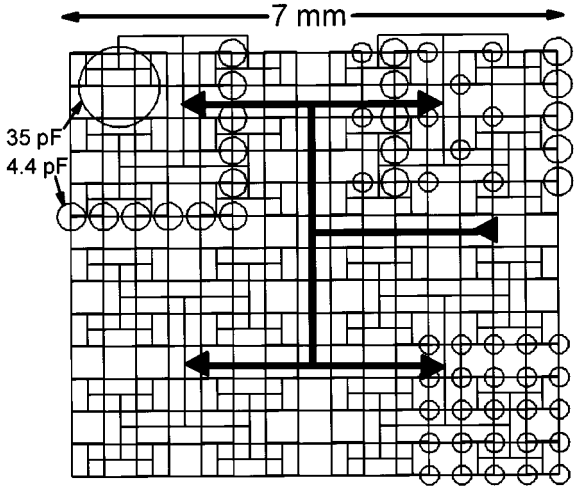

Fig. 1. Chip A, the first test site network, with loads represented by circle diameter. The largest is 35 pF.

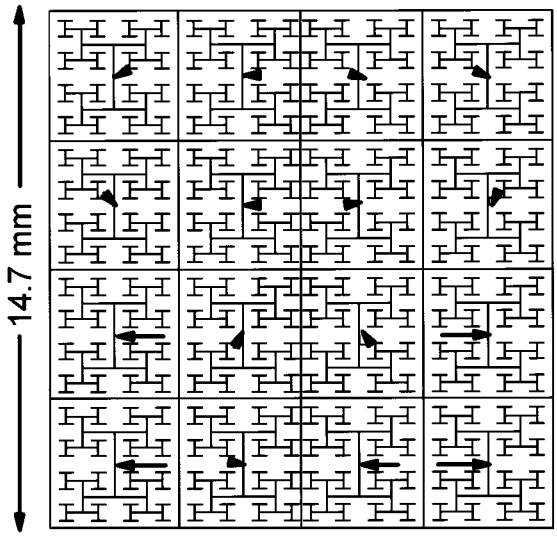

Fig. 2. Chip B final trees with arrows pointing from actual sector buffer locations to the ideal location. The largest displacement is 1.25 mm.

chip F that used one primary grid and two satellite grids [6]. This network topology consisting of a number of tunable trees driving a common grid combines many of the advantages of both trees and grids. Trees have low latency, low power, minimal wiring track usage, and the potential for very low skew. However, without the grid, trees must often be rerouted whenever the locations of clock pins change, or when the load capacitance values change significantly. The grid provides a constant structure so that the trees and the grid they are driving can be designed early to distribute the clock near every location where it may be needed. The regular grid also allows simple regular tree structures. This is important as it facilitates the design of carefully designed transmission line structures with well-controlled capacitance and inductance [8]–[11]. The grid reduces local skew by connecting nearby points directly. The tree wires are then tuned to minimize skew over longer distances, as described below.

A. Design and Modeling

In its simplest form, a clock distribution using this strategy consists of symmetric H-trees driving a simple grid having uniform wire spacing. In the simplest case, the clock loads attached to the grid would be uniform enough that a low-skew low-uncertainty global clock signal might result from the symmetry of the H-trees even without tuning. Unfortunately, negotiating ideal buffer placement on a large microprocessor is often impossible in practice. As both chip size and clock frequency continue to increase, the number of sector buffers needed to drive the grid everywhere with acceptable transition times increases.

The simplest criterion for choosing the size of buffers and the interconnect design is to minimize delay. The faster the distribution is, the more robust it will generally be. However, other design constraints must also be considered. Minimizing delay can result in long wires with slow transition times at the far end, that may increase skew and jitter in the presence of noise or process variations. However, if an attempt is made to keep slew rates too fast everywhere, a large number of small buffers would be needed, which would probably result in larger process variations, and might be very difficult to place where needed. In the presence of inductance, a 10%–90% transition time criterion is not very useful, because of the wide variety of waveforms encountered near the 10% and 90% points. We have found it more useful to use a 30% to 70% transition time definition for global clock waveforms. Typically, a maximum 30%–70% transition time approximately 10% of the cycle time has been achievable, although this is becoming more difficult, and slower transition times may be acceptable in some cases.

