

Quantifying Sources of Error in McPAT and Potential Impacts on Architectural Studies

Sam (Likun) Xi, Hans Jacobson*, Pradip Bose*, Gu-Yeon Wei, and David Brooks

Harvard University, School of Engineering and Applied Sciences

**IBM Corporation, T. J. Watson Research Center*

{samxi, guyeon, dbrooks}@eecs.harvard.edu, {hansj, pbose}@us.ibm.com

Abstract

Architectural power modeling tools are widely used by the computer architecture community for rapid evaluations of high-level design choices and design space explorations. Currently, McPAT [31] is the de facto power model, but the literature does not yet contain a careful examination of its modeling accuracy. In addition, the issue of how greatly power modeling error can affect architectural-level studies has not been quantified before. In this work, we present the first rigorous assessment of McPAT's core power and area models with a detailed, validated power modeling toolchain used in current industrial practice. We find that McPAT's predictions can have significant error because some of the models are either incomplete, too high-level, or assume implementations of structures that differ from that of the core at hand. We demonstrate that large errors are possible when using McPAT's dynamic power estimates in the context of voltage noise and thermal hotspots, but for steady-state properties, accurately modeling leakage power is more important. Based on our analysis, we are able to provide guidelines for creating accurate McPAT models, even without access to detailed industrial power modeling tools. We conclude that in spite of its accuracy gaps, McPAT is still a very useful tool for many architectural studies, and its limitations can often be adequately addressed for a given research study of interest.

1. Introduction

Architectural power modeling tools like Wattch [5] and McPAT [31, 32] have enabled researchers to perform fast, integrated design space explorations of multicore and manycore CPU configurations. As the current de facto power modeling framework, McPAT has seen widespread adoption in the architecture community, but to date, a thorough validation of its area and power models for a contemporary high-performance processor does not exist in the literature. McPAT's authors only had access to published data on peak power for the various cores they validated, so the validation of McPAT in the existing literature [31, 32] was very coarse-grained. For example, for three of the four cores examined, they validate total core peak power but not for units within the core. Also, peak power is not relevant for many architectural studies evaluating application-specific behavior. Therefore, a seemingly accurate result could be

masking significant error canceling. More importantly, error in power modeling can greatly impact the conclusions drawn from modeling studies that rely on accurate power models, but it is unclear how significant this effect is.

There has been a significant body of work for creating performance simulators [3, 8, 28, 39, 43] and power models [5, 25, 30, 31, 36, 45], but validation of these models tends to be coarse-grained and emphasizes the methodology of creating the power model. Fine-grained validation requires access to detailed design data often not available in academia, and one notable example of such validation is work by Govindan et al. for the TRIPS microarchitecture [16]. In this work, we rigorously assess McPAT's area and dynamic power models for the cores of a conventional general-purpose microprocessor, the IBM® POWER7™ server multicore chip. With privileged access to POWER7 design documentation, we construct three McPAT models that cumulatively improve accuracy with respect to a proprietary power model. Although we focus on dynamic power, we will touch on leakage too.

Our results show that McPAT's power and area models can have significant error because they are either incomplete, too high-level, or represent an implementation of a structure which differs from that of the core at hand. Incomplete models result in McPAT only modeling a *subset* of the total area and power for a component. The subset is mostly comprised of caches, CAMs, and other SRAM array-based structures and does not account for many examples of control logic. We do not introduce any new models in this report, but we show that fixing the other types of error resulted in significant improvements to power and area estimates. Only a few of the specific errors we found are POWER7-specific; in fact, most of them would affect a generic out-of-order superscalar CPU.

Power models like McPAT are used for other relevant studies, such as voltage noise and heat dissipation. We perform two simple case studies in these contexts to quantify their sensitivities to power error. Our results show that *steady-state* properties, like overall chip temperature and static IR drop, are quite resilient to dynamic power error. *Spatial* properties, like thermal hotspots, benefit from improvements to McPAT because average power overestimates have been mitigated. *Temporal* properties, like inductive noise amplitude, also benefit from improved average power estimates because these

reductions simultaneously shrink overestimates of transient power swings. We conclude that leakage power accuracy is more important for steady-state modeling, but dynamic power accuracy is more important for temporal and spatial modeling.

Finally, we discuss why the inaccuracies we report may not have manifested in prior work using McPAT. We also provide specific guidelines that future studies using McPAT can observe to avoid being misled by power error. Our guidelines can be useful even without having proprietary power modeling tools. That being said, researchers will hugely benefit from having validated models released by industry, and we hope that this vision will come to fruition.

2. Power Modeling Approaches

Architectural power models are widely used to estimate the power consumption of a microprocessor, where high-level architectural and microarchitectural parameters (e.g. cache sizes, page size, and pipeline depth/width) and activity factors (e.g. cache accesses, total instructions) are specified to the power modeling tool, which abstracts away the underlying implementation details. These high-level abstractions, which represent a tradeoff between detail and flexibility/ease of use, enable an architect to quickly evaluate design decisions and explore various design spaces. McPAT and Wattch are two well known examples of these models.

Both of these tools are analytical, meaning that they use analytical equations of capacitance to model dynamic power. In contrast, empirical models, like PowerTimer [4] and ALPS [19], use pre-characterized power data and equations from existing designs. For structures like control logic that are difficult to model analytically, empirical models and/or fudge factors are often used instead. The differences between analytical and empirical models have been described in past work [6,33].

The IBM power modeling tool, which we refer to as DPM (detailed power model) and use as the point of comparison for this work, is an empirical model that tips the balance towards painstaking detail. For example, it can track misaligned cache accesses to precisely calculate the extra energy required for the operation, and it can compute branch detection power by knowing how many branches were in the group of fetched instructions. Base energy values and power computation equations are manually updated by circuit and layout designers. Such detail not only enables very fine grained power modeling, but enables the tool to accurately compute clock-gating factors, which have been shown to be critical to accurate power modeling [25]. DPM has been validated against circuit simulations to within 5% accuracy. However, such detail also precludes high-level design space explorations because the model is so closely tied to a specific implementation. In this work, we quantify the error in architectural power models arising from this tradeoff.

Component	Parameters
I-cache	32KB, 4-way set associative, 8-way banked
Branch predictor	32K entry tournament predictor indexed by global history hash; 128-entry address cache for indirect branches; 16-entry jump return address cache
Frontend	Fetch up to 8, decode up to 6, issue up to 8
General purpose RF (GPR)	32 architected and 112 physical 64b registers, partitioned based on SMT mode
Vector RF (VRF)	64 architected and 172 physical 128b registers, partitioned based on SMT mode
ROB	20 instruction groups, each composed of up to 6 instructions
Register renaming	80 shared between GPR and VRF, 140 more for other types of registers
Issue queues	48-entry unified queue for FXU, LSU, and VSU instructions; 12-entry queue for branches; various others. Managed by an age tracking matrix
Load/store	Two pipelines; 32-entry load and store queues
D-cache	32KB, 8-way set associative, 8-way banked
D-TLB	128-entry 1st-level TLB; 512-entry 2nd-level TLB
Execution units	2 fixed-point, 4 floating-point, 1 vector, 2 load-store, 1 branch, 1 condition register, 1 decimal

Table 1: POWER7 core configuration [48,53].

3. Assessment Methodology

In our assessment, we created and compared three models of the POWER7 core, described below:

- **MR0**, a *no-revisions* model based on data published by Sinharoy et al. [48]. This model mimics the typical McPAT use case where all parameters are derived from published reports (see Table 1). The validation method in the original McPAT report [31] is an example of this use case.
- **MR1**, a *revised version 1* model that represents the most accurate core configuration parameters possible. Parameters were available through privileged access to detailed design documentation. This level of detail is typically absent (and sometimes impossible) in other power modeling studies.
- **MR2**, a *revised version 2* model that incorporates source code changes to fix modeling assumptions in McPAT which are incorrect for POWER7 (and pertain to generic out-of-order CPUs) and could not be fixed purely through the available parameters. These changes are intended to be applicable for any general purpose chip and are not in any way POWER7-specific.

This methodology lets us quantify how much improvement McPAT can show with the best configuration possible and how much more it could improve if source code were directly modified. In future work, one could create a hypothetical "MR3" that adds modeling for missing components, like datapath control logic. However, such a model would only apply to a specific implementation, and validating this model could depend on manually writing RTL; this task would only be compounded if this model were made parameterizable. As our goal was to keep modifications in the spirit of McPAT, we do not introduce any logic models in MR2, but we will briefly mention some preliminary work towards a POWER7-specific MR3 in Section 6.

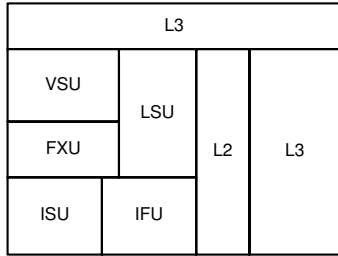


Figure 1: POWER7 chiplet simplified floorplan. The decimal floating unit (DFU) is omitted, and the wraparound L3 cache is split into two parts.

