Design and Implementation of the POWER6 Microprocessor

Benjamin Stolt, Yonatan Mittlefehldt, Sanjay Dubey, Gaurav Mittal, Mike Lee, Joshua Friedrich, and Eric Fluhr

***Abstract*—The IBM POWER6 processor is a dual-core, 341 mm^2, 790 million transistor chip fabricated using IBM's 65 nm partially-depleted SOI process. Capable of running at frequencies up to 5 GHz in high performance applications, it can also operate under 100 W for power-sensitive applications. Traditional power-intensive and deep-pipelining techniques used in high frequency design were abandoned in favor of more power efficient circuit design methodologies. The complexity and size of POWER6, together with its high operating frequency, presented a number of significant challenges for its multi-site team to complete the design on an aggressive schedule. This paper describes some of the circuit methodology and implementation innovations used in the development of POWER6, with particular emphasis on custom, synthesized, register file and SRAM design, as well as the electrical characterizations performed in the lab.**

***Index Terms*—ABIST, array, circuit design methodology, clocking, custom circuits, dual-core, latch, LBIST, microprocessor, power, POWER6, register file, RLM methodology, 65 nm SOI process, SRAM, timing.**

I. Introduction

THE POWER6 processor is the latest generation in the POWER line of PowerPC processors. Fabricated using IBM's 65 nm partially-depleted SOI process, the 341 mm^2 POWER6 chip contains over 790 million transistors and 1953 signal I/Os connected using 4.5 km of wire on 10 copper metal layers (Fig. 1). Each chip includes two dual-threaded SMT processor cores implemented in a 13 FO4 design capable of running at speeds up to 5 GHz. In addition, a private 4 MB L2 cache per core, a shared 32 MB L3 cache controller, two integrated memory controllers, an on-board I/O controller and nest support for large-scale SMP are included on the chip. In order to provide mainframe-like reliability, enhanced error-detection and system monitoring capabilities are managed through a new recovery unit that provides full checkpointing facilities. This is supplemented by complete ECC protection of large caches and architected state, parity protection on more than 99% of register files and 70% of dataflow circuits, along with extensive control checkers. In addition, improved virtualization support and decimal floating-point execution capability provide a rich set of features, while remaining binary compatible with previous POWER designs.

POWER6 achieves over twofold improvement in frequency over POWER5 to a large part due to circuit improvements in IPC-critical logic over the previous 22 FO4 design, while maintaining the POWER5 14 stage fixed-point pipe depth [1] (Fig. 2) and without the use of power-hungry dynamic circuits outside of register files and arrays. Each of the two cores features highly tuned fixed-point pipelines supporting back-to-back dependant 64-bit execution and low-latency floating point pipelines that

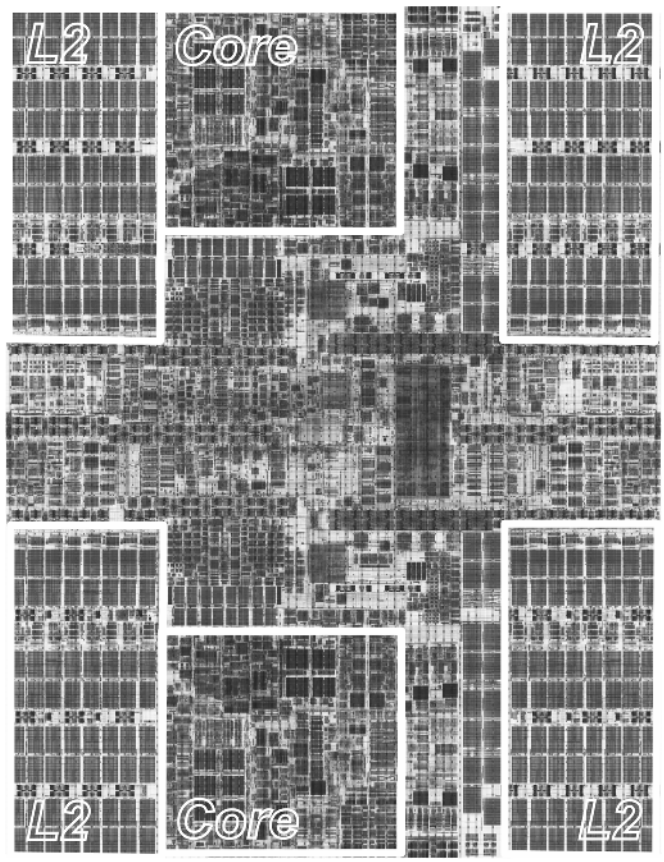

Fig. 1. POWER6 chip with cores and L2 caches highlighted.

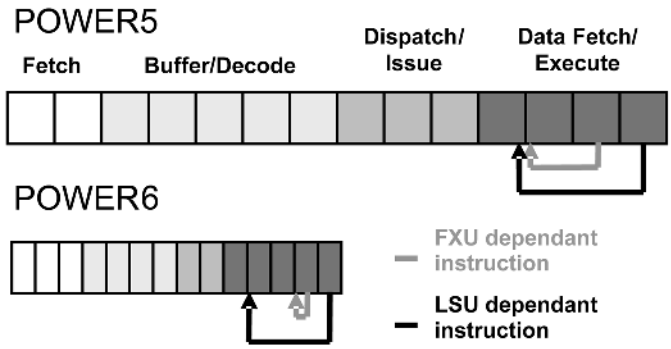

Fig. 2. POWER5 compared to POWER6 fixed-point pipeline.

