

# Chip integration methodology for the IBM S/390 G5 and G6 custom microprocessors

by R. M. Averill III  
K. G. Barkley  
M. A. Bowen  
P. J. Camporese  
A. H. Dansky  
R. F. Hatch  
D. E. Hoffman  
M. D. Mayo  
S. A. McCabe  
T. G. McNamara  
T. J. McPherson  
G. A. Northrop  
L. Sigal  
H. H. Smith  
D. A. Webber  
P. M. Williams

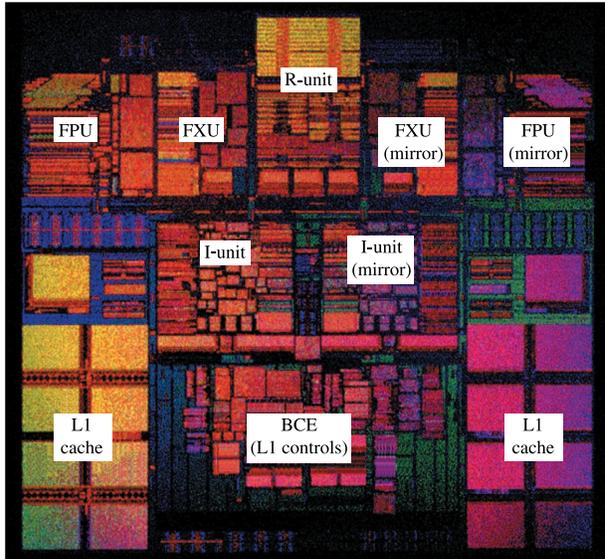
**High frequency processor designs operating at more than 500 MHz and of significant architectural complexity require custom physical design constraints from the inception of the design. New technology introductions such as copper interconnect wiring improve performance but also add complexity in ground rules and wiring. A concept known as chip integration, which includes a combination of critical physical design techniques such as floorplanning, power distribution, high-speed clock design, wiring methodologies, circuit macro floorplanning, chip-level timing/extraction, noise prevention, electrical analysis, design verification, and time to market, is given prioritized design consideration throughout all phases of the implementation. This concept is a key requirement necessary to achieve high frequency of operation, meet area targets within the defined architecture, and ensure a robust and reliable design for transistor counts from 10 to 100 million.**

## Introduction

The IBM S/390\* G5 microprocessor is based on the G4 [1, 2] microprocessor, with the addition of several architectural performance improvements [3]. With the benefit of a density improvement in CMOS technology from a 0.5- $\mu\text{m}$ -linewidth technology to the CMOS 6X 0.25- $\mu\text{m}$  technology, the G5 microprocessor is 28% smaller than the G4 version, with a 3X increase in transistor count. The size of the G5 microprocessor is 14.6 mm  $\times$  14.7 mm, with a total of 25 million transistors, 18 million for the array section and seven million for the logic section. The G6 is a remapping of the G5 processor into CMOS 7S 0.225- $\mu\text{m}$  technology with copper back-end-of-line (BEOL, an industry term for the final stage of interconnect wiring).

The G5 micro consists of several units which form the core central processor (CP) of the system. These functional units are the instruction unit (IU); the branch control element unit (BCE), which contains the cache and its associated controls; an execution unit (EU), which contains both the fixed-point unit (FXU) and the floating-point unit (FPU); and a register checkpoint unit (RU), used for full error checking and processor recovery.

©Copyright 1999 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.



**Figure 1**

Micrograph of the S/390 G5 microprocessor.

Unique to this architecture is the concept of mirrored IU and EU, which required careful floorplanning and area utilization by the chip integration team. A micrograph of the G5 processor is shown in **Figure 1**.

### Overview of methodology

Once a pipeline is established and the initial architecture is defined, the corresponding floorplan is begun in parallel with the detailed logic design. As suggested by feedback from the original floorplan, architectural tradeoffs are given high consideration, because physical design parameters greatly influence several metrics (area, timing, power, and noise) that are necessary to achieve high-frequency microprocessor operation.

The chip integration tasks consist of several phases throughout the design cycle. Before any detailed physical design can commence, floorplanning of the functional units, with consideration of critical signal wires, is analyzed to ensure that the key design objectives are met. Many initial physical design tasks such as power distribution, clock distribution, I/O wiring, and critical signal wiring are completed in advance. Careful planning of the macro-level custom bit stacks establishes the cycle-time goals and the placement and size of the various units. Robust design methodologies [4, 5] are practiced throughout the design cycle to ensure that hierarchical concepts are managed correctly. Timing methodologies, including a detailed extraction process, are followed to ensure high-frequency timing closure. Noise-avoidance

techniques are practiced both by the macro designers and the integrators, with the results going through thorough noise analysis. Final chip electrical checking ensures that power, voltage drop, and electromigration issues are addressed and within the desired specifications. Full-chip verification ensures that proper ground rules are followed, and schematic-to-physical convergence is obtained to allow manufacturability of complex designs. This paper focuses on these techniques, describing how to successfully assemble a complex microprocessor and ensure that proper analysis is done to verify the integrity of the design.

### Chip integration design issues and floorplan convergence

Initial floorplanning of high-performance microprocessors requires careful evaluation of several design points. As expected, floorplans typically evolve over time as the specifics of architecture and logic function are defined. Many factors can influence the progress of a floorplan to closure, with some aspects of the design dominating and taking precedence over others. Size and aspect ratio of the chip, technology choice, available levels of metal, local interconnect availability, power, cycle time, architectural design point, time to market, quantity of I/Os, voltage budgets, clock skew budgets, electromigration, noise budgets, circuit styles, and local cache access times are just some of these factors. Usually the architectural design point is chosen and verified by using informal floorplan budgets to check for initial feasibility. Subsequent analysis then refines the ultimate criteria for the chip.

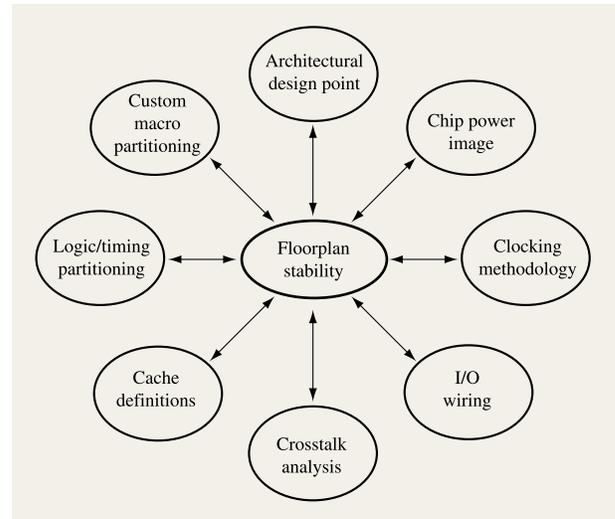
Chip integrators work closely with the chip architects during the initial floorplanning stages. Floorplanned objects such as functional units, local caches, and I/O blocks should be analyzed for aspect-ratio and size tradeoffs. Chip size and aspect ratio may be flexible, but these dimensions are generally dictated by packaging and/or cooling requirements. All global buses and critical control signals should be identified with first-pass cycle partitioning. This identifies all first-pass, timing-critical buses and signals. Placement of functional units and local caches should be made to minimize all critical bus lengths as much as possible. Various cache sizes and interleaves are usually evaluated during this time to maximize silicon utilization. In parallel, all cycles should be cross-sectioned and verified through a circuit simulator. Small refinements in latch boundaries can be adjusted during the floorplanning process, but cross sections are necessary to verify that macro partitionings of bit stacks are correctly designed, with full custom design techniques. This usually requires the largest portion of the total design time, so early accurate partitioning is required. Factors contributing to a stable floorplan are shown in **Figure 2**.

Chip floorplanning for the G5 CP chip was implemented as a hierarchy of three levels: The top level consists of CP floorplannable blocks; the second comprises the functional units, which are collections of macros that perform certain functions (i.e., FPU); and the third consists of the macros themselves, each of which performs a certain action (i.e., adder, multiplier, etc.). All wiring was accomplished concurrently. This allowed for designs in the various units in the CP to be wired in parallel for earlier timing convergence. Time to market, cycle time, clock skew budgets, and noise avoidance were the primary design criteria for this design. There was some flexibility in chip size, which allowed for optimum unit sizing and aspect ratio selection. This in turn reduced wiring lengths between units on critical buses and signals. Functional units were placed at this time to meet the above criteria as well as possible. After several iterations, the chip and unit dimensions usually solidify.

Three nonfunctional floorplannable objects—chip image, clocking wires, and I/O wires—must be added to the CP floorplan when the first-pass analysis of the plan is performed. First, the chip image, which consists of the CP power image of the top two or three levels of power and I/O (or C4) locations, is added to the floorplan. Once the chip image is placed, the clocking wires are analyzed to determine worst-case size requirements and are added to the plan. The I/O wires are placed last of the three, and before any other signal wires, because of their size. These three objects are placed in this order because the chip image has the least flexibility, while the I/O wires have the most. These objects are usually abstract representations of the finished product and are deliberately chosen to be worst-case representations in terms of their demand on resources. This helps to minimize subsequent reworking of the floorplan.

The CP chip image defines the chip active area, or design space, and is the basis for all of the remaining chip integration tasks. Included are the signal and power C4s and the power distribution grids on M4 (metal layer 4), MQ (metal layer 5), and LM (metal layer 6), which are continuous across the chip. After C4 I/O assignment is done, the unused signal C4s are converted to dummies, and their contact pads are eliminated. The power distribution grid is symmetrical about the center y axis of the design, which simplifies floorplanning because of the mirrored units and odd/even caches. Structures required by manufacturing, such as the chip guard ring, are also included in the image data. (The purpose of the guard ring is to provide a low-resistance path to ground for surge currents and a metal seal against ionic contaminants.)

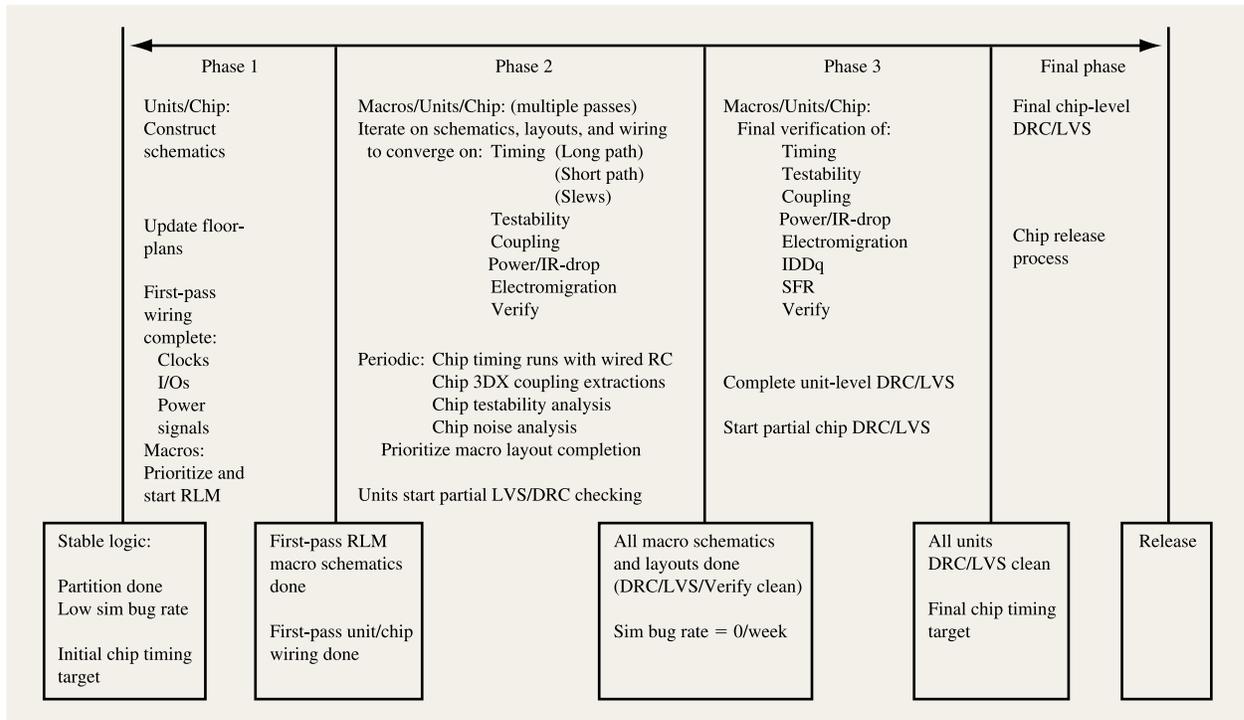
Once the floorplan has been established, a strategy must be developed to handle the convergence of several additional factors. The G5 microprocessor has followed



**Figure 2**

Floorplanning stability criteria.

the same basic strategies as its G4 predecessor. Since most of the development is done concurrently, a plan was developed to handle the multiple requirements imposed on the design. The more important requirements include frequent modification of the logic design arising from cycle simulation, timing convergence, noise avoidance, layout-to-schematic (LVS) convergence, design-rule checking (DRC), and floorplan evolution. The sequential process from concept to manufacturing release is divided into several “wiring snapshots,” which define intervals of time within which a functional unit will synchronize with its logic changes. Since the logic is effectively “frozen” during this time frame, units are free to work to a “stable” design point. Several overall timing updates are obtained during a snapshot and fed back to each functional unit. Unit designers are free to pick additional logic changes at convenient points should the scope of the change dictate immediate action. The final snapshot assumes very-close-to-stable logic and timing, with emphasis on final design checking and analysis. The emphasis, previously focused on floorplanning, logic, and timing, now shifts to LVS and DRC completion, together with CP-level final checks, to complete all required physical design verification for release to manufacturing. This strategy has been demonstrated to work for a moderate number of functional logic changes coupled with timing changes to achieve a cycle time of 500 MHz for G5 and 670 MHz for G6. A flow diagram for the final phase of the chip integration process is shown in **Figure 3**.



**Figure 3**

Chip integration final release flow.

### Power distribution

As operating voltages decrease to support CMOS scaling at submicron dimensions, and on-chip current loads increase or at best remain constant because of increased clock frequencies, more stringent demands are placed on the chip power-distribution system. This is because voltage tolerances must be reduced proportionally with supply voltage in order to support chip performance objectives. Therefore, the primary function of on-chip power distribution is to provide a supply voltage with minimal  $V_{DD}$ -ground drop from C4 contact to circuit port. Equally important is the need to minimize the voltage difference between any two circuits on the chip, since this condition has a direct impact on the circuit noise margins. This voltage gradient is usually termed spatial delta-V.