In this report, we use the following abbreviations for the core units, shown in the POWER7 floorplan in Figure 1:

- IFU: Instruction fetch unit (includes decoder).
- ISU: Instruction sequencing (i.e. scheduling) unit.
- LSU: Load-store unit.
- FXU: Fixed-point unit.
- VSU: Vector-scalar unit (floating-point).

These three models are compared against DPM. Unlike DPM, McPAT does not model every macro of each unit. Therefore, McPAT’s predictions are compared against the *subset* of the total DPM measurement for a unit representing components actually modeled by McPAT (henceforth referred to as subset area and power). Recall that McPAT primarily models caches, SRAM arrays, and CAMs; it does not account for many control logic elements. As an example, McPAT models storage components of the instruction issue queues, but it does not model logic that dispatches instructions to these issue queues. Only the former would be included in the subset, but both would be included in the total.

Performance statistics are generated by an IBM performance simulator for POWER chips called M1, described by Srinivas et al. [49]. Performance models like M1 target a goal of 2% error. M1 has been validated against RTL simulations and is used in a regression test suite, so it is continually maintained. It dumps thousands of statistics for both DPM and McPAT to use. V_{dd} is set to 1.01V and f_{clk} to 4.0GHz.

We evaluated these three models for area and power using 20 benchmarks selected from SPEC2006 and SPEC2000. We selected 14 from SPEC2006 using the guidelines described by Phansalkar et al. [40]. From SPEC2000, we selected six workloads that represent control-flow complexity, memory/compute-bound behavior, and a mixture of these qualities. A synthetic stressmark called `vsx` used by the POWER7 design team to measure the chip’s TDP is also included.

Area estimates are compared against actual areas measured from detailed floorplans. Dynamic power estimates are compared against those produced by DPM. Because DPM only models core units and the private L1 cache, the L2 and L3 caches are excluded from our validation, along with uncore components like the memory controllers and interconnect. For our case studies, we use an alternative method to account for L2/L3 cache and uncore power.

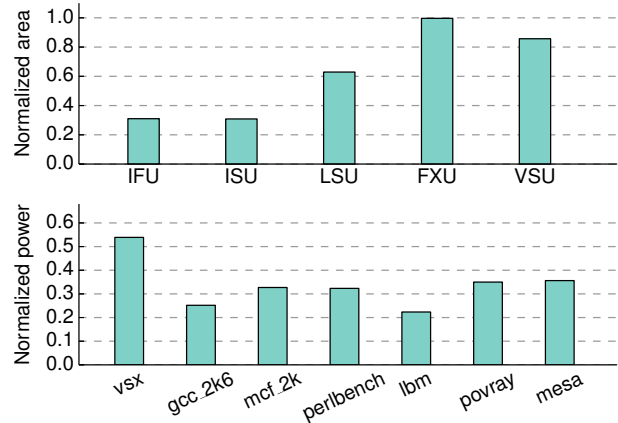


Figure 2: The subset of DPM total POWER7 area and power that is represented by McPAT’s models.

4. Assessment Results

In this section, we compare the power estimates from MR0, MR1, and MR2 against DPM for each macro modeled by McPAT, identify and categorize the sources of error, and either show how the error was addressed or explain why a fix was not attempted. Units are broken down into macros modeled by McPAT, so for the rest of this section, we only compare McPAT with DPM *subset* area and power, as we can’t compare McPAT’s predictions for macros it doesn’t even model. Note that figures are organized by units instead of by error type.

Our investigation reveals two overarching problems in opposite directions with McPAT: McPAT only models a subset of the total core, but this subset is globally overestimated, creating significant error canceling. By “error”, we refer to any deviation of McPAT’s estimate from DPM’s. These errors can be divided into four categories, listed roughly in decreasing order of importance: *abstraction error*, which arises from incomplete or missing models; *modeling assumption error*, in which assumptions about the underlying implementation of a microarchitectural structure differ from that of the CPU at hand; *input error*, which arises from incorrectly specified parameters; and *coding error*, which are programming mistakes.

4.1. Model Abstraction Errors.

Abstraction errors in McPAT are usually due to one of two reasons: either the model for a structure is incomplete or missing, or the parameters are too high-level to capture important low-level details. Incomplete models create the subset vs. total problem, and insufficiently detailed models create error within the subset’s power estimates.

4.1.1. Incomplete/missing models. Figure 2 shows that for the IFU, ISU, and LSU, subset area and power account for less than 40% of the totals in POWER7. FXU and VSU subset areas are high because McPAT accounts for most of those functions. Some of the specific unmodeled macros are listed in Table 2. Notice that the majority of these unmodeled macros can be classified as control logic. The important observation

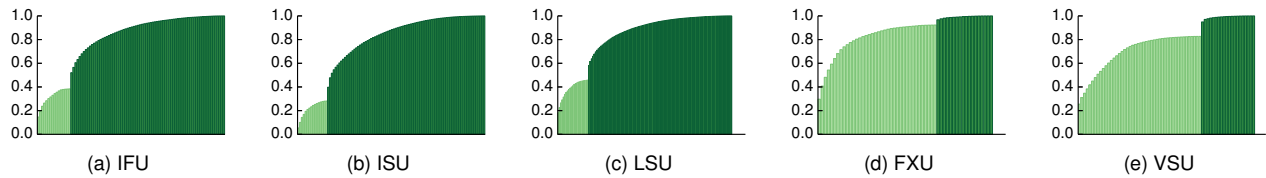


Figure 3: Power cumulative distribution function of each unit. Power from macros in the McPAT subset are in light green, and those not in the subset are in dark green. Each bar represents power consumed by a single macro plus the power from all macros consuming less power than it. Power is normalized to the total of the unit.

Unit	Top power-consuming unmodeled macros
IFU	Branch control logic (e.g. history management) Special purpose registers Instruction cracking and fusion Hardware thread management I-cache prefetcher control logic
ISU	Instruction age tracking logic Issue queue data management Instruction group tag register files Per-thread dispatch and execution state Instruction dispatch to issue queue control logic
LSU	D-cache prefetcher control logic Load/store queue thread and data management Load/store queue age tracking Cache line replacement policy logic
Global	Automated built-in self testing

Table 2: Major core macros that McPAT does not model in rough order of power cost (per unit). Only one of these macros is specific to POWER7. The FXU and VSU are not included because for the most part, only the globally unmodeled macros apply to them.

is that the fraction of functional blocks occupied by control logic is much greater than what might be suggested by an architectural block diagram and thus can have a major impact on abstraction error. Fortunately, Figure 3 shows that out of the unmodeled macros (in dark green), simply accounting for about one third of them will bring the subset fraction up to 80%. In other words, McPAT only needs to model a few more macros to account for the large majority of the unit’s power.

Despite the small fraction constituted by the subset, McPAT tends to overestimate its power and area. Figures 4 through 6 show that MR0’s estimates almost always exceed the subset’s values, and at the unit and macro levels, overestimates can exceed an order of magnitude. MR1 and MR2 improve upon MR0 by providing more accurate configuration parameters and fixing sources of error, which is why in Figure 4, each revision causes power and area estimates to decrease. In a few cases, fortuitous error canceling also leads to the surprising result that MR0 is actually quite close to DPM total power. For instance, on *omnetpp*, *bwaves*, *mcf_2k* and *swim*, MR0’s core power is merely between 0.1% and 4% off of DPM total (data not shown). These were the only four workloads out of 21 to exhibit this behavior, so we have little reason to believe that this would be the case in general for other workloads. In addition, overestimation has important ramifications for the voltage and thermal case studies that we later perform. As a

whole, these results emphasize that proper power modeling validation *must be done at the unit level or lower*. Simply validating total core and/or chip power, the general approach taken by much previous work [22, 28, 31, 36], is misleading because it hides a large amount of internal error. These figures also clearly show that MR1 is in general not a huge improvement over MR0. That is, most of the error observed in MR0 is *not* input error. Thus, directly addressing the other modeling errors through source code changes in MR2 is necessary.

Figure 4 shows that many of the benchmarks exhibit similar error relationships between the three models regardless of the workload, so we will present macro-level analysis using only 7 of the 21 workloads surveyed. These workloads were selected to span the entire range of power variance observed.

4.1.2. Insufficient modeling detail. While incomplete models create the subset problem, insufficient modeling detail creates power error within the subset itself. The two primary examples of this are McPAT’s assumptions for read and write ports and perfect clock-gating and data-gating.

Read/write port errors. The register files are good examples. McPAT defines a parameter called `peak_issue_width`, which specifies the maximum number of instructions that can be issued from the issue queues in a cycle¹. POWER7 can issue up to six instructions per cycle from the unified issue queue, so McPAT defines an integer register file with *twelve* read ports and six write ports, a worst-case approximation. McPAT’s assumption is that every instruction issued per cycle needs two of its own read ports on the register file, but this is not necessarily true (for instance, some instructions only need one register operand for execution if there is an immediate). Also in POWER7, instruction dispatch and issue rules limit the number of simultaneous accesses to such structures. Thus in MR2, we manually specified the number of read and write ports. This change contributed to a large fraction of register file area and power reduction, seen in Figures 10a and 10b. For processors that have separated issue queues, setting this parameter to the maximum of any individual queue may be a simpler solution.