Inevitably, some of the sector buffers cannot be placed close to their desired locations due to large blockages in memory arrays and custom data-flow layouts. Fig. 2 shows an example of ideal and actual buffer locations for chip B. To reduce this buffer placement problem, one microprocessor floor plan (chip C) placed all the sector buffers in the vertical centerline of the chip. Although the trees in this case are not H-trees, the same methodology and tools were used. This particular variation required significantly more horizontal wiring tracks to achieve desired transition times on the grid, but very few vertical wiring tracks.

In general, the trees from the central chip buffer to the sector buffers are length matched using wire folding. However, the trees driven by the sector buffers are not length matched as this would greatly increase the wiring tracks and power used. These trees are tuned instead primarily by wire-width tuning, as described in detail below. It is illustrative to compare three-dimensional renderings of these global clock networks. In Figs. 3–5, a third dimension of time or delay is added to a two-dimensional X–Y layout such as Figs. 1 and 2. Buffers, which contribute delay but do not transport the signal in the X–Y plane, appear as vertical lines. Symmetric binary trees appear as stacked "V"-shaped structures. The first hardware, test chip A, is shown in Fig. 3. The product chip C with the sector buffers placed in the chip centerline is shown in Fig. 4. Only one quadrant of chip E is shown (Fig. 5). If chip area remained constant, the total delay from the single central chip buffer to the mesh would tend to decrease with each technology generation. However, due to the large chip size and increased function causing a much greater

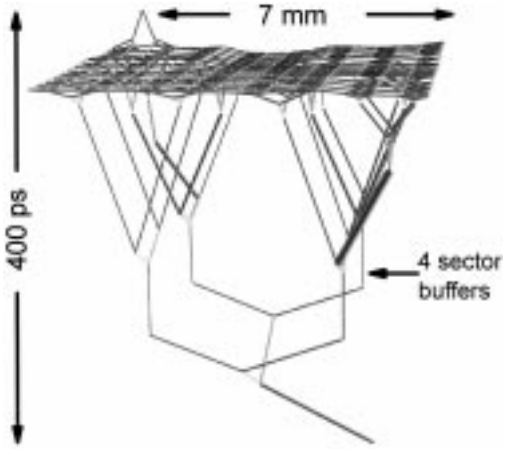

Fig. 3. X–Y-time rendering of chip A corresponding to Fig. 1. Note delay peak on grid due to original design capacitive point load of 16 pF (actually 35 pF).

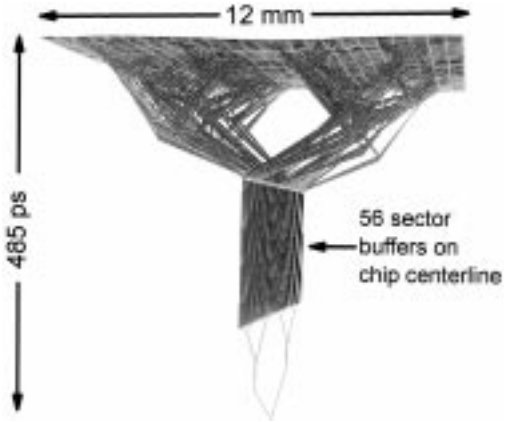

Fig. 4. X–Y-time rendering of chip C with 56 sector buffers in chip centerline.

clock load on chip E, the total delay increased significantly on chip E.

As clock frequencies increase, the global clock signal transition times must be reduced to maintain full swing and signal quality. Even copper on-chip transmission lines are fairly lossy (due to significant wire resistance), so to achieve sharp transition times it is generally necessary to reduce the delay and the wire length of the final tuned trees from the sector buffers to the final grid. This trend is easily seen in Figs. 3–5.

B. Transmission Lines

All the long global clock wires consisted of carefully controlled transmission line structures using the top two (thickest) levels of metal. This includes all the trees and the clock grid. The routing, tuning, and special-purpose extraction included adjacent shields and inductive return paths. Both wire widths and spaces were optimized to achieve fast transition times, minimal overshoot, well-controlled environmental capacitance, low power, and low skew. Analytical models for resistance, capacitance, and inductance were developed for the well-controlled structures used in the global clock distribution. Critical interconnects were split into as many as eight parallel wires each surrounded by V_{dd} and Gnd return paths to optimize *LRC* and

