Appears in the Proceedings of the 14^{th} International Symposium on Low Power Electronics and Design

End-to-End Validation of Architectural Power Models

Madhu S. Sibi Govindan
Dept. of Computer Sciences
University of Texas at Austin
sibi@cs.utexas.edu

Stephen W. Keckler
Dept. of Computer Sciences
University of Texas at Austin
skeckler@cs.utexas.edu

Doug Burger
Microsoft Research
dburger@microsoft.com

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 using the TRIPS system as a case study. We use the TRIPS processor because we have ready access to the TRIPS architectural simulators, RTL simulators, and hardware. Access to all three levels of the design provides key insights that are missing from previously published power validation 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 errors in the simulator, we conclude with recommendations on where to focus attention when building architectural power models.

Categories and Subject Descriptors

C.0 [**General**]: Modeling of Computer Architecture; B.6.3 [**Logic Design**]: Simulation; B.0 [**Hardware**]: General

General Terms

Design, Experimentation, Measurement

Keywords

Architectural Power Models, Validation, and Measurement

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. The most commonly used power model in academic architectural studies is Wattch [2]. Other high-level analytical power models are listed in the survey by Najm [11]. Despite substantial effort by researchers to build such power models, validating these models has proven difficult at best. The absolute power estimates of Wattch are validated to within 30% for three industrial designs [2]. Despite such validation efforts, applying such models to novel architectures and new process technologies invariably results in errors.

In this paper, we evaluate the accuracy of existing architectural power models by presenting a case study with the TRIPS microprocessor [3]. We use the TRIPS system in this study because we have ready access to TRIPS architectural simulators, RTL simulators, and hardware. 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. Using feedback from the hardware and RTL models, we reduce the accuracy gap between the baseline architecture power model and the hardware. Despite poor absolute accuracy, the baseline architectural power models have good *relative accuracy* to begin with (10%) and the relative accuracy improves with absolute accuracy (to 3%).

The following are the key contributions of this paper: (1) To the best of our knowledge, our work is the first to leverage all three design layers (architectural, RTL, and hardware) for validating architectural power models. (2) Although conventional wisdom *qualitatively* identifies sources of errors in architectural power models, our paper *quantitatively* identifies these errors when applied on a new architecture. (3) We observe that the *relative accuracy* in architectural power models is still very good, despite poor absolute accuracy, which bodes well for using architectural power models to make high-level design trade-offs.

2. RELATED WORK

We distinguish this paper from other work in power model validation by leveraging power estimates from both RTL models and hardware power measurement for purposes of validation. Chen et al. [4] 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. [12] built a validated power model for Alpha 21264 processor 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 and performance simulator of the PowerPC 405GP. In that paper, the authors use simple microbenchmarks on a hardware prototype to build an energy look-up table, which is incorporated into an architectural simulator for energy estimation. In contrast, our work leverages commonly used tools like CACTI [17], Wattch, and HotLeakage [18] for our architectural power models with the goal of making those power models more accurate. Kim et al. [8], while discussing the challenges for architectural power modeling, provide guidelines for architectural power modeling. While our work has some similarities, we additionally quantify the various sources of inaccuracies in architectural power modeling by comparing with real hardware. Finally, the work by Mesa-Martinez et al. [9] 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, but we also use RTL power models for validation.

ISLPED'09, August 19–21, 2009, San Francisco, California, USA.

