An Analytical Model for the Upper Bound on Temperature Differences on a Chip

Shervin Sharifi Tajana Simunic Rosing
Computer Science and Engineering Department
University of California San Diego (UCSD)
9500 Gilman Dr.
La Jolla CA 92093-0404
{shervin, tajana}@ucsd.edu

ABSTRACT
The main contribution of this work is an analytical model for finding the upper bound on the temperature difference among various locations on the die. The proposed model can be used in many applications, such as estimation of maximum temperature variations on the die and estimating the maximum placement error in temperature sensor placement algorithms. The model also identifies the conditions under which these maximum temperature variations might happen, which is very helpful for generating test data for thermal stress tests and for augmenting different benchmarks. Experiments show that maximum temperature differences can be underestimated as much as 9ºC. Based on this model, a temperature sensor placement algorithm is also proposed which is able to guaranty a maximum temperature error due to placement of the sensor. The ability of the proposed model to estimate point to point maximum temperature difference can improve the efficiency and accuracy of the sensor placement technique so that we can reduce the number of thermal sensors needed by about 16% on average.

Categories and Subject Descriptors
B.8.0 [Performance and Reliability]: General; C.4 Performance of Systems]: Modeling Techniques.

General Terms
Measurement, Performance, Reliability

Keywords
Temperature, Thermal Management, Temperature Variation, Temperature Difference, Sensor Placement

1. INTRODUCTION
High temperatures in new generations of VLSI circuits and embedded systems require that thermal considerations be taken into account during design, manufacturing and test [2]. High temperatures and temperature variations result in several issues such as degrade reliability, slow down devices, increase resistances and leakage power, etc. 8. Spatial and temporal temperature variations happen as a result of differences between functionality and structural differences and computational activities across the chip and workload variations during the time. Temperature variations as high as 50ºC across the die in a modern microprocessor are reported in [2].

Temporal and spatial temperature variations may result in performance mismatches, which can in turn lead to performance or functional failures. For example, since resistances scale with temperature, differences between temperatures of two regions of the die cause difference between resistances at those two regions which may result in clock skew problems in clock networks. Moreover, according to [2], more than 50% of all integrated circuit failures are related to thermal issues. A comprehensive framework proposed in [5] analyzes the effects of temperature on reliability of multi-core systems. It is shown in [4] that spatial and temporal temperature gradients determine the device reliability at moderate temperatures; and to achieve satisfactory reliability, resolving the thermal hotspots alone is not adequate. Temperature variations are also important in clock tree design and optimization [19]. Increase in temperature decreases the carrier mobility, and increases the resistance of interconnects. These affect the device performance and increase the delays in the system. Temperature variations on the die can change timing characteristics of the wires and interconnect networks. Therefore, the clock signals which are the most timing sensitive signals are really vulnerable to temperature variations on the die.

Such issues make analysis of the temperature variations and gradients an important issue. To estimate the magnitude of these variations, we need to know the maximum temperature difference between various points on the die. The maximum temperature difference under different workloads can be found by extensive simulations, which would incur significant overhead, while our method provides an upper bound on the temperature differences with practically no overhead. For systems which interact with other systems, even the same workload can result in completely different behavior due to interactions with other systems, which make simulations of a set of benchmarks even less reliable in determining such parameters. In this paper, an analytical model is proposed for finding an upper bound for the temperature difference between various locations of the die. An absolute upper bound is calculated which does not depend on the workloads or interactions between systems.

As an application of this model, we also present a temperature sensor placement technique based on this model. An efficient sensor placement technique can affect the cost of the system and effectiveness of dynamic thermal management techniques. For DTM techniques to be effective, they need to capture the changes in the temperature caused by power consumption variations due to runtime workload dynamics. Many runtime DTM techniques require accurate real time temperature information [6]. Lower or

GLSVLSI'08, May 4–6, 2008, Orlando, Florida, USA.

higher temperature estimations than the actual temperature may cause late or early activation of the thermal management techniques which can result in degraded reliability or degraded performance [7]. One of the important causes of inaccurate temperature measurement is the sensor placement error. Thermal sensors are often placed at locations other than the hotspots or other locations of interest since these areas on the die are usually also areas where silicon real estate is at premium. Thus, there can be a disparity between sensor readings and the actual temperature at the location of interest [7]. Adding more sensors on the die can resolve this problem to some extent at significant hardware overhead. An efficient sensor placement algorithm can increase the accuracy of temperature monitoring while reducing the hardware overhead. Our sensor placement technique uses our model to find the maximum temperature difference between the point of interest and potential sensor locations. This is used to limit the difference between the point of interest and the sensor location and guaranty the desired accuracy requirements.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 explains the details of the model. Section 4 explains the sensor placement technique based on this model. Section 5 demonstrates the experimental results and Section 6 concludes the paper.

