

# End-to-End Validation of Architectural Power Models

Madhu Saravana Sibi Govindan, Stephen W. Keckler, Doug Burger

Computer Architecture and Technology Laboratory  
Department of Computer Sciences  
University of Texas at Austin  
sibi, skeckler, dburger@cs.utexas.edu

September 3, 2008

## Abstract

*While researchers have invested substantial effort to build architectural power models, validating such models has proven difficult at best. In this paper, we examine the accuracy of commonly used architectural power models on a custom ASIC microprocessor. Our platform is the TRIPS system for which we have readily available high-level simulators, RTL simulators, and hardware. Access to all three levels of the design provides insight that is missing from previous published studies. First, we show that applying common architectural power models out-of-the-box to TRIPS results in an underestimate of the total power by 65%. Next, using a detailed breakdown of an accurate RTL power model (6% average error), we identify and quantify the major sources of inaccuracies in the architectural power model. Finally, we show how fixing these sources of errors decreases the inaccuracy to 24%. While further reductions are difficult due to systematic modeling error in the simulator, we conclude with recommendations to improve architectural level power modeling.*

## 1 Introduction

Power dissipation is one of the primary constraints for modern microprocessors, affecting all aspects of the system, including architecture, logic, and circuit design. Designers typically construct an architectural power model with cycle-accurate performance simulators to investigate power/performance trade-offs early in the design cycle. One such model widely used in many architectural studies is Wattch [2]. The Wattch model is integrated into the SimpleScalar microprocessor simulator [3]. Other high-level analytical power models are listed in the survey by Najm [12]. Despite the substantial effort of researchers to build such power models, validating these models has proven difficult at best. For example, although the Wattch power models are validated against three industry designs [2], applying these models to new designs and technology nodes invariably results in errors.

In this paper, we evaluate existing architectural modeling techniques by applying them to TRIPS [4], a new general-purpose architecture and identify and quantify the major sources of inaccuracy. We show that applying common power modeling methodologies to the TRIPS architecture underestimates the hardware power by 65% on the average. Using a detailed power breakdown obtained from a validated Register Transfer Level (RTL) power model of the same processor, we identify, classify and quantify the major sources of inaccuracy in the architectural power models. We classify the errors into the following categories: (1) *modeling errors*, which include poor estimates of latch and combinational logic gate counts, (2) *technology scaling errors*, which include extrapolation of gate capacitance from high level models in one process technology to the target technology, and (3) *abstraction errors* which include inaccurate estimates of architecture-level activity factors. Using insight and feedback from

the hardware and RTL models, we are able to reduce the accuracy gap between the baseline architecture power model and the hardware.

While absolute accuracy is important for power modeling, capturing relative changes in power due to differences in application behavior or changes in the microarchitecture is also critical in an effective architecture power model. We find that the relative accuracy in the baseline power model is good, tracking the changes in hardware power measurements to within 10%; the refined power models improves the relative accuracy to within 3%. We conclude this paper with recommendations for architectural power modeling based on our experience.

## 2 Related Work

We distinguish this paper from other work in power model validation by leveraging both power estimates from RTL power models and direct hardware power measurement for purposes of validation. Using the baseline TRIPS processor, we illustrate how lack of detailed design data from previous designs can affect modeling accuracy. Chen et al. [5] present a technique to validate architectural-level power estimation of a processor with a 16-bit DSP engine and a 32-bit RISC core. Their work also uses gate-level power estimates to validate the architectural-level estimates. Natarajan et al. [13] built a validated power model for the Alpha 21264 to analyze the energy implications of speculation and pipeline over-provisioning. They leverage detailed power breakdowns of Alpha 21264 published in literature for their model validations.

Shafi et al. [14] discuss a methodology to build a validated power/performance simulator of the PowerPC 405GP. In that paper, the authors describe how they used simple microbenchmarks running on a hardware prototype to populate an energy look-up table. This table is incorporated into the cycle accurate simulator for energy estimation. Our work, in contrast, leverages commonly used tools like CACTI [19], Wattch, and HotLeakage [20] for our architectural power models, with the goal of identifying how to make power models accurate early in the design cycle. Kim et al. [9], while discussing the challenges for architectural power modeling, provide guidelines for architectural power modeling. While our work has some similarities, we also quantify the various sources of inaccuracies in architectural power modeling by comparing with real hardware. Finally, the work by Mesa-Martinez et al. [10] validates architectural power models by using thermal models built with an infrared camera. Our work is similar in that we use real hardware for validation, however we use RTL power models for aiding in validation.