Another such parameter is `number_instruction_fetch_ports`. McPAT uses this one parameter to set the number of read/write ports for the I-cache and instruction buffer, among others. But in POWER7, the I-cache and in-

¹This interpretation of “peak issue width” agrees with how this parameter is used by the example models in McPAT.

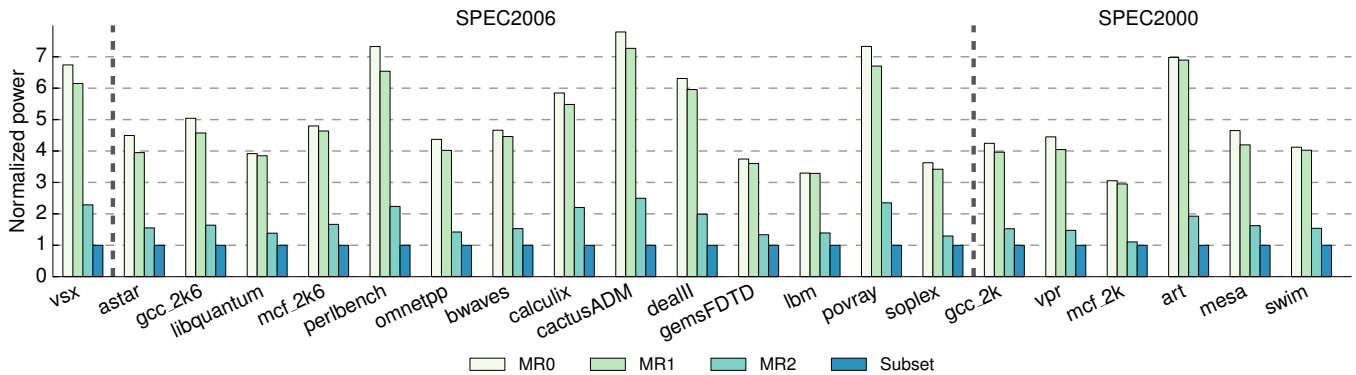


Figure 4: Core power for 21 different benchmarks, normalized to DPM subset power. Each successive revision improves accuracy and thus brings McPAT’s estimates closer to DPM subset power. Note that the selected workloads from SPEC2006 have about the same power variance as those from SPEC2000.

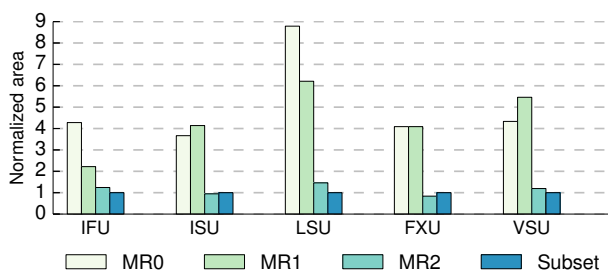


Figure 5: Core area, normalized to DPM subset. Each revision brings estimates closer to DPM subset area.

struction buffer have different port configurations because of their different physical structures. In MR2, we added additional instruction buffer parameters to resolve this issue.

Perfect clock-gating and data-gating. Clock-gating prevents switching activity on clock signals from drawing power while a macro is not in use, and data-gating does the same for data signals in combinational logic. Assuming perfect clock- and data-gating means that any components not in use at any given cycle consume zero dynamic power. This assumption is reasonable for structures like SRAM arrays and makes some sense for McPAT since the vast majority of McPAT’s modeled structures are caches, arrays, or CAMs. However, logic circuits cannot always be perfectly clock- and data-gated due to design complexity or timing reasons, and it has been shown that accurately capturing clock-gating factors is critical to accurate power models [25]. As a result, McPAT tends to underestimate dynamic power for components that include nontrivial amounts of logic. Even though POWER7 components do consume close to zero dynamic power when not being accessed, McPAT’s estimates in these scenarios are smaller still. This causes underestimates for the branch target buffer and vector register file on *gcc* and *mcf*. In our execution traces, *mcf* rarely uses the VRF, so all three models estimate effectively no VRF dynamic power on this benchmark (Figure 11). The only exception to the perfect data-gating assumption is the pipeline latch model, because McPAT models both switching and holding power for flip-flops in pipeline latches.

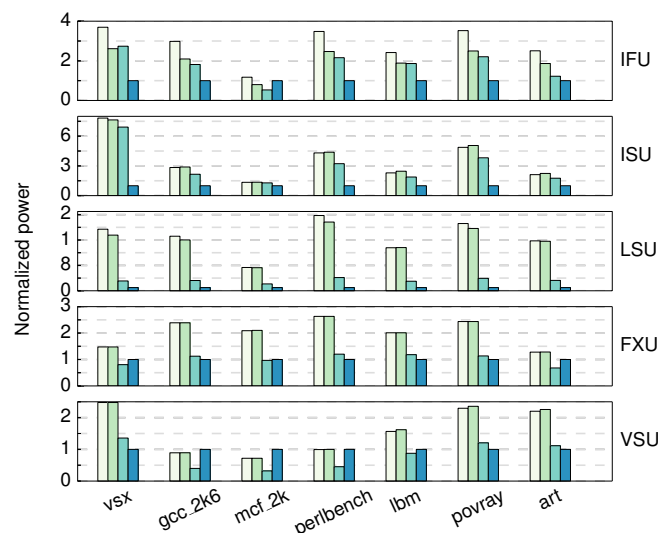


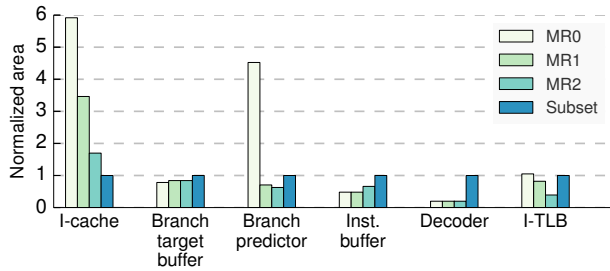
Figure 6: Core power breakdown by unit, normalized to DPM subset power. Power error relationships across the models are largely consistent regardless of the benchmark.

4.2. Modeling Assumption Errors.

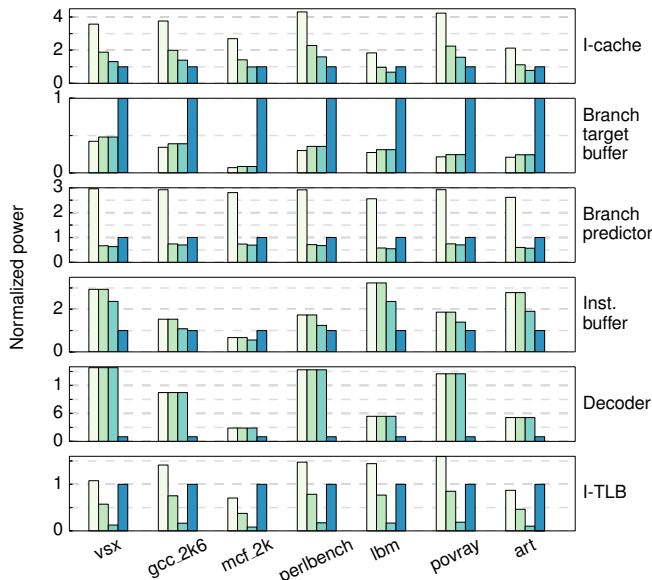
Modeling assumption errors occur when the CPU’s implementation of a microarchitectural structure differs from McPAT’s modeled implementation. McPAT’s SMT and control logic models are the main contributors to this error type.

4.2.1. McPAT’s SMT model. McPAT provides a set of assumptions regarding what hardware structures are shared, partitioned, or duplicated [32]. However, if some of those assumptions are incorrect for the CPU at hand, there isn’t always enough flexibility to compensate with the provided parameters. POWER7 is a four-way SMT design that shares some of the resources McPAT assumes are duplicated, so deduplicating hardware where appropriate in MR2 always resulted in at least a fourfold reduction in *area* for that component.

These corrections do not necessarily translate to a fourfold improvement in power for the same unit. One of two reasons can usually explain this:



(a) Area breakdown.



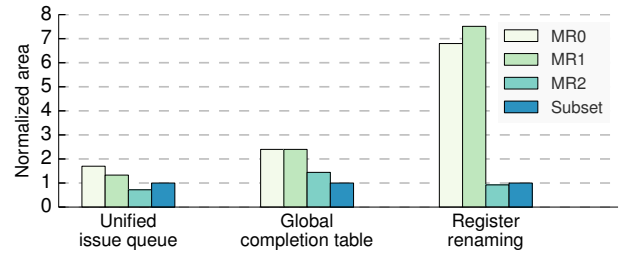
(b) Power breakdown.

