

Power Mapping and Modeling of Multi-core Processors

Kapil Dev
School of Engineering
Brown University
Providence, RI 02912
Email: kapil_dev@brown.edu

Abdullah Nazma Nowroz
School of Engineering
Brown University
Providence, RI 02912
Email: abdullah_nowroz@brown.edu

Sherief Reda
School of Engineering
Brown University
Providence, RI 02912
Email: sherief_reda@brown.edu

Abstract—We propose new techniques for post-silicon power mapping and modeling of multi-core processors using infrared imaging and performance counter measurements. An accurate finite-element modeling framework is used to capture the relationship between temperature and power, while compensating for the artifacts introduced from substituting traditional heat removal mechanisms with oil-based infrared-transparent cooling mechanisms. We use thermal conditioning techniques to build leakage power models for the die. Utilizing the power maps identified from infrared mapping, we develop empirical power models for different processor blocks based on the measurements from the performance monitoring counters (PMCs), and utilize the PMC-based models to analyze the transient power consumption. In our experiments, we capture thermal images from a quad-core processor under different workload conditions, and then we reconstruct the dynamic and leakage power maps for different blocks. Our results show good accuracy in mapping and modeling, revealing good insights into the trends of power consumption in multi-core processors.

I. INTRODUCTION

Power is a major design challenge due to the highly-complex nature of modern chips. This complexity makes accurate pre-silicon power modeling a very difficult task [3], [12]. Furthermore, workloads and process variability alter the power consumption during runtime, making it harder to accurately estimate power consumption during design time. In recent years, post-silicon power mapping has emerged as a technique to mitigate the uncertainties in design-time power models and enable effective post-silicon power characterization [5], [11], [13], [4]. Many of these techniques rely on inverting the thermal emissions captured from an operational chip into a power map. This highly-versatile approach faces numerous challenges, including the need for accurate thermal to power modeling; the need to remove artifacts introduced by the infrared experimental setup; and variabilities introduced by leakage power.

In this paper we propose a complete framework for post-silicon power mapping and modeling that solves many of the open challenges in this area. Our framework is capable of identifying the dynamic and leakage power per-block of multi-core processors under different workloads, while simultaneously analyzing the impact of process variability on leakage and capturing the relationship between the performance monitoring counters (PMCs) and per-block power consumption. Our method can be used to validate and calibrate design-time power and thermal models. The contribution of this paper are as follows.

- 1) We propose a numerical technique that uses accurate finite-element modeling (FEM) to translate the measured thermal maps captured from infrared-transparent heat sink systems to corresponding thermal maps of traditional metal and fan sinks, and then inverts the translated thermal maps to power maps. The proposed technique compensates for the thermal artifacts introduced by oil-based setup and can substitute for experimental techniques to match thermal behavior of different sinks [10].
- 2) We use thermal conditioning to devise spatial leakage variability models. The leakage models enable us to decompose the per-block power consumption into its dynamic and leakage components. Once estimated for a given chip, these leakage models can be used to compute leakage power map for any workload readily from its thermal-map alone, hence simplifying the overall power-mapping process.
- 3) We collect PMC values while simultaneously performing infrared-based power mapping. The PMC values are correlated with the power maps to identify the PMCs that are directly responsible for the power consumption of each block. Unlike previous works, [12], [2], [1] which had no access to the actual per-block power consumption, we develop per-block mathematical models by relating the measured PMCs to the per-block power consumption as calculated by the infrared power mapping framework. We use the PMC-based models to analyze the transient power consumption of each processor block.
- 4) We apply our proposed framework on a real quad-core processor to get detailed dynamic and leakage powers for different blocks (e.g. cores, L2-caches, etc.) while executing workloads using multiple SPEC 2006 benchmarks. Proposed PMC-based models are used to estimate power dissipation in each blocks of the processor. Our results provide useful insights into the distribution of power in multi-core processors.

The organization of the paper is as follows. Section II describes the proposed framework for post-silicon power mapping and modeling. Section III describes techniques for leakage power mapping using thermal conditioning. We devise in Section IV empirical models that relate the per-block power consumption to the measurements of PMCs. We provide an extensive set of experimental results in Section V. Finally, Section VI summarizes the main conclusions of this work.

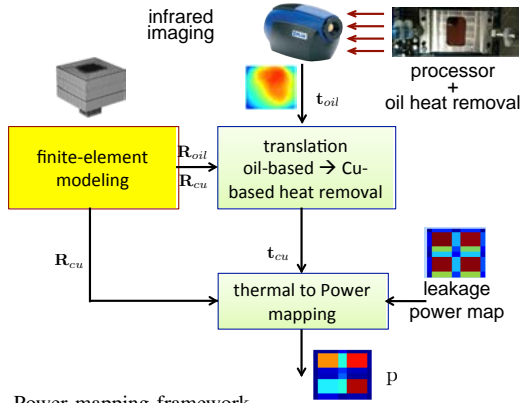


Fig. 1. Power mapping framework.