## 3 Overview of the TRIPS System

This section provides a brief overview of the TRIPS architecture and the prototype system used for this study. Due to space limitations, we refer the reader to [4, 8] for a detailed description of the TRIPS architecture and microarchitecture.

The TRIPS microprocessor is an implementation of the TRIPS ISA, which belongs to a class of ISAs called EDGE [4]. Figure 1 shows an annotated die photo of the TRIPS chip. Each TRIPS chip consists of two processor cores (marked as Processors 0 and 1) and a 1-MB Non-Uniform Cache Access (NUCA) L2 cache organized as 16 memory banks [8]. The processors and the NUCA L2 are connected using an on-chip network. The figure also shows the major microarchitectural units of the processor, including the register file, instruction fetch unit, L1 instruction cache, L1 data cache, and the 4x4 array of execution units. Each of these units is partitioned into smaller identical tiles which communicate with each other using well-defined control networks. The chip additionally has several data controller tiles, including two SDRAM controllers (SDC), two Direct Memory Access (DMA) controllers, an External Bus Controller (EBC) and a Chip-to-Chip (C2C) controller.

The TRIPS prototype chip is designed in a 130 nm IBM ASIC process with about 170 million transistors. To keep the design simple the TRIPS prototype chip does not implement any form of clock gating. We activate only

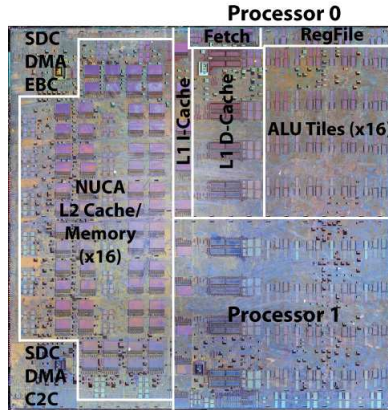


Figure 1: Annotated Die Photo of the TRIPS Chip.

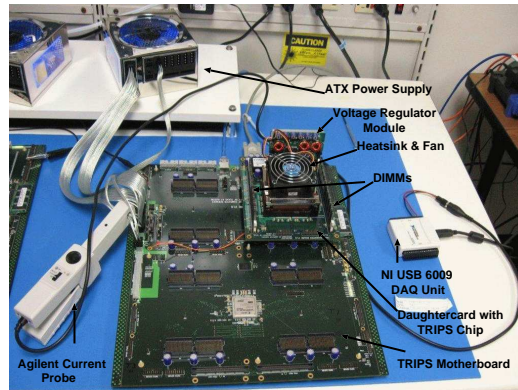


Figure 2: TRIPS Circuit Boards and Test Apparatus.

one of the two processors on the chip for this study, but account for the clock tree and idle power of the unused processor when estimating the total power (because measured hardware power is the power dissipated by the entire TRIPS chip).

Figure 2 shows a photograph of the prototype system used in this study. Each motherboard can support up to 4 TRIPS chips. Each chip gets mounted onto the motherboard via a daughtercard. The daughtercard contains one Voltage Regulator Module (VRM) that steps down the 12 V ATX power supply to 1.5 V for the TRIPS chip, a heat-sink and fan assembly, and two 1-GB DDR SDRAM DIMMs. The DIMMs receive a 2.5 V power supply from the regulator. We use the following system parameters for all our experiments: 1.5V chip power supply, 366 MHz chip clock frequency and 133/266 MHz for the DIMMs. The photo also shows the power measurement infrastructure, which is discussed further in Section 4.

The advantages of using the TRIPS prototype in this paper are twofold. First, we have readily available cycle-accurate architectural simulators, both pre-synthesized and post-synthesized TRIPS RTL netlists, and the actual hardware for direct power measurement. Second, being a new architecture, the TRIPS design can clearly illustrate the modeling errors when existing power models are applied to a new design.

