

Measuring Power and Temperature from Real Processors

Francisco J. Mesa-Martinez Michael Brown Joseph Nayfach-Battilana Jose Renau

Dept. of Computer Engineering, University of California Santa Cruz

<http://masc.soe.ucsc.edu>

Abstract

The modeling of power and thermal behavior of modern processors requires challenging validation approaches, which may be complex and in some cases unreliable. In order to address some of the difficulties associated with the validation of power and thermal models, this document describes an infrared measurement setup that simultaneously captures run-time power consumption and thermal characteristics of a processor. We use infrared cameras with high spatial resolution ($10 \times 10 \mu m$) and high frame rate (125Hz) to capture thermal maps. Power measurements are obtained with a multimeter at a sampling rate of $1KHz$. The synchronized traces can then be used in the validation process of possible thermal and power processor activity models.

1 Introduction

Temperature and power consumption have become first order design parameters for most modern, high performance architectures. Elevated operational temperature and power consumption present possible limits to performance and manufacturability. Due to the importance of this data, modern architects have extended their performance-centric processor simulation infrastructures to accommodate models of power consumption [1, 8] and thermal behavior [2, 7].

Wattch [1] and several Wattch-like tools are used routinely in the modeling of dynamic power consumption in modern processors. Wattch builds on top of CACTI [8], a popular power model for SRAM-like structures. The modeling of both static and leakage-associated power consumption is usually carried out using packages such as HotLeakage [9], which builds on top of the HotSpot [7] thermal model and the Wattch [1] power model.

Each of these tools has been individually validated to a varying degree, but the validation of a final integrated

model is not an easy process. This is mainly due to the fact that modern processors do not provide a sufficient (if any) means to validate such models. This limitation stems from the nature of the verification process for these kinds of tools.

Validation of real-time processor metrics demands the measurement of real-time responses from the processor itself. Designers obtain this real-time data using performance-monitoring structures – such as performance counters. Using these structures, designers can compare the real-time data collected from the processor with that predicted by the simulation environment. For example, different performance counters can provide metrics such as the average number of instructions per cycle (IPC) and instruction cache miss rate, or more detailed statistics like load-store queue replays. Those statistics make it possible to validate 3architectural simulators with existing processors.

However, This is not the case for power and thermal models. Unlike performance statistics, modern processors lack structures to gather power and thermal metrics. However, adding a sufficient number of power counters to obtain the needed level of granularity would consume a significant amount of area, and impact power consumption and processor performance.

To validate processor power models, the architecture community would like to observe the actual temperature and power behavior of proposed high performance systems. Without the measurement of real-time responses from the processor, the best efforts of the architecture community are reduced to guess work and approximations when modeling the power and thermal behavior of proposed architectural designs. Further, the cumulative impact of many power and thermal approximations may have a significant effect on the resulting accuracy of the simulated systems. Therefore, many architects do not trust the absolute results predicted for the behavior of their systems using current tools – relying instead on relative thermal or power behavior trends.

This paper describes an infrastructure to directly measure temperature and power consumption of modern processors. The proposed measuring system uses infrared cameras to capture transient temperature fluctuations. Power consumption is gathered by isolating the current used by the processor during run time. The data gathered can in turn be used to validate the accuracy of many power and thermal models.

The measuring setup described in this paper captures thermal maps and power consumption from modern processors. As a result, this work shows a unified measuring setup that captures temperatures on modern high-performance processors; develops image processing filters to increase the accuracy of thermal images; and measures floorplan temperature on a real chip.

2 Infrastructure

In our approach, processor temperature is measured using an infrared camera, while power consumption data is obtained by isolating the current used by the CPU directly at run time. The resulting system is capable of measuring power and temperature characteristics of modern high performance processors, all done with a very fine degree of granularity. Figure 1 displays the major components of the measuring setup.

