

Accurate Temperature Estimation for Efficient Thermal Management

Shervin Sharifi , ChunChen Liu, Tajana Simunic Rosing
Computer Science and Engineering Department, University of California, San Diego
{shervin, chl084, tajana}@ucsd.edu

Abstract

In this work we present a method for accurate estimation of temperature at various locations on a chip considering the inaccuracies in thermal sensor readings due to limitations mainly due to thermal sensor placement and sensor noise. This technique enables accurate estimation of temperature at different locations on the chip with only a limited number of sensors in an efficient way. We utilize Kalman filter (KF) for temperature estimation and for elimination of sensing inaccuracies as well. The computational complexity is reduced by using steady state Kalman filter during normal operation of the chip and reducing the order of the thermal model by a projection based model order reduction method. Our experimental results show that this technique typically reduces the standard deviation and maximum value of temperature estimation errors by about an order of magnitude.

1. Introduction

High temperatures caused by ever increasing integration density and power consumption of VLSI circuits have caused a number of issues such as slower devices, longer delays, complicated timing and noise analysis, increased leakage power and degraded reliability [1]. According to [2], more than 50% of all integrated circuit failures are related to thermal issues. Temperature on the chip can be controlled using various off-chip, on-chip, static or dynamic techniques [2]. One of the most important aspects of dynamic thermal management is obtaining accurate temperature information and capturing the variations in the temperature caused by power consumption variations due to runtime workload changes. Many runtime DTM techniques require accurate real time temperature information [3]. The accuracy of the thermal measurements directly affects the performance of the thermal management and the performance of the CPU [6]. Temperature estimations lower than the actual temperature can result in late activation of DTM which may result in higher packaging cost or reliability degradation. Estimations higher than actual temperature can result in early activation of DTM which degrades performance.

Dynamic thermal management techniques usually rely on the temperature obtained from on-chip thermal sensors or on techniques which estimate the temperature based on the power consumption of functional units. The problems

associated with the open loop temperature estimation techniques and also inaccuracies in the temperature sensors call for efficient techniques which can provide accurate temperature information even in the presence of significant noise in thermal sensor readings. The temperature sensing inaccuracies are caused by variety of factors including sensor placement, process variation, degradation of sensors, power variations, etc. Thermal sensors are often placed at locations other than the location of interest since hot spot areas on the die are usually also areas where silicon real estate is at premium. Thus, there can be a significant disparity between sensor readings and the actual temperature at the location of interest [4]. According to [4], there can be temperature differences about 10°C between the sensor and the hotspot. The technique proposed in this paper provides an efficient way for accurately estimating temperatures at different locations on the die using imprecise readings provided by a limited number of sensors available on the chip. The experimental results show significant reductions in temperature estimation errors with very minor typical overhead of a few hundreds of microseconds for reduced order models. The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 explains the details of the technique. Section 4 demonstrates the experimental results and Section 5 concludes the paper.

2. Related Work

DTM techniques need accurate and efficient temperature measurements at runtime. Different thermal modeling and simulation methods have been proposed for various levels of abstraction [2]. One of the most widely used models for temperature estimation at micro-architectural level is HotSpot [6], which is based on building a multi-layer thermal RC network of the given chip. Conventional integration-based transient simulation is performed in [7] to calculate the temperature at each execution interval. Accurate temperature estimation techniques like HotSpot [6] usually suffer from the high computation cost which typically cannot be afforded at runtime. In [8], the authors exploit periodic behavior of programs and the resulting periodicity in temperature of micro-architectural modules to speed up transient thermal simulation using spectrum analysis. Another technique proposed in [8] performs runtime thermal simulation based on the observation that the average power consumption of architecture level modules in microprocessors is the major contributor to the variations in the temperature. Therefore