These two conditions are used as electrical design constraints on the power-distribution system. Another constraint is electromigration concerns, associated primarily with the via density of the power grid. Electromigration must be considered early in the design, since via density demands will dictate wider wire widths under certain loading conditions. If such problems are found late in the design cycle, significant rework of the chip floorplan and net topologies may jeopardize a timely

delivery. Therefore, it is important to evaluate these constraints early, under adverse conditions, with a “correct-by-design” power distribution.

To quantify these effects, a dc model is used as a design tool to specify wire widths, periodicity, and via density options. Customized code is used to automatically generate a resistive-mesh model for a given grid design point. The grid design point is not arbitrary. The lower-level grids (M1–M2) are specified by circuit layout considerations, while upper levels (M4–MQ–LM) are more flexible in wire width and periodicity options. Given the nature of the C4 chip attachment process, it can be assumed that power from the package is delivered evenly across the chip area, and that all power and ground C4s are at the same potential. Given this assumption, the maximum and minimum voltage drops are contained within the boundary of adjacent  $V_{DD}$  and ground C4s, as shown in **Figure 4**. This simplification reduces model complexity significantly, making it possible to perform circuit simulations within minutes for a particular design point and loading condition.

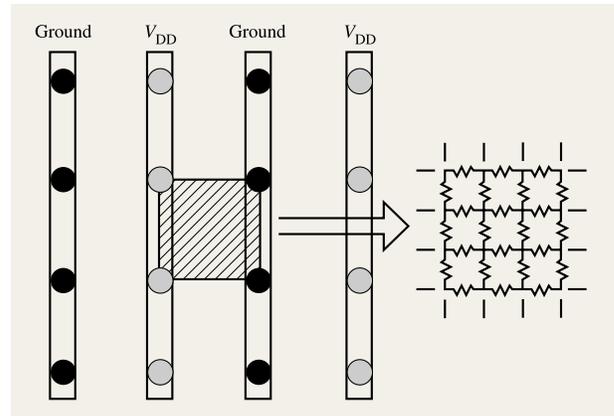
Since loading conditions are difficult to predict, two configurations are considered—the uniform load and the “hot-spot” load. In both cases, the current density is

computed using assumptions for chip power and size of active area. (It should be pointed that chip power is difficult to predict during the early stages of any design that includes major micro-architecture updates. Since G5/G6 was a remapping of G4 with minor architecture updates, power consumption was extrapolated from hardware measurements.) Using this information equivalent, current sources are placed within the model, either in a uniform manner or clustered in a worst-case location from  $V_{DD}$  to ground along the M1 strips. The differential  $V_{DD}$ -ground voltage is monitored at all observable points along the M1 strips to assess the voltage drops and spatial delta-V. The currents through all resistive elements within the model that include vias are also monitored and compared to electromigration specifications for additional consideration. This process is iterated for several design points until one satisfies the constraints without significantly affecting wiring resources.

The assumption of using a dc current to represent a CMOS load is not accurate in a general sense. Under steady-state conditions in the presence of significant decoupling capacitance (including both added and intrinsic), the dc assumption is reasonable. However, under transient conditions, ac noise dominates. This noise is very difficult to predict during the early design stage, since ac load models must be developed and coordinated within the instruction sequence of the processor, which is not well known in the early phases of the design. (This problem is evaluated in more detail during the post-physical-design-verification phase of the design, as described in the section on electrical analysis.) Given this effect, voltage drops predicted by the dc model are limited to approximately 15–20 mV for a given design point.

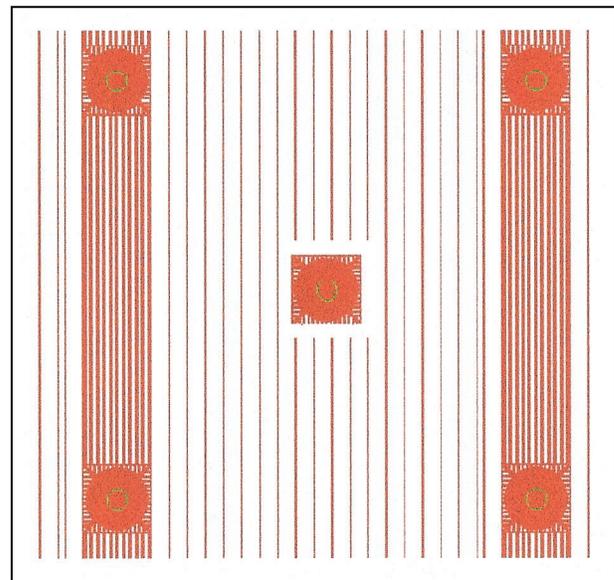
The signal-to-power ratio within a given bay defined by adjacent  $V_{DD}$  and ground stripes is an additional consideration for the reduction of signal coupling noise in the power distribution. As described in the section on noise containment, “quiet” wiring tracks are used in conjunction with power rails to shield signal lines from other nets. This is particularly true on the LM layer, where C4 rails would suffice for power delivery; however, because of coupled noise concerns, a significantly reduced signal-to-power ratio was implemented on this layer.

The on-chip power distribution starts with the use of C4 distribution pads that run vertically, starting with ground and alternating between power and ground. The C4 pad spacing is  $450\ \mu\text{m}$  in each direction for both power and ground. Wide vertical-last metal buses distribute power. Interstitial-last metal lines, shown in **Figure 5**, help with controlling inductance and were added between the wide buses that also alternate between power and ground. The interstitial spacing ensures a maximum of 5:1 signal-to-power ratio on all last-metal wiring. The power distribution network is a continuous grid over the entire



**Figure 4**

DC power distribution model with  $V_{DD}$  and ground C4s.



**Figure 5**

G6 C4 power distribution shown with interspersed copper LM interstitials. “Cheesed” power LM stripes shown with narrow spacings on left and right sides of figure.

chip for M4, MQ, and LM to ensure minimal voltage-drop differences across the chip. The M1, M2, and M3 power is also a grid structure but is not continuous over the chip (**Figure 6**). It consists of islands over the macros and is connected to the M4 above. The area between the macros is used for signal wiring. Electrical analysis was done via extensive simulation to ensure a robust and reliable power

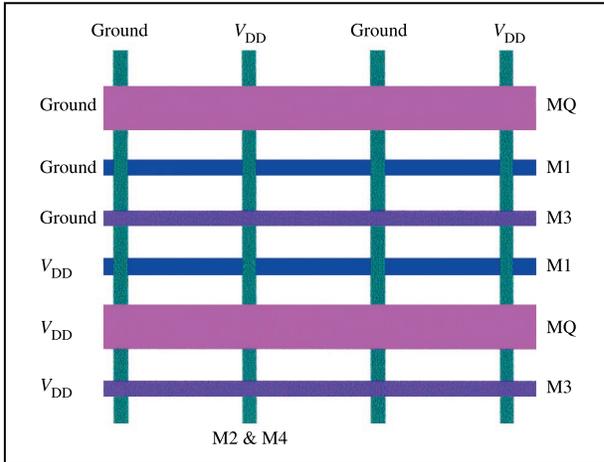


Figure 6

G5/G6 power grid.

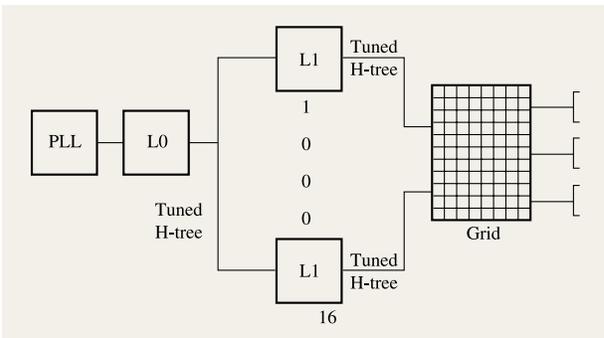


Figure 7

Clock hierarchy distribution.

network while maintaining adequate routing resource to wire the chip.

### Clock distribution

The first component used for CP clock distribution is a centrally located phase-locked loop (PLL) that multiplies a slow external reference frequency by a factor of 8 to generate the high-frequency (500 MHz for G5) global clock (CLKG) that is distributed throughout the CP chip.

Two levels of buffers and a combination of H-trees and a grid distribute the clock throughout the chip, as shown in **Figure 7**. The final level of distribution is a set of local connections from the grid to clock pins on the macros.

Routing of the clock signal within the macro is under

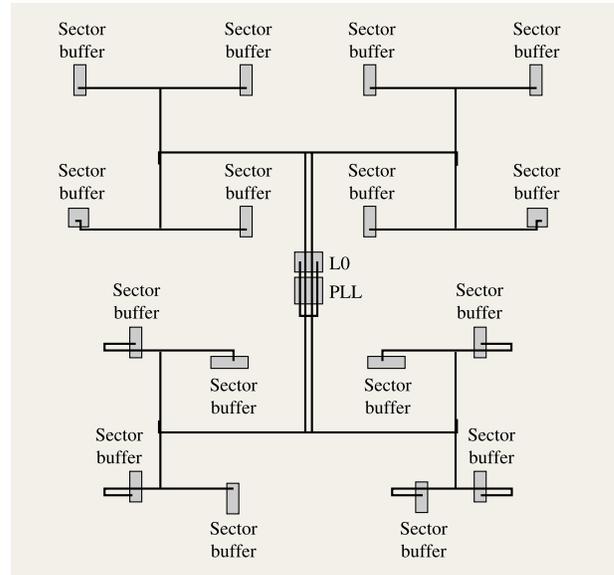


Figure 8

PLL clock distribution via H-trees to the 16 L1 sector buffers.

control of the macro designer and is verified by extraction and timing. The primary requirement for the clock distribution was to deliver a global clock to all of the macros with minimal skew and delay as well as to maintain similar clock waveforms across the chip [6]. By having all clock splitters receive CLKG signals with similar waveshapes, the gate delay of the receiving gates should also be similar and predictable. If the CLKG waveshapes vary, the gate delays, which are not modeled by the timing methodology, will also vary.

Since the clock distribution is a global function and covers the entire chip, space was reserved for the buffers and the distribution network early in the design process. For the clock distribution, the chip was divided into 16 sectors, with each sector having its own buffer (sector buffer). Ideally, the sector buffers would be located in the center of each sector, but floorplanning requirements forced several sector buffers to be offset from the sector center.

The PLL and central clock buffer are both located near the center of the chip. An H-tree distributes a low-skew clock signal from the central buffer to each of the 16 sector buffers, as shown in **Figure 8**. The H-tree is generated by hand, and the wire lengths and widths are chosen to obtain minimal skew and delay. The H-tree was built on the top two levels of metal, which allowed low-resistance nets. Each wire in the H-tree was shielded at the sides by power and ground, and the main trunk and

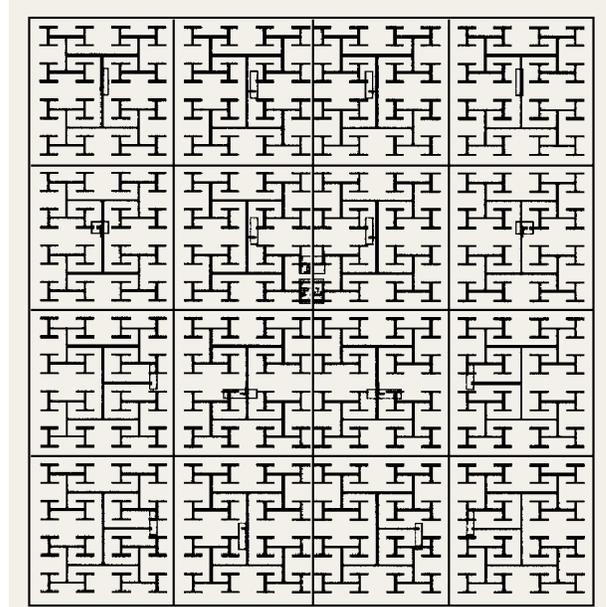
first branch of the first-level H-tree was shielded below. This carefully designed shielding and routing of the H-trees resulted in predictable capacitive and inductive characteristics. The shielding also reduced noise coupled into nearby signal lines from the clock lines. A proprietary clock tuning and optimization program was used to adjust the widths of the H-tree wires to obtain a clock signal at the sector buffer inputs that had minimal skew and delay with good rise and fall transition times. Distributed RLC models were used for simulation of the H-tree wires throughout the tuning process and during final clock verification. Simulation of the first-level H-tree showed a mean delay of 450 ps with 2.9 ps of skew at the sector buffer inputs.

The sector buffers have three stages of inverters. Surrounding the inverters are banks of decoupling capacitors to reduce the delta-I noise generated by the clock switching.

Each sector buffer drives an H-tree, which in turn drives the intersection points of the grid, as shown in **Figure 9**. The grid consists of 32 horizontal wires and 32 vertical wires across the entire chip (**Figure 10**). The spacing of the wires was approximately  $0.45 \text{ mm} \times 0.45 \text{ mm}$ , with vias connecting the wires at each of the 1024 intersection points. The individual second-level H-trees are routed identically except for the connection from the sector buffer to the center of the H-tree, where some variation was required to account for the fact that not all of the sector buffers could be placed in the center of their respective sectors. Like the H-trees, the grid is shielded on both sides to reduce the coupling of noise into nearby signal wires. On one side, the shield is provided by placing the grid wire adjacent to a power or ground bus. The other side is shielded by a wire that is connected to the ground. The grid connects all of the sectors into one common network.