Manuscript received April 17, 2007; revised September 27, 2007.

The authors are with International Business Machines Corporation, Austin, TX 78757 USA (e-mail: bstolt@us.ibm.com).

Digital Object Identifier 10.1109/JSSC.2007.910963

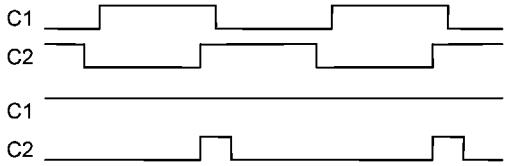

Fig. 3. Clock modes—regular and pulsed mode.

TABLE I
CLOCKING MODES

Latch mode	Clocking	Benefits
Regular mode	1.0 FO4 overlap between C2 rise and C1 fall.	Default
Delay C1 mode	C1 fall delayed; 2.1 FO4 overlap between C2 rise and C1 fall.	Frequency improvement
Delay C2 mode	C2 rise delayed by 2.5 FO4	Frequency debug
Max pulse width mode	C1 is held high and C2 is 5.2 FO4 pulse.	Power savings Frequency improvement
Mid pulse width mode	C1 is held high and C2 is 4.2 FO4 pulse	Power savings Frequency improvement
Min pulse width mode	C1 is held high and C2 is 2.9 FO4 pulse	Power savings Frequency improvement

include VMX and DFU execution capabilities. An enhanced nest design supports the high-frequency cores with twice the L2 cache and double the I/O bandwidth of POWER5 [1]. The balanced design allows for clock frequencies up to 5 GHz in high performance applications but can also operate under 100 W for power-sensitive applications due to dynamic power management and extensive use of device tuning.

II. CIRCUIT DESIGN METHODOLOGIES

A. Latch Design

The majority of state-saving devices used in POWER6, outside register files and SRAMs, are scannable master–slave flip-flops (FFs). In normal operation, each of these is controlled by two opposite phase, slightly skewed clocks, C1 and C2 that drive the master latch (L1) and slave latch (L2), respectively [2]. In order to reduce chip power, most flip-flops can be run in pulsed mode where C1 is held high while C2 is pulsed (Fig. 3). Since only one clock signal is active in this mode, switching power is reduced. Table I describes various latch modes and their clocks. Delay C1 mode allows cycle stealing during the C2 rise and C1 fall overlap, which provides the capability to shift cycle boundaries and tune frequency in the hardware. Pulsed mode allows even more cycle stealing at the cost of extra padding needed to meet tighter hold time requirements. Designs were padded for minimum pulsewidth mode (2.9 FO4), while mid (4.2 FO4) and max (5.2 FO4) pulsewidth modes were supported to provide maximum flexibility when the chip was tuned in the lab (see Section IV). Finally, a Delay C2 mode, which delayed the C2 rise, was available for debugging frequency limiting paths.

B. Library Cells

One of the driving forces behind the efficient design methodology of POWER6 was the RodRunner pcell-based gate library

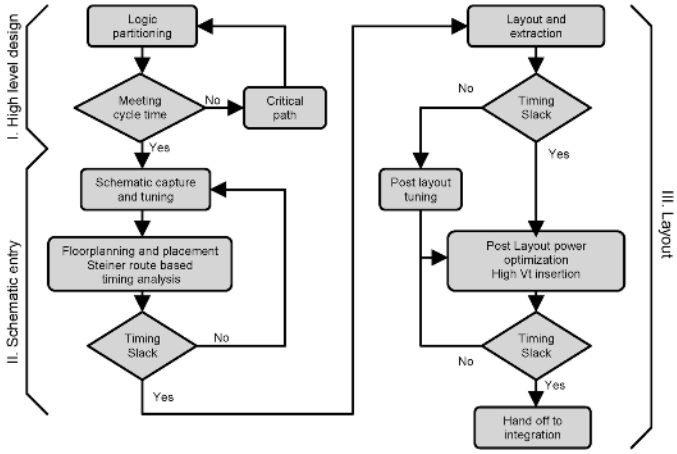

Fig. 4. Custom design flow.

that provided fine device size granularity while retaining the advantages of cell-based layout design. In addition, the resources required to create the full cell library were greatly reduced because layouts for each cell were generated and updated automatically.

For synthesized random logic macros, the use of RodRunner cells allowed a very large library of standard cells to be created giving synthesis maximum flexibility. Over 500 unique cells were available for each of three V_T types supported in the 65 nm technology, without the enormous overhead that would normally be associated with maintaining a library of that size. As many as four different beta ratios were available for each size cell with two- and three-input cells usually having multiple tapering ratios available as well.