piecewise constant average power inputs can be used to speed up the thermal analysis. Techniques such as those introduced in [8] need to continually perform temperature estimation, thus cause significant overhead. In addition, since these on-chip temperature estimation techniques are not complemented by thermal sensor measurements, the estimates can easily deviate from the actual temperature values. Due to the shortcomings of temperature estimation techniques, many DTM techniques utilize real time temperature measurements obtained by thermal sensors [3]. One of the major problems in direct use of on-chip temperature sensors is the sensor imprecision and noise [4]. There are several factors that cause inaccuracy in temperature measurement. Placement of thermal sensors is limited by a number of factors such as routing and I/O considerations. Sensor placement methods like [17] try to assign optimal locations to the sensors such that different hotspots are covered by thermal sensors, they can not resolve the problem of temperature difference between the hotspot and sensor completely, especially when the number of sensors is limited, which is usually the case. In addition, hotspots tend to move around the chip as a function of the workload. Thermal diodes are usually used as temperature sensors and the voltage reading over the diode must be calibrated to get the actual temperature. This calibration is usually done at test time [4]. Even a small variation in the offset can lead to large errors in temperature measurement. Variations in process parameters introduced during manufacturing result in sensor reading inaccuracies as well (e.g. threshold voltage variation on the die) [4]. Errors are introduced in the process of analog to digital conversion due to quantization, and limitations of design and technology. Changes in power supply voltage can also affect sensor readings. Finally, sensor accuracy is affected by its reliability degradation. While the average sensor error may be low, the sensor might deliver single readings with high deviations from the actual temperature. If the readings from the sensors are directly used with complete trust, such variations may cause several serious problems.

The technique presented in this paper estimates at run time the temperature at inaccessible locations of the die with very minimal overhead in the presence of sensor noises. Thus, it can be used at runtime for temperature aware scheduling and other online thermal management techniques. It can be activated as needed rather than continuously; for example when the temperature at a unit on the chip is approaching a threshold. This way it provides an easy trade off between accuracy and overhead. Finally, it can adapt to the changes in the measurement noise characteristics, which is very important since mean time to failure of thermal sensors is shorter than that of assets they are supposed to protect, therefore the characteristics of sensor inaccuracies may change during the lifetime of the chip [5]. To the best of our knowledge, it is the first proposed technique for estimation of temperature at inaccessible points of the chip based on the temperature readings at available sensors. We next outline our temperature estimation technique, followed by results and conclusions.

3. Accurate Temperature Estimation

Our technique accurately estimates the temperature at various locations on the die by using temperature measurements obtained from a few on-chip sensors. Issues related to sensor placement and dynamic change of hotspots can be addressed by our method. To use our method, we first need to complete a sequence of off-line setup steps shown in Figure 1.a, followed by run-time implementation showed in Figure 1.b. The setup phase (Figure 1.a.) starts by creation of chip's equivalent thermal RC network using models described in [6] [7]. The linear dynamic system generated in this way is usually too large and complex for an on chip software implementation. Therefore, model order reduction is used to reduce the size of the model and generate a much smaller yet accurate linear system. Kalman filtering estimates the temperature at different locations on the chip based on the inaccurate temperature readings at sensor locations and inaccurate power consumption estimates. Thus, in the next step, the calibration is performed where KF is applied to the reduced order model of the system. The calibration ends when the KF reaches its steady state.

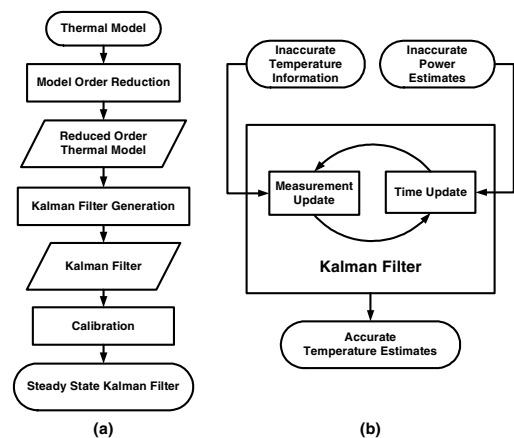


Figure 1. The proposed technique
(a) Sequence of steps in off-line setup
(b) Temperature estimation by Kalman filter

After this, the resulting steady state KF is used during the normal operation to actually perform the temperature estimation (Figure 1.b). The KF estimates the temperature in a predict-correct manner based on inaccurate information of temperature and power consumption. Time update equations project forward in time the current temperatures and the error covariance estimates to obtain a priori estimates for the measurement step. The measurement update equations incorporate the new measurements into the a priori estimate to obtain an improved a posteriori estimate of the temperature.