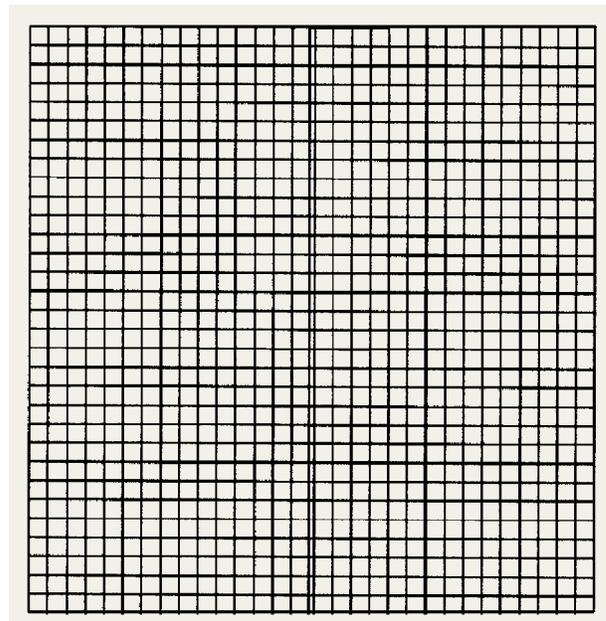
Each macro contains one or more clock splitters that are connected to the grid by local wiring. Guidelines were developed and software written to verify that the connection between the macro and grid would have a delay of less than 5 ps. In fact, almost all connections were made with less than 3 ps of delay. Initial tuning of the clock distribution ignores the delay associated with local clock wires but includes the capacitive load from those wires. Final simulations include the local clock wires and give waveforms and measurements at the macro input pins.

The clock tuning program is again used to adjust each of the 16 second-level H-trees to obtain a mean delay and minimal skew. The clock loads are extracted from the chip and clustered at the endpoints of the H-trees. The grid is removed and modeled as a lumped capacitance at the endpoints of the H-trees. The tuning process consists of multiple runs of the clock tuning program to adjust the widths of the wires to optimize skew and delay. The final



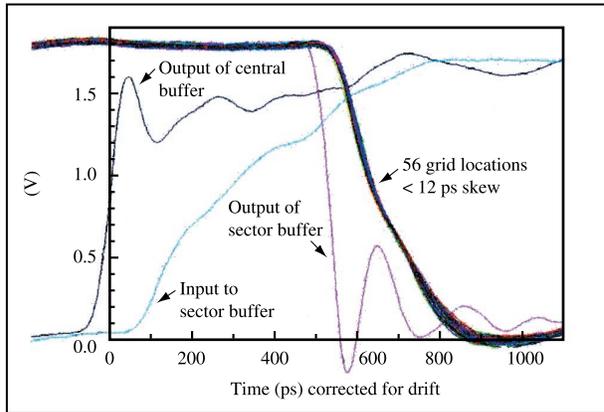
**Figure 9**

Sector buffers driving H-trees which drive grid intersection points.



**Figure 10**

Clock grid.



**Figure 11**

Grid clock skew < 12 ps of 56-picoprobe measured waveforms.

step in tuning is to simulate with the grid connecting all of the second-level H-tree endpoints. The results will be close to the final solution. Simulations with this grid and with an “unconnected” grid show the value of the grid approach. The skew on the grid was reduced from almost 75 ps to 15 ps by “shorting” all 1024 drive points together with the grid. By observing the skew distribution across the chip, manual tuning of the second-level H-trees improved the simulated skew to 11.2 ps on the grid. Final verification of the clock distribution network was a simulation of the second-level H-trees, the grid, the interconnection wires to each macro pin, and the macro load. The result of this simulation for typical conditions was a mean delay of 319 ps with a skew at the macro pins of 10.2 ps.

Simulation has shown that the grid-tree approach is effective in reducing the sensitivity of the clock distribution to ac load variation (ACLV) and skew in the first-level H-tree. Another benefit of the grid, as shown by simulation and experiments, is its insensitivity to load variation. This attribute was extremely critical in reducing the time to market, since changes that affected a macro’s clock distribution had little or no effect on the global clock distribution network of H-trees and the grid. In addition, floorplanning of the clock distribution network at the chip level also allowed the unit-level floorplan to change several times throughout the design cycle without requiring regeneration of the clock distribution.

To verify the clock distribution, a series of delay measurements were made on a CP chip. The chip was powered on and held in a state with all of the clock splitters off in order to reduce power consumption. A pulse generator provided an external reference clock. One picoprobe was placed on the clock grid to provide a

“reference” to correct against drift. A second picoprobe was moved around the clock grid, and a series of measurements was taken, along with a measurement of the reference point. With very careful positioning of the picoprobes to ensure good electrical contact, 56 points were measured around the chip. After correction for long-term drift in the measuring equipment, the 56 points showed less than 12 ps of global CLK skew on the grid (**Figure 11**). Although significant distortion was introduced by the picoprobes, the delay measurements are valid because all spots on the grid have similar waveshapes. In a system environment with 500-MHz clocks, power-supply noise and thermal gradients act as additional sources of skew. However, the measurement of 12 ps of skew on the grid showed the effectiveness of the grid-tree approach to delivering a low-skew clock.

For the G6 processor, the clock distribution was designed to support a clock frequency greater than 600 MHz. At this frequency, the two-level clock distribution no longer provided good rise and fall transitions at the inputs to the sector buffers. An additional level of buffers (quadrant buffers) was added between the central buffer and the sector buffers. With minimal floorplanning changes, the four additional quadrant buffers were added in open areas on the chip, and the associated H-trees were designed.

### Hierarchical methodology: Contract generation

As previously mentioned, one of the major benefits of hierarchical design is the ability to do concurrent design at all of the levels of the hierarchy. However, one item that must be properly managed is the utilization of wiring resources between the levels of hierarchy. If this management is not properly done, hierarchical design can actually become detrimental.

The methodology for management of wiring resources is based upon the concept of contracts. Contracts partition a common wiring resource between one level of the hierarchy and another, in order to designate which wiring tracks can be used. This might be done by a simple patterned agreement in which one level of hierarchy uses every third wiring track on a certain metal layer and the other level uses the remaining two out of three. This can lead to inefficiencies, since many times the tracks may not follow a set pattern or the tracks are not allocated in the ideal locations. Therefore, wiring contracts were developed from real wire utilization. These wiring contracts also take into account the accumulated effects of different types of predesigned structures, such as I/O wiring, power, and clock. A sample hierarchical wiring contract is shown in **Figure 12**.

Critical nets are given special consideration when deciding upon the use of wiring resources. Another major decision in the methodology was to make the chip symmetric about the center y axis. This allowed for

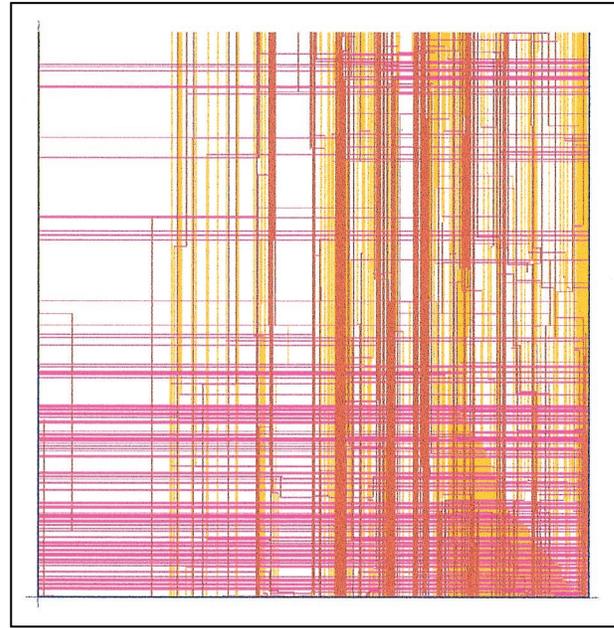
optimum use of wires on both sides of the chip in the dual-unit areas. The power grid and clock tree were also designed with symmetry from the center of the design. Crosstalk avoidance was incorporated into the wiring contracts, with spacing being added for long wires and allowances for changing wiring planes when deemed necessary. Late design changes to wiring that affect the chip, units, or macros were relatively easy to accommodate with this contract methodology, since all work could be done concurrently.

For wiring contract purposes, the three-level hierarchy previously described for the CP chip is condensed to two levels, a global level and a macro level. The global level is further subdivided into a chip, or CP, level and a unit level. The chip level contains units as well as chip-level macros, buffers, and I/O macros; the unit level contains macros that were designed with both custom and semicustom design techniques. Each unit has a set contract that is a vertical slice from the chip data, from C4 to silicon, encompassing the unit area. There are corresponding layout views wherever appropriate that allow for complete unit LVS/DRC checking.

Although contracts allowed both the chip and unit integrators to work with constraints, macro designers also need a way to pass constraints to the units. A concept called abstract generation is used to allow custom macro designers to pass “bottom-up” constraints to the units, and to allow unit integrators to pass “top-down” constraints to the semicustom designers, using a standard cell place-and-route process based upon completed layouts. This process also gives the unit integrators the capability of verifying that the macro layout designer did not violate the original macro wiring contract and pin positions.

Complex layout abstraction and blockage modeling is essential for achieving a “correct-by-design” error-free design approach. The design layout rules evolve at such a rapid pace that the “automatic routing” tools cannot handle many of the detailed shape-to-shape relationships and constraints. The abstraction process has two main purposes: data volume reduction and proper data modeling for the wiring tools. The latest copper BEOL technology layout rules create situations for which, without proper blockage modeling, it would be nearly impossible to achieve DRC-correct designs using automatic routing tools. Manual corrections for DRC violations are at times difficult or impossible to implement. Two examples in which abstraction techniques are used to produce “error-free” designs are as follows.

In the first example, the width of a wire (power, clock, I/O, etc.) is greater than some value defined in the layout rules. When this occurs, any adjacent wire must be kept some distance greater than the minimum spacing away. There are ranges of widths and spaces to which the design must adhere. The second example involves a “metal overlap



**Figure 12**

Sample hierarchical wiring contract.

past via” requirement for which the metal line is greater than a certain width and spaced less than some specified distance from an adjacent wire on the same metal layer.

The abstraction/blockage modeling code recognizes the above situations (as well as many others) and adds “artificial” blockages where required to guide the routing tools and ensure an error-free design.

### Chip-level wiring

Chip-level wiring was completed at two levels of hierarchy on the G5 microprocessor. The top level, or CP, was wired in conjunction with the unit wires, which were considered as wires above the macro level but below the CP.

Wiring these levels concurrently required the use of the previously mentioned contracts between macros and units, units and CP, power and units/CP, clocks and units/CP, and I/O and units/CP. Effective and early creation of these contracts was critical to completing the task at hand. Such contracts are nonmanufacturing layers that are designed as blockages or “keep-outs” for routers. As the G6 microprocessor is implemented, copper interconnects have special requirements that must be considered early in the design phase and make the contracts even more critical in the overall design process.

Minimization of wire length and macro-to-unit crossovers is important to timing convergence, and an accurate or slightly conservative RC global wire extractor

is necessary. This wire extractor should be able to handle a mix of actual and estimated wires. Estimated wires are used where there are no partially complete wire segments on a net. Usually the estimated RC calculated is more optimistic than its real counterpart. In timing-critical microprocessor designs, it is important to replace estimated with real wires to ensure the validity of the CP timing runs. Often, wiring congestion and blockages show that multiple wiring passes are required to converge on a timing-stable design.

Once first-pass placement of functional units occurs, a wiring-congestion analysis should be completed. To complete this analysis, the I/O wires, clock structures, and power grid must be designed to a point at which they can be analyzed. I/O wires from the chip should be handled separately, because these usually have special requirements. All multichip module (MCM)-mounted chips incorporate C4 solder ball technology. Only slightly more than half of the signal C4s were used on this design. Electrostatic discharge (ESD) requirements make I/O wires unique because of their size and length. ESD and output impedance limitations generally require these wires to be wider (three to four times the width of normal signal wire) and have reasonable lengths. The floorplan should accommodate I/O-macro placement to support these requirements. Generally, I/O-macro placements can be positioned uniformly along the edges of the floorplan and fanned inward toward the center. (Some exceptions to this will occur for C4s directly in the middle of the chip.) Fanning inward of I/O signals alleviates wiring congestion in the middle of the chip, where it tends to be the worst. This is compounded by the wider I/O signal wires.

The chip power and C4 structure image must be designed early in the floorplanning stages to stabilize functional unit placement. Since this is a uniform structure, its impact on the units is a factor with regard to "white space" between the units. CP wiring is typically done between and over units. The white space is the area between units used to complete the top-level global wires. Since these wires must sometimes be high-performance (low-delay) wires, thicker and wider top-level metal must be shared with the image. This could and usually does affect unit placement to a small degree. Other wires that can affect global wiring can be global preroutes, wire shielding, clock meshes, clock H-trees, and clock shielding.

Copper wiring is a new technology that was used in the G6 microprocessor. Special consideration must be given to chips that use this new back-end-of-line (BEOL) technology. The softness of this interconnect metal requires a reduction in the usability of maximum-width lines. Wide power buses have to be "cheesed"<sup>1</sup> to provide the local oxide densities necessary to prevent local

"dishing" at metal planarization steps. C4 final metal design requires special patterns to handle local I/O currents while maintaining local copper density requirements. Generally, wide shielding must be broken up into multiple smaller widths. As much wider lines are needed, such as I/O, clocking, and shielding, care must be used when routing them next to smaller-width wires. Often, increased spacing is desirable to increase yield on the part. These density requirements do not outweigh the benefits of copper wires, but they must be considered early in the design process.

### Floorplanning for bit stacks and standard cells

Bit stacks require special handling in high-performance microprocessors. Typically, several iterations of circuit simulation of cross sections and repartitioning of logic blocks in cooperation with the system architect will result in a well-defined pipeline, which typically comprises full-custom circuits. The unit integrator must then map out estimated dataflow macro sizes and do several types of wiring analysis:

- Vertical wiring resource analysis. The orientation of macro stacks usually dictates the "vertical orientation" of the dataflow. An analysis of vertical resource and bus mapping must be made so that all macro designers understand which shared layers of metal are available. Dataflow wiring, wire shifts, and lower-level vertical power striping must be defined. Macro cell size is usually the controlling factor for vertical-level power striping.
- Horizontal wiring resource analysis. Since most level-1 metal is dedicated to macro design, lower-level horizontal metal must be evaluated for control wiring, wire shifts, and horizontal power distribution.
- Dataflow macro estimates. Until the macro designers provide inputs for the area of their macros, dataflow macro size and aspect ratio can be estimated. Driving impedance on output buses, bus length and width, and receiver loads are factors that can affect macro sizes.

Standard cell floorplanning is performed somewhat differently from that for bit stacks. Dataflow circuits are typically full-custom (static with some dynamic) circuits, whereas control macros are derived from a library of "books." Given the row height of these cells, synthesis techniques can predict gate and book usage for a given macro. Depending on macro size, 25 to 50% additional area will be required for wiring in the floorplan. Pin placement is optimized to minimize wire length. This is not as important in the dataflow for vertical wiring, since pin placement is usually optimized by the horizontal orientation of dataflow macros on a given bit pitch.