## 4 Experimental Methodology

This section explains the architectural and RTL power modeling methodology. We also describe the infrastructure for hardware power measurement and our approach to isolating the power dissipated by various system components.

We use two types of benchmarks for this study. First, we run a smaller microbenchmark suite on all three levels: architectural, RTL and hardware. These microbenchmarks consist of key loops extracted from the SPEC CPU2000 [15] suite. We use these results for a detailed analysis of modeling inaccuracies and to validate our architectural power models. The low RTL simulation speed restricts us to this microbenchmark suite where each benchmark runs for 100 to 200K cycles. Second, using the insights gained from the microbenchmark results, we refine our architectural power models. We use these refined models on the EEMBC benchmark [6] suite and compare the results to measured hardware power. For our hardware runs, we suitably increase the iteration counts of the benchmarks to ensure meaningful power measurements. Also, our hardware results report the average of three runs of each benchmark.

### 4.1 Architectural Power Models

**Simulation Methodology:** Our architectural power modeling methodology shown in Figure 3 has two steps. First, we run the benchmark binary on a cycle-accurate simulator that models the TRIPS processor core (excluding the L2). At the end of this simulation, we get access counts of various microarchitectural structures in the core and a trace of all generated L2 addresses. Second, we run this L2 address trace through a cycle-accurate L2 simulator to obtain access counts of the structures in the L2 subsystem. We follow this two-step methodology (for both architectural and RTL) because full-chip RTL simulations are extremely slow. We use the same unified L2 and DIMM model for both architectural and RTL simulators of the processor core.

**Power Models:** The base architectural power is derived via commonly used power modeling methodologies. We build CACTI [19] models for all major structures such as caches, SRAM arrays, register arrays, branch predictor tables, load-store queue CAMs, and on-chip network router FIFOs to obtain a per-access-energy for each structure. This per-access-energy combined with the access counts from the simulator provides the overall energy dissipated in these structures.

The power models for integer and floating point ALUs and clock tree are derived from Wattch [2] using linear technology scaling from the built-in 350nm technology of Wattch. We model global clock drivers, global clock tree interconnect, pre-charge transistors and pipeline latches. We estimate the number of latches in each tile based on a detailed microarchitecture specification. The per-latch capacitance estimates are derived from Wattch as well.

To estimate control logic and interconnect power, we use rules-of-thumb to estimate the control logic gate counts and the average gate capacitance. From our experience with our ASIC design, we assume the tile gate count is about four times the tile latch count. Given the gate counts, we use a proprietary rule-of-thumb in the IBM ASIC documentation to estimate the total gate capacitance. Using these estimates and models based on Rent’s rule [16], we estimate the control logic and interconnect access energies of the various tiles. These energies combined with various event counts of the tiles provide the total energy dissipated for control logic and interconnect.

We build leakage power models for all array structures based on HotLeakage [20]. Leakage power estimates for non-array structures are based on gate-count estimates and average transistor density estimates. We use an analytical power model for the DIMMs obtained from Micron for both architectural and RTL power models [11].

### 4.2 RTL Power Modeling

Figure 4 describes our RTL power modeling methodology. First, we run the benchmark on a Synopsys VCS [18]-based processor-level RTL simulator, which uses a pre-synthesized RTL netlist of the design. This simulation produces a set of Switching Activity Interchange Format (SAIF) files. Next, we feed the L2 address trace obtained