II. POWER MAPPING FRAMEWORK

Figure 1 illustrates the proposed power mapping framework. In our setup, the processor’s regular fan and metal heat spreader are removed and replaced by an infrared-transparent heat sink with silicon windows. Laminar mineral oil flow is pumped through the heat sink on top of the processor’s die with high flow rate to remove its heat [5], [11], [14]. During runtime, realistic workloads are applied to the processor and the steady-state or averaged thermal map is captured with the infrared camera. We will use \mathbf{t}_{oil} to denote the vector that corresponds to the captured thermal map of the processor.

Replacing the fan and metal heat spreader with an oil-based infrared transparent heat sink changes the thermal map of the die [6]. The changes in the thermal map have negligible impact on dynamic power, but they change the leakage power characteristics. Previous approaches attempted to compensate for this effect by altering the design of the infrared-transparent sink until the measurements from the processor’s internal sensors matched the measurements from the sensors when the regular metal heat sink is applied [10]. Moreover, it is still required to compensate for the directional effect due to the oil flow; a simple linear compensation technique, as used in [10], would not work well for the application-dependent temperature profiles of real processors. Therefore, instead of experimental approach, we apply a numerical approach that translates \mathbf{t}_{oil} to the thermal map, \mathbf{t}_{cu} , that would have resulted if the regular fan and copper (Cu) heat spreader were applied. Our easier accurate approach eliminates the need for any experimental modifications.

For power mapping, it is necessary to have an accurate modeling matrix \mathbf{R} that relates temperature to power at the steady state. This modeling matrix can be estimated experimentally [5], [4] or numerically using FEM methods [8]. We propose to use FEM methods to accurately estimate two modeling matrices (only one time effort per processor design): \mathbf{R}_{oil} for the case of the oil-based heat sink, and \mathbf{R}_{cu} for the case of traditional heat spreader and fan-based heat sink. We use the FEM tool COMSOL to capture the exact models, encompassing all physical factors such as, cooling fluid temperature, fluid flow rate, heat transfer coefficients, and chip geometry. Generating \mathbf{R}_{oil} matrix takes longer time (about 2.5 hours on our desktop computer having Intel i7 CPU running at 2.8GHz and with 8GB memory) than generating \mathbf{R}_{cu} matrix (less than 10 minutes); this is because for the oil-based system, we need to simulate both fluid-flow and heat-transfer physics simultaneously, while for the Cu-based

system, we just need to simulate the heat-transfer physics. To verify the accuracy of \mathbf{R}_{oil} , we applied a known power map \mathbf{p}_k and verified that the numerical results $\mathbf{R}_{oil}\mathbf{p}_k$ match the infrared-based thermal image. To verify the accuracy of \mathbf{R}_{cu} , we confirmed the matching between the measurements from the processor’s thermal sensors to the corresponding elements in the vector $\mathbf{R}_{cu}\mathbf{p}_k$ when the metal heat spreader is applied. Previous approaches to model \mathbf{R} in simulation were only done for metal heat spreader with the objective of speeding thermal simulation runtime, where the model matrix \mathbf{R} is used to substitute lengthy FEM-based thermal simulations [8].

If a power map \mathbf{p} is simulated using both FEM models, then obtain the following two equations:

$$\mathbf{R}_{oil}\mathbf{p} = \mathbf{t}_{oil} \Rightarrow \mathbf{p} = (\mathbf{R}_{oil}^T\mathbf{R}_{oil})^{-1}\mathbf{R}_{oil}^T\mathbf{t}_{oil} \quad (1)$$

$$\mathbf{R}_{cu}\mathbf{p} = \mathbf{t}_{cu} \quad (2)$$

Substituting Equation (1) into Equation (2), we get

$$\mathbf{t}_{cu} = \mathbf{R}_{cu}(\mathbf{R}_{oil}^T\mathbf{R}_{oil})^{-1}\mathbf{R}_{oil}^T\mathbf{t}_{oil} \quad (3)$$

Given \mathbf{t}_{oil} , Equation (3) provides a means to translate the oil-based thermal image to a thermal image that would have resulted from using traditional metal heat spreaders. Our numerical translation method is quite generic and it could be used to translate thermal maps between any two cooling systems. The thermal map, \mathbf{t}_{cu} , is then numerically inverted to yield the per-block power maps, where we use the leakage power map as lower bound constraint. The procedure to estimate the leakage power map is given in Section III. In particular, we solve following constrained optimization problem to reconstruct the power map of the die.