Early partitioning of control macros into small-to-medium-size macros helps in their placement. The same

<sup>1</sup> *Cheesing* is a term used to represent the process of removing metal stripes from within a wide metal wire (i.e., metal with holes in it).

rules that apply to global wiring apply to wiring the control section of a functional unit. Extreme aspect ratios (greater than 5:1 or 6:1) should be avoided. They may be floorplanned correctly at the global level, but may present wiring challenges at the macro level.

### Chip timing and extraction

Achieving high frequencies in deep-submicron processor designs is a highly iterative process. In general, a methodology [7] that provides fast iterations with high predictability between iterations will allow the highest frequencies. During the G5 and G6 design cycles, iterative snapshots were made of the full chip timing including wire RC delays, and these were analyzed on a weekly basis. Global wires and macros were tuned with each iteration. Static transistor-level timing analysis was used to create macro timing abstracts containing records for all input-to-latch, input-to-output, and output-to-latch paths in a macro. The macro timing abstracts and wire data were then passed into a global timer for full unit and chip analysis. The unit and chip timing analysis created macro-level timing assertions that were used to drive synthesis and custom macro tuning. The timing assertions consisted of macro input arrival times and slew rates, and macro output required times and capacitance loads.

To achieve high timing predictability between timing iterations and of course to reach the final cycle-time goal, it was extremely important that the accuracy of the static timing tool be measured and corrected for any flaws. To accomplish the measurements, hundreds of logic paths were analyzed, comparing the delay and slew results between the plan-of-record static timer and our dynamic circuit simulator using FET models from the foundry and adjusting the static-simulator technology file to achieve our desired accuracy. To achieve accurate macro timing results, every macro was extracted by breaking down all FETs into single devices and all interconnects into environmental multiple-RC poles, and the static-timing simulator was applied to the resulting extracted netlist. Interconnect parasitic accuracy was established by developing a method in which the extraction technology file capacitance coefficients were developed from multiple structures given to a field solver. This data was automatically added to the extraction technology file. The extraction technology file was developed to handle both timing and noise extraction. The technology file contains capacitance coefficients which are generated by utilizing a 3D field solver. This allows the extractor to build coupling capacitances that may be due to same-layer or different-layer coupling. The net result was that the timing predicted by the methodology fell within 5% of the timing measured on the final hardware.

Because of the quick turnaround time required for timing analysis during the design phase, a statistical global

extraction tool was introduced. During the last phases of the design, full environmental extraction is employed, and the results are typically found to be bounded within the range predicted by the statistical extraction.

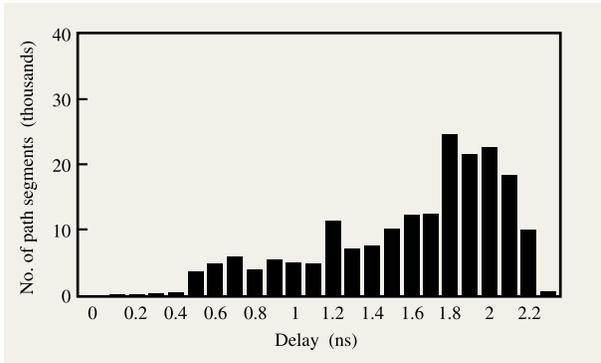
An extraction methodology was used on a flattened full model of the chip. This methodology includes all interconnects down to the macro level. Macro-level shapes are viewed as coupling into the global interconnect, but are not themselves checked. The extraction tool used is an IBM proprietary tool known as 3DX, and again the capacitance coefficients are computed by analyzing structures with a 3D field solver. It has many advanced features that allow it to handle the data volume associated with a flat chip; for example,

- A scan line algorithm can be used to reduce the number of shapes that must be in memory at any one time.
- The job can be broken into multiple parts, which can be run simultaneously on multiple machines.
- Several modes of operation allow tradeoffs between run time and accuracy.

A complementary tool is also used to calculate the resistance, which is added to the model.

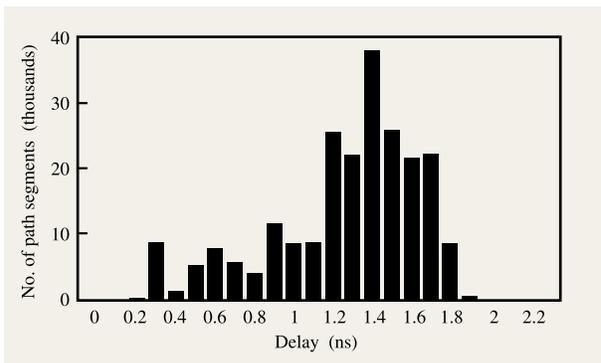
The three-level hierarchical design method subdivided the microprocessor into many units, with each unit further subdivided into macros. This hierarchy permits many quick timing iterations because the global timing analysis can be incrementally updated with new macro placement or timing fixes, and local timing analysis can be performed in parallel with global timing analysis. This methodology also provides good timing predictability between iterations, since placement of units on the chip and placement of macros within a unit are under the complete control of chip/unit floorplanners. The physical placement of gates is restricted to within the boundary of a macro, and the macro/unit pin placement provides a stable global wiring solution with predictable resistance and capacitance. It is worth noting here that accurate hierarchical pin placement on the macros/units relative to other macros and to macros in other units is highly desirable in achieving a good basis for overall timing and slew closure. Histograms of path delays for the G5 and G6 processors are shown in **Figures 13** and **14**.

**Figure 15** shows the G5 microprocessor timing closure. It is typical in the early design phase to be at 2× the timing target, since functionality takes priority. The floorplan and unit pin placement were optimized as the first-pass logic was completed. Critical nets were placed on low-resistance wide wires during the floorplan and pin-optimization stages. During the end of the design phase, all ac operational nets are subject to meeting slew criteria at all sinks in every net. Given this criterion and the size of the chip, it was a requirement that some nets in the



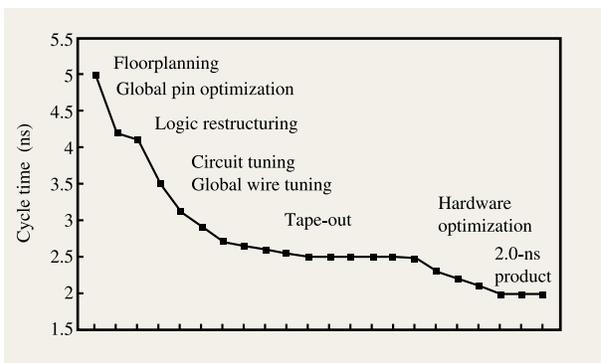
**Figure 13**

Histogram of G5 processor path delays.



**Figure 14**

Histogram of G6 processor path delays.



**Figure 15**

G5 processor timing closure.

critical path be attributed as wide wires and/or contain repeaters. Only after the logic was functionally complete did the logic team begin to focus strongly on timing reduction. In general, the logic designers minimized the number of macro crossings in a critical path by duplicating critical logic and optimizing latch points in the design. Custom macro tuning was done in parallel with logic design and integration. Local-control macro timing was improved by applying Shannon expansion techniques and coding the VHDL for optimal synthesis. Logic restructuring for timing accounts for most of the timing closure phase, and intermacro logic restructuring was completed several months before tape-out to allow faster physical design completion. Logic restructuring for timing *within* a macro, however, continued until tape-out. Circuit tuning and global wire tuning were done in parallel with logic restructuring to achieve the final cycle-time goal.

Initial G5 hardware was fully functional at the target cycle time. Final product hardware was 14% faster than initial hardware as a result of hardware optimization. The hardware optimization results are shown in **Figure 16**. Typical chips were analyzed to determine critical paths, and further design optimization resulted in a 65-ps improvement at the final tape-out. The clocks were tuned to give 90-ps improvement. A combination of increased voltage and sorting provided an additional 170-ps improvement, along with increased product yield and margin. An effort was made to minimize the wire delay in the critical paths. This “design for sorting” approach was achieved by tuning the floorplan and logic to minimize the amount of global wire in the critical paths. (A silicon-dominated critical path benefits more from  $L_{\text{eff}}$  sorting than a wire-dominated critical path.)

The G6 timing goal was achieved by circuit tuning and an automated low- $V_t$  methodology. The low- $V_t$  devices have a lower threshold voltage and drive higher current, resulting in a 10% speed improvement over nominal threshold devices. Because the low- $V_t$  devices have a smaller noise margin, they must be used selectively in static logic; the low- $V_t$  process inserted such devices into only critical paths. The low- $V_t$  process accounted for many parameters such as early-mode slack, number of critical paths through a gate, latch nodes, and whether the gate was receiving a global wire. This process allowed us to quickly achieve the G6 timing goal by creating critical paths with greater than 90% low- $V_t$  devices. The low- $V_t$  transistor count is only 1.25% of the total number of transistors on the chip, and 4% of the logic transistors.

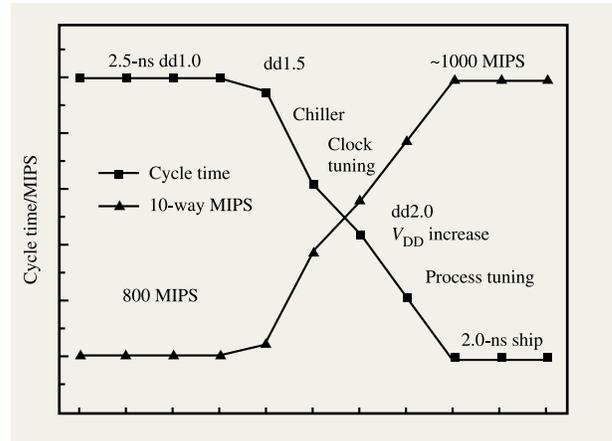
### Noise prevention and coupling analysis

Continued advances in silicon CMOS technologies have yielded dramatic increases in both circuit speed and wiring

density. These improvements have occurred from scaling of both front-end device and back-end line parameters, as well as from reductions in power-supply voltage. However, this scaling process has also resulted in a trend toward increased wire geometry aspect ratios (thickness/width) combined with shrinking wire dimensions (width and spacing), which creates increased coupling capacitance to neighboring wires. This effect, coupled with lower circuit noise margins associated with reduced power-supply voltage, has made chip functionality more sensitive to interconnect parasitic effects such as false logic switch, and latch disturbances. A graphical and numerical demonstration of minimum ground-rule differences between two IBM technologies, CMOS 5X and CMOS 6X, is given in **Figure 17**. From the figure it can be seen that there is significant overlap of coupling noise and circuit noise margin in CMOS 6X. Another effect associated with this phenomenon, but not shown in the graph, is the net delay variation, or noise “jitter,” which could affect chip performance and reliability by adding or subtracting wire delay not accounted for in traditional static timing tools.

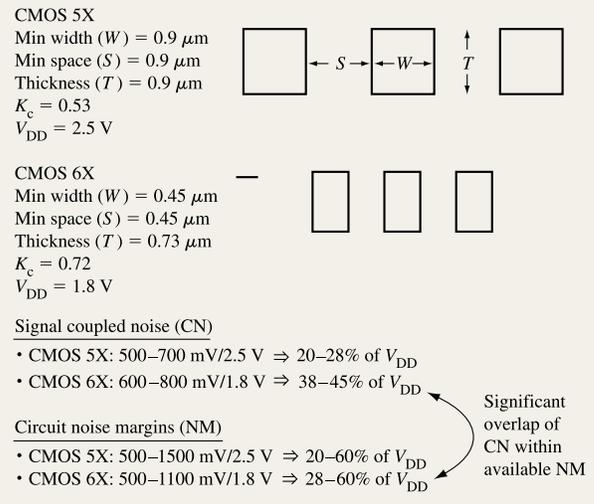
Coupling capacitance between nets is by far the dominant contributor in calculating noise, but the resistance of the line shunting the affected (“victim”) driver, the size of the driver, and the topology of the nets relative to one another can also be factors in determining the overall noise on any given net. The interactions between the nets and the active components connected to the nets with signals switching on those nets will force the victim driver to change its impedance in trying to react to noise generation by the noisy sources on the victim net. It should be pointed out that the noise magnitude can actually saturate or level off if the coupling capacitance is modeled as a multiport RC model (victim and perpetrator line resistance included).

In addition to capacitive coupling noise [8], inductive noise effects were also observed during hardware testing of the G4 processor. One example had dynamic nodes with ac noise margins of the order of 800 to 1000 mV, with capacitive coupling from both in-plane neighbors and vertical coupling from low-resistance data buses on the top LM layer. Careful electrical analysis of the failing circuit and its neighboring active nets using a full-wave field solver [9] and circuit simulation demonstrated inductive coupling as a significant noise source from vertical neighbors. The analysis identified a poor local power distribution structure (signal-to-power ratio  $\approx 40:1$ ) in conjunction with low-resistance global nets as the environment which manifested this problem. Wire fixes which included reducing the S:P ratio were performed in a metal-only design change. Additional work in the area of inductive effects within on-chip interconnects [10, 11] have also indicated these effects.



**Figure 16**

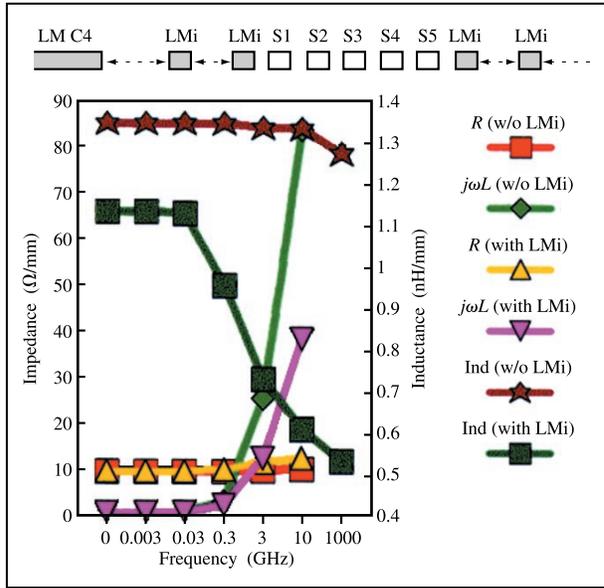
Hardware optimization to improve cycle time.



**Figure 17**

Technology scaling impact of coupling noise on noise margins.