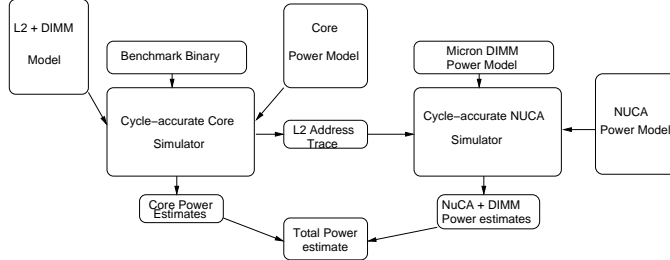


Figure 3: Architectural Simulation Methodology

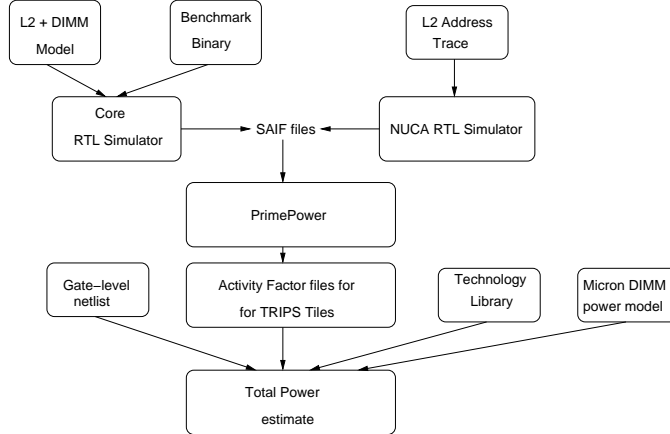


Figure 4: RTL Simulation Methodology

from the architectural simulations (Figure 3) to the NUCA RTL simulator to obtain the L2 cache SAIF files. These SAIF files represent the toggle counts of the various nodes in the pre-synthesized netlist of the design. We use Synopsys Primepower [17] to propagate these toggle counts to a post-synthesized, gate-level netlist and obtain an average switching activity for each tile in the core and the L2 subsystem. Combining this average activity factor for each tile with the total capacitance estimate from the gate-level netlist and the IBM Standard Cell library, we estimate the average dynamic power. We obtain the capacitance of the gates and global clock buffers from IBM cell library. We again estimate the interconnect capacitance using Rent’s Rule as published in [16]. We also obtain the PFET and the NFET widths of various IBM cells from the library to estimate the leakage power.

### 4.3 Hardware Power Measurement

Figure 2 shows the hardware power measurement infrastructure attached to the TRIPS board. We use an Agilent 1146A clamp-on current probe for measuring the power consumption of the TRIPS daughtercard. The voltage output of the probe is sampled by a National Instruments (NI) USB 6009 Data Acquisition System at the rate of 10 KHz and is logged to a PC using the NI Data Logger program.

**Motherboard Power:** The 12-V supply of the ATX power supply, in addition to powering the daughtercards, also supplies power to DDR termination voltages on the motherboard. We measure this power after removing the daughtercard and note it as 2.5 Watts. The fan and heatsink assembly consumes about 0.8 Watts. Thus we deduct 3.3 Watts power from all the measured power.

**DRAM DIMMs:** To measure the power consumed by the DDR DIMMs on the daughtercard, we unplug the DIMMs, reset the TRIPS chip, disable the PLL (Phased Lock Loop) needed for the DIMMs, run the chip at 366MHz, and measure the power. We repeat the experiment with the DIMMs plugged in and their PLL enabled to generate 133/266MHz clock and measure the power. The difference between these power measurements is about 3.6 Watts and is attributed to the DIMMs. We also repeat the experiment with the chip running at 100 and 200 MHz to verify that the results match.

**Voltage Regulator Module:** As mentioned before, a VRM on the daughtercard supplies 1.5 V for the TRIPS chip. To accommodate for the typical 85-90% efficiencies of VRMs [1], we derate the measured power (after deducting the 3.2 Watts for the motherboard and 3.6 Watts for the DIMMs) by 10%. Finally, we report the total power as the sum of the derated chip power and the DIMM power.

**Frequency Dependence:** Finally, we attempt to isolate the clock tree portion of the total power. To this end, we run the chip in the idle mode at 100 and 366 MHz and measure the dissipated power. Since the chip is idle in both cases, we use the linear dependence between clock frequency and power with these two data points to isolate the clock tree power. We interpolate the clock tree power model and confirm that it matches the measured power at 200 MHz. In total, we estimate the clock tree to consume 18.3 Watts at 366 MHz. The absence of clock gating in the TRIPS chip is attributed to the relatively high clock tree power.