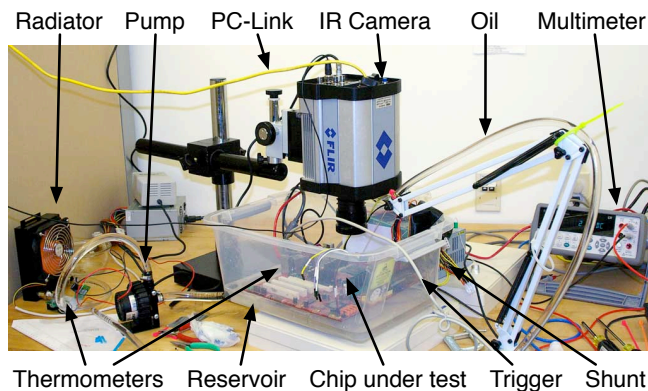


Figure 1: Measuring setup

As stated above, the proposed **measuring setup** (Section 2.1) captures the chip temperature with an infrared (IR) camera. An infrared-transparent heat sink is used to allow the IR camera to obtain the processor die temperature. This transparent heat sink is capable of dissipating up to 100W, thus it is aptly suited for most modern high performance processors. The setup is ca-

pable of capturing up to 125fps with a $10 \times 10 \mu m$ spatial resolution, and it can be applied to multiple chips with relative simplicity. The IR camera frame rate (125fps) can be increased up to 10KHz as long as the bandwidth of the camera stays under 1GB/s. E.g: If the measurement experiments require it, it is possible to capture 1000fps (1KHz) with a 100x80 resolution.

Modern IR cameras have resolutions in excess of 640x400 pixels, with a precision of 25mK error per pixel or less. Nevertheless, cameras suffer from a significant distortion of several degrees Kelvin error between pixels and need calibration for each specific lens, objective, and/or temperature range setup. To solve these problems and increase accuracy, we introduce a correction filter using **thermal image processing** (Section 2.2). The **power measurement** setup used to isolate dynamic CPU power consumption is explained in Section 2.3.

2.1 Measuring Setup

To generate an accurate thermal map for a given chip, we need to measure the temperature at multiple points. To do so, an infrared camera is used to measure temperature as close as possible to the transistor junction.

To keep the processor in operational conditions, an IR transparent heat sink must be implemented. To do so, we create a mineral oil (Fluka Mineral Oil 69808) flow on top of the silicon substrate. Even though water has around 2.5 times the specific heat of mineral oil, we can not use it because it is not transparent to the infrared spectrum. Several oils like olive oil are partially transparent on the infrared wavelengths. Fluka oil is designed for infrared spectrography and delivers excellent infrared pictures. Turbulent flow can remove more heat than laminar flow, however it is more complicated to correctly model. For that reason, on the L2 side of the processor, we add filling with the same height as the silicon. The oil impacts on the filling and generates a flow from L2 to the core. We keep the oil flowing as fast as possible to minimize heat transfer from L2 to the core. Our measurements show less than 0.1C oil temperature increase from side to side of the chip.

The oil temperature is continually monitored with multiple digital thermometers (Dallas DS18B20) connected to the measuring computer. The setup is capable of dissipating up to a 100W. We keep 2 liters in the oil reservoir and connect a small radiator to guarantee minimal temperature oscillations during each run.

A **detailed thermal map** is obtained with an infrared camera (FLIR SC-4000). Using the PC-Link (Gigabit Ethernet), the camera is set up to capture and transfer 125fps with 320x200 spatial resolution. This camera operates on the 3-5 μ m wavelength (MWIR) a range of light where silicon is transparent. As a result, the IR camera is capable of measuring the temperature “through” the chip being tested. Modern high performance processors are manufactured using flip chips – exposing the silicon substrate. Since the camera can measure temperatures through the silicon substrate, using flip-chips¹ greatly simplifies the task of measuring junction temperatures. Although the SC-4000 has 25mK sensitivity per pixel, it requires extensive calibration in order to obtain accurate thermal measurements.

2.2 Thermal Image Processing

Due to their operational characteristics, infrared cameras are calibrated to compensate for different material emissivities, lens configurations, temperature ranges for the object/material measured, and a host of other factors. One approach for calibration is to have the infrared measuring device calibrated for the specific setup by the manufacturer. However, this ignores the temperature range of the object and increases the likelihood of measurements being made outside of the calibrated range. To solve this problem, we perform an in-house calibration.