$$\mathbf{p}^* = \arg_{\mathbf{p}} \min \|\mathbf{R}_{cu}(\mathbf{p} - (\mathbf{R}_{oil}^T\mathbf{R}_{oil})^{-1}\mathbf{R}_{oil}^T\mathbf{t}_{oil})\|_2 \quad (4)$$

such that, $\forall i: p_i \geq p_{lkg,i}$

where, \mathbf{p}^* is the reconstructed power-vector, $p_{lkg,i}$ denotes the leakage power in the i^{th} die-block, and p_i denotes i^{th} element of \mathbf{p} , i.e., the power in the i^{th} block of the die. By solving the above optimization problem (used MATLAB *lsqlin* function), we obtain the total power of each block for the die. The dynamic power of each block is readily obtained by subtracting the leakage power from the reconstructed total power. Using the $p_i \geq p_{lkg,i}$ constraint helps in ensuring that dynamic power for all blocks is always positive. A total power constraint could also be added to reduce the reconstruction-error further [13], [4]. Our power mapping framework provides the dynamic, leakage, and total powers for each block of a processor given its thermal map. While we used the proposed framework for power mapping and modeling of a multi-core processor, the framework is quite generic and could be applied on other type of integrated circuits as well. Unless otherwise stated, we use the translated metal heat spreader based thermal maps for all our analysis onwards.

III. MAPPING LEAKAGE VARIABILITY

Leakage power, especially its dominant sub-threshold component, depends exponentially on temperature. But within the typical chip operation range 25 - 85 °C, it has a quadratic dependency on temperature, which can be modeled by second-order Taylor series expansion at a reference temperature. In order to compute the chip’s spatial leakage power map, we divide the die area into a grid with large number of locations

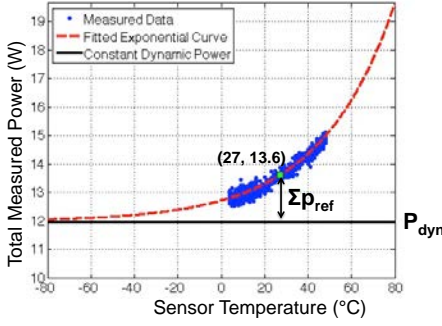


Fig. 2. Measured power vs. average chip temperature, while keeping dynamic power unchanged. P_{dyn} denotes the dynamic power and $\sum p_{ref}$ denotes the total leakage power at reference temperature 27 °C.

n . For each location i , we develop a second-order Taylor expansion model for leakage power, $p_{lkg,i}$, as a function of the average temperature, t_i , of location i . The expansion around a reference power, $p_{ref,i}$, and temperature, $t_{ref,i}$, is given by

$$p_{lkg,i} = p_{ref,i} + \alpha_{1,i}(t_i - t_{ref,i}) + \alpha_{2,i}(t_i - t_{ref,i})^2 \quad (5)$$

where $\alpha_{1,i}$ and $\alpha_{2,i}$ are the model coefficients for location i that depend on the voltage, process variability, and structure of devices. The total leakage power, P_{lkg} is the sum of leakage of the chip's n locations, which can be written as:

$$P_{lkg} = \sum_i^n p_{ref,i} + \sum_{i=1}^n [\alpha_{1,i}(t_i - t_{ref,i}) + \alpha_{2,i}(t_i - t_{ref,i})^2]$$

which can be re-arranged as

$$\Delta P = \sum_{i=1}^n \alpha_{1,i} \Delta t_i + \alpha_{2,i} \Delta t_i^2, \quad (6)$$

where $\Delta P = P_{lkg} - \sum_i p_{ref,i}$ and $\Delta t_i = t_i - t_{ref,i}$. Note that ΔP , which is the change in total power, is readily obtained using an external multimeter that measures the total power of the processor, and Δt_i is measured using the thermal maps provided from our infrared imaging or from the translated thermal maps as described in Section II.

To learn the model coefficients, we repeat the thermal conditioning experiment m times with different ambient temperatures, and for each experiment, we measure the change in total power and change in the thermal map. The j^{th} thermal conditioning experiment provides a thermal image which consists of $\Delta t_{j,i}$ at each chip location i and an incremental total leakage power ΔP_j , which creates an instance of Equation (7).

$$\Delta P_j = \sum_{i=1}^n \alpha_{1,i} \Delta t_{j,i} + \alpha_{2,i} \Delta t_{j,i}^2 \quad (7)$$

The results from the m thermal conditioning experiments can be assembled into a system of equations as follows

$$\begin{bmatrix} \Delta t_{1,1} & \Delta t_{1,1}^2 & \cdots & \Delta t_{1,n} & \Delta t_{1,n}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Delta t_{m,1} & \Delta t_{m,1}^2 & \cdots & \Delta t_{m,n} & \Delta t_{m,n}^2 \end{bmatrix} \begin{bmatrix} \alpha_{1,1} \\ \alpha_{2,1} \\ \vdots \\ \alpha_{1,n} \\ \alpha_{2,n} \end{bmatrix} = \begin{bmatrix} \Delta P_1 \\ \vdots \\ \Delta P_m \end{bmatrix} \quad (8)$$