## 5 Power Comparison Results

For an architectural power model to be useful, it must be accurate (1) in its estimate of absolute power consumption, and (2) in its estimates of the relative power consumed across different programs or architectural changes. Figure 5 compares the base architecture power estimates (the bar labeled **Base**) to RTL estimates and hardware power. We observe that the baseline architectural power model underestimates the total power by 65% compared to the hardware power. We also find that the RTL power estimates are much more accurate and within 6% of the measured hardware power. As shown in Table 1, we break down the RTL power estimates into power categories not visible via hardware measurement to identify the root source of errors in the baseline architecture power model and to derive improvements to the power model. At the end, our tuned power model is within 24% of the hardware power and tracks reasonably well across the different benchmarks.

### 5.1 Sources of Inaccuracy

Table 1 shows a breakdown of the average power estimate of the microbenchmarks into major categories like dynamic power due to combinational logic, array structures, ALUs, interconnect, clock tree including the latches and clock buffers, leakage power and power dissipated in the DIMMs along with the fraction of the total error caused by each category in Column 4. Using this breakdown, we focus our attention on the major sources of error namely latches, clock buffers and control logic power. Errors in these categories of power can stem from underestimates in counts (latch counts, gate counts, etc) and underestimates in capacitances.

**Latch Counts:** We estimate the number of latches based on a detailed microarchitecture specification for each tile in the TRIPS design. Upon a detailed analysis, the architectural model underestimates the latch counts by 53%. First, the architectural estimates are based on microarchitectural specifications which invariably change during actual RTL design. Second, certain structures in the TRIPS design like Load-Store Queue Content-Addressable Memories (CAMs), FIFOs, etc, which are expected to be custom SRAM arrays, had to be implemented out of discrete latches due to lack of suitable dense structures in the ASIC library. These latches, which account for 40% of the actual latch count, are not included in the initial architectural estimates. After accounting for these additional latches, the architectural latch estimates underestimates the latch count by an additional 13%. We attribute to this error to the mismatch between architectural specifications and the actual RTL design.

**Latch Capacitance:** The architectural latch capacitance estimates come from Wattch, after suitable technology scaling and the RTL estimates come from the IBM Standard Cell library. The architectural models underestimate the per-latch capacitance by 40%. First, the estimates of Wattch are based on the Alpha processor family, a custom-designed processor whereas TRIPS is based on a conservative ASIC design methodology. Second, the technology scaling involved in the estimates of Wattch is another source of inaccuracy. The errors in latch counts and latch capacitances contribute 54% of the overall error (Row 4 in Table 1).

Category	Arch(W)	RTL(W)	Fraction of Total Error
Control Logic + Arrays + ALUs	1.91	5.94	0.21
Interconnection	0.47	1.27	0.04
Clock Buffers	0.13	3.30	0.16
Latches	4.21	14.56	0.54
Leakage	1.36	1.91	0.03
DIMMs	3.44	3.61	0.01
Total	11.52	30.84	1.00

Table 1: Detailed Power Breakdown

**Clock Buffer Counts:** The number and capacitance of clock buffers in our architectural power model come from Wattch. The architectural models underestimate the number of clock buffers in the design by 33%. Additionally, IBM requires Level-Sensitive Scan Design (LSSD) based latches for testability [7]. Due to this requirement, the final TRIPS clock tree has many clock-splitters [7] (about 30K splitters), which are not accounted for in the initial architectural power estimates. This mismatch in the number of clock-splitters causes an average error of about 16% in the total power estimate (Row 3 in Table 1).

**Control Logic Power:** Modeling the dynamic power of complex combinational (or control logic) is a major challenge for architectural power because it is hard to accurately estimate gate counts/capacitance and average activity factors. As mentioned in Section 4, we estimate the control logic capacitance based on rules-of-thumb for gate-counts and gate capacitances. A detailed analysis shows that the capacitance estimates based on rules-of-thumb underestimate the actual capacitance by 35%.

The real challenge for control logic power is estimating the average activity factor at the architectural level [9] because of the inherent difference in the level of abstraction between architectural and RTL models. We estimate the control logic power based on an event-based model in the architectural simulator. Despite including most events relevant to the power model, this approach underestimates the average activity factor by 65%. These differences combined cause a 21% error attributed to both the control logic and the array power (Row 1 in Table 1).