Indium antimonide (InSb) sensors, like the one found in the FLIR camera used in the measurement setup, have a high sensitivity per pixel (25mK). This corresponds to the camera’s optimal accuracy once it is correctly calibrated. To compensate for the camera error, we perform two controlled measurements: one with cold (16°C) and one with hot (71°C) mineral oil on top of an inactive (off) processor’s silicon substrate.

The camera specifications indicate that a linear (“Temp” = $A * \text{IR Temp} + B$) correction should be applied for each camera pixel. Our image filter automatically generates a linear correction factor to compensate for the inaccuracies. A secondary filter is used to compensate for the optical distortion induced by the lens setup and to register the camera.

¹Low power chips tend to be wire-bond, while more high-performance chips tend to be flip-chip.

2.3 Power Model

To measure the overall power consumption, we intercept the 12V wires that provide power to the voltage regulators (VR) on-board. To have a low overhead and high accuracy measurement, we use a shunt (LTS 25-NP). Shunts provide higher accuracy than clamping and other inductive approaches to measure current, at the cost of it being a more intrusive approach. However, simply measuring the power supplied by the processor 12V cable is not enough. The reason is that the voltage regulators (VR) from the motherboard are not 100% efficient. To have accurate absolute power measurements, we discount the power wasted due to VR. Note that to know the efficiency, it is necessary to find the VR specifications.

The voltage reported by the shunt is measured with an Agilent 34410A multimeter. This multimeter is capable of sampling at 1KHz and storing over 50 seconds of execution. To read the power measurements, we use a TCP/IP ruby script.

3 Evaluation

The main objective of the work is to propose an infrastructure to capture temperature with high resolution and obtain the associated power. This evaluation shows the type of data generated by our approach.

Figure 2-(a) shows the temperature and power profile for the first 20 seconds of execution (2500 frames) of the *applu* benchmark from SpecFP2000. The solid lines correspond to a run where the oil is heated to 33.5C. The dashed lines correspond to a run with a 22C oil temperature. As expected even though the initial oil temperature difference is just 11.5C the total temperature difference after 20 seconds is slightly higher (13C). The power consumption is higher on the high temperature (HT) run than on the low temperature (LT) run. The reason is that as we increase the oil temperature the leakage power also increases. The plot shows that on average for a 13C increase, the total power consumption increases by 5.3%.

Figures 2-(b) and 2-(c) show the average temperature for several floorplan blocks as the *applu* and *apsi* benchmarks execute respectively. The floorplan blocks, starting from highest to lowest average temperature, are the register file, data cache (D\$), floating point unit (FP0), clock generator, memory controller (MC), and instruction cache (I\$). For both applications, the temperature