To combat this problem for G5/G6, the chip power distribution was augmented with interstitial LM  $V_{DD}$ /ground returns placed within the power C4 rails using a 5:1 signal-to-power-ratio grid rule, as discussed earlier in the section on power distribution. Such techniques reduce the inductive noise impact by reducing the effective loop inductance associated with signal/return structure. The effect of lowering the loop inductance creates a shift in the frequency domain, resulting in a higher crossover frequency for the resistance and inductive reactance.



**Figure 18**

Interstitial LM grid used to reduce inductive effects.

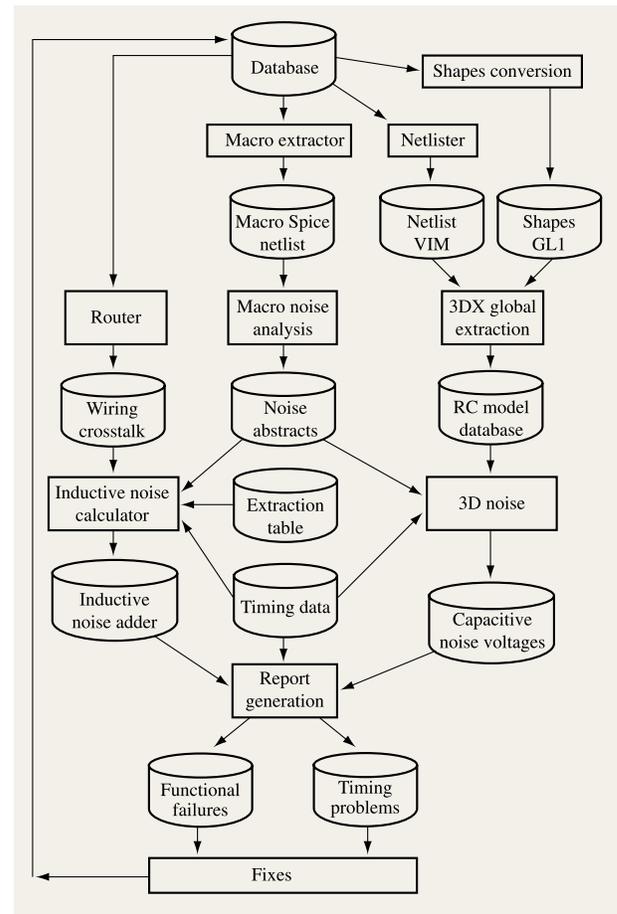
This is demonstrated in **Figure 18**, where a 2D field solver [12] was used to resolve the impedance of an LM signal line with and without interstitial returns. The crossover frequency is an indicator of how inductive effects become prevalent in noise prediction. Below this frequency, RC modeling of noise suffices, while above this frequency, RLC models must be considered.

The concept of reducing signal-to-power ratio also yields benefits by providing additional quiet tracks for signal lines, because there are fewer instances in which a victim net can be sandwiched by two active nets. Although there is no guarantee that a net wired next to a power or ground line would pass a noise check, the additional shields aid in the reduction of noise. This premise is applied judiciously across the chip/unit power distribution design with wide wire geometries [13] as a measure of noise avoidance.

The design of the G5/G6 floorplan took into consideration a correct-by-construction approach in the analysis techniques used to validate the design assumptions. On the basis of this premise, a comprehensive noise-verification process was developed to estimate the total noise for all global nets. The total noise consists of the crosstalk noise generated by either capacitive or inductive coupling or both, as well as simultaneous switching or delta-I noise. Functional failure criteria are established on a net-by-net basis. The impact of noise on nets with minimal timing slack is also

evaluated. This process, shown in **Figure 19**, is outlined below:

1. Macro pin data for such receiver noise margins and driver output impedance is established for all signal I/O.
2. Customized code is augmented with router net topology to account for inductive noise on nets.
3. 3D full-chip RC extraction is performed.
4. Noise simulations of RC networks are done to establish capacitive coupling noise on nets.
5. Results of steps 3 and 4 are combined and compared to noise limits established in step 1 to determine functional failures.
6. Results of step 5 are used with closed-form equations to account for the delay impact of noise on nets with minimal timing slack.



**Figure 19**

Global noise-verification process.

It should be pointed out that limitations of the current 3D extraction process resulted in the inductive and capacitive coupling noise voltages being calculated independently, using distinctly different calculation methods. The results are combined to assess the total coupling noise on a per-net basis.

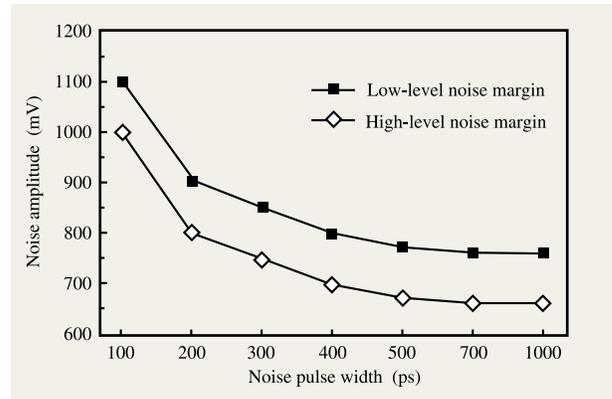
In order to describe the exact procedures, some definition of terms is required. The net of interest is called the “victim” net, since it is being disturbed by other nets that are switching. The switching nets are referred to as “perpetrators” (subsequently abbreviated to “perps”). The victim net is assumed to be nonswitching, or quiet, when it is vulnerable to a functional failure. The victim net is assumed to be switching when it is being analyzed for a timing failure (late mode) which can be caused by being slowed down by the perps switching in a direction opposite to that of the victim net. A failure can also be caused by the victim net being sped up (early mode) by the perps switching in the same direction as the victim net.

- *Noise attributes at macro I/O terminals*

Before noise calculations are performed, terminal noise attributes such as driver output impedance, receiver input capacitance, and noise limits must be determined for all macro signal I/O pins.

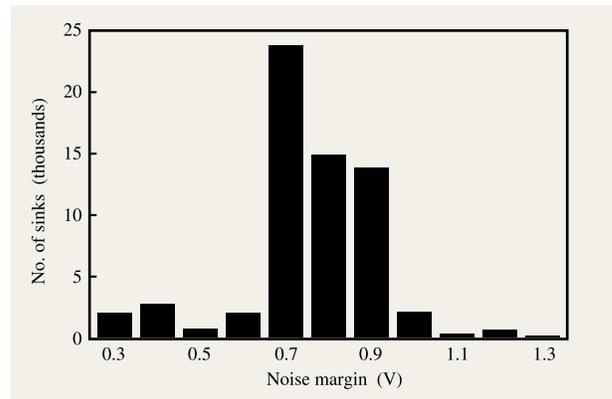
The driver output impedance and receiver input capacitance values are used in conjunction with the global net extraction process to accurately calculate the noise waveforms. The noise limit is a budgeted value allocated for the coupled noise check. Other noise sources such as delta-I which create spatial power-supply gradients from driver to receiver positions are precalculated [14] and factored into the receiver noise margin calculation.

The receiver noise margins, as well as the terminal parameters of the macro I/Os, are calculated by an internally developed tool using static analysis techniques. This tool operates on a transistor-level extracted netlist and has been optimized to perform a small-signal sensitivity analysis of channel-connected components [15]. Stability criteria based on small-signal analysis, such as a change of  $V_{out}/V_{in} < 1$  are used to compute the receiving-circuit ac noise margin. A typical noise-margin curve for an inverting circuit in CMOS 6X is shown in **Figure 20**. The I/O parameters are extracted directly from the netlist. In the case of driver output resistance, an equivalent device size is determined for the channel-connected components and converted to a small-signal resistance using precomputed coefficients. These calculations are performed for all macros on the CP chip, with resulting data placed in a noise “abstract” file for each macro to be used for subsequent global noise analysis. A histogram of noise margins at a pulse width of 500 ps is shown in **Figure 21**.



**Figure 20**

Typical ac noise-margin curve.



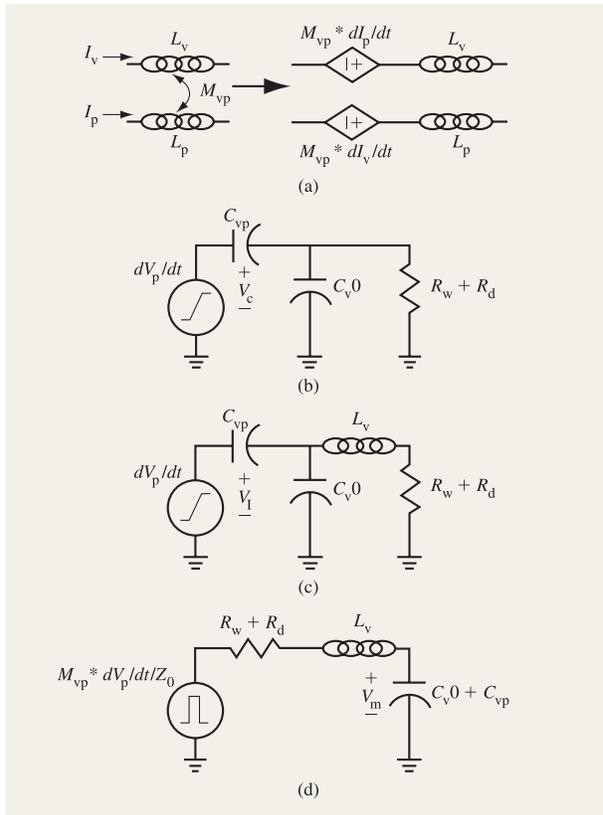
**Figure 21**

Histogram of receiver noise margins.

- *Inductive-coupling noise estimation*

The calculation of inductive coupling noise is a function of several physical and electrical parameters, such as victim/perp net topologies, power distribution, and driver slew rates. The primary reason for such detailed information is to relate the current and voltage magnitudes, rate of change, and topology orientation of the perp net to those of the victim.

The coupling interaction is based on close parallel wiring proximity of the victim and perp nets. There can be many such interactions across the extent of the victim net, with as many different perp nets. Inductive coupling becomes a noticeable proportion of the total coupling noise when the perp wire resistance falls below 25  $\Omega/\text{mm}$  [11]. These low-resistance conditions occur at the top



**Figure 22**

(a) Lumped-element equivalent circuit of a pairwise partial-inductance model. (b) One-pole capacitive circuit. (c) Merged circuits (a) and (b) result in two-pole RLC self-inductance model. (d) Merged circuits (a) and (b) result in two-pole RLC mutual-inductance model.

wiring levels of the chip, where long unit-to-unit interconnections (~10–15 mm) are prevalent.

To simplify this problem, pairwise coupling networks are formulated for each victim/perp coupling interaction, such that closed-form equations can be applied in conjunction with an inductance and capacitance extraction process. The capacitance extraction is an important component in computing the inductive noise adder because of the injected current of the capacitive coupling and the loading effect of the total capacitance.

The following steps form the basis of the inductive noise analysis routine:

1. Physical extraction of pairwise coupling interactions is performed for all nets within a unit. This crosstalk report contains the  $xy$  values, metal-level names, and spacing for each coupled segment associated with a victim net. This is true for both in-plane and interplane

interactions. The procedure to generate a crosstalk report can be performed on the physical database alone and does not require a schematic, as do full-extraction programs such as 3DX. The benefit of this is that the noise-calculation procedure still operates in the presence of net shorts, since no logical-to-physical checking is required. This allows the noise analysis to be performed early in the design cycle. A unit will typically contain 2000 to 8000 nets which have to be considered as potential victim nets. As an example, one unit has 6806 nets, of which 3222 nets have significant coupled noise. The 3222 victim nets have a total of 27465 victim-perp interactions, of which 14654 are significant enough to be analyzed for noise voltage. The average number of significant perps per victim net is almost 5. Of the 14654 victim-perp coupling interactions, 65% are same-plane, and 35% are in different planes. As shown by this example, several thousand unit nets create the need for a substantial quantity of noise analysis.

2. Using the routing database, every net is traced through every wire segment from the driver pin to every receiver (input) on the net. This is not a trivial task for a complex net that contains, e.g., 100 wire segments and 10 receivers. The tracing is required to determine certain electrical parameters such as wire resistance from the driver to each coupling section on the victim net and signal flow direction. Also, the perp nets have to be traced in order to calculate transition-time degradation from the driver output node to the given coupling section where the perp is coupling to a victim net.
3. The electrical extraction parameters at each coupled wire segment are calculated using an extraction-table-lookup technique that is a simplified version of the process described in [16]. The process used here assumes that orthogonal wiring layers are 50% occupied and neglects loading from adjacent nets outside the pairwise interaction. The extract table lists the coupling capacitances, self-capacitance, mutual inductance, and self-inductance as a function of the wiring level, wire widths for the victim and perp wire segments, and the spacing between the wire segments. These combinations include wires on different wiring planes as well as on the same plane. These electrical extraction parameters are not as accurate as a full three-dimensional extraction program, but still provide a good estimate for crosstalk calculations.
4. Other necessary electrical parameters such as perp rise and fall times, arrival times, driver equivalent output resistances, and receiver input capacitances are read from various output files created from the various design databases. When the parameter values are not available, default values are substituted.

5. Inductive noise voltage is calculated using a combination of one-pole RC and two two-pole RLC closed-form equations. The two-pole RLC equations are derived using an equivalent lumped-element circuit of a pairwise partial inductance model which is shown in **Figure 22(a)**. Merging this equivalent circuit with the one-pole capacitive coupling circuit [8] shown in **Figure 22(b)** results in two two-pole RLC networks [**Figures 22(c)** and **22(d)**]. The noise voltage  $V_m$  generated by the mutual inductance is due to the  $dI/dt$  on the perp net. This current change impact is represented as a pulse with amplitude based on the ratio of perp  $dV/dt$  and transmission-line impedance,  $Z_0$ . The pulse width is a function of the perp rise time. The self-inductance represents an additional load in series with the resistance of the net. The noise voltage  $V_I$  is computed for the capacitive coupling case in the presence of this loading effect.