Figure 7: Instruction fetch unit breakdowns.

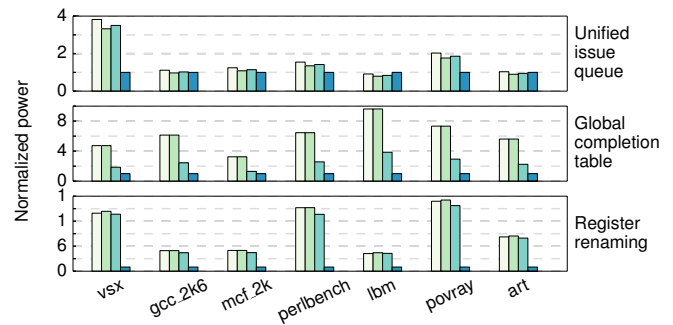
- For a cache structure, McPAT sometimes duplicates the area of the element without increasing its energy per access (which may be the desired effect).
- The element that was duplicated accounts for only a small fraction of the total unit power.

The register renaming unit is a good example of both reasons (Figure 8b). McPAT’s SMT model assumes separate renaming tables per thread [32] whereas POWER7 uses shared renaming tables. Duplicated renaming tables in McPAT are modeled by computing area and energy per access for a single renaming table, whose area is then multiplied by the number of hardware threads. In MR2, we deduplicated hardware, reducing area error greatly, but power was unaffected. It turns out that MR2 estimates the renaming table power to within 10%, but the renaming unit includes an analytical logic model for dependency checking logic that overestimates power nearly tenfold. This logic, which is not duplicated, has large errors because the POWER7 implementation greatly differs from McPAT’s model implementation.

One way to address this duplication problem without modifying source code is to specify $1/N$ of the actual number of entries, where N is the number of hardware threads. When



(a) Area breakdown.



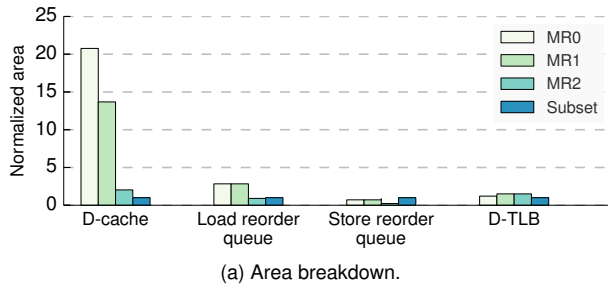
(b) Power breakdown.

Figure 8: Instruction sequencing unit breakdowns.

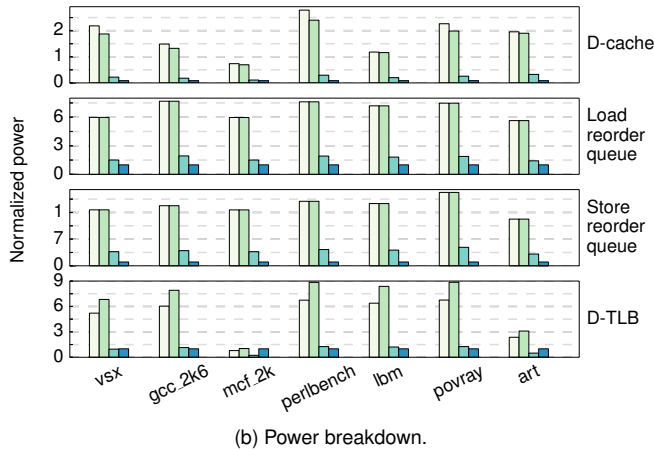
McPAT multiplies area by N , it is effectively modeling an N -way physically partitioned structure. There are cases where this structure is true or approximately true in POWER7 (such as the load/store queues), but this is not always the case. Consider the register renaming free lists: the number of entries is set by the number of physical registers, so this method would mean specifying $1/N$ of the total number of physical registers. There are two problems with this. First, this solution assumes that the free lists, and by extension the physical register files, are composed of N array instances. This is not true for POWER7 and not necessarily true in general. Second, specifying fewer physical registers would reduce the width of the register renaming table entries.

To illustrate these problems, we took this $1/N$ approach for the general purpose register file (GPR) and vector register file (VRF) in MR0 and MR1. As shown in Figure 10, the GPR area is $7\times$ too high, but the power estimate is accurate. VRF results are similar (Figure 11): area is $3\times$ too high, but power is more accurate. MR2 was able to achieve good area *and* power estimates for both the GPR and VRF by modeling them as being built with one array per pipeline, eliminating the duplication per hardware thread. In contrast to the renaming tables, physical register files are duplicated per integer/floating-point *pipeline*, and McPAT models this correctly.

4.2.2. Control logic and arithmetic logic. These are known to be difficult to model, especially at the architectural level [16, 31]. Since the vast majority of McPAT’s modeled structures are caches, array-based structures and CAMs, the subset area is merely 25-50% of the total area of the IFU, ISU, and LSU, as shown by Figure 5. McPAT does use analytical models for some control logic, such as dependency checking logic,



(a) Area breakdown.



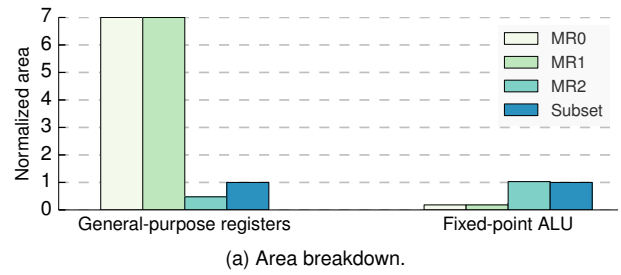
(b) Power breakdown.

Figure 9: Load-store unit breakdowns.

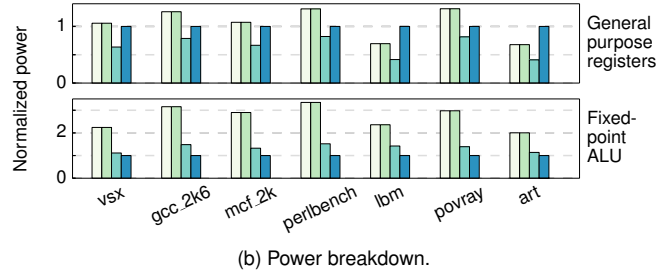
instruction decoder, and aspects of instruction issue selection. As in Wattch [5], the decoder is modeled as a n to 2^n bit decoder, which does not represent functions like instruction cracking or group formation/instruction fusion, so area and power error is large for MR0, MR1, and MR2.

Like other analytical power models [5, 47], McPAT models the ALU [35] and FPU [38] empirically. Base power and area is obtained from published data and scaled for the activity factors, technology node, and operating point. Since these models are based on older designs from different architectures, MR0’s predictions show significant inaccuracies for POWER7 (Figures 10 and 11). MR2 simply replaced the base area and power values with ones measured from detailed floorplans and microbenchmarks, respectively. As expected, area error dropped to 1% and power error to within 20%. Of course, these changes would not be appropriate for ALUs that don’t look like those in POWER7. One possible solution would be to have a library of base area and power values for different ALU designs. We discuss this problem further in Section 6.

4.2.3. POWER7-specific details. A few modeling assumption errors are due to details specific for POWER7. For instance, the global completion table (GCT) tracks instruction groups by tags and other metadata instead of storing entire decoded instruction (or μop) words. Also, McPAT assumes that each entry in the instruction buffer stores `peak_issue_width` instructions (six in POWER7) whereas POWER7 stores four instructions per entry. Adding the appropriate additional parameters in MR2 improved area and power estimates.



(a) Area breakdown.



(b) Power breakdown.

Figure 10: Fixed-point unit breakdowns.

4.3. Input Error.

When creating MR0, we often needed to guess the values of certain parameters. These values were corrected in MR1 and made significant improvements in a few cases, but input error is not a major contributor to overall error. This can be seen in that MR1 is often only slightly different from MR0, but much different compared to MR2.

The IFU is a good example of this error category. Published data stated the I-cache was 16-way banked when it was actually 8 [48]. This cut I-cache area and power error in half for most benchmarks. The branch target buffers turned out to be 2-way set associative, which improved area errors to 5%, but power estimates are greatly underestimated due to clock- and data-gating assumptions. The I-TLB was also 2-way set associative instead of fully associative, but this was a rare case where fixing parameters increased both area and power error. We were unable to pinpoint the cause.

4.4. Coding Errors and CACTI Issues.

There were only two instances where coding errors in McPAT caused large power modeling errors. The physical register files were intended to be modeled as shared components [32] but were actually modeled as duplicated components. Also, the register renaming tables stored 33 Bytes for each entry instead of 33 bits. Fixing these bugs in MR2 reduced register file and renaming table error significantly. Both were present in McPAT v1.2 and have been reported to the developers.

We had to manually tweak CACTI’s optimization weights in order to reduce I-cache and D-cache power and area error while staying close to the cycle time target. These weights represent tradeoffs between energy and delay. The presented data are the best results we were able to achieve. This is not a “bug”, but it is still worth mentioning.