A key benefit to RodRunner was the ability to make any DRC or methodology (METH) updates in a single location within the RodRunner cell. This change was instantly picked up across all instantiations of the cell, including in the standard cell library. While this occasionally required minor updates to existing layouts to ensure compatibility, these could be performed with minimal effort. This also allowed technology updates, which could affect transistor strengths and beta ratios, to be easily compensated for by the designer or Einstuner [6] (IBM's device tuning tool).

C. Custom Methodology

The tools used for the custom methodology maximized the possible number of iterations on a circuit, allowing the designer to rapidly approach an optimal solution. The methodology could be split into three design phases as illustrated in Fig. 4: high level design, schematic entry and placement, and layout.

During the high level design phase, physical abstracts were used to floorplan a macro as well as develop a pin/wiring contract with integrators. Early timing abstracts were generated based on circuit designer estimates of logic implementation.

Schematic entry and placement could be performed simultaneously with an innovative new tool called PIP [6] (Placement with Instance Parameters), a GUI for a library of Skill functions used to place cells. PIP allowed circuit designers to more accurately and easily floorplan and time their macros. This combined

with STEP [6] (STeiner Estimated Parasitic), allowed fairly accurate wire models to be included in early timing abstracts. A circuit topology checker could then be used to verify the circuits to ensure they met project design rules prior to layout implementation.

During the layout phase, only routing was needed as all cells had been previously placed. The use of RodRunner allowed automatic optimization tools, such as Einstuner, to easily update both schematics and layouts to improve timing, area and power by optimizing device sizing and beta ratios. Additionally, the LAVA engine [6], which performed leakage calculations based on analyzing channel-connected components, could change the $\mathrm{V_T}$ of cells, to either reduce leakage power on noncritical paths or increase the speed of a failing paths. Tools to add decoupling capacitors, gate arrays (see Section II-E) and redundant vias or to tweak n-well/rox layers could then be run on the completed layout to improve yield and performance. A number of physical checks, including DRC and LVS, methodology, DFT, extraction, power and transistor-level timing, were performed to validate the design, followed by electromigration, noise and IR drop analysis to ensure circuit reliability.

D. RLM Methodology

The synthesized random logic macro (RLM) methodology was designed to have as much commonality with customs as possible to allow maximum sharing of tools and checkers. RLMs were designed with the same bit image as customs and were generally allowed unrestricted use of M1–M3 while M4 (and higher for special cases) was shared with the unit via contracts. Pins were required to follow a more restrictive set of placement and spacing rules to provide the highest possible pin density while still ensuring accessibility to pins for both the unit and the RLM by automated routing tools.

The RLM process was broken into three major phases: synthesis and placement, routing, and physical validation. The first step was performed in an IBM tool suite that combined logical synthesis, mapping, placement and timing capabilities in an integrated framework called PDSRTL [6]. Given a VHDL design, macro dimensions with pin locations, and a set of timing contracts, PDSRTL optimized the design for timing, power, area and electrical constraints. A carefully tuned set of default parameters yielded high quality results for the majority of the designs, while at the same time these could be customized to adapt to characteristics of individual RLMs.

The second step took the fully placed RLM, pre-routed wide clock nets based on LCB and latch placement, and added fill cells (see Section II-E) before the design was run through a grid-based routing tool. A fully redundant via set could be used on 95% of the designs for increased yield, with the remaining RLMs using a mixed via set. The final routed design was translated into a standard layout by removing floorplanning information and replacing abstracts of all the standard cells with actual layouts.

At this point, the methodology aligned with the custom macro methodology and the same physical checks are performed to validate the design. Typically, RLMs were clean by construction and only required minimal tweaking to pass all requirements. Like for customs, Einstuner and LAVA $\mathrm{V_T}$ substitution were available for post-layout tuning.

E. Filler Cell and ECO Methodology

The aggressive schedule of the POWER6 design required physical design (PD) to already be in late stages while verification work was still ongoing, resulting in an unusually large number of engineering change orders (ECOs). The RLM flow was capable of automatically taking a modified (and optionally placed) netlist, merging it into the existing design and running incremental routing to update the layout. In past designs, once the front-end-of-the-line (FEOL) layers were locked and no further changes to cell sizing or placement were possible, back-end-of-the-line (BEOL) or wire-only ECO capability was severely limited by the number of spare cells of each type and their location in the macro. In POWER6, a special set of gate-array cells, each containing a single PFET and NFET device, were used to fill the unused area in both RLM and custom designs. When used as fill, these cells remained disconnected to have no impact on power, but in a BEOL ECO, they could be combined and replaced by functional cells that had the exact same FEOL layers, but connected the transistors to form any type of static gate. This capability, combined with spare latches that were scattered throughout all RLMs and some customs, allowed even the most complex changes to be completed using BEOL layers only.

For extremely complex changes where VHDL did not easily correspond to the netlist, an experimental process was introduced for RLMs that made the PDSRTL tool aware of the gate-array methodology. By describing a delta-VHDL, a designer could essentially graft a new cone of logic into the design. PDSRTL would swap fill cells for gate-array cells as needed and, where possible, reuse existing cells to map the new logic. While results were generally not as efficient as manual ECOs, this new approach proved to be extremely valuable for complex situations where an ECO would have otherwise been unfeasible.