**Others:** The architectural power models turn out to be fairly accurate for other power components like the interconnect power. However, since the TRIPS chip is implemented at 130nm technology leakage power is not a major fraction of the overall power. The analytical models for the Micron DIMM are also reasonably accurate and are within 4% of the measured DIMM power.

## 5.2 Discussion

We classify the errors identified above into three categories.

*Modeling errors* mainly include estimation errors in the power models. For example, our architectural models underestimate the number of latches, clock-splitters, and gate counts of control logic due to various reasons mentioned above. Possible causes of such errors include artifacts of the design methodology (additional latches and clock-splitters in our case) or a mismatch between specifications and actual RTL design. While a few of the above-mentioned modeling errors are specific only to ASIC designs, this class of errors affects power models for custom designs as well.

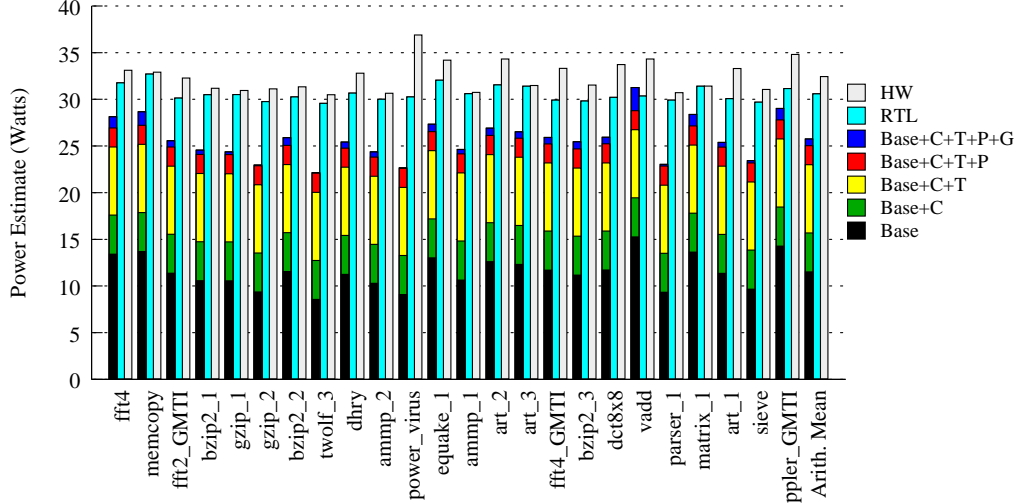


Figure 5: TRIPS Estimated and Measured Power.

*Technology scaling errors* are caused by errors in the capacitance estimates of the power models. The 40% underestimate of the per-latch capacitance in our model is an example of a technology scaling error. The assumption of a simple linear scaling model and differences in design methodologies (custom versus ASIC) are typical causes of technology scaling errors. Technology scaling errors are a common problem to all architectural models irrespective of design methodologies.

*Abstraction errors* arise from a lack of detail in the architectural simulators. Errors in the estimation of activity factor at the architectural level and differences between the architectural and RTL performance models are abstraction errors. Architectural simulators tend to trade-off detailed modeling to the speed of simulation which is a major source of abstraction errors.

In our architectural power models, technology scaling errors are the most important contributor to the overall error followed by abstraction errors and modeling errors. While addressing technology errors and modeling errors might be possible with a detailed analysis, abstraction errors are a fundamental challenge to accurate power models.

### 5.3 Improved Architectural Models

Using the insights gained from the above analysis, we evaluate a series of architecture power models that incrementally fix classes of errors to improve accuracy. Figure 5 shows the power estimates of the architectural power models for the microbenchmark suite. For each benchmark, the graph shows three bars: architectural power estimates, RTL power estimates, and measured hardware power. The architectural bar has five segments, each representing a different architectural power model. **Base** represents our baseline architectural power model as explained in Section 4. As discussed before, our **Base** model underestimates the total power by 65%, while the absolute RTL estimates are reasonably accurate.