2. RELATED WORK

Due to the reliability, performance and cost issues caused by high temperatures and temperature variations, accurate estimation of temperature is becoming increasingly important. Usually extensive simulations are performed in order to estimate the maximum temperature variations. To the best of our knowledge, no model has been proposed to estimate the temperature difference between various locations on the die. The technique proposed in [17] is a special case of this problem and proposes a model for estimating the temperature at distance *d* from a hotspot. It estimates the maximum temperature differential between a hotspot and another location based on their distance and processor packaging information. This model is based on the assumption that the temperature of the hotspot decays exponentially with the distance from a hotspot. For a given maximum temperature error, a maximum distance from the hotspot is calculated. The points within this distance from the hotspot will have a temperature difference to the hotspot within the desired accuracy. Selection of activity factor parameter is not easy and also depends on the application. Therefore the results will not be exact and a pessimistic selection of the distance must be used to guaranty the maximum error. Moreover, when calculating the maximum temperature difference to the hotspot, the result depends only on the distance from the hotspot. In other words, in this model, for all of the points at equal distance from the hotspot, same maximum temperature difference to the hotspot is assumed, while this is not correct, i.e. for the points at the same distance from hotspot, the maximum temperature difference to the hotspot could be very different. This could be due to the temperature effect of other power sources on the temperature around the hotspot. Figure 1 shows the contour map of maximum temperature difference relative to a point of interest in a multi-processor SoC which is used in our experiments and consists of 6 *XScale*® cores [11]. It is clearly shown that the maximum temperature differences around the region of interest are not symmetric. Even when there is only one power source, the aforementioned assumption could be wrong, e.g. due to the location of this power source on the chip.

Several techniques have been proposed for on-chip placement of thermal sensors. These techniques are usually based on the identification of the hotspots on the die and placing the sensors such that they appropriately cover these hotspots. [16] introduces a systematic technique for thermal sensor allocation and placement in microprocessors. This technique identifies an optimal physical location for each sensor such that the sensor's attraction towards steep thermal gradient is maximized. The problem with this approach is that it does not consider the accuracy of the sensors and does not guaranty a maximum error in the thermal sensor readings. The technique proposed in [18] determines the number and positions of the sensors required for thermal monitoring on a FPGA for an arbitrary design that uses distributed fine-grain reconfigurable logic. This technique relies on the model introduced in [17] to calculate the range of the hotspot which is the maximum distance from the hotspot that a sensor can be placed while still maintaining the intended accuracy. We propose the concept of observability area instead of the range. The observability area is the area around a point of interest in which the maximum temperature difference to that point is always within maximum tolerable error. Due to the location of power sources and the effect of neighbor power sources, the observability area of a point of interest is not usually circular, therefore in these cases considering it as a circle – as done by some previous methods – may be incorrect.

The next section describes the details of our model.

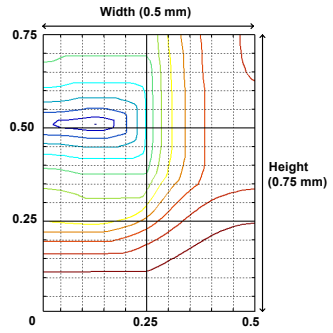

Figure 1 Contour map of maximum temperature difference relative to a point of interest

3. DESCRIPTION OF THE MODEL

Calculating the upper bound of the maximum temperature difference between various locations on the die is an important step during the design process as it enables better placement of temperature sensors and helps with evaluation of potential reliability issues. Our algorithm starts with the evaluation of the effect each power source has on the temperature variations. This can be done by simulation or by using analytical methods. Following this step, the maximum temperature difference between pairs of locations is calculated by exploiting the LTI characteristics of the system.