3.1 Temperature Estimation

Temperature values at different locations on the chip depend on factors such as power consumptions of functional units, layout of the chip and the package characteristics.

Analysis and estimation of temperature requires a thermal model which represents the relation between these factors and the resulting temperature. The differential equations describing the heat flow have a form dual to that of electrical current. This duality is the basis for the microarchitectural thermal model proposed in [6] and is further explained in [7] and [1]. The lumped values of thermal R and Cs represent the heat flow among units and from each unit to the thermal package.

We model the temperature at grid cell level [8] which enables more accurate and fine grained temperature estimates. An analytical method is proposed in [1] to determine the proper size of a grid cell. The thermal network is represented in state space form with the grid cell temperatures as states and the power consumption as inputs to this system. The outputs of this state space model are the temperatures at the sensor locations which can be observed by sensor readings. We define C_i and G_i as thermal capacitance and thermal conductance matrices, D as the input selection matrix which identifies the effect of power consumptions at current time steps on the temperature at next time step and F as the output matrix which identifies the sensor grid cells at which temperatures are observable. u is the vector of power consumption values at power consuming components on the die and T is the vector of temperature values at different grid cells. The units for temperature and power are centigrade degree and Watt. The system can be represented as:

$$\begin{aligned} \frac{dT}{dt}(t) &= -C_i^{-1}G_i T(t) + C_i^{-1}Du(t) \\ S(t) &= FT(t) \end{aligned} \quad (1)$$

Since sensor measurements can be inaccurate and exact power consumptions of the functional units at run time are not available, we use Kalman filter (KF) for temperature estimation. KF uses a form of feedback control to estimate a process in a predict-correct manner with time and measurement update phases. Time update equations project forward in time the current state of the system and the error covariance estimates to obtain a priori estimates for the measurement step. The measurement update equations incorporate the new measurements into the a priori estimate to obtain an improved a posteriori estimate.

We use Kalman filtering to both estimate the temperature and to filter out any thermal sensor noise. In order to apply the KF to our model, we convert the continuous time differential equations in (1) to corresponding discrete time equations in (2). Here H , J and F are the state matrix, input matrix and output matrix of the system respectively. Furthermore, at time n , $T[n]$, $u[n]$ and $S[n]$ are the state vector representing temperatures at different grid cells, input vector of functional block power consumption and output vector of temperatures at sensor locations respectively.

$$\begin{aligned} T[n+1] &= HT[n] + Ju[n] \\ S[n] &= FT[n] \end{aligned} \quad (2)$$

Applying KF requires knowledge of the power consumption at different functional units. Accurate estimation of power consumptions of each component at each

time step is not practical in runtime. On the other hand, [8] shows that most of the energy in the power traces is concentrated in the DC component. In other words, the trend of temperature variations is determined by the average power in a certain amount of time. This is especially true for power traces with very large DC components and smaller high frequency harmonics [8]. Based on this fact, we use the average power consumption of each component as an estimation of the actual power consumption at that time.

Introduction of noise due to inaccuracies of the modeling the process, $w[n]$, and the measurement noise, $v[n]$, enables us to rewrite the system formulation as:

$$\begin{aligned} T[n+1] &= HT[n] + Ju[n] + Gw[n] \\ S_v[n] &= FT[n] + v[n] \end{aligned} \quad (3)$$

Let $\tilde{T}[n|n-1]$ represent the estimate of $T[n]$ given the past measurements up to $S_v[n-1]$. Also let $\hat{T}[n|n]$ represent the updated estimate based on the last measurement $S_v[n]$. Where P is the error covariance matrix, the time-update equations for our system would be:

$$\begin{aligned} \tilde{T}[n+1|n] &= H\tilde{T}[n|n] + Ju[n] \\ P[n+1|n] &= HP[n|n]H^T + GQ[n]G^T \end{aligned} \quad (4)$$

Given the current estimate $\tilde{T}[n|n]$, the time update predicts the state value at the next sample $n+1$ (one step ahead). Then the measurement update adjusts this prediction based on the new measurement $S_v[n+1]$. The measurement update equations for this system are:

$$\begin{aligned} \hat{T}[n|n] &= \tilde{T}[n|n-1] + M[n](S_v[n] - F\tilde{T}[n|n-1]) \\ M[n] &= P[n|n-1]F^T(R[n] + FP[n|n-1]F^T)^{-1} \\ P[n|n] &= (I - M[n]F)P[n|n-1] \end{aligned} \quad (5)$$

M is called Kalman gain or innovation gain. It is chosen to minimize the steady state covariance of the estimation error given the noise covariance $Q = E(w[n]w[n]^T)$ and $R = E(v[n]v[n]^T)$.