F. Timing Methodology

In order to achieve timing closure on the POWER6 chip, parallel development at all levels of the design, rapid iteration and early timing estimation were essential in addition to highly accurate timing models. Hierarchies on POWER6 included chip, core, nest, unit, and macro levels. Using assertions or timing contracts to describe boundary conditions such as arrival times, slews and capacitance loads, a top-down methodology was used, allowing each level of the design to be analyzed and iterated independently of the others.

Only the basic best case and nominal case timing corners needed to be run separately to evaluate timing at any given level of hierarchy, due to parallel modeling of all clock phases, voltage levels and both pulsed and nonpulsed modes in a single run. A third run that modeled actual hostile capacitances and their impact on timing was used late in the design for detailed noise coupling analysis.

RLMs were treated in the same way as customs in unit, core and chip timing environments through the use of transistor-level

timing abstracts. This is a departure from previous POWER designs, where RLMs were modeled using standard cells and associated delay equations, and yielded much more consistent and accurate timing information.

III. Circuit Challenges and Implementation

A. Custom Design

Various techniques were used to efficiently implement custom macros. To increase circuit robustness and reduce design time, only static logic was used outside of arrays and register files. To keep the design flexible, macros were designed modularly, allowing large pieces of the circuits to be easily repositioned if timing or wiring constraints required. Circuits were also designed hierarchically with reusable pieces to lessen the layout effort. Since most circuits used PIP to place gates during schematic entry, layout work consisted mainly of wire routing, a process further helped by the use of a tool that highlighted flight lines for unconnected signals. This enabled circuit designers to complete layout tasks traditionally performed by mask designers.

Designers were able to manually tweak timing assertions and set gates as "no touch" to force Einstuner to focus on a particular section of the circuit, useful when the designer knew which paths were most critical. Additionally, a slack highlighting tool was available to display timing slacks directly on schematics to streamline timing takedown.

B. RLM Design

Due to the very low cycle time, high complexity and aggressive schedule of the core, POWER6 created some challenges in partitioning the design and choosing which parts could be synthesized. A number of situations were encountered, where normally noncritical control logic that was functionally unstable late in the project became critical due to the large amount of function that had to fit into a cycle. The logical uncertainty made such macros well-suited to be RLMs, but required the RLM flow to handle much harder timing problems.

These challenges were partially overcome by user control of how an RLM was synthesized and applying custom design concepts without sacrificing the automation and rapid turnaround time expected of synthesized designs. The designer could instantiate individually sized cells or fully custom blocks (e.g., an adder), optionally defining exact location. If placement was more flexible, a target region could be specified instead, allowing localized movement for legalization or fitting critical cells. Alternatively, a preferred implementation could be specified in logic to seed initial synthesis and mapping, while still allowing changes needed in later optimizations. Finally, individual nets could be preserved and wire codes defining width/spacing specified.

In addition, a certain amount of tuning could be done using controls passed to the PDSRTL flow. These ranged from full sets of parameters (e.g., a "dataflow" mode, that could emulate the top-to-bottom flow of a dataflow macro) to overriding slew, slack and density targets, to restructuring options that controlled how the VHDL was translated and initially mapped.

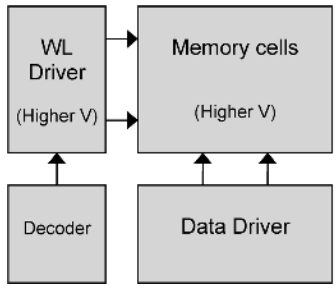

Fig. 5. Dual voltage rails.

An effective strategy for tackling the timing takedown on complex RLMs was rapid iteration of the design in a feedback loop with the unit timing environment where only small changes were introduced on each pass. Techniques such as slack apportionment and innovative tools for analyzing timing and placement allowed targeted changes to be implemented. To avoid having to wait for regular unit releases and PD updates, logic, timing contracts and physical attributes were often edited by hand or script without worrying about them being PD clean.

C. Register File Design

High frequency targets in register files were achieved using partially decoded addresses with a local multiplexed circuit and highly automated tools for functional verification, timing, and electrical checks. Both static and dynamic circuit methodologies were used. In core register files with less than 32 entries, a static circuit style was implemented, while those with more than 32 entries favored a dynamic circuit implementation, based on power, design complexity, robustness and resource required. In register files outside the core that operated at a lower frequency, static circuits were generally used, except in specific cases where cycle time could not be met. In these situations dynamic circuits were used as well. Design styles were decided upon in the early concept stage using circuit cross-sections and SPICE runs as references. Using Verity [6] (IBM functional verification tool), verification was simplified and run time substantially reduced. This method was a logical based comparison of the circuit with VHDL, rather than transistor and stimulus-based analysis, which would have taken much longer to verify. To time the register files, a method known as "donut" was used, in which only the first and last rows and the first and last columns were extensively simulated. The remaining paths were timed using the arrival times of word lines and bit lines.

D. High-Speed Array Design

One of the biggest challenges in achieving the high frequency was the SRAM design, which normally lags behind the other circuits. A dual-power rail technique was used to compensate for the longer channel lengths and higher threshold voltages. The word line drivers and memory cells were connected to a separately tunable higher voltage level than the rest of the circuitry (Fig. 5). This allowed the transfer pass gates of the memory cells to drive harder for faster reads and writes. This unique voltage control in the SRAM cell also helped to find the optimal tradeoff between performance and stability, without affecting the rest of the processor's circuits.