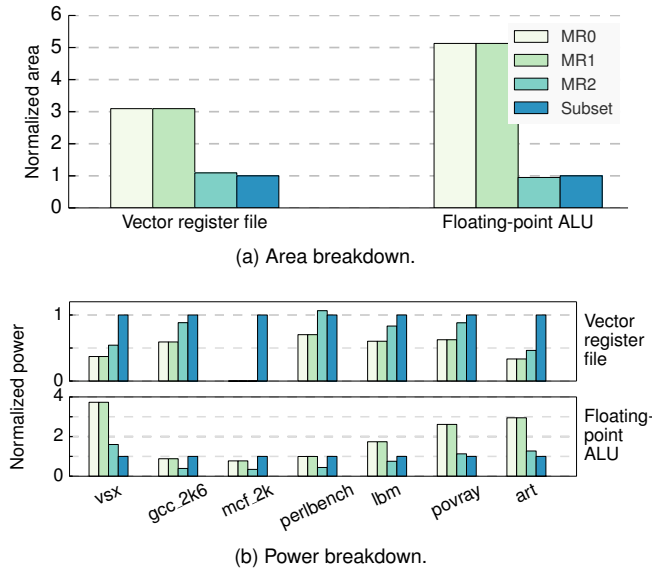


Figure 11: Vector scalar unit breakdowns. See Section 4.1 for an explanation of VRF power on `mcf_2k`.

Finally, we note that CACTI’s 45nm technology model is for planar bulk devices, but POWER7 is built on an SOI process. As there is no ITRS model for 45nm SOI or 32nm bulk, we ran our analysis on CACTI’s 32nm SOI model and compared it with the 45nm bulk results. Despite changing both technology feature size and device topology, average power only changed about 20-30%. We believe that this does not impact any of our conclusions given the magnitude of the errors we observed and the fact that nothing we’ve discussed thus far pertains to a specific technology.

5. Case Studies

Accurate power modeling is critical because many studies rely on it to evaluate other systems of interest, such as power-aware scheduling, or properties of a chip like thermal hotspots and voltage noise. We demonstrated in the previous sections that McPAT’s power and area models possess a significant amount of error unless carefully tuned for the target platform. How much power modeling error can these studies tolerate until the conclusions drawn become wrong or misguided?

In the following case studies of thermal hotspots and voltage noise, we show that error in McPAT’s dynamic power predictions only has a small effect on a chip’s *steady-state* properties like average chip temperature and static IR drop magnitude, suggesting that accurately modeling leakage power may be more important for these metrics. However, dynamic power error has a much larger effect on *temporal* properties, like the amplitude of transient inductive noise, and *spatial* properties, like the locations of thermal hotspots and greatest IR drop. For such properties, dynamic power error can result in even larger error in the studies, potentially leading to wrong conclusions. For these studies, it is important to carefully tune McPAT’s dynamic power models in order to obtain accurate results.

5.1. Thermal Hotspots

Heat dissipation is an important design consideration for modern microprocessors. Excessive heat can result in reduced performance as the chip tries to stay within its thermal budget, and over time, reliability of the chip can suffer. Past thermal studies have investigated temperature induced reliability degradation [50], thermal-aware task scheduling [2, 11], DVFS for mitigating thermal emergencies [12], as well as optimal floorplanning across a chip [21, 44]. Here, we examine thermal hotspots on the POWER7 chip to quantify thermal error deriving from power error.

We constructed a full chip model using HotSpot [23], with parameters derived from the POWER7 chip and package. We selected four representative SPEC2000 workloads, simulated selected regions of each for 100 million cycles, then duplicated the resulting performance traces 40 times for a total of 4 billion cycles. Transient power was computed using MR0, MR2, and DPM. This single threaded power trace is duplicated across all eight cores (a multiprogrammed SPEC workload) during thermal modeling. We chose MR0 over MR1 because it represents the kind of model most users will have – MR1 requires proprietary data and Section 4 showed that MR0 and MR1 are similar because the overall effect of input error is small. Some important details about our methodology are mentioned below:

- When computing chip temperature with DPM data, we use total power instead of subset power, because subset power is such a small fraction of total power that the computed temperature would be unrealistically small.
- Based on our experiments, we find that McPAT’s leakage power estimates are approximately $2\times$ smaller than DPM’s. Leakage power is primarily a function of technology, but accounting for technology parameters is beyond the scope of this paper, so we normalize all McPAT power results by substituting its leakage numbers with those from DPM.
- DPM does not model power for the L2 and L3 cache or any uncore components. For these components, we use leakage power obtained from circuit simulations of the synthetic stressmark. This means that overall chip temperatures will be lower than what would be observed in practice.

Figure 12 demonstrates temperature characteristics consistent with the power models we have described. As a whole, MR0 appears more accurate than MR2; Figure 13 shows that MR0 exhibits only 5% mean error for three of the four benchmarks, while MR2’s error is consistently higher ($\sim 10\%$). The exception is `art`, because the total core power predicted by MR0 on this benchmark is dominated by the LSU. As a whole, this suggests that if leakage power is accurately modeled, McPAT can be used to predict average chip temperature well.

However, MR0 invariably identifies the LSU as the thermal hotspot because its LSU power estimates are $3-6\times$ greater than DPM’s (Figure 9b). In contrast, DPM shows that no single unit within the core is primarily responsible for heat production.

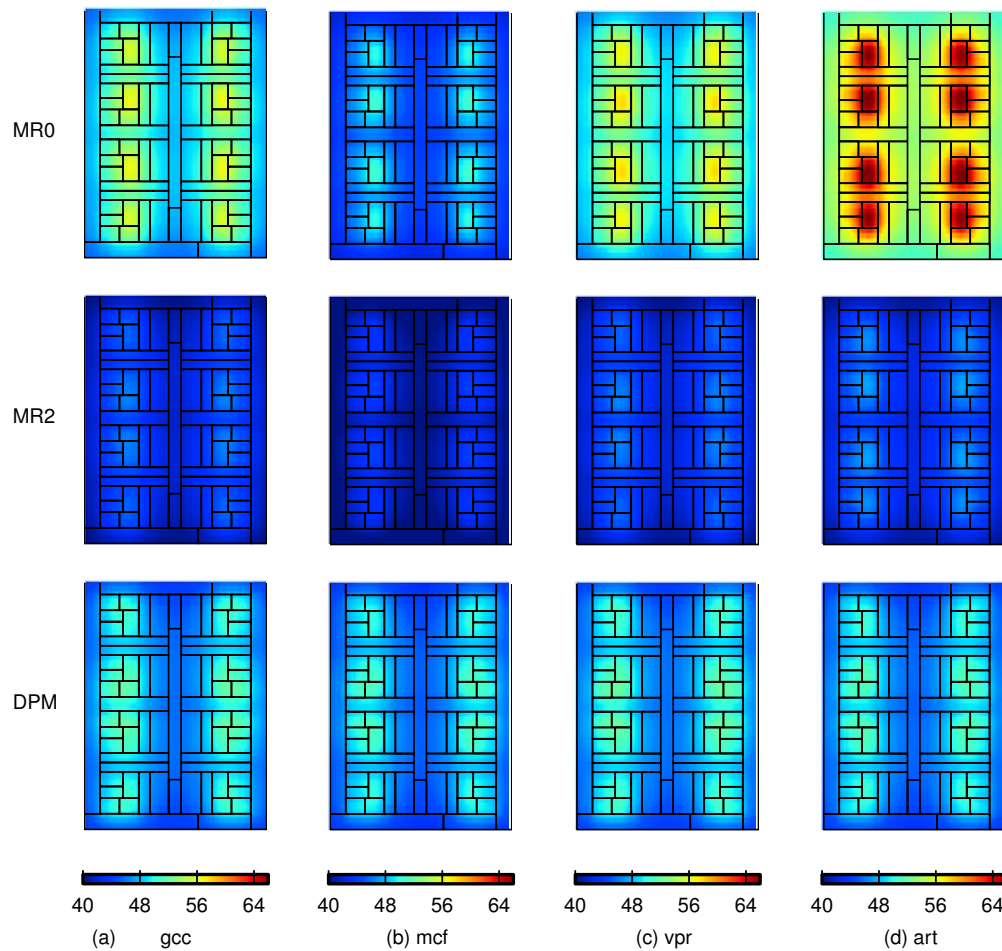


Figure 12: Steady state thermal distribution across the POWER7 chip in °C. MR0 shows much greater variance across workloads than MR2 and DPM, but MR0's overall chip temperature is more accurate.

MR2 agrees with DPM in this regard, even if its estimates are universally much smaller because MR2 has eliminated a lot of error canceling. Furthermore, MR0 exhibits significant variation across workloads, whereas MR2 and DPM are far more consistent. This is an important qualitative error that was addressed by MR2's cumulative changes.