3.2 Reducing Computational Complexity

Considering k as the size of the dynamic model, computational complexity of the KF is $O(k^3)$ due to the matrix inversion in calculating Kalman gain $M[n]$. This causes high computational overhead when the size of the model gets larger. We introduce two techniques that significantly reduce the computational complexity of the model. One of these techniques reduces the size of the model used in KF, while the other reduces the number of computations required for KF.

3.2.1 Steady State Kalman Filtering

The time scales at which the sensor noise characteristics change are much larger than the time scale at which we study the system (months or years compared to seconds). Thus we assume the system and noise covariances are time-invariant. As a result, we can use steady state KF in which it is not necessary to compute the estimation error covariance or Kalman gain in real time [9]. The steady state KF reduces the computational overhead from $O(k^3)$ to $O(k^2)$ while still providing good estimation performance [9]. A calibration step is needed prior to run-time operation in order to get the

KF to steady state. In our experiments, we show that use of steady state KF reduces the computational complexity to several orders of magnitude without significant effect on accuracy (Figure 4).

3.2.2 Model Order Reduction

The model order reduction enables us to find a low-dimensional but accurate approximation of the thermal network which preserves the input-output behavior to a desired extent. We use a projection based implicit moment matching method (PRIMA) [11] which is used to find a mapping from the high-dimensional space of the given state-space model to a lower dimensional space. Krylov subspace vectors are used instead of moments. For a square matrix of dimension N and a vector b , the subspace spanned by the vectors $[b, Ab, \dots, A^{q-1}b]$ is called a Krylov subspace of dimension m generated by $\{A, b\}$ and is denoted by $Kr(A, b, q)$. With thermal capacitance and conductance matrices represented by C_i and G_i respectively, the circuit formulation shown in equation (1) can be represented in this form:

$$sC_r T = -G_r T + Du \quad (6)$$

The reduced order model is generated using congruence transformation, where $C_r = V_q^T C_i V_q$, $G_r = V_q^T G_i V_q$, $D_r = V_q^T D$, $X_r = V_q^T X$:

$$sC_r T = -G_r T + D_r u \quad (7)$$

The projection matrix $V_q = \{V_1, V_2, \dots, V_q\}$ is obtained by Arnoldi process such that

$$\begin{aligned} \text{Span}\{V_1, V_2, \dots, V_q\} &= Kr\{-G_i^{-1}C_i, DU\} = \\ \text{Span}\{DU, -G_i^{-1}C_i DU, \dots, (-G_i^{-1}C_i)^{q-1}DU\} \end{aligned} \quad (8)$$

and

$$\begin{aligned} v_i^T v_j &= 0 \quad \text{for all } i \neq j \\ v_i^T v_i &= 1 \quad \text{for all } i \end{aligned} \quad (9)$$

This approach matches moments up to order q . The larger the number of matched moments, the closer is the behavior of the reduced order model to the original system, but at the cost of higher processing time.

Because of the moment-matching properties of Krylov-subspaces, the reduced transfer function will agree with the original up to the first q derivatives on an expansion around some chosen point in the complex plane (usually $s=0$). In addition, due to the congruence transformation, the reduced model inherits the structure of the original model (7), which means the passivity is preserved.

There are some other model order reduction techniques which are designed for linear circuits with multiple sources. For example, EKS [12] can match higher moments compared with PRIMA in MIMO system, but its limitations on the inputs makes it inappropriate for our case. In our case, as experimental results show, PRIMA works well. The reason is that our network consists of only R and Cs. Moreover, the topology of the network is such that it operates as a low pass filter which eliminates the high frequencies in the inputs. Therefore PRIMA with a few number of moments around frequency $s=0$ provides us with enough accuracy and acceptable overhead compared to RHS-model order reduction

methods like EKS [12]. Although methods like EKS can be more accurate, they impose some limitations on the inputs and more importantly incur more computational overhead. The effectiveness of PRIMA is shown in Table II, where matching a small number of moments provide enough accuracy for our application.