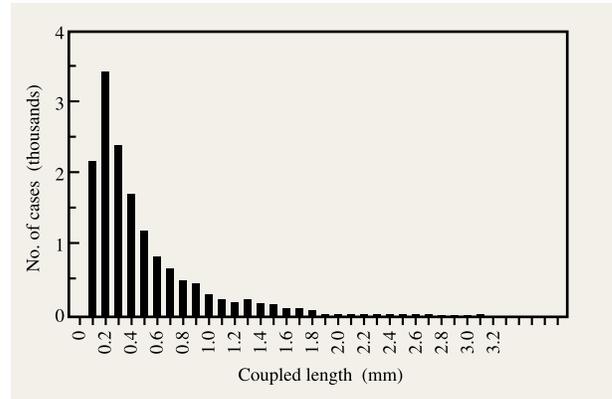
6. Closed-form equations are easily derived for both RLC circuits and use all of the electrical values described previously in steps 3 to 5. To compute the inductive noise adder such that it is consistent with the RC computations obtained from a full-chip 3D extraction, the noise from the one-pole RC equation is subtracted from the sum of the noise computed from the RLC circuits:

$$\text{Inductive noise adder} = V_m + V_I - V_c.$$

It should be pointed out that this is a bounding calculation that assumes a worst-case condition for addition of  $V_m$  and  $V_I$  regardless of signal direction. Also, this adder is augmented to support a more detailed and accurate RC calculation, which is the foundation of the noise-verification process used for G5/G6.

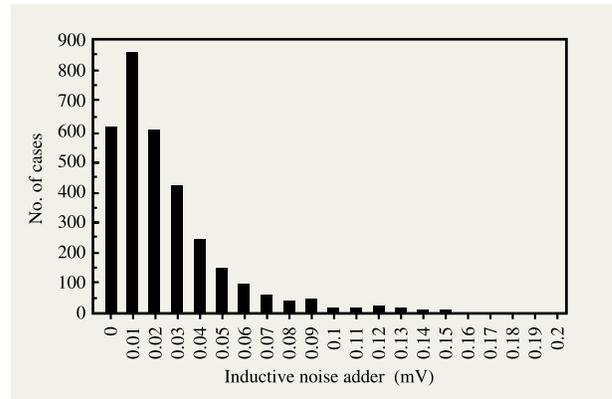
7. For selected potential failing nets, a more accurate analysis can be performed by automatically generating a detailed circuit simulation model using RLC transmission-line models.

**Figure 23** is a histogram of coupled length for each of the 14654 analyzed victim-perp interactions. **Figure 24** is a plot of the total inductive coupled-noise voltage for each of the 3222 victim nets. Typically, a coupled-noise histogram shows a long tail. Figure 23 has 37 cases with coupled length greater than 3 mm each, or 0.25% of the nets. Figure 24 has 95 cases out of 3222 victim nets with total inductive noise greater than 100 mV each, or 3% of the nets at the unit level. Data for the chip-level global nets (unit-to-unit) is discussed in the next section. Typically, a small number of the nets cause the majority of the noise problems.



**Figure 23**

Histogram of coupled length for each victim/perp interaction.

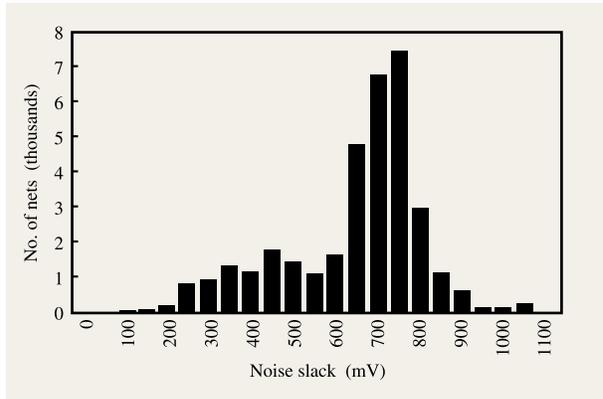


**Figure 24**

Histogram of inductive noise adder.

• *Capacitive-coupling noise estimation*

After the 3D RC extraction has been completed on all global nets, the data is used as input to a proprietary tool known as 3DNoise, which builds a complete RC coupling model for each net in the design using pin attributes found in the noise “abstracts” in conjunction with the 3D RC extraction data. Perp driver transition times, which are found in timing reports, are used in conjunction with a fast circuit-simulation technique [17] to accurately assess the capacitive coupling noise at each receiver on the net. This simulation is performed for all global nets. In the absence of any other data, 3DNoise will build a model with default driver resistance and pin caps. It will then switch all perp coupling capacitances simultaneously, with



**Figure 25**

Histogram of noise slack.

assumed transition times, and note the capacitive coupling noise at each sink of the net in question (victim). If the voltage at any of the sinks exceeds a default noise margin, this sink is reported as a possible functional noise failure. This “default mode” of 3DNoise is very useful in early stages of the design, when more detailed information is not available.

Additional procedures are applied to reduce pessimism in the calculations, but require additional analysis and/or postprocessing time:

- In general, the global noise analyzer assumes that the given perp transition occurs at each coupled section. This is a worst-case assumption, since it ignores the transition-time degradation that occurs along the path of the perp. When a net exceeds its noise tolerance, it is re-analyzed by including the perp secondary resistance in the RC network. This reduces the transition time at the coupled sections, thus reducing noise.
- In determining perp timing windows, it is currently assumed that the victim is vulnerable to noise at any time throughout the cycle, and without timing windows, it is assumed that all perp noise pulses line up in such a way that they are all added into the noise at the sink. A timing window is defined as that range of time within which a signal can be switching in the perp net, or the interval from the earliest possible time at the driver to the latest possible time at the sinks. If two perp windows do not overlap, they cannot possibly both affect the noise at the same time.

• *Noise impact on chip functionality and performance*

The total coupling noise on a net is the summation of the computed capacitive and inductive crosstalk noise. This

noise can cause latch disturbances created by feed-through noise from pass gates and/or dynamic half-latch disturbances, which could render the chip useless independently of cycle time. The capacitive and inductive data are combined in a report and merged with noise-margin information from the noise abstracts to provide a list of potential functional failures (nets that cause circuits to switch falsely). A key metric from this report is noise slack. Noise slack is defined as the noise margin at a sink minus the noise injected at that sink. A negative number indicates the possibility of a functional failure. For nets with multiple sinks, the sink with the smallest noise slack is reported. As can be seen in **Figure 25**, no functional failures are currently predicted. However, when the design was first analyzed, several hundred possible failures were first reported; these were then fixed using a combination of the techniques described below. Another important point is that the distribution of coupled noise tails off quickly. **Figure 26(a)** shows the top 197 (0.56%) nets on the chip. In analyzing noise, the tail of the curve is always the section of interest. **Figure 26(b)** shows the possible effect of timing windows. Without timing windows, there are 575 nets in the tail, which can greatly increase the time spent sifting through them to find real problems.

The effect of noise on timing is also of concern. The required cycle-time goals can be jeopardized by the interaction of the noise generated by the perps with their timing positions. Because of the interaction of signals as they switch opposite one another, signal edge integrity can be subject to many distortions. This can cause receiving gates to turn on much later than expected, causing the cycle-time goal to be missed. It can also be demonstrated that the paths can be sped up when coupling signals are switching together, which reduces the capacitance and results in a nonfunctional chip. Since the delay of any one signal is solely based on the interactions of the signals in its proximity, the overall signal delay can reach a saturation point. It was found through simulations of worst-case signal interactions [1] that overall path timing can be delayed by as much as 350 ps.

To determine the effect on timing, a window of vulnerability about each sink of a net is determined. Four windows of vulnerability are calculated, and the timing delta ( $\Delta t$ ) is calculated for each window. The four windows are early-mode rise and fall and late-mode rise and fall. For each of these, the windows are determined by the arrival time and the transition time at the sink in question:

$$\begin{aligned} \text{Start} &= \text{mode\_direction\_arrival\_time} \\ &\quad - 0.8 * \text{mode\_direction\_transition\_time;} \\ \text{End} &= \text{mode\_direction\_arrival\_time} \\ &\quad + 0.4 * \text{mode\_direction\_transition\_time.} \end{aligned}$$

Then, for each perp, the active window (earliest arrival time at the driver to latest arrival time at any of the sinks) is determined. If this perp window intersects the victim's window of vulnerability, an equation is used that is dependent on the victim transition time and the noise being injected by the perp to derive a timing delta. The top four perps (largest delta- $t$ ) are summed, and the remaining are averaged by RMS. **Figure 27** indicates the importance of checking noise effects, as they affected timing on this chip by as much as 275 ps.

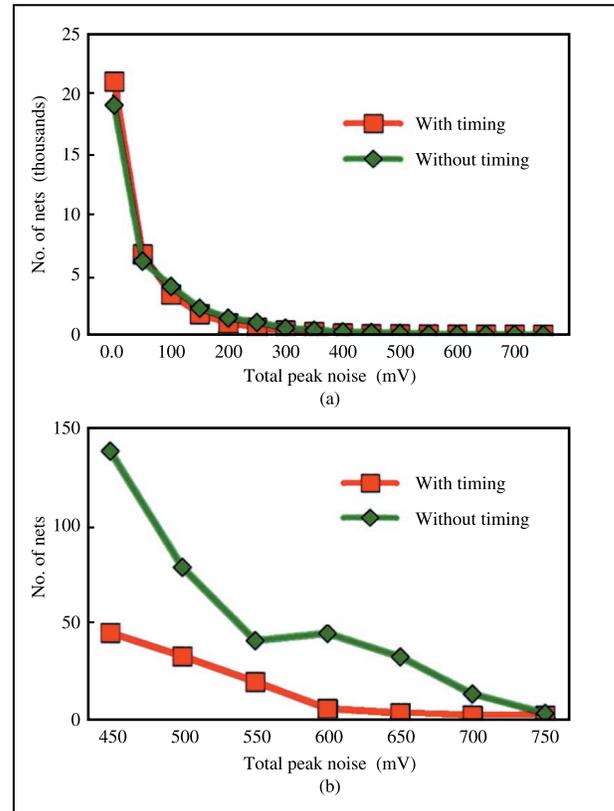
When G5 was first analyzed, several hundred nets were found that were susceptible to noise-induced timing failures. The potential problems of susceptible nets were corrected by using a combination of the techniques described below:

- Separate the wires. If the larger perps can be moved such that they do not couple for as great a distance, this will reduce the noise generated. Some care must be taken, however, that the problem is not merely switched to another net. This technique solves a majority of the problems.
- Decrease the victim driver resistance. A larger victim driver will reduce the noise “seen” by its sinks. The tradeoff is increased  $dI/dt$  noise, and also the potential to make this net become a stronger perp to other nets.
- Increase the transition time of the perp. If the perp switches more slowly, the noise it generates will be reduced, but this obviously slows the net down.
- Increase the noise margin of the sink. Making a macro less noise-sensitive is something that can be considered early in the design, but it is somewhat difficult at the stage when good noise numbers are being generated. It is a good practice to have the macro designers be aware of noise problems as macros are being built, so that no macros end up with low noise tolerance.
- Add buffers to the victim. Cutting a net in half reduces the potential coupling capacitance by half. Buffers also tend to have very good noise tolerance. This option is excellent at minimizing “churn” in functional units late in the design cycle, when buffers are added at the CP level.

### Full-chip electrical analysis

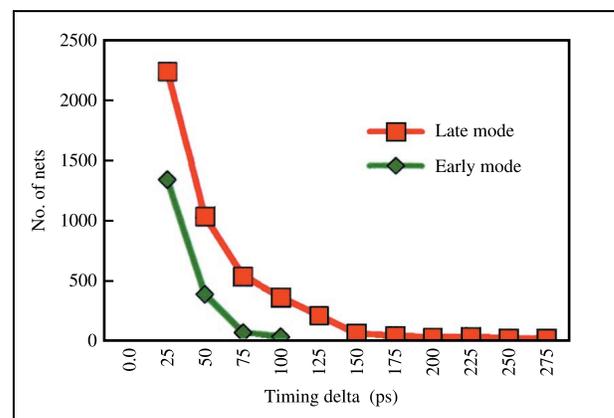
The methodology for electrical analysis used for G4 [5] was the basis for G5 and G6. Electrical analysis consists of predicting and calculating the power dissipation of the design, verifying the integrity of the power grid, and ensuring that electromigration checks are performed [18, 19].

The methodology uses the concept of power points and the abstraction of electrical data across the design hierarchy. Power points are those points within a macro for which current is determined. For this design, the



**Figure 26**

Histograms of (a) peak noise and (b) peak noise above 400 mV.



**Figure 27**

Noise effects on timing.

contact level where the circuit contacted the metal for power distribution was identified as the point at which to

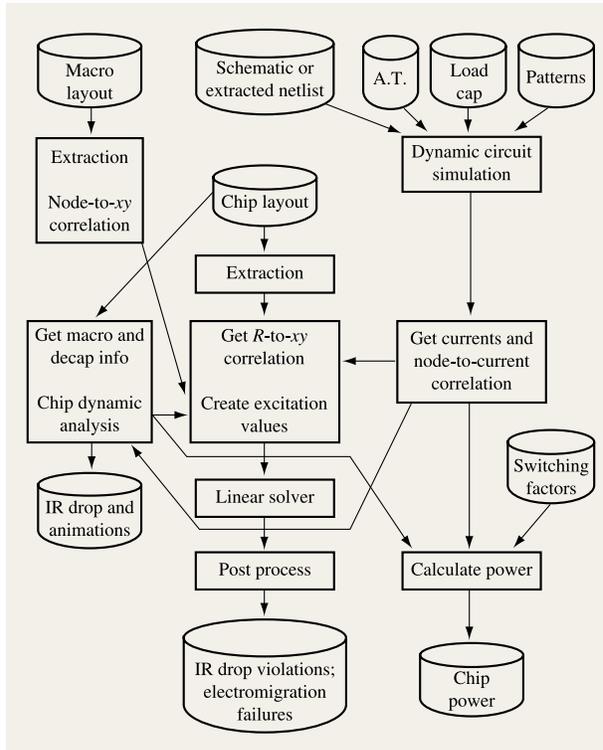


Figure 28

Electrical analysis process flow.

determine current draw. The currents associated with power points were then abstracted for use at the chip level of analysis.

As seen in **Figure 28**, the first step in the methodology is for the macro designer to run a dynamic electrical analysis tool to determine macro power and the currents for the power points. The inputs to this analysis are the netlist for the macro, the arrival times of the input signals, the capacitance load on the output pins of the macro, and a pattern file for the inputs. An extracted netlist must be used to generate power-point currents for subsequent voltage drop and electromigration analysis, but to determine chip power only, the netlist can come from a schematic. Since the schematics are ready earlier in the design cycle than the layouts, and since early power dissipation numbers are wanted, a schematic-based analysis is done followed by an analysis on the extracted netlist. The submission of these many extraction runs on the layouts and the analysis of the netlists were fully automated.