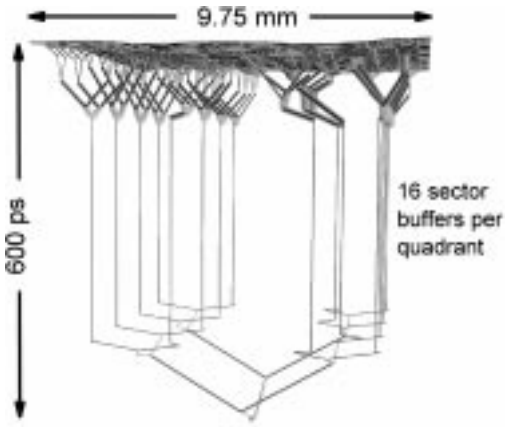

Fig. 5. X–Y-time rendering of one quadrant of chip E. The entire network had 64 sector buffers and a total simulated delay of 1.2 ns.

coupling parameters. Chips C–E and F used copper interconnects. For the much shorter wires connecting the grid to the local clock blocks, simpler *RC* models were often used.

C. Tuning

The clock load distribution on microprocessors is very nonuniform, with some regions containing very high clock load capacitance densities, and other large regions (such as the interior of memory arrays) containing little or no clock load. The clock grid, which was usually fairly uniform over the chip, contributes a grid wire capacitance that is uniform over the chip, but the clock density from the local clock blocks is often larger, and is very nonuniform. To reduce skew due to both nonideal buffer placements (Fig. 2) and nonuniform loads, a quick efficient tuning method was needed. Typically, 10 to 20 complete tuning cycles were done for each microprocessor, each with increasing detail and accuracy of buffer placement, wiring, and clock loads. In addition, many smaller experiments and optimizations were performed to study alternatives or solve specific problems.

A new tuning method was required to allow rapid tuning of a large network with more than 50 000 *RLC* circuit elements and 70 buffers that requires more than an hour for a single full simulation. The tuning method involves approximating the actual network by a similar network that can be separated into a number of smaller independent linear networks. The resulting small independent linear networks can then be rapidly tuned in parallel.

To create the separable network, an artificial network is generated where all the grid loops are first cut such that a similar fraction of the grid is driven by each tree endpoint. This artificial cut-grid network has much more skew than the actual network, because the uncut grid connects together nearby points, and thus provides a local delay-smoothing effect, reducing local skew. To create a more accurate representation of the actual uncut network, this smoothing effect must be included in the separable cut-grid network. To approximate this smoothing effect in the cut-grid network, the capacitive loads are smoothed using a diffusion-like algorithm. Thus a single large network (Fig. 1, with actual loads represented by circles) is approximated by the more

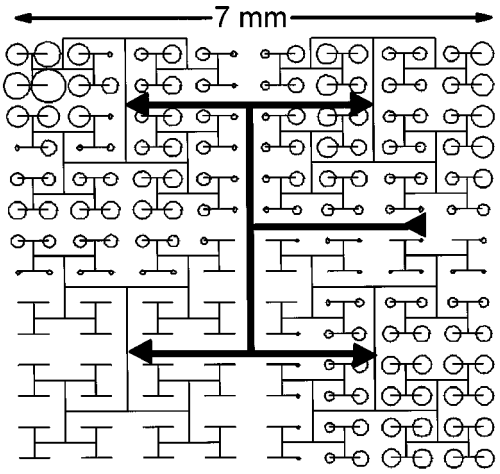

Fig. 6. Chip A cut-grid network corresponding to Fig. 1, with diffused loads. The diffused load capacitances are proportional to the circle diameter.

easily tuned cut-grid network with smoothed capacitive loads (Fig. 6).

In summary, to allow efficient tuning of the large single network resulting from the strategy with many trees driving a single grid, an approximate network was created by cutting the grid. To approximate the smoothing effect of the grid which is lost, the capacitive loads were smoothed instead. The resulting approximate network with cut-grid smoothed loads was then separated into a number of much smaller independent tree networks, which could then be tuned in parallel on a number of processors in less than an hour even for the largest design.

In some cases, the amount of metal width for a particular segment of the global clock distribution varied across the chip by more than a factor of 10 as a result of the tuning process.