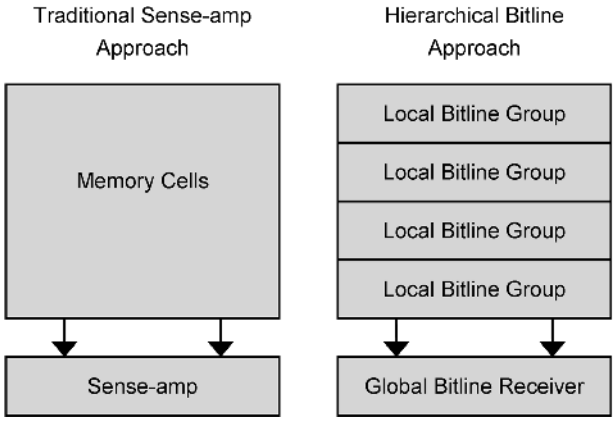

Fig. 6. Bit line read methods.

Hierarchical bit lines were also used in POWER6 to improve array performance. By staging the bit lines into hierarchies, the local bit line capacitance a memory cell saw could be small enough for the cell to fully (or almost fully) switch the bit line. The signal was then propagated hierarchically to downstream logic (Fig. 6). Using hierarchical bit lines also reduced design complexity as analog behavior simulation of the sense amplifier was avoided. In addition, the use of troublesome body contacts [3] was avoided by not using sense amplifiers.

A fast clock pulse generator, which allowed faster array access at the cost of the setup time requirement of the input signals, was also used to increase SRAM frequency. By using a self-timed pulse, the duty cycle of the clock pulse is optimized to favor the evaluate phase.

E. Timing

As the design progressed, various levels of timing accuracy were used. In early phases, zero wire delay timing helped with logical partitioning. Macros were initially estimated using timing abstracts constructed using designer cross sections and cone size calculation tools. As floorplans became more stable, Steiner distances and wire codes were introduced. Timing abstracts improved to include estimated placement, loading and sizing and were eventually replaced with extracted models. At the unit level and above, wire models moved to global routes and later detailed routes as the design progressed. Final timing analysis was done using unit-level 3-D extracted parasitic models and, for noise coupling analysis, a flat 3-D extracted model. Any of these varying levels of detail could be mixed, allowing different parts of the chip to be at different stages of the design, but still allow core and chip level timing analysis to continue using the highest level of accuracy available.

F. Integration and Clock/Power Distribution

From an integration perspective, POWER6 was designed in four levels: macro, unit, core/nest, and chip. Through the use of abstracts and wiring contracts established early in the design, these levels of hierarchy could all be designed concurrently with extensive track sharing across almost all 10 layers of metal. Abstracts also allowed designers to efficiently communicate floorplans and pin locations, leading to more accurate

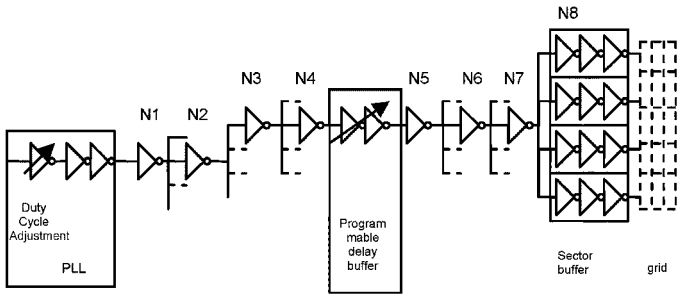

Fig. 7. Global clock distribution topology.

floorplan andwiring estimates very early in the design. Combined with the timing information generated from Steiner based routing abstracts, optimal repeater solutions could more easily be generated across all levels of hierarchy, enabling faster timing closure for the entire project.

A low-skew, high-frequency global clock distribution network was designed to support the high operating frequency. Grid-tree methodology was employed, which was similar to prior POWER designs and high-end gaming chips [4]. The clock output of the PLL was distributed using inverters and shielded high-level wires to local clock sector buffers evenly distributed throughout the chip (Fig. 7). The sector buffer in turn drove a part of a large-area clock grid through a local H-tree, which was tuned based on actual clock load to achieve minimal local clock skew. The local clock buffers were connected to the clock grid using reserved tracks to facilitate incremental update without affecting other signal wires.

POWER6 was designed to use four power domains to enable a wide range of operation, isolating logic, array and I/O supply grids from each other. All logic including flip-flops and noncritical array circuits operated at V_{DD} voltage (nominal at 1.1 V). Timing critical logic and voltage sensitive memory cells were provided a higher voltage V_{CS} supply (nominal at 1.25 V) to support higher performance and increase manufacturing yields. The off-chip interface and PLL (I/O) required a higher invariant voltage (V_{IO}; nominal at 1.8 V) for chip signaling. A fourth V_{SB} domain was used for chip power-up. These independent domains allowed circuit-type specific tuning of voltage to minimize power consumption while achieving higher frequency [5].