Temperatures at different locations on the die depend on several factors such as the power consumptions of the functional units, the layout of the chip and the characteristics of the materials used in the chip. The differential equations used to describe the heat flow have a form similar to that of electrical current. This duality is the basis for the microarchitectural-level thermal model proposed in [7] and further explained in [8], [9] and 8. The thermal network generated by the model includes thermal resistors and capacitors. Temperature can be modeled at the level of a

functional block, or the die can be divided into regular grid cells – as shown in Figure 1- to obtain more fine grained estimates. Each grid cell has its own corresponding node in the thermal network whose voltage represents the cell's temperature, while the power consumptions of different components are applied as current sources to the thermal network. The lumped values of thermal Rs and Cs can be computed to represent the heat flow between the units and from each unit to the thermal package. The method for calculating the equivalent thermal R and Cs is described in [9]. Given the layout and the thermal characteristics of a chip, we divide it into a grid of *r* rows and *c* columns as shown in Figure 1. The proper size of the grid cells and the number of rows and columns of the grid can be determined by the method proposed in 8. Since the thermal resistors and capacitors are linear components, this thermal network can be considered a linear time-invariant dynamic system. We exploit the LTI characteristics of this system as a basis for calculating an upper bound for the temperature difference. First, we explain the idea using a simple case of a single input and single output system, and then extend to the thermal networks with multiple inputs and outputs.

Representing the power input to the thermal circuit as $p(t)$, the temperature output of the system $f(t)$ can be represented as:

$$f(t) = h(t) * p(t) = \int_0^t h(\tau)\, p(t-\tau)\, d\tau \qquad (1)$$

where $h(t)$ is the impulse response of the system. The maximum and minimum power consumed at each functional unit (p^{Max} and p^{Min} respectively) are known and we also know that the power consumed at a functional unit is always non-negative:

$$0 \le p^{Max} \le p(t-\tau) \le p^{Min} \qquad (2)$$

Let's suppose $H+$ and $H-$ are respectively the sets of intervals where the impulse response $h(t)$ takes non-negative and negative values. Therefore, the equation (1) can be rewritten as:

$$f(t) = \int_0^t h(\tau)\, p(t-\tau)\, d\tau = \int_{H+} |h(\tau)|\, p(t-\tau)\, d\tau - \int_{H-} |h(\tau)|\, p(t-\tau)\, d\tau \qquad (3)$$

Based on (2), we can write:

$$p^{Min} \int_{H+} |h(\tau)|\, d\tau \le \int_{H+} |h(\tau)|\, p(t-\tau) d\tau \le p^{Max} \int_{H+} |h(\tau)|\, d\tau$$
$$p^{Min} \int_{H-} |h(\tau)|\, d\tau \le \int_{H-} |h(\tau)|\, p(t-\tau) d\tau \le p^{Max} \int_{H-} |h(\tau)|\, d\tau \qquad (4)$$

This results in:

$$f^{Min} = p^{Min} \int_{H+} |h(\tau)|\, d\tau - p^{Max} \int_{H-} |h(\tau)|\, d\tau = p^{Min} \int_{H+} h(\tau)\, d\tau + p^{Max} \int_{H-} h(\tau)\, d\tau$$
$$f^{Max} = p^{Max} \int_{H+} |h(\tau)|\, d\tau - p^{Min} \int_{H-} |h(\tau)|\, d\tau = p^{Max} \int_{H+} h(\tau)\, d\tau + p^{Min} \int_{H-} h(\tau)\, d\tau \qquad (5)$$

Equation (5) provides an upper bound on the value of the output f of a single input single output system based on its impulse response. Suppose a system with m power sources represented as $p_1(t), \ldots, p_m(t)$. p_i^{Max}, p_i^{Min} are maximum and minimum input values (min and max power consumptions of the corresponding functional unit). Considering a single output, and $h_i(t)$ as the response of the output to the impulse on input i, the LTI characteristics of the system imply:

$$f(t) = \sum_{i=1}^{m} f_i(t) = \sum_{i=1}^{m} h_i(t) * p_i(t) \qquad (6)$$

Therefore, for the minimum and maximum values of the function, we will have:

$$f^{Min} = \sum_{i=1}^{m} f_i^{Min}$$
$$f^{Max} = \sum_{i=1}^{m} f_i^{Max} \qquad (7)$$

This can be extended to the systems with multiple outputs as well. Equation (7) holds for each output of the system.

Our model for calculating the upper bound for temperature difference between two different points on the chip is based on equations (5) and (7). The function f is defined to be the difference between temperatures of the two different locations of interest on the die. The temperature difference between points a and b is represented by $Td(a,b)$. Impulse response of this function $h_{(a,b),i}(t)$ to input i is:

$$h_{(a,b),i}(t) = h_{a,i}(t) - h_{b,i}(t) \qquad (8)$$

where $h_{a,i}(t)$, $h_{b,i}(t)$ are impulse responses of temperature at grid cells a and b respectively.