III. Measurements

Early experience showed the importance of measuring the internal signal waveforms [9], and measurements have since been done whenever possible. Direct probe measurements were performed on two test chips and one product microprocessor. Diced chips were prepared by removing passivation above the top metal layer. Small probe pads were sometimes incorporated into the design without increasing wiring tracks used. For clock skew measurements, the clock was driven directly, without the use of a PLL, at low frequency. The frequency of operation was limited by the probe card and the need for a relatively quiet power supply. A two-probe measurement method was used employing two identical 500-Ω probes. One probe remained stationary at a point on the grid, and the other roving probe was first calibrated by probing nearby. Using two channels of a 20-GHz digital sampling oscilloscope, waveforms from both probes were accumulated simultaneously. The roving probe was returned periodically to the calibration point to check for drift. The digital waveform data was later processed to correct for small amplitude and voltage offset variations, smoothed, and the delay differences at $V_{dd}/2$ calculated. Even with the limited waveform fidelity of the probes used, this method produced skew delay measurements with an RMS error less than 5 ps,

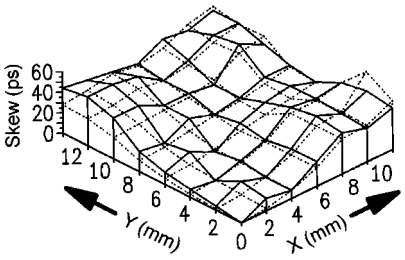

Fig. 7. Measured and simulated skew from test chip A, using the corrected load capacitance model.

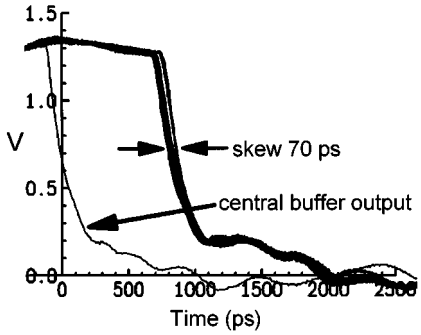

Fig. 8. Measured waveforms at 26 locations (Fig. 9) with 70 ps skew on chip E, Power4 1-GHz test chip.

as long as both probes were measuring very similar waveform shapes.

The clock network on test chip A (Fig. 1) was designed and tuned for an artificial clock load distribution created using multiple instances of a gate capacitor cell in place of the local clock buffers of a real chip. The measured skew was 50 ps, much larger than the originally simulated skew of 14 ps. This was eventually traced to the fact that the capacitor cell was mislabeled with a capacitance a factor of 2.2 smaller than the actual effective capacitance. With corrected capacitances in the model, the RMS grid delay difference between the measurements and the simulation model reduced to 7.5 ps (Fig. 7).

The most challenging design was for test chip E having a chip area of 379 mm^2, where the global clock network was required to drive 7050 global clock pins (where each pin resulted in a tap off the global clock grid) with a total pin capacitance of 792 pF. The chip was required to function at 1 GHz. Direct probe skew measurements were made at low clock frequency (50 MHz) due to the limited V_{dd} current available from the special cantilever probe card used that allowed probing of most of the internal chip area.

Three samples of chip E were probed, yielding significant chip-to-chip differences. The sample with the largest skew was measured more extensively. Fig. 8 shows the 70-ps skew measured at nominal voltage on this chip, at the locations shown in Fig. 9. To investigate the source of this skew, V_{dd} was reduced until the buffer delays doubled. This resulted in a doubling of the measured skew, implying that the skew is primarily due to buffer delay variations, rather than errors in the interconnect model or interconnect process variations. This is consistent with the expectation that in a well-designed global clock network, across-chip transistor channel-length variations are the primary source of skew. Two other samples of chip E showed 22

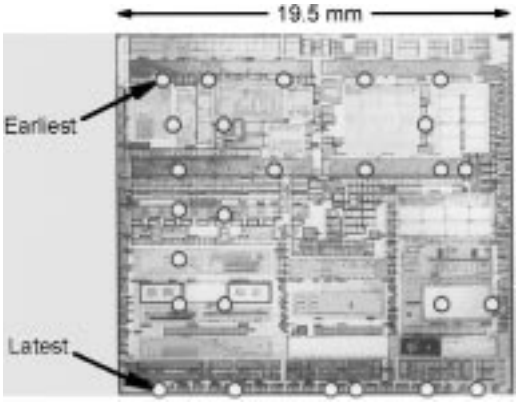

Fig. 9. Chip E with 26 circled locations probed for waveforms in Fig. 8.

and 40 ps of skew, respectively, when measured at the four corners and chip center.