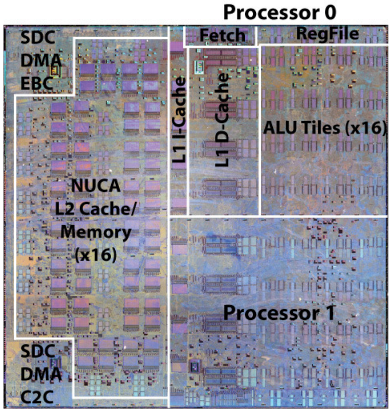

Figure 1: Annotated Die Photo of the TRIPS Chip

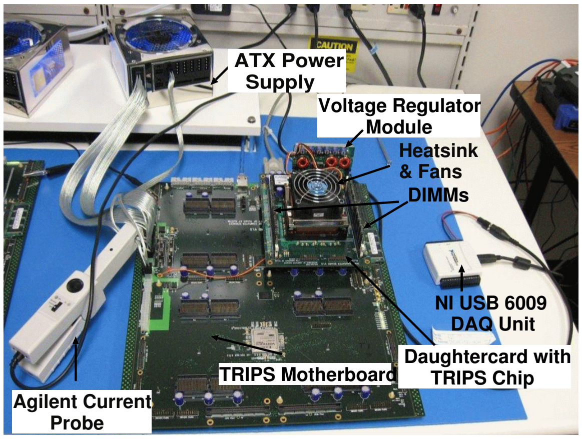

Figure 2: TRIPS Circuit Boards and Test Apparatus

3. OVERVIEW OF THE TRIPS SYSTEM

The TRIPS microprocessor is an implementation of the TRIPS ISA, which belongs to a class of ISAs called EDGE [3]. 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 [7]. 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, two DMA controllers, an External Bus Controller, and a Chip-to-Chip controller.

The TRIPS prototype chip is designed in a 130 nm IBM ASIC process with approximately 170 million transistors. To keep the design simple, the TRIPS prototype chip does not implement any form of clock gating. We activate only 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. 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 TRIPS motherboard can support up to 4 TRIPS chips. Each chip is mounted onto the motherboard via a daughtercard. The daughtercard contains one Voltage Regulator Module (VRM) that steps down the 12V 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 2.5V 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.

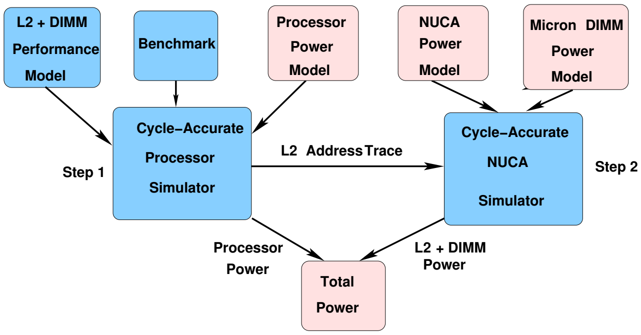

Figure 3: Architectural Simulation Methodology

4. EXPERIMENTAL METHODOLOGY

We use two types of benchmarks in this study. First, we run a smaller microbenchmark suite, consisting of key loops extracted from the SPEC CPU2000 suite, on all three levels: architectural, RTL and hardware. 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 suite [5] and compare the results to the measured hardware power. For our hardware runs, we suitably increase the iteration counts of the benchmarks to ensure meaningful power measurements, and we report the average of three runs of each benchmark.

4.1 Architectural Power Models

Simulation Methodology: Our architectural simulation 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 collect 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. Since full-chip RTL simulations are extremely slow, we follow this two-step methodology to allow reasonably long simulations. Although architectural simulators are orders of magnitude faster than RTL simulators, we use the same two-step methodology at both levels to ensure consistency in the performance models. We use the same unified L2 and DIMM model for both architectural and RTL processor simulators.

Power Models: The base architectural power is derived via commonly used power modeling methodologies. We build CACTI [17] 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]. We derive these models at 130nm by using linear technology scaling from the built-in 350nm technology of Wattch. We model global clock drivers,

Tile Name	G/L Ratio
Chip-to-Chip Controller	3.56
DMA Controller	14.23
External Bus Controller	9.50
Instruction Cache	2.23
SDRAM Memory Controller	3.96
Data Cache and Load-Store Queues	5.54
Execution Tile (Issue Logic, ALUs)	8.57
Global Control Tile	4.15
L2 Cache Banks	3.45
Register Tile	5.19
L2 Router Tile	4.41

Table 1: Control Logic Ratios

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.