We solve the above system of equations using least-square regression to find the $2n$ α first-order and second-order model coefficients. To compute the total reference leakage power, $\sum_i p_{ref,i}$ in Equation (6), one can change the ambient temperature of the chip while keeping the dynamic power constant

(by running a stable workload), and measuring the total power consumption and the average chip temperature simultaneously. To estimate the total reference leakage power, an exponential model of the measured power to the chip's average temperature can be used to extrapolate it to the point where leakage power tapers off. As shown in Figure 2, for our experimental quad-core processor, we estimate the total reference leakage power at 27 °C as 1.6 W, and stable dynamic power, P_{dyn} as 12 W. For a particular chip, these coefficients need to be computed only once, and then the same coefficients are used for estimating fine-resolution leakage maps for any thermal map of the chip as given in the framework of Figure 1.

Process Variability Mapping. In a typical power mapping experiment, the temperature of location i is plugged into Equation (5) to estimate the leakage power of location i . If it is desired to estimate the inherent spatial leakage variability arising from process variability, then a fixed temperature could instead be plugged into the equations of all chip locations. By using the same temperature everywhere, the leakage variations that arise will be due to the coefficients $\alpha_{1,i}$ and $\alpha_{2,i}$ which are dependent on the inherent process variability, assuming a fixed operating voltage.

IV. POWER MODELING USING PMCS

A popular approach for modeling total power is through the use of *performance monitoring counters* (PMCs) [12], [2], [7], [9], [1]. Performance counters are embedded in the processor to track the usage of different processor blocks. Examples of such events include the number of retired instructions, the number of cache hits, and the number of correctly predicted branches. In contrast to previous works, where the PMCs are related and modeled to total chip power or simulated power, we relate actual power of each circuit block as estimated through infrared-based mapping to the runtime PMCs. This gives accurate per-block PMC models and enable us to directly isolate the PMCs responsible for power consumption for each block. The PMC based models can be then used to model the transient power consumption and in situations where no infrared imaging system is available as in the case of end users.

Our infrared-based power mapping technique directly obtains the power consumption of each circuit block under different workload conditions. We propose to simultaneously collect the measurements of the PMCs, while collecting the infrared imaging data. The post-silicon power estimates are then used to derive fitted empirical models that relate the performance counters to the power consumption of each block. For instance, if m_1 , m_2 , and m_3 are three PMCs correlated to the power estimates, p_i , of block i , then an empirical model, \hat{p}_i , can be described as $\hat{p}_i = c_0 + c_1 m_1 + c_2 m_2 + c_3 m_3$, where c_0 , c_1 , c_2 and c_3 are the model coefficients, which have to be determined by fitting the observed power estimates of each block with the PMC measurements on a training set of workloads. The fitting is done using least-square estimation, where it is desired to minimize the modeling error, $(\hat{p}_i - p_i)^2$ over the training data. The main steps of our power modeling procedure are summarized in Figure 3.

The fitted PMC models enable us to substitute the post-silicon power mapping results in situations where infrared imaging is difficult. These include, for example, systems deployed in user environments where access to infrared imaging is not easy, or for high-resolution transient power mapping.

Procedure: PMC-based power modeling procedure
Input: Infrared-based power estimates for each block and associated PMC measurements
Output: Power Models for each block as a function of PMC measurements

- 1) For each circuit block i :
 - a. Identify the PMC measurements that are strongly correlated or anti-correlated with power estimates of i .
 - b. Use least-square estimation to fit a linear model that estimates the power of i as a function of the strongly-correlated PMC measurements.

Fig. 3. Algorithm to compute PMC-based models.

Infrared-based transient power mapping is inherently limited because of the low-pass filtering of power variations and the limited sampling rate of infrared cameras [13]. PMC-based modeling circumvents the transient analysis limitations of infrared imaging. We illustrate the use of PMC-based models for transient power modeling in Section V.

V. EXPERIMENTAL SETUP AND RESULTS

Our experimental system consists of a motherboard fitted with a 45 nm AMD Athlon II X4 610e quad-core processor and 4 GB of memory. The motherboard runs Linux OS with 2.6.10.8 kernel. The floorplan of the processor with 11 different blocks is shown in Figure 4. We treat each core as one block, as we could not find public-domain information on the make-up of blocks within each core. It is worth mentioning that our proposed technique of power mapping is generic and will work for any arbitrary layout details we use for reconstructing power maps. The processor has 4×512 KB L2 caches, but it lacks a shared L3 cache. The area in the center is occupied by the northbridge and other miscellaneous components such as the main clock trunks, the thermal sensor, and the built-in thermal throttling and power management circuits. The periphery is composed of the devices for I/O and DDR3 communication. The processor supports four distinct DVFS settings. Except for the R-matrix verification experiment, we set the DVFS to 1.7 GHz.