IV. Discussion

One key difference between this network and other recently published work [11] is the choice of using a single grid. This choice of a single grid works well only if all the trees have very similar delay. If the trees all driving the single grid are not well tuned to have the same delay, or if there are significant model errors or process variations, then different parts of the grid will try to switch at different times. If the delay differences are large enough, then sector buffers may fight each other, causing large currents and poor signal quality. Monte Carlo simulations show that the skew within a particular small area of the chip is likely to be smaller if the grid is partitioned such that a small independent grid drives that small area.

Many simulations of process and model errors as well as hardware measurements were performed to guide the decision whether to use a single grid. The problem of adjacent sector buffers switching at different times is less severe than it might at first appear when using this strategy. Note that adjacent sector buffers are not connected together directly by the grid wires, but are isolated from each other by the final tree wires. This contributes to the fact that even with large process variations that cause significant skew across the chip, the resulting local skew is relatively small. This is because the switching time of any single point on the grid results from a weighted average of the switching times of a number of nearby sector buffers.

While the choice of a single grid works well for the chips described here up to the gigahertz range, this may not be feasible choice in the future for large chips running at much higher frequencies.

V. Conclusion

A tunable, efficient, yet robust global clock distribution strategy has been successfully implemented and measured on several high-performance microprocessors consisting of a single grid driven by a number of tunable trees. A "divide and conquer" tuning method was developed for these very large networks, achieving rapid turnaround and very low skew despite nonuniform loads and other inevitable asymmetries.

While more asynchronous techniques may soon be required for the largest high-frequency chips, the simplicity and robustness of synchronous clocking will continue to be attractive, at least for significant portions of large chips. Thus, the need for large low-uncertainty global clock networks will continue.

Acknowledgment

The authors would like to thank P. Cook, A. Deutsch, A. Ruehli, B. Rubin, S. Walker, J. S. Neely, T. Bucelot, and M. Rosenfield of the IBM Research Division, as well as D. Hoffman and C. Anderson of the IBM Enterprise Systems Group, for many valuable interactions.

References

[1] D. W. Dobberpuhl *et al.*, "A 200-MHz 64-b dual-issue CMOS microprocessor," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1555–1567, Nov. 1992.

[2] G. Northrop *et al.*, "A 600-MHz G5 S/390 microprocessor," in *ISSCC Tech. Dig.*, Feb. 1999, pp. 88–89.

[3] A. G. Aipperspach *et al.*, "A 0.2-μm 1.8-V SOI 550-MHz 64-b PowerPC microprocessor with copper interconnects," *IEEE J. Solid-State Circuits*, vol. 34, pp. 1430–1435, Nov. 1999.

[4] T. McPherson *et al.*, "760-MHz G6 S/390 microprocessor exploiting multiple V_t and copper interconnects," in *ISSCC Tech. Dig.*, Feb. 2000, pp. 96–97.

[5] B. McCredie *et al.*, "A 1-GHz Power4 test chip design," presented at the Hot Chips 11, Symp. High-Performance Chips, Palo Alto, CA, Aug. 1999.

[6] P. Hofstee *et al.*, "A 1-GHz single-issue 64-b PowerPC processor," in *ISSCC Tech. Dig.*, Feb. 2000, pp. 92–93.

[7] C. Webb *et al.*, "A 400-MHz S/390 microprocessor," *IEEE J. Solid-State Circuits*, vol. 32, pp. 1665–1675, Nov. 1997.

[8] A. Deutsch *et al.*, "When are transmission-line effects important for on chip interconnections," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1836–1846, Oct. 1997.

[9] P. J. Restle, K. A. Jenkins, A. Deutsch, and P. W. 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.

[10] P. Restle, A. Ruehli, and S. G. Walker, "Dealing with inductance in high-speed chip design," in *Proc. DAC*, Jun. 1999, pp. 904–909.

[11] S. Rusu and S. Tam, "Clock generation and distribution for the first IA-64 microprocessor," in *Proc. DAC*, June 1999, pp. 904–909.

Phillip J. Restle (M'87) received the B.A. degree in physics from Oberlin College, Oberlin, OH, in 1979, and the Ph.D. degree in physics from the University of Illinois, Urbana, in 1986.