In summary, even though MR0 can possess over 200% total core dynamic power error, it only results in 10% overall chip temperature error. Some of this error is suppressed because of accurate leakage data in the power trace and heat diffusion from the core to the surrounding uncore area. Nonetheless, we can conclude that while dynamic power modeling accuracy may not be critical for estimating average temperature, it is much more important for analyzing spatial properties.

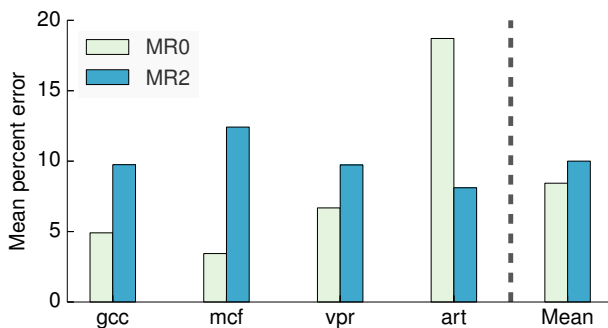
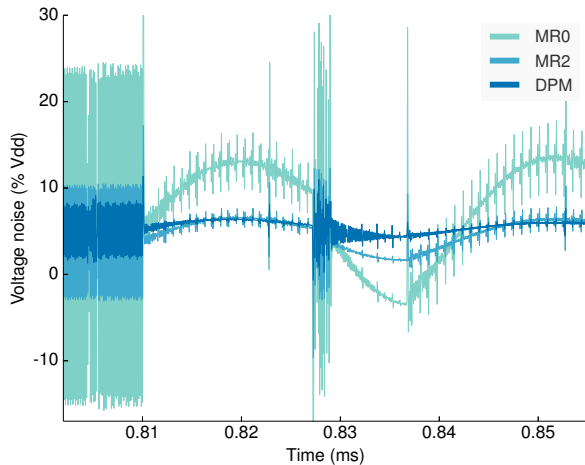


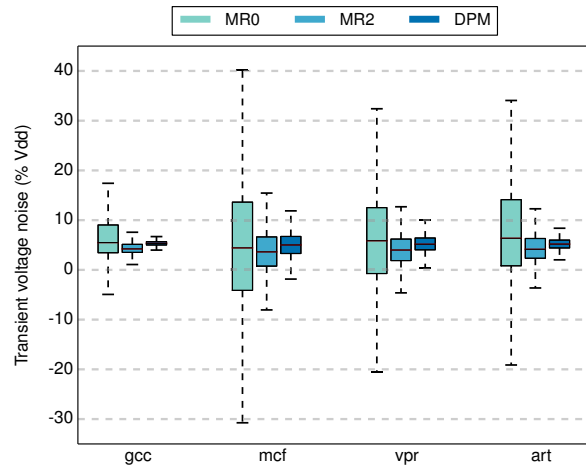
Figure 13: Overall chip temperature error with respect to DPM. Note that MR2 error is always negative, but absolute value error is easier to visually compare.

5.2. Voltage Noise

Voltage noise, comprised of static IR drop and transient inductive noise, is an important phenomenon in contemporary chips, because aggressive power and clock gating can produce large fluctuations in supply current that can then induce fluctuations in supply voltage. Significant voltage drops can result in timing violations for logic circuits. To mitigate effects of voltage noise, researchers have proposed various runtime strategies [17, 20, 26, 29, 41, 55], optimal placement of available C4 pads [51], and more. In this case study, we quantify the amount of error in voltage noise that derives from power error.



(a) Snapshot of a voltage noise trace for the three power models from `gcc_2k`. MR0 shows considerably greater inductive noise than DPM, whereas MR2 has much less error.



(b) Distribution of voltage noise for all benchmarks and all power models with standard box and whisker heights. Consistent with Figure 14a, MR2 is a significant improvement over MR0.

Figure 14: Inductive noise example and general characteristics.

To evaluate voltage noise characteristics, we constructed an on-chip power distribution network and package model using VoltSpot [55], with parameters derived from the POWER7 chip and package. Because VoltSpot is a very fine grained modeling tool and the C4 pads on POWER7 are very densely packed, we only model a single *chiplet* - core, L2 and local L3 cache - rather than the full chip like in the thermal study. The power grid and C4 pad specifications are approximations of the actual structure and layout in POWER7. Note that although PDN parameters are derived from the physical hardware, it is beyond the scope of this work to correlate voltage noise computed by VoltSpot with those from actual hardware.

We used the same four workloads from the thermal study and simulated representative regions of them for 40 million cycles each. Transient power was computed using MR0, MR2, and DPM. Like the previous study, we compare MR0 and MR2's results with DPM total instead of subset because subset voltage noise is unrealistically small.

5.2.1. Transient voltage behavior Transient voltage noise is produced when periodic current swings trigger localized Ldi/dt resonance as well as chip-wide LC resonance if the current swings occur near the global resonance frequency of the PDN. Here, we assess the accuracy of transient voltage noise predictions using power traces produced by McPAT.

Figure 14a shows a snapshot of a voltage trace from `gcc` that includes two distinct phases of the application with distinct transient characteristics. The snapshot demonstrates a trend in both phases that persists throughout all benchmarks: MR0's power estimates result in huge supply voltage swings, whereas MR2's transient voltage noise is much more muted. From Figure 14b, we see that in all cases, MR0 exhibits the greatest variance in voltage noise amplitude by far. Based on the whisker heights, MR0 predicts anywhere from 20 to 70% swing, whereas MR2 ranges from 8-24% and DPM from 4-

14%. MR2's maximum predictions are merely 6% Vdd higher than that of DPM, compared to 54% for MR0.

These results raise a question: MR2's power estimates were tuned to match DPM *subset* rather than *total* power, so why does MR2 produce accurate voltage noise results when compared with DPM total power? The reason is that *transient* power fluctuation, not average power, creates inductive noise. While MR2's average power is smaller than average total power, MR2's transient power swing amplitude is much more accurate. This case study demonstrates that for inductive noise studies, accurately capturing transient power, rather than average power, is much more significant.

5.2.2. Static IR drop Static IR drop is caused by the impedance of the power delivery network, and in today's systems, the primary solution is the use of a voltage guardband. IR drop is effectively static on the time scale of processor activity due to the presence of the power grid itself and decoupling capacitance, but sustained nonuniform activity and current draw can create significant variations across a chip. We did analyze IR drop for POWER7, but we found the exact same conclusions as we did for thermal hotspots: MR0 is slightly more accurate than MR2 for overall IR drop but much worse for spatial properties. Therefore, we omit the data for brevity.

6. Discussion and Guidelines

Despite the amount of existing work using McPAT, few (if any) have mentioned modeling inaccuracies like the ones we describe. One possible reason is that many past studies have used cores that did not trigger some errors we observed. For example, older and simpler cores like Atom and Penryn have lower issue widths, so studies using them [13, 18, 42, 51, 55] avoid read/write port overestimates, one of the major error sources we observed. Penryn cores also do not support SMT,

Microarchitecture	Intel Nehalem	AMD K10	IBM POWER7
SKU	X7560	1090T	–
Cores	8	6	8
Base clock (GHz)	2.26	3.2	3.3
SMT	2	–	4
Issue width	4	9	8
Pipeline depth	16	12	17
Icache, Dcache	32K, 32K	64K, 64K	32K, 32K
L2 (per core)	256K	512K	256K
L3 (shared)	24M	6M	32M
Die area (mm ²)	684	346	567

Table 3: Comparison of server-class Intel, AMD, and IBM processors on 45nm technology nodes, showing that POWER7 is not a microarchitectural outlier. Specifications are taken from published documentation [15, 34, 48].

eliminating the duplication of hardware error. This being said, POWER7 is not unusual from a power modeling perspective. Table 3 shows the high-level microarchitectural parameters that define three different 45nm server-class processor cores. From such a view, there is no fundamental reason why one should not use McPAT to model a POWER7-like chip. Furthermore, server-class cores, like Haswell, are becoming more complex in order to meet single-thread growth targets while simultaneously attaining lower power budgets for mobile applications. Therefore, without knowledge about the modeling gaps in McPAT, power modeling studies using this tool are likely to become progressively more inaccurate over time.

As mentioned in Section 2, one method of analytically modeling hard-to-model control logic is to assume that its power is correlated with that of a cache structure and add a fudge factor to compensate. McPAT does not use this method in that it does not have explicit fudge factors to account for missing control logic models². It is unfair to argue that these fudge factors are implicit through the subset (caches/arrays) overestimates because CACTI is not meant for logic modeling.

We investigated correlation between subset and nonsubset power on POWER7 as a first step towards creating MR3. For this analysis, we used power traces from four SPEC2006 benchmarks. As shown by Figure 15, the covariance between the power traces is close to +1 for all four benchmarks, indicating that subset and nonsubset power are highly correlated and that the fudge factor method does have merit. However, this method is difficult to use because these fudge factors likely require RTL simulations to ascertain. Indeed, Figure 15 shows that the ratio of subset to nonsubset power varies as much as 0.4 between units and up to 0.3 within a unit between benchmarks. The error bars denote standard deviation of the ratio over the time series power trace, so variation of up to 0.3 exists even during a single benchmark’s execution.