In the **Base+C** model, we fix most of modeling errors introduced by latch and clock-splitter counts. However, we include neither the underestimate of latches (13%) due to differences between the specifications and the RTL nor the underestimate of buffers in the clock tree (33%). Also, the technology models for capacitance and the control logic power estimates are from the original **Base** model. The **Base+C+T** model fixes all the technology scaling errors in the latch capacitance and clock buffer capacitance by using estimates from the IBM Standard Cell library. In the **Base+C+T+P** model, we include the additional 13% latches and 33% clock buffers to fix all errors in the clock tree power. In the **Base+C+T+P+G** model, we replace the gate count estimates for various tiles based on rules-of-thumb by the actual gate counts of the tiles.



Figure 5 shows the incremental accuracy improvement for the various architectural power models. The **Base+C** model, which fixes the modeling errors related to the clock tree, reduces the overall error by 13% compared to **Base**. Fixing the technology scaling errors in the **Base+C+T** model provides an additional error reduction of 22%. The **Base+C+T+P** model with a perfect clock tree model reduces the overall error by 6%. Finally, the actual gate counts in the **Base+C+T+P+G** model reduces the error by a small amount of 2%. The marginal reduction in error in the **Base+C+T+P+G** model is due to two reasons: (1) the original rules-of-thumb for control logic capacitance estimation are reasonably accurate, and (2) the actual gate counts for a few tiles are less than the rule-of-thumb estimates, which tends to negate the accuracy improvement of actual gate counts. Thus, power estimates obtained using the **Base+C+T+P+G** model are within 21% of measured hardware power for the microbenchmark suite. We also apply the **Base+C+T+P+G** models to the EEMBC suite and observe that on an average the architectural estimates are within 24% of hardware power.

Differences in the power models for control logic, interconnects, leakage, and the DIMMs cause the remaining discrepancy between modeled and measured power. We identify that about 89% of the remaining error is caused by lack of detailed, bit-level switching activity data - a type of abstraction error - in the architectural power models for control logic (64%), interconnects(17%) and the DIMMs(8%). We attribute the remaining 11% error to architectural leakage models which lack detailed transistor width data: a combination of modeling and abstraction errors.

While inaccuracies remain in the absolute power, the architecture power models track the changes in power consumption across the benchmarks much more closely. We measure this *relative* power by measuring relative increase or decrease in power on a benchmark from the arithmetic mean across all the benchmarks for both the power models and the hardware. If the relative increase or decrease the architectural models closely tracks that of the hardware, then models track well. The results show that all the architectural power models track the hardware results very closely, and that on average **Base** tracks the hardware to within about 10%. The average relative accuracy improves to within 3% with **Base+C+T+P+G** model. However, some programs such as *power\_virus*, exhibit large absolute error and large relative error (25%).

## 6 Conclusions

In this paper, we developed and evaluated a series of architecture-level power models for a new processor. Our experience shows that applying commonly used power modeling methodologies results in a more than a factor of two underestimate in absolute power consumption. The underestimate stems from errors in estimating latch count, gate count, clock tree, and logic gate capacitance. While refining these estimates with feedback from the final design improves the accuracy to within 24%, yet more empirical data from the final design is needed. These results point to the difficulty in building architecture power models from the ground up and provide guidance on where to focus attention in architecture-level power models:

- Clock Tree: Because of the dominance of the clock in power modeling, architects must do a careful job in clock tree power modeling. Accurate estimates of latch counts are critical, and must take into account anticipated changes—more latches and clock splitters in our case—due to artifacts of the design methodology, especially for ASIC designs. Very early clock-tree design combined with estimates from previous generations can definitely help this process. Clock tree power estimation will be even more difficult for designs that implement clock gating and dynamic voltage/frequency scaling. While our work is only a step in this direction, more research is needed for designs with clock gating.
- Technology models: While the power models in existing high-level tools such as Cacti and Wattch may have once been validated with a particular technology node, most architects employ simple scaling rules to estimate feature size and capacitance is smaller technologies. While this scaling may be appropriate in

some cases, our experience with an ASIC technology indicates that actual gate capacitances were higher than anticipated. Because custom technologies at small feature sizes may not match linear scaling, more detailed models of such technologies would improve power model accuracy.

- **Unstructured Logic:** In comparison to memory and regular datapath structures, estimating size and complexity of the control logic is challenging and often overlooked in architecture power models. In our experience, estimating the gate count of various units in the processor is key to power estimation of combinational logic power. Developing good rules of thumb will greatly assist in the accuracy of future power models.