We image the processor using a mid-wave FLIR 5600 camera with 640×512 pixel resolution. We also intercept the 12 V supply lines to the processor and measure the current through a shunt resistor connected to an external Agilent 34410A digital multimeter, which enables us to log the total power measurements of the processor. To implement thermal conditioning in our experimental setup, we use a thermoelectric device and a fluid monitoring device in line with the oil flow [14]. By changing the voltage and current of the thermoelectric

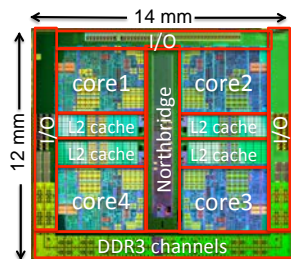


Fig. 4. Layout of the quad-core AMD Athlon II X4 processor.

	memory bound	processor bound
Integer point	<i>omnetpp</i>	<i>hmmmer</i>
Floating point	<i>soplex</i>	<i>gamess</i>

TABLE I. SELECTED SPEC CPU2006 BENCHMARKS.

device, we can either cool or heat the fluid to any desired temperature. Thus, we setup a feedback control system to control the fluid temperature to any desired set point.

A. Power Mapping Results

The goal of the first experiment is to demonstrate the results of power mapping for the processor using different number of workloads and different workload characteristics. Our workloads come from widely used SPEC CPU2006 benchmark suite. We selected four benchmark applications, which cover both integer point and floating point computations and processor-bound and memory-bound characteristics. These benchmarks are listed in Table I.

We ran 15 different cases of workload sets. For each experiment, we captured the steady-state thermal image using an infrared-camera and reconstructed the underlying power maps from the translated thermal maps to the Cu-based spreader as proposed in Section II. We decomposed the total power maps into dynamic and leakage power dissipation of each block of the processor and analyzed the spatial leakage variability as described in Section III. For example, the reconstructed maps for four sample cases are shown in Figure 5. The third row shows a case, where we ran *soplex*, *gamess*, and *hmmmer* benchmarks on cores 1, 2, and 3 respectively. Second column shows the equivalent temperature maps for Cu-system for each workload case. The third column shows the reconstructed total power dissipation in each block for the four cases. It is clear from the reconstructed power-maps that they agree with the intuitive expectation that cores running processor-bound applications (i.e., *hmmmer* and *gamess*) are having higher power consumption than the idle cores or cores running memory-bound workloads. Similarly, fourth and fifth column show the per-unit reconstructed dynamic power and leakage power for four different workloads. The figures also show that the L2 cache power is mainly dominated by leakage power with a small amount of dynamic power.

The per-block power results for 15 sample workload cases are presented in Table II. We also report the total dynamic power, total leakage power, and the sum of leakage and dynamic power. The results show that the leakage power is on

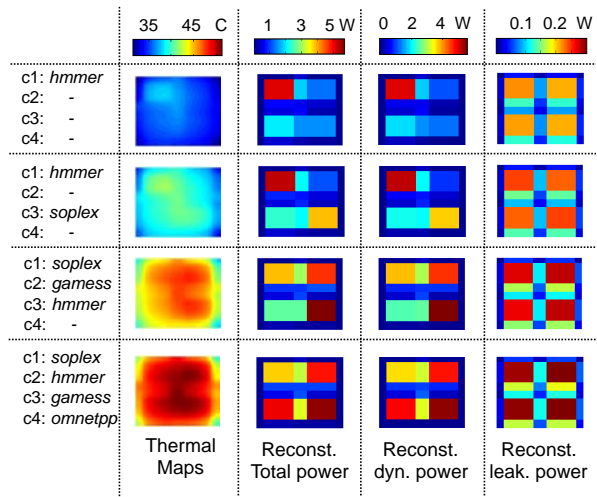


Fig. 5. Thermal maps, reconstructed total-power, dynamic-power and leakage-power maps.