In order to support this many power domains, functions were carefully grouped together to eliminate the need for providing more then two voltage rails and ground to any portion of the chip. To establish the appropriate ratio of the various power grids in a chip region, current demands for each placed object were analyzed and where dual voltage domains existed, V_{IO} or V_{CS} stripes were substituted for V_{DD} based on the current splits and IR drop requirements. A robust metal grid ensured efficient power distribution, while controlled collapse chip connections (C4) were optimally placed across the chip based on actual circuit power requirements calculated using an IBM tool called ASF [6]. AC performance was improved by filling white space with decoupling capacitors.

IV. Electrical Characterization

POWER6 supports a wide variety of systems, such as low-power blades, midrange systems, high-performance enterprise

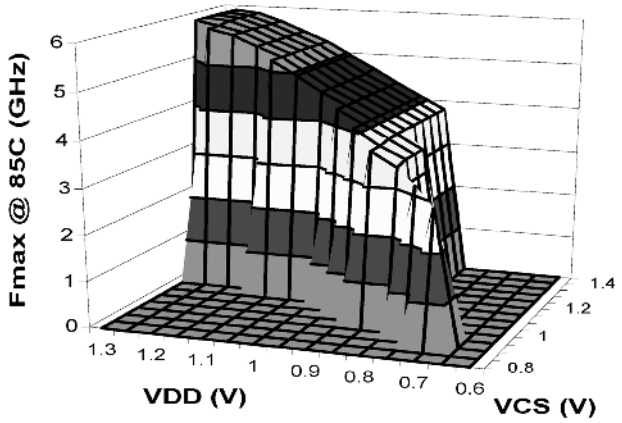

Fig. 8. Maximum frequency versus V_{DD} versus V_{CS}.

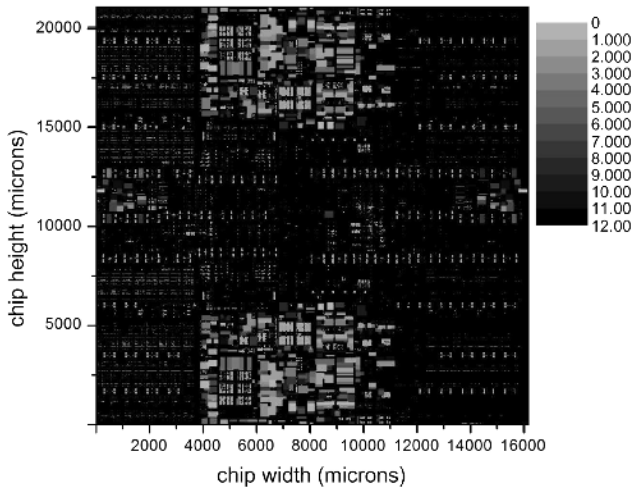

Fig. 9. Location of POWER6 critical paths (DD3 chip) for slacks <12 ps.

class servers and scientific systems, as it operates throughout a wide logic supply V_{DD} (0.75 V to 1.3 V) and SRAM supply V_{CS} (0.9 V to 1.45 V) range. The chip operates at clock frequencies up to 5 GHz in high performance applications and can also operate under 100 W for power-sensitive applications. Fig. 8 shows the operating range of a typical POWER6 chip [7]. V_{DD} and V_{CS} are optimized for each chip within the power and frequency constraint of each target system. This optimization is done at post-package module test. V_{DD} and V_{CS} for each target system are stored in an EPROM chip on the module. While this per chip optimization also increases the test time per chip, this is more than offset by the increased yield. This customized per chip voltage enables support for frequency and power targets in a variety of systems.

Extensive lab debugging, using a functional exerciser (Trash), Logic Built-In Self Test (LBIST) and Array Built-In Self Test and Repair (ABIST), was required to improve performance and yield at the V_{DD} and V_{CS} ranges. Chip logic, excluding the cores, such as memory subsystem, was designed with extra cycle-time margin. Fig. 9 shows that the cycle time critical logic paths are mainly in the two cores of the chip [7] (note that this image was regenerated for newer hardware than was available in [7]). This ensured that manufacturing variations on under-designed paths would not artificially limit

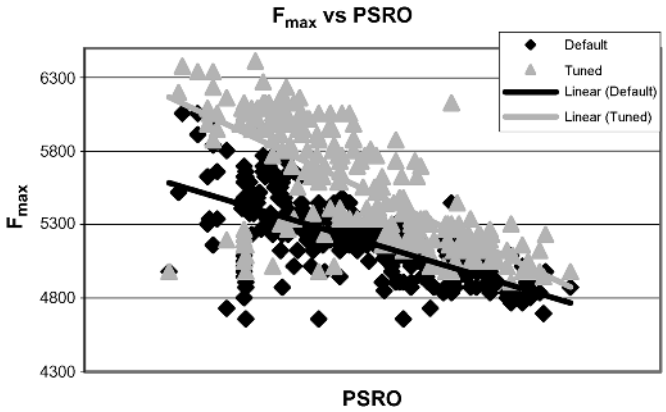

Fig. 10. F_{max} versus per-stage ring oscillator delay (PSRO).