Based on our power model error analysis, we suggest the following solutions for improving power model accuracy:

²The only exception to this is that McPAT estimates pipeline latch, common data bus, and layout overhead with fudge factors.

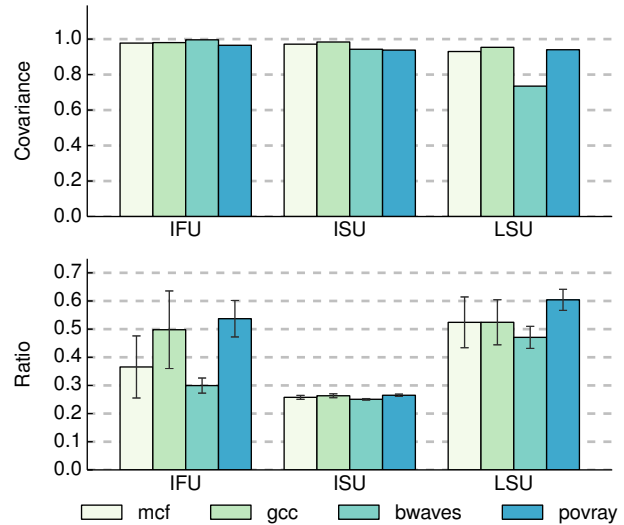


Figure 15: Covariance and ratio between subset and nonsubset components on four SPEC2006 benchmarks. The FXU and VSU were excluded because their subsets cover over 80% of the logic.

- Abstraction error:** Users of McPAT must specify important parameters like read/write ports as accurately as possible with the available data. In general, we need to build more detailed models of microarchitectural structures. Alternatively, having more collective experience for generalizing fudge factors for key units would be valuable.
- Modeling assumption error:** Users of McPAT should take care to correctly model shared resources for SMT when appropriate. Modeling of control and arithmetic logic is difficult at the architectural level, but we believe there is potential in building semi-empirical models by characterizing hardware from open source projects like RISC V [52], FabScalar [9], chip generators [14], and OpenSPARC [46].
- Input error:** Users should carefully specify modeling parameters, but it is comforting to find that in our study, input error was not a big contributor to overall power error.

In addition, we emphasize two guidelines about how power models can and should be validated when tools like DPM are not available:

- Validate at unit level using measured power.** We demonstrated that validating a power model at the core or chip level hides a large amount of internal error at the unit level and below. Targeted microbenchmarking is a well-known technique for characterizing fine-grained power [1, 27, 45].
- Validate leakage power.** Leakage is primarily a function of area, technology, and voltage/frequency, so it can be captured by more detailed analytical models and/or circuit simulations. Power-gating factors are important too, but researchers will usually have to settle for educated guesses.

Finally, the academic community would greatly benefit from the availability of validated power models for contemporary commercial chips and/or assistance with

power/performance validation studies as described in this work. There have been some industrial performance and power simulators that have been released, like Turandot [37] and PowerTimer [4] from IBM and XTREM from Intel [10], but many of these tools were designed for processors that are greatly outdated by current standards, and the release of updated core models would be extremely helpful. The power models developed in this paper can be downloaded at <http://vlsiarch.eecs.harvard.edu/mcpat>. Note that these models should only be used for core dynamic power analysis, *not* for uncore or leakage studies.

7. Related Work

Govindan et al. [16] validated a Wattch power model for a prototype TRIPS processor [7] against RTL simulations and hardware measurements to categorize and quantify the types of modeling error observed. We distinguish our work from theirs because POWER7 is a commercial superscalar server multicore chip whose design is much more complex than the TRIPS prototype, so it can potentially reveal edge cases in power models that a simpler CPU would not. Furthermore, the TRIPS architecture is an EDGE ISA, but McPAT was designed for modeling conventional pipeline models rather than graph-based execution models.

Zhai et al. [54] describe a power model called HaPPy that relies on a feature of recent Intel CPUs called Running Average Power Limit (RAPL) [24]. Although RAPL only provides total power for all cores on the chip, Zhai combines this information with performance counters to deduce information about hyper-threads and core-level activity. This technique could be extended to reveal additional unit-level data.

Mesa-Martinez et al. [36] presents a genetic algorithm that creates a power model by correlating a set of power equations to measured chip temperature. Their approach is unique, but their power model is validated with a multimeter that only measures total chip power, and as we have shown, power model validation at the chip or core level is insufficient. However, this method can provide unit-level power estimates, and one could extend the validation to the unit-level using techniques like microbenchmarks and/or HaPPy.

Jacobson et al. [25] describe methods for picking the best utilization metrics to use in cycle-accurate simulators and guidelines for designing power model abstractions. They focus on quantifying how sensitive power error is with respect to the pipeline event counters that are used in the models, whereas we examine the sensitivity of power error on the accuracy of microarchitectural parameters provided to the model.

8. Conclusion

In this work, we performed the first highly detailed assessment of McPAT's area and power models with the IBM POWER7™ core, using a proprietary power modeling tool as ground truth. We found that McPAT's predictions can have

significant error primarily due to abstraction errors and differences in modeling assumptions. When using McPAT to perform other studies like voltage noise and thermal hotspots, we find that studies focusing on temporal or spatial properties are most greatly impacted by dynamic power error, but those focusing on steady-state properties might be unaffected as long as leakage power is accurately modeled. Finally, we discuss specific guidelines researchers can observe to avoid such errors and ways to improve architectural power models going forward.

9. Acknowledgements

We thank Dean Tullsen and Chris Batten for their helpful feedback. We also thank the anonymous reviewers for their suggestions, Michael Healy and Thomas Strach of IBM for their guidance on voltage noise and PDN modeling, Runjie Zhang for the validated HotSpot model, and the POWER7 design team for their help. This work is sponsored by Defense Advanced Research Projects Agency, Microsystems Technology Office (MTO), under contract no. HR0011-13-C-0022. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This document is: Approved for Public Release, Distribution Unlimited.

References

- [1] R. Bertran, A. Buyuktosunoglu, M. S. Gupta, M. Gonzalez, and P. Bose, "Systematic Energy Characterization of CMP/SMT Processor Systems via Automated Micro-Benchmarks," in *International Symposium on Microarchitecture*, 2011.
- [2] A. Bhattacharjee and M. Martonosi, "Thread Criticality Predictors for Dynamic Performance, Power, and Resource Management in Chip Multiprocessors," in *International Symposium on Computer Architecture*, 2009.
- [3] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 Simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, 2011.
- [4] D. Brooks, P. Bose, V. Srinivasan, M. Gschwind, P. Emma, and M. Rosenfield, "New methodology for early-stage, microarchitecture-level power-performance analysis of microprocessors," *IBM Journal of Research and Development*, vol. 47, no. 5, 2003.
- [5] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," in *IEEE International Symposium on Computer Architecture*, 2000.
- [6] D. M. Brooks, P. Bose, and M. Martonosi, "Power-Performance Simulation: Design and Validation Strategies," *ACM SIGMETRICS Performance Evaluation Review*, vol. 31, no. 4, Mar. 2004.
- [7] 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*, vol. 37, no. 7, 2004.
- [8] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, Nov. 2011.
- [9] N. Choudhary, S. Wadhavkar, T. Shah, H. Mayukh, J. Gandhi, B. H. Dwiel, S. Navada, H. Najaf-abadi, and E. Rotenberg, "FabScalar: Composing Synthesizable RTL Designs of Arbitrary Cores within a Canonical Superscalar Template," in *International Symposium on Computer Architecture*, 2011.
- [10] G. Contreras, M. Martonosi, J. Peng, R. Ju, and G.-Y. Lueh, "Xtrem: A power simulator for the intel xscale core," in *Languages, Compilers, and Tools for Embedded Systems*, 2004.