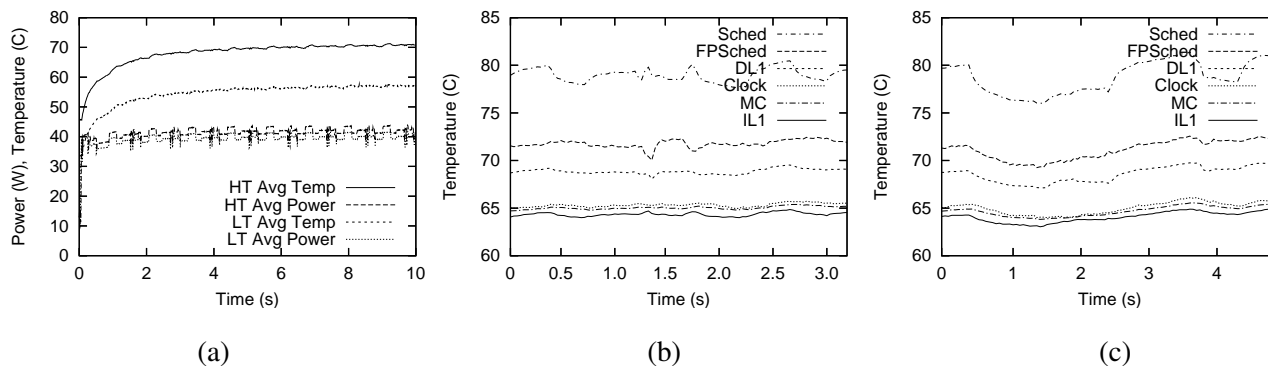


Figure 2: Thermal and power data for the first 20 seconds of applu (a); temperature profile for the memory controller (MC), register file (RF), instruction cache (I\$), data cache (D\$), and floating point unit for applu (b) and apsi (c).

across blocks is somewhat correlated. However, there are several situations where the data cache temperature increases while the register file temperature decreases (1.5s for applu (Figure 2-(b)), 4s for apsi (Figure 2-(c)). In both cases, we find a phase where the access to the register file decreases and the rate of memory operations increases. For most of the applications, the register file is close to 10C hotter than any other floorplan block.

3.1 Thermal Imaging

This section shows the raw thermal images and provides additional insights on the image processing performed on this paper.

As section 2.2 states, the IR camera does not have the same accuracy over all the pixels. The central pixels have a higher accuracy than side pixels. To compensate for this error, we perform a linear correction on a per-pixel basis. Figure 3-(a) shows the corrected thermal image for a single frame, including a floorplan overlay.

The overlay on Figure 3-(a) does not cover the whole picture. The upper part of the figure shows part of the L2 cache. The picture seems to indicate that the pixels outside the die visible on the left and lower part are as hot as the die itself. The measurements on these areas have two artifacts. First, the emissivity is different outside the die area. Second, the fluid has turbulence outside the die. This turbulence creates fluctuations in the thermal measurements. As a result, measurements outside the die area are not accurate for our setup.

Figures 3-(a) and 3-(b) show that there is temperature variability inside the floorplan blocks. It is this variability that prompted an extension to our thermal model so that each floorplan block could be modeled

with fine granularity. Therefore, although we report the average temperature for each floorplan block, our thermal model internally computes multiple temperature points for each block.

Figures 3-(b) and Figure 3-(c) show the frame from *crafty* (SpecINT2000) with the maximum temperature measured. On this frame the register file reached 84C. The average temperature per floorplan block is shown on Figure 3-(c).

4 Related Work

Real power consumption measurements are a very useful tool to the computer architecture community. Isci et al [5] measure the overall power consumption with a multimeter. Together with the activity rate captured from the processor performance counters, they provide the total power breakdown for each processor floorplan area. A major difference between the setup described by Isci and our power measurement setup is the use of a shunt instead of clamp amp meter to increase measurement accuracy. In addition, we also take into account the efficiency of the on-board voltage regulator. A bigger difference, is that our setup also captures temperature.

A related work by Sung et al [3] builds on top of models that use performance counters to generate detailed thermal maps. Their work compares a less compute intensive regression model against a HotSpot thermal map.

Hamann et al [4] introduce a system to measure the temperature on a chip with a infrared camera. Their setup is similar to ours but they do not provide enough details on the materials/components used. Neverthe-