TABLE II
SCALED FREQUENCY AND POWER FOR VARIOUS FLIP-FLOP (FF) MODES AT MANUFACTURING TEST

Latch Mode	Scaled Freq.	Scaled Power
90% of FF in Max Pulsed Width mode	1	1
90% of FF in Mid Pulsed Width mode	1	1
100% of FF in Delay Capture clock C1 mode	0.982	1.05
100% of FF in Regular Mode	0.95	1.05

the frequency of these optimized paths. It also enabled the use of the core-centric functional exerciser, Trash, which loads an instruction program from L2, as a main tool for performance characterization.

POWER6 has on-chip programmable clocks to shift cycle boundaries. SRAM arrays have pulsed clocks with a programmable clock pulsewidth and launch delay. Master–slave FF banks can be programmed in delay data capture clock mode or pulsed mode to allow cycle stealing. On-chip clock tuning resulted in up to a 10% frequency improvement as shown in Fig. 10. Pulsed mode resulted in 5% frequency and 5% power improvement (Table II).

Process splits to stress PFETs, NFETs, array cells and decaps were evaluated to improve performance and yields, resulting in a process tolerant POWER6 design. Fig. 11 shows the robustness of the design with V_T process splits. POWER6 yields are high even at $V_{DD} = 0.7$ V. Minor mask changes based on split lot evaluation resulted in significant yield improvement.

Using LBIST, which is based on Level Sensitive Scan Design (LSSD), a pseudorandom sequence was scanned into latches and functional clocks were activated for a specified number of cycles. The latch values were scanned out, compressed and matched with expected values. The scan chain was broken into 71 subsections in order to facilitate fast scanning of the test sequence. The LBIST fault coverage was very high as almost all the logic, excluding arrays, used scannable master–slave FFs. Although it provided good coverage, a high transient droop of the voltage supply rails occured during LBIST. This limits the accuracy of LBIST in POWER6 for evaluating peak frequency, resulting in the use of Trash for such characterizations. Fig. 12

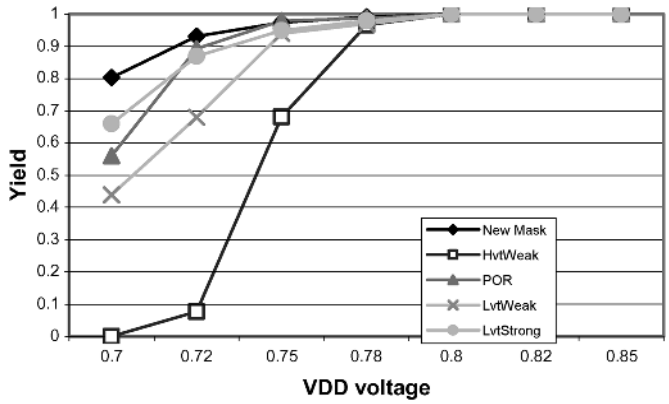

Fig. 11. Yield normalized with good chips at 1.1 V versus voltage at low frequency. "New mask" is at POR with design fixes. Weak high V_T (HV_T Weak), POR, weak low-V_T and strong low-V_T results are with old design.

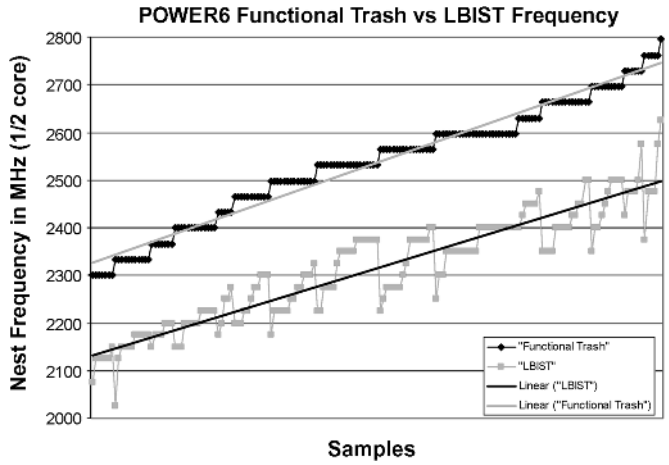

Fig. 12. POWER6 functional test versus LBIST frequency.

shows an example of LBIST frequency versus Trash frequency. Due to higher fault coverage than functional exercisers, LBIST was used at production testing to find chip failures. Eleven LBIST tests were run to find circuit faults, including static DC and frequency sensitive one and two-cycle AC tests. DC tests were run at system frequency, while one-cycle AC tests were run at 70% of system frequency and two-cycle AC tests at low frequency. One-cycle AC tests helped to find resistive faults and two-cycle AC tests were used mainly for finding hold time fails in the chip.

ABIST was used for array testing and repair, due to LBIST and Trash limitations. Faulty cells were identified and defective rows repaired with redundant rows in the array. At-speed ABIST was used for production testing of SRAM performance and fault coverage.

V. Conclusion

The POWER6 processor was chiefly designed and fabricated across seven IBM sites in two countries. The two, dual-threaded SMT cores per chip provide reliability through enhanced features, while the nest supports wide scale SMP. The innovations of the design team are credited with producing a server chip that can run at 5 GHz, while maintaining the pipeline depth of previous generations of POWER processors. New tools and methodologies were designed to assist in achieving the aggressive goals. Extensive lab characterization allowed for higher yields, while discovering areas on the chip to improve performance in later mask revisions.