While estimating absolute power consumption is particularly difficult, we did find that the relative power from the architecture models tracked the power measured in hardware reasonably well across the programs in our benchmark suite. This observation bodes well for architecture studies that seek to compare relative power consumption across different applications and architecture configurations as long as the modeling, abstraction, and technology modeling errors in the power models are shared in common mode across the configurations.

## References

- [1] W. L. Bircher, M. Valluri, J. Law, and L. K. John. Runtime identification of microprocessor energy saving opportunities. In *International Symposium on Low Power Electronics and Design*, pages 275–280, August 2005.
- [2] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *International Symposium on Computer architecture*, pages 83–94, May 2000.
- [3] D. Burger and T. M. Austin. The simplescalar tool set, version 2.0. *SIGARCH Computer Architecture News*, 25(3):13–25, 1997.
- [4] D. Burger, S. W. Keckler, K. S. McKinley, M. Dahlin, L. K. John, C. Lin, C. R. Moore, J. Burrill, R. G. McDonald, W. Yoder, and the TRIPS Team. Scaling to the End of Silicon with EDGE architectures. *IEEE Computer*, 37(7):44–55, July 2004.
- [5] R. Y. Chen, R. M. Owens, M. J. Irwin, and R. S. Bajwa. Validation of an architectural level power analysis technique. In *Design Automation Conference*, pages 242–245, June 1998.
- [6] <http://www.eembc.org>.
- [7] J. Engel, T. Guzowski, A. Hunt, D. Lackey, L. Pickup, R. Proctor, K. Reynolds, A. Rincon, and D. Stauffer. Design methodology for IBM ASIC products. *IBM Journal of Research and Development*, 40(4):387–406, July 1996.
- [8] C. Kim, D. Burger, and S. W. Keckler. An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches. In *International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 211–222, October 2002.
- [9] N. S. Kim, T. Austin, T. Mudge, and D. Grunwald. Challenges for architectural level power modeling. In *Power Aware Computing*, pages 317–337. Kluwer Academic Publishers, Norwell, MA, 2002.
- [10] F. J. Mesa-Martinez, J. Nayfach-Battilana, and J. Renau. Power model validation through thermal measurements. In *International Symposium on Computer Architecture*, pages 302–311, June 2007.

- [11] Micron Technology Incorporated. Calculating DDR Memory System Power. <http://download.micron.com/pdf/technotes/ddr/TN4603.pdf>, 2001.
- [12] F. N. Najm. A survey of power estimation techniques in VLSI circuits. *IEEE Transactions on Very Large Scale Integrated Systems*, 2(4):446–455, December 1994.
- [13] K. Natarajan, H. Hanson, S. W. Keckler, C. R. Moore, and D. Burger. Microprocessor pipeline energy analysis. In *International Symposium on Low Power Electronics and Design*, pages 282–287, August 2003.
- [14] H. Shafi, P. J. Bohrer, J. Phelan, C. A. Rusu, and J. L. Peterson. Design and validation of a performance and power simulator for powerpc systems. *IBM Journal of Research and Development*, 47(5/6):641–651, 2003.
- [15] <http://www.spec.org>.
- [16] D. Stroobandt and J. V. Campenhout. Accurate interconnection length estimations for predictions early in the design cycle. In *VLSI Design, Special Issue on Physical Design in Deep Submicron*, volume 10, pages 1–20, 1999.
- [17] Synopsys, Inc. Primepower: Full-chip dynamic power analysis for multi-million gate designs. [www.synopsys.com/products/power/primepower\\_ds.pdf](http://www.synopsys.com/products/power/primepower_ds.pdf).
- [18] Synopsys, Inc. VCS: Comprehensive RTL verification solution. <http://www.synopsys.com/products/simulation/simulation.html>.
- [19] D. Tarjan, S. Thoziyoor, and N. Jouppi. Cacti 4.0. Technical Report HPL-2006-86, HP Labs, 2006.
- [20] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. HotLeakage: A temperature-aware model of subthreshold and gate leakage for architects. Technical Report CS-2003-05, University of Virginia, Department of Computer Science, March 2003.