He joined the IBM T. J. Watson Research Center, Yorktown Heights, NY, in 1986, where he worked on CMOS parametric testing and modeling, CMOS oxide-trap noise, package testing, and DRAM variable retention time. Since 1993, he has concentrated on tools and designs for VLSI clock distribution networks contributing to IBM's servers and high-performance ASIC designs. He holds four patents, has written 15 papers, and has given several invited talks and tutorials on high-frequency on-chip interconnects.

Dr. Restle received IBM awards for the S/390 G4 and G5 microprocessors in 1997 and 1998, and for the invention of a high-performance VLSI clock distribution design methodology in 2000.

Timothy G. McNamara received the B.E. degree in electrical engineering from the State University of New York, Stony Brook, in 1983 and the M.S. degree in computer engineering from Syracuse University, Syracuse, NY, in 1990.

Since joining IBM in 1983, he has designed high performance clock systems for IBM's large bipolar mainframes and CMOS servers. He is currently a Senior Engineer in S/390 Custom Microprocessor Design. He currently holds eight U.S. patents and has authored or co-authored a number of technical papers.

David A. Webber received the B.S. and M.S. degrees in electrical engineering from the University of Illinois, Urbana, in 1981 and 1983, respectively.

He joined IBM in 1983 in Hopewell Junction, NY, working on the design of automated test equipment. Since 1993, he has been working on circuits, clocking and integration for S/390 CMOS microprocessors. He is currently an Advisory Engineer in IBM S/390 Microprocessor Development.

Peter J. Camporese received the B.S. degree in electrical engineering from the Polytechnic Institute of New York, Brooklyn, NY, in 1988 and the M.S. degree in computer engineering from Syracuse University, Syracuse, NY, in 1991.

He joined the IBM Data Systems Division, Poughkeepsie, NY, in 1988, where he has worked on system performance, circuit design, physical design, and chip integration. He is currently a Senior Engineer in the IBM Server Division and a Technical Leader for the physical design and integration of CMOS zSeries microprocessors. He holds ten U.S. patents and is coauthor of several papers on high-speed microprocessor design.

Kwok F. Eng received the B.S.E.E. degree from The Cooper Union, New York, in 1978.

After a year of study for the M.S.E.E. degree, he joined IBM, Kingston, NY, in 1979, where he has worked on device characterization, circuit design, physical design, chip design, and EDA tools. He is currently an Advisory Engineer in charge of physical design tools for the z-series chip design group for IBM, Poughkeepsie, NY.

Keith A. Jenkins (SM'98) received the Ph.D. degree in physics from Columbia University, New York, NY, for experimental work in high-energy physics.

He is currently a Senior Engineer at the IBM Thomas J. Watson Research Center, where he is a member of the Communications Technology department. At the IBM Research Division, he has done research in a variety of device and circuit subjects, including high-frequency measurement techniques, electron-beam circuit testing, radiation-device interactions, low-temperature electronics, and SOI technology. His current activities include evaluation of the performance of VLSI circuits and phase-locked loops and investigations into substrate coupling in mixed-signal and RF circuits.

David H. Allen received the B.S. degree in electrical engineering from the University of Kansas, Lawrence, in 1980, and is currently working toward the Master of Computer Engineering degree at the University of Minnesota, Minneapolis.

He was involved in the design of high-speed memory, logic, and custom circuits at Micron Technology, Boise, ID, VTC, Bloomington, MN, and Intel, Portland, OR, before joining IBM's Enterprise Server Group, Rochester, MN, in 1992. Since then, he has developed custom 64-b PowerPC processors for AS/400 and RS/6000 servers, and has been involved in developing custom circuit techniques, clocking and latch design, and processor design methodology. Currently he is leading the implementation of a $>$1-GHz PowerPC processor design in SOI technology.

Michael J. Rohn received the B.S. degree in electrical engineering from University of Illinois, Urbana-Champaign, in 1984.

He joined IBM, Endicott, NY, in 1984, where he worked on system packaging analysis, bus design, chip electrical analysis, and system-level noise analysis on various system-370 products. Additional responsibilities included clock distribution definition at both the chip and system level. In 1989, he joined IBM's AS/400 division, where he has been responsible for library circuit designs, chip electrical analysis, bus analysis, and physical design team lead on four generations of 64-b PowerPC processors.