Analytical estimation of combinational or control logic power is challenging at the architectural level. As one of the key contributions of this work, we propose simple rules-of-thumb to estimate control logic power. We assume that the total gate count for a TRIPS tile is a constant (about four) times the number of latches in the tile. Table 1 shows the gate-to-latch ratio of various TRIPS tiles based on a detailed analysis of the post-synthesized netlist. We observe that the rule, despite being simple, holds for most of the TRIPS tiles with notable exceptions being DMA (Direct Memory Access Controller), EBC (External Bus Controller) and the Execution Tile, which are control-logic intensive and have relatively less storage when compared to other tiles. Excluding the DMA and the EBC, which are not used in this study, the arithmetic mean of the gate-to-latch ratio is 4.56 with a standard deviation of 1.8. Although such simple rules must be fine-tuned before applying to other architectures, the key take-away is that applying even simple rules-of-thumb for control logic improves the accuracy of architectural power models significantly.

Given the gate counts, we use another rule-of-thumb, similar to equation (2) in [13], to estimate the total gate capacitance. The value of C_{avg}, a high-level estimate of the average gate capacitance, is obtained from the documentation of IBM 130nm ASIC process. Using these gate capacitance estimates and models based on Rent's rule [15], 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 control logic and interconnect energies. We build leakage power models for all array structures based on HotLeakage [18], and leakage models 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 [10].

4.2 RTL Power Modeling

Figure 4 describes our RTL power modeling methodology. First, we run the benchmark through a Synopys VCS [16]-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 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 Synopys Primepower [16] 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

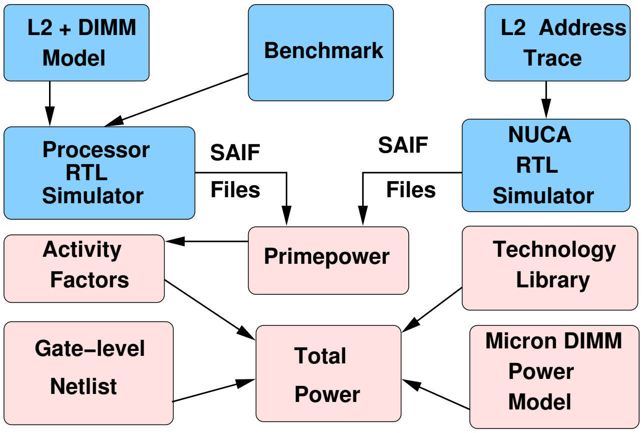

Figure 4: RTL Simulation Methodology

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 [15]. We also obtain the PFET and the NFET widths of various IBM cells from the library to estimate the leakage power. We use this methodology because (1) architectural versus RTL analysis requires fine-grained breakdown of the total power into different categories like clock tree, latches, and control logic for a head-to-head comparison of architectural and RTL power models, and (2) tools like Primepower have limited flexibility in generating such fine-grained power breakdowns.

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 by data logging software.

Motherboard Power: The 12V 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, about 3.6 Watts, is attributed to the DIMMs. We repeat the experiment with the TRIPS chip running at 100 and 200 MHz to further verify DIMM power consumption.

Voltage Regulator Module: As mentioned before, a VRM on the daughtercard supplies 1.5V 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%. We report the total power as the sum of the derated chip power and the DIMM power.

Frequency Dependence: Finally, to isolate the clock tree portion of the total power, 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

	Arch	RTL	Fraction of
Category	(Watts)	(Watts)	Total Error
Control Logic + ALUs + Arrays	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 2: Detailed Power Breakdown

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 configurations. Figure 5 compares the base architectural power estimates (the bar labeled **Base**) to RTL estimates and hardware power. While the graph plots the arithmetic mean of estimated and measured power for all 24 benchmarks in our suite, it only plots 12 individual benchmark samples for clarity. Our baseline architectural power model underestimates the total power by 65% compared to the hardware power whereas the RTL power estimates are much more accurate and within 6% of the measured hardware power. We use the RTL power estimates to validate and improve our architectural power models to within 24% of the hardware power, comparable to Wattch, which was within 30% of published industrial data.