The next section shows how temperature estimation accuracy can be traded off with computational complexity.

4. Experimental Results

We use a multi-processor SoC comprised of 6 XScale cores [13] to evaluate with MiBench Ver 1.0 [14] benchmarks for the evaluation of our technique. MiBench has software from the automotive/industrial, network and telecommunications segments [15]. Idle times between task arrivals are modeled using Pareto distribution [16] with a timeout-based dynamic power management policy. XScale power values are used to estimate each core's power consumption [13]. Parameters used for package are: convection capacitance 140.4 J/K, convection resistance 0.1 K/W, spreader thickness 10^{-3} m, and initial temperature of 333°K.

Our method consists of both off and on line parts. The off-line implementation is done in Matlab, while the run-time part is implemented in C++ which runs on XScales. In our experiments, the temperature values are obtained from grid mode HotSpot [10]. The chip is divided into a grid, as shown in Figure 2. The temperature values of the grid cell containing the sensors are observable, while the temperature at other grid cells are assumed to not be observable and must be estimated using our technique. We used a 18x12 grid for our experiments, but one of the advantages of using PRIMA model order reduction technique is that the size of reduced model only depends on the number of power sources and the number of matched moments, not the number of grid cells. Therefore, increasing the granularity of the grid in order to increase the accuracy does not result in higher computational overhead. An analytical method is presented in [1] to find the appropriate number and size of the grid cells. No specific sensor technology is assumed in this work. The readings from the temperature sensors are used as starting temperature values. Figure 3 shows an example of temperature estimation by our method.

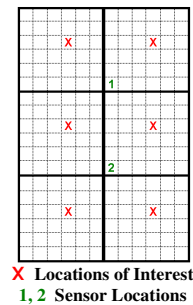
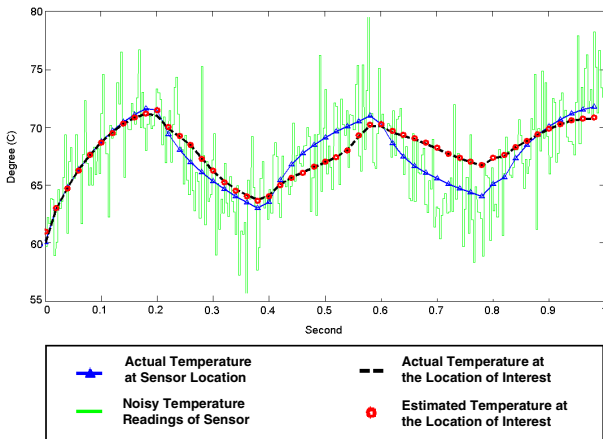


Figure 2. Grid cells, locations of interest and sensor locations

Table 1 Error statistics for limited number of sensors

Number of Sensors	Sensor Measurement Errors (°C)		Temperature Estimation Error (°C)	
	Mean Absolute Error	Std. Dev.	Mean Absolute Error	Std. Dev.
2	3.74	4.72	0.77	1.28
3	3.72	4.60	0.76	1.27
4	4.41	5.50	0.75	1.27
5	3.29	3.94	0.76	1.27

Actual temperatures at the sensor locations and locations of interest are the obtained from the grid mode HotSpot. Temperature readings at the sensor locations are generated by superimposing the noise on the actual temperature values at the sensor locations. To model the inaccuracies observed at thermal sensors, different Gaussian noise and bias values are superimposed on the actual temperature values. Processes generating these noises are assumed to be stationary between different calibrations. As Figure 3 shows, the estimated temperature closely follows the actual temperature at the location of interest. Accurate estimates of the temperature prevent early or late activation of DTM techniques due to significant sensor error and noise. In order to evaluate the ability of our estimation technique, we examine the mean absolute error and the standard deviation of the error as the location of interest. These values are averaged over all the locations of interest in the MPSoC. First we evaluate the ability of the proposed technique to estimate the temperature at different locations on the chip using a few number of sensors. We estimate the temperature at 6 locations of interest using only 2, 3, 4 and 5 sensors. The configuration for 2 sensors is shown in Figure 2. Each sensor is placed such that it is at about the same distance from the hotspots it must cover. The step size is 100ms and 3 moments are matched. As can be seen in this table, the technique is able to estimate the temperature at the locations far away from the limited number of sensors. In order to evaluate the other effective parameters, we assume that we have as many sensors as locations of interest, but not near the location of interest.