	core 1	core 2	core 3	core 4	Reconstructed total power (W) for each block										Total power (W)				
					core 1	L2-1	core 2	L2-2	core 3	L2-3	core 4	L2-4	I/O	N. B.	DDR3	dyn	lkg	dyn+lkg	meas
omnetpp	-	-	-	-	3.61	0.66	1.41	0.38	1.58	0.22	2.07	0.46	0.73	4.21	0.95	14.11	2.18	16.28	16.75
hmmmer	-	-	-	-	5.10	0.68	1.27	0.39	1.47	0.21	1.99	0.46	0.68	4.52	0.76	15.29	2.24	17.53	18.42
soplex	-	-	-	-	3.69	0.70	1.33	0.39	1.58	0.21	2.06	0.44	0.70	4.30	0.91	14.13	2.18	16.31	17.04
gamess	-	-	-	-	4.86	0.74	1.27	0.37	1.36	0.22	1.92	0.45	0.67	4.37	0.71	14.72	2.21	16.93	18.16
omnetpp	-	soplex	-	-	3.50	0.73	1.25	0.48	3.85	0.30	2.28	0.45	0.71	5.25	0.86	17.31	2.36	19.66	19.78
omnetpp	-	hmmmer	-	-	3.61	0.76	1.13	0.52	5.47	0.22	2.28	0.46	0.71	5.71	0.74	19.14	2.46	21.60	21.56
omnetpp	-	gamess	-	-	3.72	0.77	1.16	0.50	5.22	0.31	2.26	0.47	0.71	5.66	0.73	19.05	2.46	21.51	21.49
hmmmer	-	soplex	-	-	5.30	0.82	1.15	0.50	3.85	0.31	2.32	0.48	0.68	5.76	0.78	19.48	2.48	21.96	21.63
hmmmer	-	gamess	-	-	5.34	0.83	1.03	0.52	5.13	0.32	2.21	0.48	0.68	6.08	0.59	20.66	2.55	23.21	23.24
soplex	-	gamess	-	-	3.89	0.82	1.16	0.52	5.33	0.31	2.33	0.47	0.71	5.80	0.74	19.59	2.49	22.08	21.85
omnetpp	soplex	gamess	-	-	3.69	0.86	3.23	0.90	5.58	0.38	2.59	0.50	0.76	6.94	0.71	23.41	2.71	26.12	24.77
omnetpp	soplex	hmmmer	-	-	3.63	0.84	3.10	0.88	5.72	0.29	2.61	0.50	0.76	6.87	0.75	23.24	2.70	25.94	24.59
soplex	gamess	hmmmer	-	-	3.91	0.94	4.71	1.09	5.71	0.31	2.64	0.50	0.71	7.61	0.61	25.90	2.85	28.75	26.89
gamess	gamess	gamess	gamess	-	5.51	1.26	4.68	1.09	4.94	0.47	6.68	0.71	0.65	9.50	0.48	32.71	3.26	35.97	33.04
soplex	hmmmer	gamess	omnetpp	-	3.81	1.07	4.90	1.03	5.62	0.40	5.03	0.59	0.72	8.67	0.55	29.33	3.06	32.38	29.37

TABLE II. POWER-MAPPING RESULTS FOR 15 TEST CASES. N.B.: NORTH BRIDGE BLOCK; DYN: DYNAMIC POWER; LKG: LEAKAGE POWER; DYN+LKG: THE TOTAL POWER OF THE RECONSTRUCTED POWER MAP; MEAS: THE TOTAL POWER MEASURED THROUGH THE EXTERNAL DIGITAL MULTIMETER.

the average about 11% of the total power. We also report in the last column the total measured power through the external multimeter after compensating for the total leakage difference between the oil-based sink and the Cu-based sink. We notice that our total estimated power through infrared-based mapping achieve very close results with an average absolute error of 0.97 W of the measured power. The differences could be either due to modeling inaccuracies or due to the fact that the measured total power also include the power consumed by the off-chip voltage regulators, and thus, it does not represent the net power consumed by the processor. We have also considered including the total measured power as a constraint to the optimization formulation given in Section II; however, the resultant power maps had some counter-intuitive results.

B. Leakage Modeling Results

To estimate the leakage profile for the AMD quad-core processor, we perform the thermal conditioning techniques described in Section III, where we increase the chip temperature from 27 °C to 55 °C by increasing the infrared transparent cooling fluid temperature from 18 °C to 45 °C, and measuring the associated changes in power consumption and thermal profiles of the chip using infrared imaging. We divide our chip into small blocks of size about 0.4 mm² resulting into approximately 418 first-order and 418 second-order coefficients. In order to maintain the stability of the least square estimation, the maximum number of coefficients i.e. the leakage power resolution is limited by the available number of instances of Equation (7). We collected approximately 2000 data points to solve our least square estimation. The total reference leakage power, $\sum p_{ref}$ in Equation (6) is estimated by changing the die ambient temperature as shown earlier in Figure 2, and using the procedure described in Section III.

To uncover the underlying leakage spatial-variability introduced by process variability, we assume constant temperature across the die, and measure the leakage power for each grid

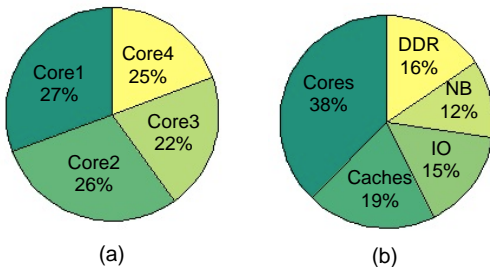


Fig. 6. a) Percentage Leakage power per core with its L2 cache, and b) Percentage Leakage power per block type.

location. Figure 6.a shows the percentage of leakage power for each core with its L2-cache. Core 1 has approximately 5% more leakage than the lowest power core. This result for instance can be used to bias the operating system scheduler to allocate applications on the lower-leakage cores before the higher-leakage cores. Figure 6.b gives the total leakage power distribution among different blocks. There is approximately 10.3% within-die variations among all the blocks.