All of the inputs for dynamic circuit analysis are automatically generated. Common signals for the entire design are preset to fixed states, and the generation program assumes that all other inputs are random in nature. The designer is responsible for determining whether these assumptions are valid and editing the pattern configuration file if needed. Designers who wish to do so can supply their own patterns. Arrival times and capacitance loads are obtained from the global timing files. For smaller macros, hundreds of patterns were analyzed, and for the largest macros, tens of patterns were analyzed.

The output of the dynamic analysis is a binary file representing waveforms for the various input patterns. A postprocessing tool was developed which analyzed the waveforms and determined the average power for the macro over all of the patterns and the current waveform for the pattern with the highest average current. The overall average power values for each macro were used to determine the total chip power. Global switching factors derived from previous logic simulation analysis or from individual macro-based switching factors from the current logic simulation were used, along with technology spread factors and voltage variance, to determine hold, nominal, and maximum power. **Figure 29** shows the macro power densities for the G5 microprocessor chip. With this technique, the estimates of chip power have been conservative by five to ten percent compared with measured power. The currents for each macro were also used in a package-level noise-analysis tool for delta-I noise due to inductance in the package pins.

For voltage drop due to current flowing through the on-chip power distribution wiring (IR drop), a two-step approach was used. A global analysis approach was used

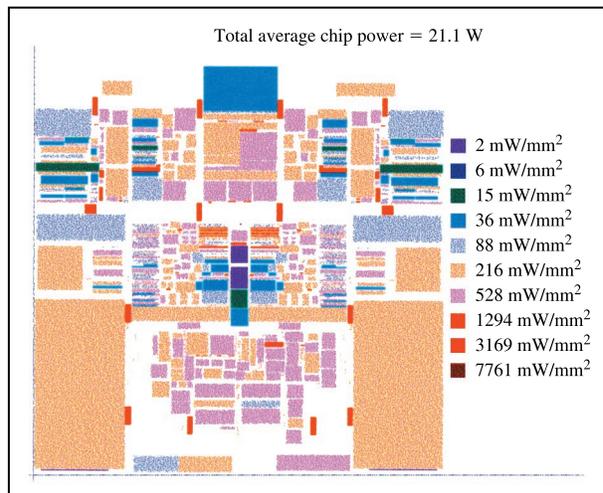


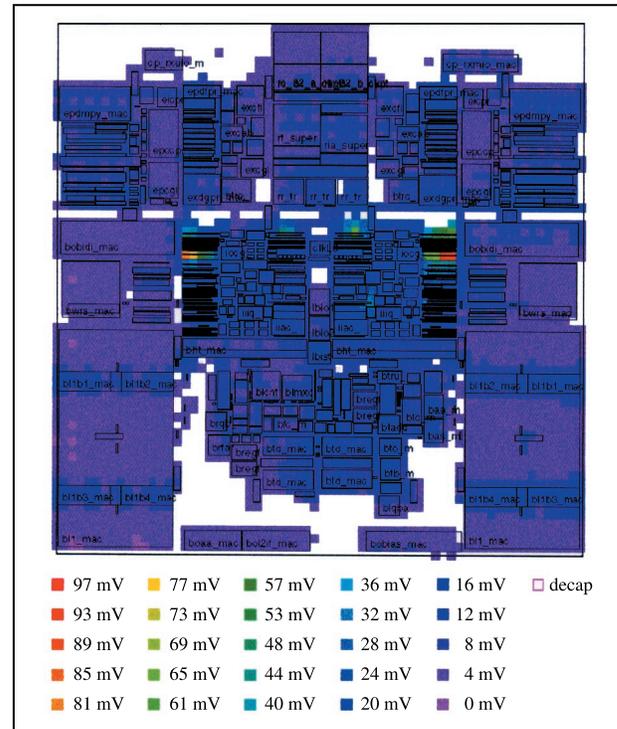
Figure 29

G5 macro power densities.

to determine the areas with large IR drops. Detailed analysis was then performed on those areas. Detailed analysis was also performed on areas where macros had high current densities. The global analysis was done by taking the on-chip power distribution layout and, through a program, generating an equivalent RLC network from chip power pins to chip power pins. It was determined that having inductance in the model accounted for 10 mV, so it was removed from modeling for better performance. The current waveforms of the macros were then distributed over the contacts in the appropriate area based on *xy* information and attached to the equivalent network. For array macros, the total current was not distributed over the memory area of the array but only over the peripheral circuitry. A dynamic electrical analysis was then done, with the results being either the average or the peak IR drop, depending upon which values of current were used. This global analysis also supported attachment of the chip model to the package model. An animation was made from these findings. **Figure 30** shows a color map of the steady-state average voltage drop for the G5 microprocessor. The maximum average voltage drop was 97 mV, with the maximum steady-state peak voltage drop being 156 mV.

For the detailed analysis, the currents for each power point from the macros in the area of the chip being analyzed were brought to the chip level and joined together to form an excitation function. This was done using the power-point *xy* coordinate which is carried along and the transform information of the macros on the chip. In order to improve the matching of currents to contacts in a section being analyzed at the chip level, enhancements were made to the programs to handle unique circuits such as decoupling capacitors. A large resistive network was created by an extraction program run against the power grid for the section of the chip being analyzed. This network, along with the currents for the power points and chip voltage points, was fed into a linear solver program. IR drop results were calculated for each power point. The extraction program also produced values for the width of wires and the sizes of vias. This information was passed into the linear solver so that the results reflected the wire level, width, and current flowing in the wire. Another program analyzed this data to determine electromigration problems. Electromigration error and warning markers were also created for import into the macro layout for ease of diagnosis and correction.

As discussed above, analysis was done on a section cut from the global power grid. This cutting of a section could create antenna shapes. These would be shapes which were attributed as part of the power grid, but the cutting isolated them from making contact with the rest of the grid. Thus, the extraction program identified them as



**Figure 30**

G5 steady-state voltage drop.

disjoint nets to the rest of the power grid. These edge effects were automatically handled with improved code for G5/G6. In rare cases, short runs of mbar (a local interconnect used for macro-level wiring) were used inside a macro to make the power grid continuous. To have this layer always present in the extraction layers would create too much data and longer run times. In G5/G6, a process was developed in which the detection of mbar would automatically run the extraction with the correct but slower extraction rules.

Also developed for G6 was a macro-level checking tool which checked for missing vias and the presence of mbar in the power distribution. A design guideline was established that prohibited the use of mbar in the power grid, and this was used to check compliance. Most electromigration problems found were due to missing vias, and this checker also improved the electromigration coverage. A more powerful global extractor was also introduced into the process for G6, increasing the size by a factor of 20 for extracting the global section, thus allowing for more detailed analysis of the chip with fewer extraction runs.

## Full-chip verification

The physical verification methodology for the G5/G6 design was in many ways fundamentally the same as that for the G4. A hierarchical internal checking tool was employed to ensure that each design was correct down to the transistor level. The same tool that was used to check the simplest of circuits was also used to check the completed top level of the design. The one major difference for G5 and G6 was that there was a far greater emphasis on reducing the DRC and LVS run size.

A reduction in run size was required for many reasons. The increased complexity of the G5/G6 designs, coupled with the increased complexity of the design-rule-checking run-sets, resulted in DRC and LVS runs that either took too long to complete or did not fit within the machine memory limit used to run the checking jobs. Also, after the normal physical verification cycle was completed, it was necessary to run an additional series of jobs. These jobs added metal-fill shapes that were needed to fulfill the pattern-density rules required by the more aggressive technologies used for the G5 and G6 designs. These shapes were added long enough after the design data were developed that extraction for timing and coupling noise analysis were not performed after this step. The overall physical verification turnaround time was therefore increased by the time needed to add and check the fill shapes. In order to keep turnaround time to a minimum, it was necessary to reduce the run times of the final DRC/LVS cycle.

A two-pronged approach was employed in order to solve the run size problem. For DRC checking, the main problems were memory usage and turnaround time. The full DRC check did not fit into memory, and would take several days to complete even if it did. It was decided that a partitioning of the DRC run-set would be necessary.

In the existing methodology, it was not feasible or desirable to partition the LVS run. The turnaround time was manageable, and LVS jobs submitted in the evening would be available for diagnosis the following morning. A reduction in the LVS portion of the physical verification cycle could be realized only by reducing the number of times it was necessary to rerun the full-chip checking job.

The DRC run-set was divided into several subsets. These subsets were designed to check groups of closely related design levels so that they might be as efficient as possible. For example, the DRC run-set used for G5 was broken up into four sub-run-sets. Two of these checked silicon design levels, one checked metal personalization levels, and the last one checked the rules that dealt with the interaction of silicon and personalization. For G6, which was designed in a technology with very complex personalization rules, the original run-set was broken up into eight sub-run-sets. This partitioning of the run-set resulted in overnight or better turnaround times for all

DRC runs. An additional benefit of this partitioning was that the reduced memory usage of the runs allowed them to run on a wider class of machines.

This method of partitioning the DRC run-set was also employed in the checking of large macros and processor subunits. In these cases, the run-set was divided into fewer subsets than were needed for full-chip checking because of the smaller design size. Using the same DRC subsets as for the full chip would probably have been counterproductive in this application. As with the full-chip DRC, the partitioning of the checking runs resulted in a quicker turnaround time due to shorter run times and greater system resource accessibility.

The primary bottleneck for LVS checking was repeated iterations of chip-level checking needed to find and remove netlist compare errors. An LVS cycle consisted of data output, the actual LVS run itself, error diagnosis, and error repair. Because of the length of time needed for data output and the running of LVS, only one cycle could be completed in a 24-hour period. It was therefore important to keep the number of cycles to a minimum to allow the designs to complete on time.

The approach taken to minimize the number of these cycles was to make as many early chip-level LVS runs as possible. A top-down checking methodology, enabled by the hierarchical design, was employed to achieve this goal. Long before any of the processor subunits had been completed, LVS was being run at the chip level. Abstracts containing top-level terminal information were used in place of the subunits, with the LVS checking tool told to ignore any schematic information nested below them. At this point, the top level of the chip consisted of power and clock distribution, I/O circuitry, and chip-level wiring. These top-level elements were therefore completely verified before any final unit checking had begun.

Once a unit had been completed, an LVS check was run which included the actual layout data for that unit, along with abstracts for each of the other units. Once a unit was deemed to be LVS-clean within the chip, it was permanently added to a "master" chip-level layout which contained all of the units which had passed LVS at the chip level, plus abstracts for those which were still experiencing LVS failures. Running LVS on this layout enabled the detection of any unit-to-unit LVS errors, and speeded the completion of final chip-level LVS checking.

The benefits of such an approach were twofold. The smaller "unit-in-chip"-level runs finished more quickly than a full-chip-level LVS run, and could run on a wider range of machines. It was not unusual to be able to submit several checking jobs a day on the same unit. The main benefit of such an approach, however, was the vast improvement in diagnostic capability when compared to a full-chip LVS run. Since only one processor unit was

being checked at a time, it was relatively easy to pinpoint errors caused by interaction with chip-level wiring. This was especially true when dealing with power-supply shorts, which had been virtually impossible to find using a full-chip approach.

After the chip had passed final DRC and LVS checks, an additional series of jobs were run in order to add metal-fill shapes to the final data. These metal-fill shapes were needed to satisfy mask house density requirements. The "filled" data version was about 50% larger than the final version, and it soon became apparent that rerunning the same checks would have a prohibitive cost, both in memory usage and turnaround time. It was determined that any additional checks should be run only if they addressed the correctness of the fill shapes themselves, since the nonfill shapes were known to be correct. Compare jobs were run to ensure that there were no changes in the nonfill data levels, and DRC jobs were submitted that checked the layout correctness of the fill shapes. To ensure that the filled design was LVS-clean, it was determined that the only danger could come from a short. Adding metal shapes to a clean design could not result in an open. Therefore, a pseudo-LVS run was made on the final filled data, ensuring that each new fill shape did not connect to any metal shapes on the same design level, or to via shapes above or below it. Successful completion of this run, along with the fill-DRC and nonfill compare jobs, resulted in data that was 100% clean and ready for final mask build.

## Conclusions

In conclusion, it is appropriate to reiterate the most important physical design criteria for the G5/G6 microprocessor chips and emphasize the methodologies and techniques employed to enable successful execution of the design. Time to market, although not a unique challenge to the physical design of large (25 million transistors) VLSI microprocessors, is a significant constraint from the point of view of practicality, and will strongly influence level of success in the marketplace. Other, more technical criteria discussed are technology (capability of the fabrication process), architectural requirements (logical definition), performance (frequency of operation), silicon area required for implementation of devices and interconnections, requirement for a high-performance, low-skew clock distribution network, requirement for a robust power distribution network, power dissipation, electrical signal integrity, including on-chip inductive effects, final physical design rule checking (DRC), and final schematic-to-physical design equivalence (LVS) checking.

The recurring theme and key enabler for timely convergence on each of these physical design criteria is the exploitation of a hierarchical design methodology. The

hierarchical design methodology has enabled concurrent, iterative execution of design subsets (macro, unit, and chip levels) while ensuring compatibility of the components (and their interactions) through judicious management of constraints across the hierarchy. The flow of constraints within this design hierarchy is organized with the capability to pass requirements and restrictions up the hierarchy (from macro to unit to chip) as well as down the hierarchy (from chip to unit to macro). Implementation at the higher levels of hierarchy is further simplified by the use of abstraction techniques for modeling the lower levels (macros). Several key examples discussed include concurrent logical and physical design progression through the use of wiring snapshots; implementation of global power distribution; implementation of global clock distribution; management of interconnect (wiring) resources through the use of "wiring contracts"; the highly iterative timing and extraction process, which employs detailed transistor-level simulations at the macro level and a global analysis utilizing abstracted macro timing models; the coupling analysis process, which employs detailed small-signal transistor-level simulations at the macro level and a global analysis utilizing abstracted macro noise models; and the power simulation flow, which also runs detailed macro-level simulations with a global analysis of abstracted detailed macro data.

One of the most significant technological advances exploited in the G6 processor chip implementation discussed here is the copper back-end-of-line for on-chip interconnections. While CMOS device scaling continues to advance well into deep-submicron levels, the performance of the interconnect technology has not scaled accordingly when global chip interconnections on larger and larger chip die sizes are considered. The lower resistivity of copper has offset some of the lag on interconnect performance advances, but imposes additional complexity in implementation and ground-rule verification.