**Figure 3 Comparison of temperature read by sensors, actual temperatures and estimated temperature****Table 2 Error statistics for different time steps**

Step Size (10 ⁻⁴ s)	Sensor Measurement Errors (°C)		Temperature Estimation Error (°C)	
	Mean Absolute Error	Std. Dev.	Mean Absolute Error	Std. Dev.
32	3.00	4.16	0.06	0.08
128	2.91	4.04	0.13	0.25
512	3.09	4.33	0.49	0.89
1028	4.29	4.27	0.97	1.24
2056	3.89	5.08	1.74	1.39

To show that the method can work for estimating the temperature at not only the hotspots, but at any location on the chip, we chose arbitrary locations for the sensors.

An important parameter which affects both the accuracy and the computational requirements of our technique is the time step at which temperature sensors are read and KF is applied. Table 2 shows the mean absolute error and standard deviation for measurement and estimation errors for different sizes of time steps. The basic time step is chosen at 10⁻⁴s and multiplied by some powers of 2.

Table 3 shows how matching different number of moments affects the accuracy of our technique. For this table, the step size is selected 50ms. Size of the model for matching 1, 2, 3 and 4 moments are 6, 12, 18 and 24 respectively. The associated overhead can be found in Table 3. As it is shown in the table, after matching the second moment, addition of more moments does not significantly improve accuracy. Therefore, the model can be reduced by matching 2 moments with minimum performance loss. Figure 4 presents the runtime overhead associated with KF. This figure allows comparison between the overhead of general KF and steady state KF. In the calibration process, i.e. before the KF reaches its steady state, the general KF is used. But since the calibration can be performed offline, its overhead does not affect the performance. The number of calibration steps depends on the thermal network and noise characteristics, which in our case is less than 100. As can be seen in this figure, the performance overhead of the steady state KF is significantly lower, and this difference gets larger for bigger models. An important point is that due to the lack of the floating point hardware on XScale, floating point instructions are emulated by the software which takes significant time. In the processors with floating point units, the overheads would be much lower.

Table 3 also accentuates the importance of model order reduction in reducing the overhead of this technique.

Table 3. Effect of number of matched moments

Sensor Measurement Errors (°C)	No. of Matched Moments	Temperature Estimation Error (°C)	
		Mean Absolute Error	Std. Dev.
-	-	3.38	4.82
Estimation Error	1	0.88	1.24
	2	0.75	1.05
	3	0.48	0.90
	4	0.68	0.88

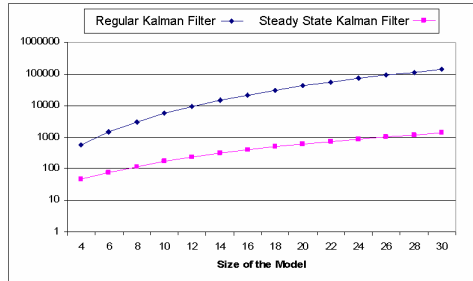


Figure 4. Runtime overheads for KF (milliseconds)

One of the advantages of the model order reduction technique used here is that the size of the generated model does not depend on the granularity of the grid, but is a product of the number of power consuming units and the number of matched moments. In our example, since there are 6 power consuming functional units, matching 4, 3, 2 and 1 moments reduces the model size to 24, 18, 12 and 6 respectively. Our simulations show that even matching 2 or 3 moments provides enough accuracy for most applications. As shown in the results, this method can be used in order to accurately estimate the temperature when higher accuracy is desired, but the number of sensors and the locations they could be placed are limited.