- [11] A. K. Coskun, T. S. Rosing, and K. Whisnant, "Temperature Aware Task Scheduling in MPSoCs," in *Design, Automation, and Test in Europe Conference and Exhibition*, 2007.
- [12] J. Donald and M. Martonosi, "Techniques for Multicore Thermal Management," in *International Symposium on Computer Architecture*, 2006.
- [13] H. Esmailzadeh, A. Sampson, L. Ceze, and D. Burger, "Neural acceleration for general-purpose approximate programs," in *International Symposium on Microarchitecture*, 2012.
- [14] S. Galal, O. Shacham, J. S. B. II, J. Pu, A. Vassiliev, and M. Horowitz, "FPU Generator for Design Space Exploration," in *International Symposium on Computer Arithmetic*, 2013.
- [15] J. Gorbald, "Intel xeon x7560: Nehalem ex review," <http://bit-tech.net/hardware/cpus/2010/04/07/intel-xeon-x7560-nehalem-ex-review/>, accessed: 2014-08-28.
- [16] M. S. S. Govindan, S. W. Keckler, and D. Burger, "End-to-End Validation of Architectural Power Models," in *International Symposium on Low-Power Electronics and Design*, 2009.
- [17] E. Grochowski, D. Ayers, and V. Tiwari, "Microarchitectural Simulation and Control of di/dt-induced Power Supply Voltage Variation," in *High Performance Computer Architecture*, 2002.
- [18] M. Guevara, B. Lubin, and B. Lee, "Market mechanisms for managing datacenters with heterogeneous microarchitectures," *ACM Transactions on Computer Systems*, vol. 32, no. 1, 2014.
- [19] S. H. Gunther, F. Binns, D. M. Carmean, and J. C. Hall, "Managing the Impact of Increasing Microprocessor Power Consumption," *Intel Technology Journal*, vol. 5, no. 1, 2001.
- [20] M. S. Gupta, V. J. Reddi, G. Holloway, G.-Y. Wei, and D. M. Brooks, "An Event-Guided Approach to Reducing Voltage Noise in Processors," in *Design, Automation, and Test in Europe Conference and Exhibition*, 2009.
- [21] M. Healy, H.-H. Lee, G. Loh, and S. K. Lim, "Thermal optimization in multi-granularity, multi-core floorplanning," in *Asia and South Pacific Design Automation Conference*, 2009.
- [22] W. Heirman, S. Sarkar, T. E. Carlson, I. Hur, and L. Eeckhout, "Power-Aware Multi-Core Simulation for Early Design Stage Hardware/Software Co-Optimization," in *Parallel Architectures and Compilation Techniques*, 2012.
- [23] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A Compact Thermal Modeling Methodology for Early Stage VLSI Design," *IEEE Transactions on VLSI Systems*, vol. 14, no. 5, 2006.
- [24] *Intel 64 and IA-32 Architectures Software Developer's Manual, volume 3*, Intel Corporation.
- [25] H. Jacobson, A. Buyuktosunoglu, P. Bose, E. Acar, and R. Eickemeyer, "Abstraction and Microarchitecture Scaling in Early-Stage Power Modeling," in *High Performance Computer Architecture*, 2011.
- [26] R. Joseph, D. Brooks, and M. Martonosi, "Control techniques to eliminate voltage emergencies in high performance processors," in *High Performance Computer Architecture*, 2003.
- [27] A. M. Joshi, L. Eeckhout, L. K. John, and C. Isen, "Automated microprocessor stressmark generation," in *High Performance Computer Architecture*, 2008.
- [28] S. Kanev, G.-Y. Wei, and D. Brooks, "XIOSim: Power-Performance Modeling of Mobile x86 Cores," in *International Symposium on Low Power Electronics and Design*, 2012.
- [29] C. R. Lefurgy, A. J. Drake, M. S. Floyd, M. S. Allen-Ware, B. Brock, J. A. Tierno, and J. B. Carter, "Active Management of Timing Guardband to Save Energy in POWER7," in *International Symposium on Microarchitecture*, 2011.
- [30] J. Leng, T. Hetherington, A. EITantawy, S. Gilani, N. S. Kim, T. M. Aamodt, and V. J. Reddi, "GPUWatch: Enabling Energy Optimizations in GPGPUS," in *International Symposium on Computer Architecture*, 2013.
- [31] S. Li, J. H. Ahn, J. B. Brockman, and N. P. Jouppi, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *IEEE International Symposium on Microarchitecture*, 2009.
- [32] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "The mcpat framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 10, no. 1, p. 5, 2013.
- [33] X. Liang, K. Turgay, and D. Brooks, "Architectural Power Models for SRAM and CAM Structures Based on Hybrid Analytical/Empirical Techniques," in *International Conference on Computer Aided Design*, 2007.
- [34] Y. Malich, "AMD K10 Microarchitecture," <http://www.xbitlabs.com/articles/cpu/display/amd-k10.html>, accessed: 2014-08-28.
- [35] S. Mathew, M. Anders, B. Bloechel, T. Nguyen, R. Krishnamurthy, and S. Borkar, "A 4GHz 300mW 64b Integer Execution ALU with Dual Supply Voltages in 90nm CMOS," in *IEEE International Solid State Circuits Conference*, 2004.
- [36] F. J. Mesa-Martinez, J. Nayfach-Battilana, and J. Renau, "Power Model Validation Through Thermal Measurements," in *International Symposium on Computer Architecture*, 2007.
- [37] M. Moudgill, P. Bose, and J. H. Moreno, "Validation of Turandot, a Fast Processor Model for Microarchitecture Exploration," in *Performance, Computing and Communications Conference*, 1999.
- [38] U. G. Nawathe, M. Hassan, K. C. Yen, A. Kumar, A. Ramachandran, and D. Greenhill, "Implementation of an 8-Core, 64-Thread Power-Efficient SPARC Server on a Chip," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, 2008.
- [39] A. Patel, F. Afram, S. Chen, and K. Ghose, "MARSSx86: A Full System Simulator for x86 CPUs," in *Design Automation Conference 2011 (DAC'11)*, 2011.
- [40] A. Phansalkar, A. Joshi, and L. K. John, "Analysis of redundancy and application balance in the spec cpu2006 benchmark suite," in *International Symposium on Computer Architecture*, 2007.
- [41] V. Reddi, S. Kanev, W. Kim, S. Campanoni, M. D. Smith, G.-Y. Wei, and D. Brooks, "Voltage Smoothing: Characterizing and Mitigating Voltage Noise in a Production Processor Using Software-Guided Thread Scheduling," in *International Symposium on Microarchitecture*, 2010.
- [42] D. Sanchez and C. Kozyrakis, "The zcache: Decoupling ways and associativity," in *International Symposium on Microarchitecture*, 2010.
- [43] D. Sanchez and C. Kozyrakis, "ZSim: Fast and Accurate Microarchitectural Simulation of Thousand-Core Systems," in *International Symposium on Computer Architecture*, 2013.
- [44] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, "A Case for Thermal-Aware Floorplanning at the Microarchitectural Level," in *Journal of Instruction-Level Parallelism*, 2005.
- [45] H. Shafi, P. Bohrer, J. Phelan, C. Rusu, and J. Peterson, "Design and Validation of a Performance and Power Simulator for PowerPC Systems," *IBM Journal of Research and Development*, vol. 47, no. 5, 2003.
- [46] M. Shah, J. Barren, J. Brooks, R. Golla, G. Grohoski, N. Gura, R. Hetherington, P. Jordan, M. Luttrell, C. Olson *et al.*, "Ultrasparc t2: A highly-treaded, power-efficient, sparc soc," in *Asian Solid-State Circuits Conference*, 2007.
- [47] Y. S. Shao, B. Reagen, G.-Y. Wei, and D. Brooks, "Aladdin: A Pre-RTL, Power-Performance Accelerator Simulator Enabling Large Design Space Exploration of Customized Architectures," in *International Symposium on Computer Architecture*, 2014.
- [48] B. Sinharoy, R. Kalla, W. J. Starke, H. Q. Le, R. Cargoni, J. A. V. Norstrand, B. J. Ronchetti, J. Stuecheli, J. Leenstra, G. L. Guthrie, D. Q. Nguyen, B. Blaner, C. F. Marino, E. Retter, and P. Williams, "IBM POWER7 multicore server processor," *IBM Journal of Research and Development*, vol. 55, no. 3, 2011.
- [49] M. Srinivas, B. Sinharoy, R. J. Eickemeyer, R. Raghavan, S. Kunkel, T. Chen, W. Maron, D. Flemming, A. Blanchard, P. Seshadri, J. W. Kellington, A. Mericas, A. E. Petruski, V. R. Indukuru, and S. Reyes, "IBM POWER7 performance modeling, verification, and evaluation," *IBM Journal of Research and Development*, vol. 55, no. 3, 2011.
- [50] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The case for lifetime reliability-aware microprocessors," in *International Symposium on Computer Architecture*, 2004.
- [51] K. Wang, B. Meyer, R. Zhang, M. Stan, and K. Skadron, "Walking Pads: Managing C4 Placement for Transient Voltage Noise Minimization," in *Design Automation Conference*, 2014.
- [52] A. Waterman, Y. Lee, D. Patterson, and K. Asanovic, *The RISC-V Instruction Set Manual, Volume I: Base User-Level ISA*, University of California, Berkeley, 2011.
- [53] D. Wendel, R. Kalla, R. Cargoni, J. Clables, J. Friedrich, R. Frech, J. Kahle, B. Sinharoy, W. Starke, S. Taylor, S. Weitzel, S. Chu, S. Islam, and V. Zyuban, "The Implementation of POWER7: A Highly Parallel and Scalable Multi-Core High-End Server Processor," in *IEEE International Solid State Circuits Conference*, 2010.
- [54] Y. Zhai, X. Zhang, S. Eranian, L. Tang, and J. Mars, "HaPPY: Hypertread-aware Power Profiling Dynamically," in *Proceedings of the 2014 USENIX Conference*, 2014.
- [55] R. Zhang, K. Wang, B. H. Meyer, M. R. Stan, and K. Skadron, "Architecture Implications of Pads as a Scarce Resource," in *International Symposium on Computer Architecture (ISCA)*, 2014.