Finally, in exploiting the advances of technology such as silicon-on-insulator (SOI) devices, low-dielectric BEOL insulators, additional BEOL metal levels, and higher and higher device densities to build microprocessor chips capable of densities of several hundred million devices on a die, the integration of the physical design will continue to be a challenge. Architectural definition must be developed concurrently with the physical implementation in mind.

## Acknowledgments

The authors acknowledge the contributions of the S/390 design teams. Key to the processor chip integration implementation were integrators D. Merrill, D. Balazich, S. Carey, R. Dennis, N. Kolesar, and T. Wohlfahrt; circuit

leaders Yuen Chan, Yiu-Hing Chan, R. Crea, F. Tanzi, D. Webber, and F. Yee; and back-end methodology developers H. Chen, E. Cho, B. Curran, A. Deutsch, P. Lin, F. Busaba, K. Eng, L. Lange, L. Lacey, S. Meotra, S. Neely, P. Restle, and P. Villerubia.

\*Trademark or registered trademark of International Business Machines Corporation.

## References

1. C. F. Webb, C. J. Anderson, L. Sigal, K. L. Shepard, J. S. Liptay, J. D. Warnock, B. Curran, B. W. Krumm, M. D. Mayo, P. J. Camporese, E. M. Schwarz, M. S. Farrell, P. J. Restle, R. M. Averill, and T. J. Slegel, "A 400MHz S/390 Microprocessor," *1997 IEEE International Solid State Circuit Conference Digest of Technical Papers*, 1997, pp. 168–169, 449.
2. C. F. Webb and John Liptay, "A High Frequency Custom CMOS S/390 Microprocessor," *Proceedings of the 1997 IEEE International Conference on Computer Design*, 1997, pp. 241–246.
3. T. J. Slegel, R. M. Averill, M. A. Check, B. C. Giamei, B. W. Krumm, C. A. Krygowski, W. H. Li, J. S. Liptay, J. D. MacDougall, T. J. McPherson, J. A. Navarro, E. M. Schwarz, K. Shum, and C. F. Webb, "IBM's S/390 G5 Microprocessor," presented at the 1998 Hot Chips Symposium, Stanford, CA, August 1998.
4. D. Hoffman, R. Averill, B. Curran, Y. H. Chan, A. Danksy, R. Hatch, T. McNamara, T. McPherson, G. Northrop, L. Sigal, A. Pelella, and P. Williams, "Deep Submicron Design Techniques for 500MHz IBM S/390 G5 Custom Microprocessor," *Proceedings of the 1998 International Conference on Computer Design*, 1998, pp. 258–263.
5. K. L. Shepard, S. M. Carey, E. K. Cho, B. W. Curran, R. F. Hatch, D. E. Hoffman, S. A. McCabe, G. A. Northrop, and R. Seigler, "Design Methodology for the S/390 Parallel Enterprise Server G4 Microprocessors," *IBM J. Res. Develop.* **41**, No. 4/5, 515–547 (July/September 1997).
6. J. Warnock, L. Sigal, B. Curran, and Y. Chan, "High Performance CMOS Circuit Techniques for the G4 S/390 Microprocessor," *Proceedings of the 1997 IEEE International Conference on Computer Design*, 1997, pp. 247–252.
7. P. Restle, P. Jenkins, A. Deutsch, and P. Cook, "Measurements and Modeling of On Chip Transmission Lines Effects in a 400MHz Microprocessor," *IEEE J. Solid State Circuits* **33**, No. 4, 662–665 (April 1998).
8. A. Danksy, H. Smith, and P. Williams, "On-Chip Coupled Noise Analysis of a High Performance S/390 Microprocessor," *Proceedings of the 1997 Electronic Components and Technology Conference*, 1997, pp. 817–824.
9. B. J. Rubin, "An Electromagnetic Approach for Modeling High-Performance Computer Packages," *IBM J. Res. Develop.* **34**, No. 4, 585–600 (July 1990).
10. A. Deutsch, W. D. Becker, G. A. Katopis, H. Smith, P. J. Restle, P. W. Coteus, C. W. Surovic, G. V. Kopcsay, B. J. Rubin, R. P. Dunne, T. Gallo, K. A. Jenkins, L. M. Terman, R. H. Dennard, and D. R. Knebel, "Design and Technology Guidelines for Short, Medium, and Long On-Chip Interconnections," *Proceedings of the 1996 IEEE Electrical Performance of Electronic Packaging Conference*, 1996, pp. 30–33.
11. A. Deutsch, H. Smith, G. A. Katopis, W. D. Becker, P. W. Coteus, C. W. Surovic, G. V. Kopcsay, B. J. Rubin, R. P. Dunne, T. Gallo, D. R. Knebel, B. L. Krauter, L. M. Terman, G. A. Sai-Halasz, and P. J. Restle, "The Importance of Inductance and Inductive Coupling for On Chip Wiring," *Proceedings of the 1997 IEEE Electrical Performance of Electronic Packaging Conference*, 1997, pp. 53–56.
12. W. T. Weeks, L. L. Wu, M. F. McAllister, and A. Singh, "Resistive and Inductive Skin Effect in Rectangular Conductors," *IBM J. Res. Develop.* **23**, No. 6, 652–660 (1979).
13. H. Smith and M. Cases, "Wiring Rule Methodology for On-Chip Interconnects," *Proceedings of the 1996 IEEE Electrical Performance of Electronic Packaging Conference*, 1996, pp. 33–35.
14. B. D. McCredie and W. D. Becker, "Modeling, Measurement, and Simulation of Simultaneous Switching Noise," *IEEE Trans. Compon. Packag., Manuf. Technol.* **19**, 461–472 (August 1996).
15. K. L. Shepard and V. Narayanan, "Noise in Deep Submicron Digital Design," *ICCAD '96 Digest of Technical Papers*, 1996, pp. 524–531.
16. Saila Ponnappaili, Jim Lehner, and Sanjay Upreti, "Method of Extracting 3-D Capacitance and Inductance Parasitics in Sub-Micron VLSI Chip Designs Using Pattern Recognition and Parameterization," filed at U.S. Patent Office September 1997.
17. C. L. Ratzlaff, N. Gopal, and L. T. Pillage, "RICE: Rapid Interconnect Circuit Evaluator," *Proceedings of the 28th Design Automation Conference*, June 1991, pp. 555–560.
18. H. H. Chen, "Minimizing Chip-Level Simultaneous Switching Noise for High Performance Microprocessors," *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1996, pp. 544–547.
19. H. H. Chen and J. S. Neely, "Interconnect and Circuit Modeling Techniques for Full-Chip Power Supply Noise Analysis," *IEEE Trans. Compon. Packag., Manuf. Technol.* **21**, No. 3, 209 (August 1998).

Received February 4, 1999; accepted for publication July 28, 1999

**Robert M. Averill III** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (averillr@us.ibm.com).* Mr. Averill is a Senior Engineer in the S/390 Hardware Development Laboratory, Poughkeepsie, New York. In 1983 he joined IBM at the East Fishkill facility, where he developed advanced VLSI test equipment. He joined the Advanced CMOS Microprocessor group in Poughkeepsie in 1994 as a circuit designer and is currently the chip integration leader for microprocessors in IBM S/390 G5 and G6 Enterprise Servers. Mr. Averill received a B.S.E.E. degree from Northwestern University in 1983 and an M.S.E.E. degree from Syracuse University in 1990.

**Keith G. Barkley** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (barkleyk@us.ibm.com).*

**Michael A. Bowen** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (bowenm@us.ibm.com).*

**Peter J. Camporese** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (pcamp@us.ibm.com).* Mr. Camporese is an Advisory Engineer in the Custom VLSI Processor Design group for S/390 hardware development. He received a B.S. degree in electrical engineering from Polytechnic University in 1988 and an M.S. degree in computer engineering from Syracuse University in 1991. Mr. Camporese joined the IBM Data Systems Division in 1988; since then he has worked on system performance, circuit design, physical design, VLSI chip integration, custom VLSI design methodology, and tools development. Most recently, he has been a technical team leader for the physical design and integration of S/390 custom CMOS microprocessors.

**Allan H. Dansky** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (dansky@us.ibm.com).* Mr. Dansky received a B.S. degree from CCNY and an M.S. degree from Syracuse University, both in electrical engineering. He is a Senior Engineer with the IBM S/390 Division in Poughkeepsie, New York, now working on S/390 microprocessor chip wire routing, chip integration, hierarchical design methodology, and crosstalk analysis. He has written many software programs to integrate CAD tools, perform electrical checking across the design hierarchy, and calculate coupled noise between adjacent wires. From 1970 to 1992, Mr. Dansky worked on the design and evaluation of bipolar and BiCMOS circuits from five circuits to tens of thousands of circuits per chip. In the late 1970s, he performed the circuit design for a 5000-circuit "S/370" microprocessor on a chip, which received wide publicity in technical journals for its advanced placement and wiring design automation programs. In the 1980s, he worked on the development of several new high-speed, low-power bipolar and BiCMOS circuits to increase the number of bipolar circuits per chip. Mr. Dansky has authored several papers; he holds nine patents in high-performance VLSI circuit design and crosstalk calculation methodology.

**Robert F. Hatch** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (rfhatch@us.ibm.com).* Mr. Hatch is a Senior Engineer in the Custom Design Department of the S/390 Division. He received a B.S. degree in engineering from Southern Illinois University in 1976 and an

M.S. degree in electrical engineering from Purdue University in 1978. Mr. Hatch joined IBM in 1978 at the East Fishkill facility, where he did custom PLA circuit design for a custom VLSI bipolar CPU chip set. His work has been in the areas of bipolar circuit design, MOS circuit design, macro design, VLSI chip design, and VLSI design tools. Mr. Hatch is currently working on the next generation of S/390 CMOS processors.

**Dale E. Hoffman** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (daleh@us.ibm.com).* Mr. Hoffman is a Senior Engineering Design Manager in the S/390 Hardware Development Laboratory. In 1981 he joined IBM at the East Fishkill facility, where he developed and managed advanced VLSI logic and memory test systems. In 1993, he joined the Advanced CMOS Microprocessor group in Poughkeepsie as a design manager responsible for back-end design methodology, chip integration, and physical design and test for high-frequency custom microprocessors in the IBM S/390 G4, G5, and G6 Enterprise Servers. He is currently the processor subsystem manager for the next generation of advanced microprocessor and system development. Mr. Hoffman received a B.S.E.E. degree from Pennsylvania State University in 1981 and an M.S.E.E. degree from Syracuse University in 1984. He holds six U.S. patents and has several publications.

**Mark D. Mayo** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (mayom@us.ibm.com).* Mr. Mayo is a Senior Engineer in the S/390 Hardware Development Laboratory, Poughkeepsie, New York. In 1980 he joined IBM at the East Fishkill facility, where he worked on the design and development of bipolar and BiCMOS gate arrays. In 1994, he joined the S/390 CMOS high-frequency microprocessor team in Poughkeepsie and led the circuit design teams for the G4, G5, and G6 buffer control element (BCE). Mr. Mayo received a B.S.E.E. degree, with high honors, from Lehigh University in 1980 and an M.S.E.E. degree from Syracuse University in 1985. He has received two IBM Outstanding Technical Achievement Awards and an IBM Outstanding Innovation Award for his circuit development work.

**Scott A. McCabe** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (mccabes@us.ibm.com).*

**Timothy G. McNamara** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (tmcnamar@us.ibm.com).* Mr. McNamara is a Senior Engineer in S/390 custom microprocessor design. He received his B.E. degree in electrical engineering from the State University of New York at Stony Brook in 1983 and his M.S. degree in computer engineering from Syracuse University in 1990. He is currently working on high-performance clock system designs for the S/390 CMOS microprocessors. Mr. McNamara has authored or co-authored a number of technical papers; he holds three U.S. patents.

**Thomas J. McPherson** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (mcpherso@us.ibm.com).* Mr. McPherson is an Engineering Manager in S/390 microprocessor development. He received his B.S. degree in electrical engineering from Rutgers University in 1990

and his M.S. degree in computer engineering from Syracuse University in 1994. Mr. McPherson joined IBM in 1990 as a logic designer for CMOS ASIC designs. Since 1994 he has worked on CMOS microprocessor designs. Mr. McPherson was the frequency leader for the G5 and G6 microprocessors and is currently working on future S/390 microprocessors.

**Gregory A. Northrop** *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (gnorth@us.ibm.com).*

**Leon Sigal** *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (sigal@us.ibm.com).* Mr. Sigal received a B.S. degree in biomedical engineering in 1985 from the University of Iowa and an M.S. degree in electrical engineering in 1986 from the University of Wisconsin at Madison. He worked at Hewlett-Packard's microprocessor development laboratory between 1986 and 1992. Mr. Sigal joined IBM in 1992 and has since been involved in CMOS S/390 microprocessor circuit design.

**Howard H. Smith** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (smithh@us.ibm.com).* Mr. Smith received his B.S. and M.S. degrees in electrical engineering from the New Jersey Institute of Technology, Newark, in 1984 and 1985, respectively. He joined IBM in 1984 as an integrated circuit engineer at the semiconductor development laboratory in East Fishkill, New York, working in the area of high-performance masterslice designs. Mr. Smith is currently an Advisory Engineer in the IBM System/390 Division in Poughkeepsie, New York, where he is responsible for electrical analysis issues associated with high-density CMOS circuit technology and package-related products. Recent assignments have included the development and coordination of on-chip noise verification processes for the S/390 processor designs. His expertise lies in the area of electrical noise modeling and prediction in system-level computer operation. Mr. Smith has co-authored several papers on system-level noise prediction, on-chip interconnects, and electromagnetic characterization of connectors and antennas.

**David A. Webber** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (webberd@us.ibm.com).* Mr. Webber is an Advisory Engineer in IBM S/390 microprocessor development. He joined IBM in 1983 in Hopewell Junction, New York, working on the design of automated test equipment. Since 1993, he has been working on circuits and integration for S/390 CMOS microprocessors. Mr. Webber received B.S. and M.S. degrees in electrical engineering from the University of Illinois in 1981 and 1983, respectively.

**Patrick M. Williams** *IBM System/390 Division, 522 South Road, Poughkeepsie, New York 12601 (patricw@us.ibm.com).*