Michael P. Quaranta received the B.S. degree in computer engineering from Lehigh University, Bethlehem, PA, in 1985.

He joined IBM's Data Technology Division in 1985 where he worked on bipolar gate array technologies. In 1992, he joined IBM's AS/400 division, where he was responsible for physical design methodology for several custom 64-b PowerPC processors. Currently he works as an Advisory Applications Engineer in IBM's Microelectronics Division.

David W. Boerstler (M'98) received the B.S. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, in 1978 and M.S. degrees in computer engineering and electrical engineering from Syracuse University, Syracuse, NY, in 1981 and 1985, respectively.

Since joining IBM in 1978, he has held a variety of assignments, including the design of high-frequency PLLs for clock generation and recovery, low-skew clock distribution network design, fiber-optic transceiver and system design, and other analog, digital, and mixed-signal bipolar and CMOS circuit design projects. He is currently a Research Staff Member with the High-Performance VLSI Group at the IBM Austin Research Laboratory, Austin, TX. His current research interests include high-frequency synchronization techniques and signaling approaches for high-speed interconnect.

He has received IBM Outstanding Technical Achievement Awards for his work on the design of the serializer/deserializer for the ESCON fiber-optic channel products and for the clock generator design for IBM's 1-GHz PowerPC microprocessor prototype. He has also received numerous IBM Invention Achievement and Technical Group Awards and is an IBM Research Division Master Inventor.

Charles J. Alpert (M'96) received the B.S. and B.A. degrees from Stanford University, Stanford, CA, in 1991, and the Ph.D. degree in computer science from the University of California, Los Angeles, in 1996.

He is currently a Research Staff Member with the IBM Austin Research Laboratory, Austin, TX. He has served on the Technical Program Committees for both ISPD and ICCAD. His research interests include placement, interconnect planning, clock distribution, and global routing.

Dr. Alpert received a Best Paper Award at both the 1994 and 1995 ACM/IEEE Design Automation Conferences.

Craig A. Carter (M'83) received the B.S. and M.S. degrees in electrical engineering from the University of Idaho, Boise, in 1983 and 1985, respectively.

He was an ASIC Design Engineer at AMI, Pocatello, ID, and Seattle Silicon, Bellevue, WA, and a CAD Engineer at Micron Technology, Boise, before joining IBM, Austin, TX, in 1997. He is currently an Advisory Engineer and member of the chip integration team for the Power4 processor.

Roger N. Bailey received the B.S. degree in electrical engineering from the University of Washington, Seattle, and the M.S.E.E. degree from the University of Illinois, Urbana-Champaign, in 1977.

He joined IBM, Rochester, MN, in 1977, where he worked on attaching peripherals to the S/38. In 1980 he transferred to IBM, Austin, TX. As a member of the original workstation team, he worked on every processor from the RT, Power to Power4 processors. He has lead a variety of groups from logic design to, most recently, the integration team for the Power4 processor. He is currently with Lucent Technologies, Austin, TX, in the Network Processor Division as a Senior Technical Staff Member.

John G. Petrovick received the B.S. degree in electrical engineering from Arizona State University, Tempe, in 1983, and the M.S. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 1985.

He joined IBM Microelectronics in 1985, where he was responsible for circuit design and physical design methodologies for ASIC products. Since then he has held a variety of positions related to the physical design and design methodologies of custom X86 and RISC microprocessors. He is currently a Senior Technical Staff Member in the Power4 Custom Microprocessor Design Group, Austin, TX.

Byron L. Krauter received the B.S. degree in physics and mathematics and the M.S. degree in electrical engineering from the University of Nebraska, Lincoln, in 1976 and 1978, respectively. He received the Ph.D. degree in electrical engineering from the University of Texas, Austin, in 1995.

He has been with IBM since 1979 and is currently working in a VLSI CAD tools development group in Austin, TX. His research interests include sparse boundary element method (BEM) techniques for inductance and capacitance extraction and on-chip interconnect analysis.

Bradley D. McCredie (M'92) received the B.S., M.S., and Ph.D. degrees from the University of Illinois, Urbana-Champaign.

He joined IBM, East Fishkill, NY, in 1991. His first assignment with IBM was in package design and analysis. In 1997, he moved to Austin, TX, to lead the Power3 Microprocesser Package Design Team. He is currently the Chief Engineer of the Power4 design team.