5. Conclusion

This paper proposes a method for accurate software estimation of temperature at different locations on the chip based on the inaccurate values obtained from a few on-chip temperature sensors. One important advantage of our method is that it can be activated only when the temperature is approaching a limit to provide more accurate estimates. KF is used for state estimation and to eliminate the noise of on-chip temperature sensors. In order to reduce the complexity, a model order reduction technique is applied. In order to further improve the efficiency, steady state KF is used that is calibrated off-line. The experimental results show that the mean absolute error and the standard deviation of the error are minimized significantly as compared to information obtained from direct reading of sensors. Lower standard deviation translates to more reliability for the temperatures obtained by this technique. Our experimental results also show significant reductions in the maximum error value which prevents false negative and positive alarms. Most importantly, this technique can be used efficiently in order to estimate the temperatures at the locations of interest where no sensor is available. Our temperature estimation technique has very low run-time overhead in orders of hundreds of microseconds which is needed in OS level schedulers that run in millisecond time scales.

6. References

- [1] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, and S. Ghosh. "HotSpot: A Compact Thermal Modeling Method for CMOS VLSI Systems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. VOL. 14, NO.5, May 2006, pp.501-513.
- [2] M. Pedram and S. Nazarian, "Thermal modeling, analysis, and management in VLSI circuits: Principles and Methods," Proc. of IEEE, Special Issue on Thermal Analysis of ULSI, Vol. 94, No. 8, Aug. 2006, pp. 1487-1501.
- [3] A. K. Coskun and T. S. Rosing and Keith Whisnant. "Temperature Aware Task Scheduling in MPSoCs," In Proc. Design Autom. and Test in Europe (DATE) , Apr. 2007, pp. 1659-1664.
- [4] E. Rotem, J. Hermerding, A. Cohen, H. Cain, "Temperature measurement in the Intel® Core™ Duo Processor," In Proc. of Int. Workshop on Thermal investigations of ICs, 2006. pp. 23-27.
- [5] K. Whisnant, K. C. Gross and N. Lingurovska, "Proactive Fault Monitoring in Enterprise Servers," In Proc. of IEEE Int'l Multiconference in Computer Science & Computer Eng., Jun. 2005 pp. 3-10.
- [6] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature aware microarchitecture," in Proc. of the Int. Symp. on Computer Architecture, Jun. 2003, pp. 2-13.
- [7] W. Huang, M. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," In Proc. Design Automation Conference, Jun. 2004, pp. 878-883.
- [8] P. Liu, H. Li, , L. Jin, W. Wu, S. Tan, J. Yang, "Fast Thermal Simulation for Runtime Temperature Tracking and Management." IEEE Trans. Computer Aided Design of Integr. Circuits (CAD), VOL. 25, NO. 12, DECEMBER 2006, pp. 130-136.
- [9] D. Simon, Optimal State Estimation: Kalman, H ∞ , and Nonlinear Approaches, Wiley, 2006.
- [10] HotSpot, <http://lava.cs.virginia.edu/HotSpot/>
- [11] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced order interconnect macro-modeling algorithm," IEEE Trans. on CAD, VOL. 17, NO.8, Aug. 1998, pp. 645-654.
- [12] J.M.Wang, T.Nuyen, "Extended Krylov Subspace Method for Reduced Order Analysis of Linear Circuits with Multiple Sources," Proc. Design Automation Conference, June 2000, pp. 247-252
- [13] Intel pxa270 processor, electrical, mechanical and thermal specification data sheet. <http://www.intel.com>.
- [14] M. R. Guthaus, J. S. Ringenberg, D. Ernst, Todd M. Austin, T. Mudge, R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite." In Proc. IEEE Annual Workshop on Workload Characterization, Dec. 2001.
- [15] G. Fursin, J. Cavazos, M. O'Boyle and O. Temam. "MiDataSets: Creating The Conditions For A More Realistic Evaluation of Iterative Optimization." In Proc. of the Int. Conf. High Performance Embedded Architectures & Compilers (HiPEAC 2007), Jan. 2007.
- [16] T. Simunic, L. Benini, P. Glynn, and G. De Micheli, "Event-Driven Power Management," IEEE Trans. Computer-Aided Design, VOL. 20, NO. 7, July 2001, pp. 840-857.
- [17] R. Mukherjee, S. O. Memik, "Systematic temperature sensor allocation and placement for microprocessors," Proc. Design Automation Conference, Jul. 2006, pp. 542 - 547.