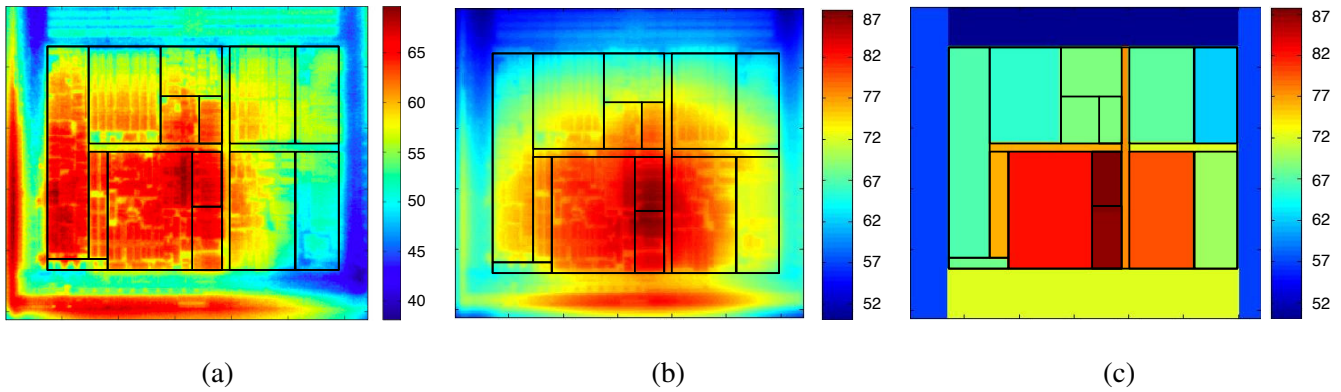


Figure 3: Full-thermal image with overlapped floorplan (a); hottest captured image (b); and its average temperature per block (c).

less, the key difference is that their setup only performs measurements for steady state thermal images and they do not provide a method to capture power and/or activity rates synchronously.

The most related work [6] employs a similar thermal and power measurement setup. However, the focus of that work is on building a novel model that relates thermal processor maps to fine-grain power consumption by using a genetic algorithm.

5 Conclusions

In this work, we show a system capable of simultaneously measuring the power and temperature characteristics of a modern high performance processor. This is all done with a very fine degree of granularity. We believe the development of such infrastructures is necessary in order to develop and validate advanced thermal and power models.

The setup was described in detail, and further data and software is available publicly. Our goal is to allow other research groups to reproduce our approach, to improve on it, and to apply it to different research topics in the computer architecture community.

Acknowledgments

This work was supported in part by the National Science Foundation under grants 0546819 and 720913; Special Research Grant from the University of California, Santa Cruz; Sun OpenSPARC Center of Excellence at UCSC; gifts from SUN, Altera, Xilinx, nVIDIA, and ChipEDA. Any opinions, findings, and conclusions or recommendations expressed herein are those of the au-

thors and do not necessarily reflect the views of the NSF.

References

- [1] D. Brooks, V. Tiwari, and M. Martonosi. Watch: a Framework for Architectural-Level Power Analysis and Optimizations. In *International Symposium on Computer Architecture*, pages 83–94, Jun 2000.
- [2] Y.K. Cheng, P. Raha, C.C. Teng, E. Rosenbaum, and S.M. Kang. ILLIADS-T: An Electrothermal Timing Simulator for Temperature-Sensitive Reliability Diagnosis of CMOS VLSI Chips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(8):1434–1445, Aug 1998.
- [3] S.-W. Chung and K. Skadron. Using On-Chip Event Counters For High-Resolution, Real-Time Temperature Measurement. In *Thermal and Thermomechanical Phenomena in Electronics Systems, 2006*, pages 114–120. IEEE Computer Society, May 2006.
- [4] H.F. Hamann, J. Lacey, A. Weger, and J. Wakil. Spatially-resolved imaging of microprocessor power (SIMP): hotspots in microprocessors. In *Thermal and Thermomechanical Phenomena in Electronics Systems, 2006*, pages 121–125. IEEE Computer Society, May 2006.
- [5] C. Isci and M. Martonosi. Runtime power monitoring in high-end processors: Methodology and empirical data. In *MICRO 36: Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2003.
- [6] F. Mesa-Martinez, J. Nayfach-Battilana, and J. Renau. Power model validation through thermal measurements. In *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, New York, NY, USA, June 2007. ACM Press.
- [7] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-Aware Microarchitecture. In *Proceedings of the 30th Annual International Symposium on Computer Architecture*, pages 2–13, Jun 2003.
- [8] S. Wilton and N. Jouppi. CACTI: An Enhanced Cache Access and Cycle Time Model. *IEEE Journal on Solid-State Circuits*, 31(5):677–688, May 1996.
- [9] 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, Univ. of Virginia Dept. of Computer Science, March 2003.