C. PMC-based Power Modeling

In our third experiment we seek to create empirical models that relate the performance monitoring counters (PMC) to the post-silicon power consumption of each block in the quad-core processor as described in Section IV. We have collected the measurements of 11 PMCs, which cover activities in different components, for our quad-core processor using `pfmon` tool. These 11 PMCs are listed in Figure 7. We computed the correlation coefficient between the measurements of the performance counters and mapped power consumption of each block, and we report in Figure 7 all the PMCs that have strong to good correlation or anti-correlation with power consumption. For example, the number of retired μ ops (PMC #3), the data cache access (PMC #4), the retired branch instructions (PMC #11), the floating point instructions (PMC #2) all provide strong correlation to the power consumption of cores. In case of I/O and DDR channels, the L2 cache misses (PMC #5) provide a strong correlation of power consumption, while PMC #2,

#1	REQUESTS_TO_L2
#2	DISPATCHED_FPU
#3	RETIRED_UOPS
#4	DCACHE_CACHE_ACCESSES
#5	L2_CACHE_MISS
#6	MEMORY_REQUESTS
#7	MEMORY_CONTROLLER_REQUESTS
#8	CPU_TO_DRAM_REQUESTS
#9	DRAM_ACCESSES_PAGE
#10	DISPATCH_STALLS
#11	RETIRED_BRANCH_INSTRUCTIONS

block power	correlates	anti correlates
cores	(3, 4, 11, 2)	-
L2 caches	(10, 1, 11, 2, 7)	-
Northbridge + misc	(3, 4, 11, 1, 9, 8, 7, 10, 2)	-
I/O + DDR channels	5	(2, 11, 4, 3)

Fig. 7. Correlation between performance counters and power consumption of processor blocks.

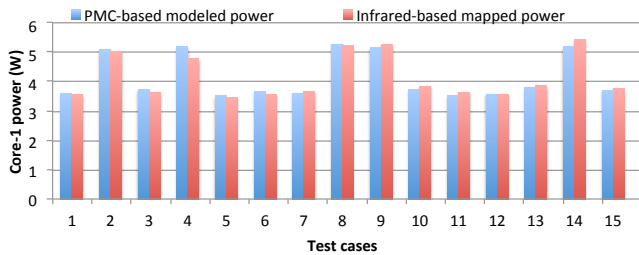


Fig. 8. Power consumption as estimated by the infrared-based system and the fitted models using the performance counters for the 15 test cases.

#11, #3, #4 provide strong anti correlation. Notice that these performance counters are strongly correlated with the power consumption of the caches and cores. That is, when the cores and caches are experiencing high activity, the I/O and DDR channels will experience low activity and vice versa.

Using the PMC measurements and their correlations with the post-silicon power mapping results, we empirically fit a power model for each processor block to its estimated power using least-square estimation as described in Section IV. The inputs to the power models are the most correlated PMCs as described in the previous paragraph. For instance, we report in Figure 8 the power consumption of Core 1 as estimated by infrared mapping and the fitted PMC models. We notice that the PMC-based fitted models for Core 1 track the power mapping results closely, with a mean absolute error of 2.4%.

To illustrate the use of PMC in transient power modeling, we utilize the derived PMC models to estimate the transient power consumption of different blocks of the processor. Figure 9 gives the power consumption of case 14 for the first 120 seconds in execution. We report in blue solid line the sum of power of all cores, the dashed blue line gives the power consumption of the northbridge, while the brown and dashed green lines give the power of IO and L2 caches respectively. Finally, the red line gives the total modeled power and the black line gives the total power from the external multimeter. We note that the PMC-based modeling is able to track the transient response accurately, following the changes in total power consumption.