5.1 Sources of Inaccuracy

Table 2 shows a breakdown of the average power estimate of the microbenchmarks into major categories 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.

Latch Counts: We estimate the number of latches based on a detailed microarchitecture specification for each TRIPS tile – we include all architecturally-visible state, latches in various pipelines, and latches used as temporary buffers in the tiles. A detailed analysis shows that our baseline 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), and FIFOs, which are expected to be custom SRAM arrays, had to be implemented as 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 13%. We attribute this error to the mismatch between architectural specifications and the actual RTL design – a common flaw in typical industry design flows where the module specifications are not updated during the later phases of the design.

Latch Capacitance: While architectural latch capacitance estimates come from Wattch, after suitable technology scaling, the RTL estimates are derived from the IBM Standard Cell library. The architectural models underestimate the per-latch capacitance by 40%. 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. 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 2).

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 LSSD-based (Level-Sensitive Scan Design) latches for testability [6]. Due to this requirement, the final TRIPS clock tree has many clock-splitters [6] (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 2).

Control Logic Power: Modeling the dynamic power of combinational logic is a major challenge for architectural models because it is hard to accurately estimate gate counts, gate capacitances, and activity factors. As mentioned in Section 4, one of the contributions of our work is proposing and evaluating new rules-of-thumb for estimating gate-counts. Although the gate-count estimate of various tiles using the rule-of-thumb is reasonably accurate, the total gate capacitance is still underestimated by about 35%. Additionally, estimating activity factors at the architectural level is challenging because of differences in the level of abstraction between architectural and RTL models [8]. The RTL power model uses fine-grained bit-level switching activity to estimate the control logic power. On the other hand, our architectural power model lacks access to bit-level switching activity, and uses microarchitectural block-level event activity as a proxy. This approximation underestimates the control logic activity factors by 65% compared to the fine-grained RTL models. The differences in capacitance estimates and activity factors combined cause a 21% error attributed to both the control logic and the array power (Row 1 in Table 2).

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 (latches and clock-splitters in our case) or a mismatch between specifications and actual RTL design. While a few of the modeling errors are specific only to ASIC designs, this class of errors affects power models for custom designs as well.

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 regardless of design methodologies. Another source of technology scaling errors is the scaling assumption of power supply voltage (V_{dd}). By sweep-