Acknowledgment

Since there are far too many contributors to this project to mention individually, the authors would like to thank the entire POWER6 team for their hard work and dedication in completing a design of this magnitude. The authors would also like to thank M. Lanzerotti and J. DiLullo for taking the time to recreate Fig. 9 for use in this paper.

References

[1] B. Curran, B. McCredie, L. Sigal, E. Schwarz, B. Fleischer, Y. H. Chan, D. Webber, M. Vaden, and A. Goyal, "4 GHz+ low-latency fixed-point and binary floating-point execution units for the POWER6 processor," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2006, pp. 436–437.

[2] B. Curran, E. Fluhr, J. Paredes, L. Sigal, J. Friedrich, Y. Chan, and C. Hwang, "Power constrained high frequency circuits for the IBM POWER6 microprocessor," *IBM J. Res. Devel.*, vol. 51, pp. 715–731, Nov. 2007.

[3] K. Bernstein and N. Rohrer, *SOI Circuit Design Concepts*. New York: Kluwer, 2000, p. 20.

[4] M. G. R. Thomson, P. J. Restle, and N. K. James, "A 5 GHz duty-cycle correcting clock distribution network for the POWER6 microprocessor," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2006, pp. 1522–1529.

[5] N. James, P. Restle, J. Friedrich, B. Huott, and B. McCredie, "Comparison of split versus connected core supplies in POWER6 microprocessor," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, vol. 604, pp. 298–299.

[6] R. Berridge, B. Averill, M. Bowen, P. Williams, P. Shephard, D. Hamid, T. Rosser, D. Thomas, D. Lewis, J. DiLullo, J. Keinert, H. Smith, R. Morel, P. Camporese, and P. Dudley, "POWER6 microprocessor physical design and methodology," *IBM J. Res. Devel.*, vol. 51, pp. 685–714, Nov. 2007.

[7] J. Friedrich, B. McCredie, N. James, B. Huott, B. Curran, E. Fluhr, G. Mittal, E. Chan, Y. Chan, D. Plass, S. Chu, H. Le, L. Clark, J. Ripley, S. Taylor, J. DiLullo, and M. Lanzerotti, "Design of Power6 microprocessor," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, pp. 96–97.

Benjamin Stolt received the B.S. and M.S. with honors in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2002.

He began work at IBM as a Logic Designer on the POWER6 Instruction Dispatch Unit, but quickly branched out into timing analysis and RLM physical design. He currently owns the POWER7 completion logic and is leading the RLM effort for the Instruction Sequencing Unit.

Yonatan Mittlefehldt received the B.S. and M.S. with honors in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2002.

He joined IBM in 2003 and began working on POWER6 as a Circuit Designer. He has designed high-speed custom circuits for the Binary Floating-point and Instruction Dispatch units on POWER6, as well as for the Instruction Fetch Unit on Z6. He is currently designing custom and array circuits for the Instruction Sequencing Unit on POWER7.

Sanjay Dubey joined IBM in 2003 and has worked on several generations of POWER processors. He was the Circuit Lead for the Recovery Unit on POWER6, and currently works on large register file design for POWER7. Prior to joining IBM, he worked with Sun Microsystems Inc. from 1997 to 2003, where he was responsible for circuit methodologies and library design, in addition to working on custom circuit and array design.

Gaurav Mittal received the B. Tech degree from the Indian Institute of Technology, Delhi, in 1996, and the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1999.

He joined IBM, Austin, TX, in 2003 and performed yield, power and frequency characterization of POWER6 and POWER6+. He was also Circuit Lead for load store unit on POWER6+ and designed circuits for load store unit and register files on POWER6. Currently, he is Memory Subsystem Power Lead on POWER7. Prior to joining IBM, from 1999 to 2003, he was a Circuit Design Engineer at Sandcraft Inc., Santa Clara, CA, working on the load store unit and TLB.

Michael Lee received the B.S. and M.S. degrees in engineering from the University of Texas at Austin in 1995 and 1999, respectively.

He joined IBM in 1995 as a Custom Circuit Design Engineer and is currently working as an Array Design Engineer. His past projects include POWER 3, 4, 5, and 6.

Joshua Friedrich was the Circuit Design Lead for the POWER6 memory subsystem. After the completion of the first pass design, he led the circuit characterization of POWER6, optimizing both frequency and yield. On past POWER processors, he has led the design of several core units and been instrumental in power reduction efforts. He is currently a Chip Power Lead focusing on power reduction efforts and energy efficiency.

Eric Fluhr received the B.S.C.S. and M.S.E.E. degrees from Georgia Institute of Technology, Atlanta, in 1994 and 1996, respectively.

He then began work at IBM, Austin, TX, on POWER3 in custom and array circuit design and circuit/logic verification. Late in POWER4, he switched to load/store logic design through POWER5. For POWER6, he led the load/store physical design team, and is currently Microprocessor Technical Lead for POWER6+.