VI. CONCLUSIONS

In this work, we have introduced multiple novel techniques that advance the state-of-the-art post-silicon power mapping and modeling. We have devised accurate finite-element models that relate power consumption to temperatures, while compensating for the artifacts introduced by using infrared-transparent heat removal techniques. A generic numerical technique is proposed to accurately translate thermal maps from one heat

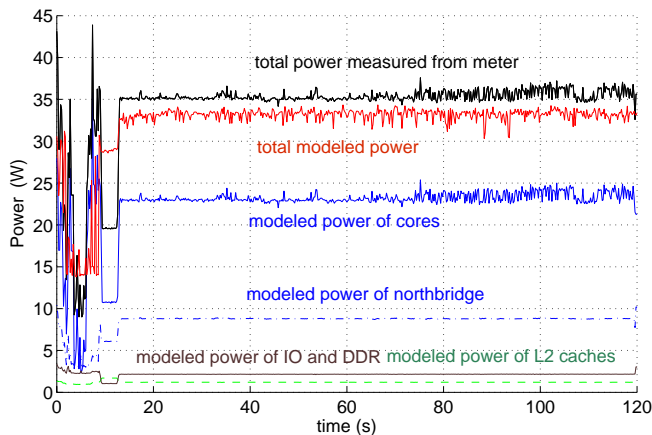


Fig. 9. Transient power modeling using PMC measurements.

sink system to another heat sink system. We have proposed techniques to model leakage power through the use of thermal conditioning. These leakage power models were used to yield fine-resolution leakage power maps and within-die variability trends for multi-core processors. We also devised accurate empirical models that estimate the infrared-based per-block power maps using the PMC measurements. We have used the PMC models to accurately estimate the transient power consumption of different processor blocks. We analyzed the power consumption of different blocks of a quad-core processors under different workload scenarios from the SPEC CPU2006 benchmarks. Our results reveal a number of insights into the make-up and scalability of power consumption in modern processors. As a future work, we are planning to leverage our post-silicon results to improve the accuracy of different design-time thermal and power modeling tools.

Acknowledgments: This work is partially supported by the NSF grant 1115424 and grant 0952866, and a gift from AMD corporation.

REFERENCES

- [1] R. Bertran, M. Gonzalez, X. Martorel, N. Navarro, and E. Ayguade, "Decomposable and Responsive Power Models for Multicore Processors using Performance Counters," in *International Conference on Supercomputing*, 2010, pp. 147–158.
- [2] K. S. M. Bhadauria and S. A. McKee, "Real Time Power Estimation and Thread Scheduling via Performance Counters," in *Proc. Workshop on Design, Architecture, and Simulation of Chip Multi-Processors*, 2008, pp. 46–55.
- [3] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," in *International Symposium on Computer Architecture*, 2000, pp. 83–94.
- [4] R. Cochran, A. Nowroz, and S. Reda, "Post-Silicon Power Characterization Using Thermal Infrared Emissions," in *ACM/IEEE International Symposium on Low Power Electronics and Design*, 2010, pp. 331–336.
- [5] H. Hamann, A. Weger, J. Lacey, Z. Hu, and P. Bose, "Hotspot-Limited Microprocessors: Direct Temperature and Power Distribution Measurements," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, pp. 56–65, 2007.
- [6] W. Huang, K. Skadron, S. Gurumurthi, R. J. Ribando, and M. R. Stan, "Differentiating the Roles of IR Measurement and Simulation for Power and Temperature-Aware Design," in *International Symposium on Performance Analysis of Systems and Software*, 2009, pp. 1–10.
- [7] V. Jimenez *et al.*, "Power and Thermal Characterization of POWER6 System," in *International Conference on Parallel Architectures and Compilation Techniques*, 2010, pp. 7–18.
- [8] T. Kemper, Y. Zhang, Z. Bian, and A. Shakouri, "Ultrafast Temperature Profile Calculation in IC Chips," in *THERMINIC*, 2006, pp. 133–137.
- [9] M. Y. Lim, A. Porterfield, and R. Fowler, "SoftPower: Fine-Grain Power Estimations Using Performance Counters," in *International Symposium on High Performance Distributed Computing*, 2010, pp. 308–311.
- [10] F. J. Mesa-Martinez, E. Ardestani, and J. Renau, "Characterizing Processor Thermal Behavior," in *Architectural Support for Programming Languages and Operating Systems*, 2010, pp. 193–204.
- [11] F. J. Mesa-Martinez, M. Brown, J. Nayfach-Battilana, and J. Renau, "Power Model Validation Through Thermal Measurements," in *International Symposium on Computer Architecture*, 2007, pp. 1–10.
- [12] M. Powell, A. Biswas, J. Emer, and S. Mukherjee, "CAMP: A Technique to Estimate per-Structure Power at Run-Time Using a Few Simple Parameters," *International Symposium on High Performance Computer Architecture*, pp. 289–300, 2009.
- [13] Z. Qi, B. H. Meyer, W. Huang, R. J. Ribando, K. Skadron, and M. R. Stan, "Temperature-to-Power Mapping," in *International Conference on Computer Design*, 2010, pp. 384–389.
- [14] S. Reda, R. Cochran, and A. N. Nowroz, "Improved Thermal Tracking for Processors Using Hard and Soft Sensor Allocation Techniques," *IEEE Transactions on Computers*, vol. 60, no. 6, pp. 841–851, 2011.