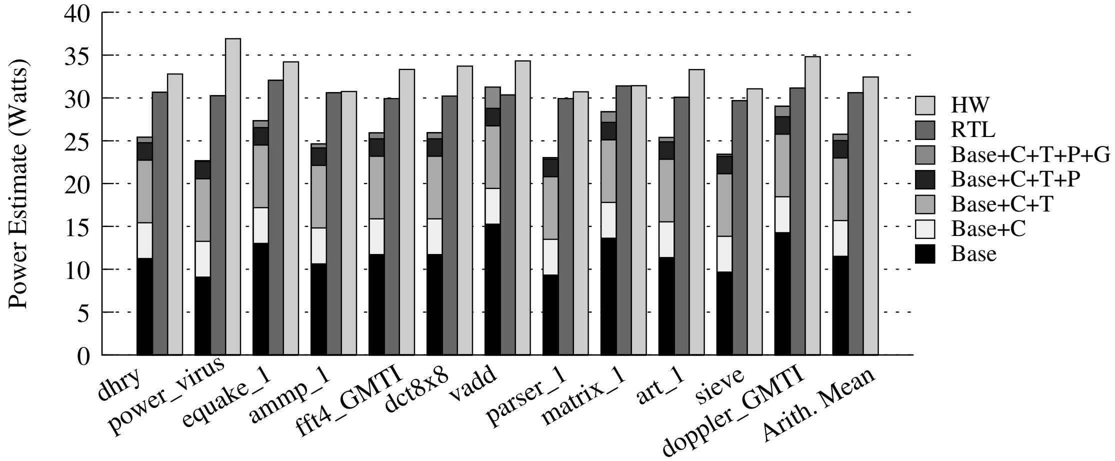

Figure 5: TRIPS Estimated and Measured Power

ing different power supply voltages in our prototype chip, and by comparing the measured hardware power to the V$_{dd}$-scaled power estimate of the architectural models, it is possible to validate this V_{dd} scaling assumption. We leave this validation for future work.

Abstraction errors arise from a lack of detail in the architectural simulators. Errors in the estimation of activity factors at the architectural level and differences between the architectural and RTL performance models are abstraction errors. Architectural simulators tend to trade-off detailed modeling for speed of simulation, which is a major source of abstraction errors. An additional source of abstraction error is the absence or presence of clock gating. When clock gating is not present (like in the TRIPS design), the architectural models only account for events in active modules whereas even inactive modules dissipate dynamic power in the hardware. However, clock gating, when present in a processor, can not be faithfully modeled at the architectural level due to differences in the level of abstraction.

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

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%), interconnect (17%), and the DIMM (8%). We attribute the remaining 11% error to architectural leakage models which lack detailed transistor widths: a combination of modeling and abstraction errors.

While inaccuracies remain in the absolute power, the architectural 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 for 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%). Such errors are caused by lack of fine-grained, bit-level switching activity in the architectural simulators.

6. CONCLUSIONS

In this paper we evaluate the accuracy of commonly used architectural power models with the TRIPS processor as a case study. Our experience shows that applying commonly used power modeling methodologies results in a more than factor of two underestimate in absolute power consumption. 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 to further improve accuracy. Despite using an ASIC design as a case-study, and leveraging feedback from RTL design, we believe that lessons learned here are applicable to other architectures as well by providing guidance on where to focus effort when building architectural 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. 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 tools such as CACTI and Wattch may have once been validated with a particular technology node, most architects employ simple scaling rules to estimate capacitance in smaller technologies. While this scaling may be appropriate in some cases, our experience with an ASIC design 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. Our experience shows that estimating the gate count of various units in the processor is key to estimation of control logic power, and even using simple rules of thumb like ours will greatly improve the accuracy of future power models.

While conventional wisdom identifies various power modeling errors, our work quantifies these errors to provide the above recommendations. Although estimating absolute power consumption is particularly difficult, we observe that the relative power from the architecture models tracked the hardware power reasonably well across the programs in our benchmark suite. Typical architectural studies compare relative power consumption across different applications and architecture configurations. Our observation on relative accuracy bodes well for such studies, provided the modeling, abstraction, and technology errors in the architectural power models are shared commonly across the various configurations.

7. ACKNOWLEDGMENTS

This research is supported by a Defense Advanced Research Projects Agency contract F33615-01-C-4106 and by NSF CISE Research Infrastructure grant EIA-0303609.

8. 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, 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.

[4] 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.

[5] `http://www.eembc.org`.

[6] J. Engel, T. Guzowksi, 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.

[7] 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.

[8] 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.

[9] 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.

[10] Micron Technology Incorporated. Calculating DDR Memory System Power. http://download.micron.com/pdf/technotes/ddr/TN4603.pdf.

[11] 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.

[12] 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.

[13] M. Nemani and F. Najm. High-level Area and Power Estimation for VLSI Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(6):697–713, June 1999.

[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] 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.

[16] Synopsys, Inc. PrimePower and VCS. www.synopsys.com.

[17] D. Tarjan, S. Thoziyoor, and N. Jouppi. CACTI 4.0. Technical Report HPL-2006-86, HP Labs, 2006.

[18] 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.