As can be seen in equation (5), calculation of $Td(a,b)^{Min}$ and $Td(a,b)^{Max}$ for each output requires only the knowledge of p_i^{Max}, p_i^{Min} and $h_{(a,b),i}(t)$. Given the thermal characteristics of the chip, $h_{(a,b),i}(t)$ can be obtained by simulation or by analytical methods. The process of finding the upper bounds is shown in Figure 2.

init Find $h_{j,i}(t)$ for the grid cells of interest (response of grid cell j to impulse at input i)

I. For all power sources i:
   a. Apply a step input to power source i while setting all other power sources to 0
   b. Find the impulse response of the grid cells of interest by differentiating the step response

loop Find f^{Min} and f^{Max} for the pairs of interest *(a,b)*

I. Calculate the impulse response of the temperature difference between two locations to each power source i

   using the equation $h(t)=h_{a,i}(t)- h_{b,i}(t)$

II. Calculate Td^{Min} and Td^{Max} using equations (5) and (7)

Figure 2 Algorithm for calculating the maximum temperature difference between two points *a* and *b*

Step 1 needs to be done just once, and then its results can be used for all pairs of interest. The calculations need not be done for all pairs, but only for the pairs of grid cells which are of interest. After simulations of step 1 which are done once, step 2 is done only for the pairs of interest.

Here we show how to generate the power trace leading to the maximum temperature difference. Based on (3), to maximize the value of $f(t)$ at t_0, for the intervals of τ where $h(\tau) \ge 0$, $p(t_0-\tau)$ must take the maximum value. For example if $h(\tau)$ is non-negative on interval $t_1 < \tau < t_2$, $p(t)$ must take its maximum value on interval $t_0-t_2 < t < t_0-t_1$. It can be easily shown that for the intervals of τ where $h(\tau)<0$, $p(t_0-\tau)$ must take it minimum value. These rules allow us to generate the power trace $p(t)$ which leads to the maximum temperature difference. Doing this for all power sources enables us to detect the configurations and scenarios which lead to maximum temperature variations between different points on the

die. This information is also helpful in augmenting the benchmarks and generating test data for stress tests. In the next section, the sensor placement method is explained which uses our model to guarantee the desired accuracy of sensor temperature readings.

4. THERMAL SENSOR PLACEMENT

As explained before, usually the thermal sensors can not be placed exactly at the locations they are supposed to monitor. This causes a disparity between the temperature at the point of interest and the temperature read at the sensor (also known as sensor placement error). In a thermal sensor placement technique, the objective is to find the minimum number of the sensors and their locations such that the placement errors at all points of interest are always less than the required accuracies. Let's suppose we have a set of points of interest $Q=\{q_1, ..., q_k\}$ with desired accuracies (or maximum tolerable errors) $E=\{e_1, ..., e_k\}$. The set of sensors would be $S=\{s_1, ..., s_r\}$. The objective is to find the minimum number of sensors and their locations such that for each q_i there exists a s_j for which always $T(q_i) - T(s_j) < e_i$.

We introduced the concept of observability area earlier which is defined as the area around a point of interest a in which the maximum temperature difference to a is always less than the maximum tolerable error. Therefore, to find the observability area, the maximum temperature difference between each point of interest and its neighboring grid cells must be found. The observable set for each point of interest is the set of grid cells which fall into the observability area.

The input to the sensor placement technique is a set of points of interest along with their desired accuracies. As shown in Figure 6 the chip is divided into a grid. First the observable area of each point of interest is found. It is the area in which the maximum temperature difference to that point of interest is less than its maximum tolerable error. The observable set is the set of grid cells which completely fall in the observable area. Given the observable sets of points of interest, the minimum set of grid cells will be found such that if sensors are placed in it, each points of interest has at least one sensor in its observable set. Given these maximum temperature difference between a point of interest and its neighboring grid cells enables us to identify its observable set. If some grid cells cannot be used as a sensor location due to some reasons like routing limitations, those cells are eliminated from the observable set. Given the observable set for each point of interest, the next step is to find the optimum number of sensors and their locations such that there is at least one sensor in the observable set of each point of interest. Considering the collection of all grid cells which are potential sensor locations as set G, C is the collection of all k observable sets (O_is) which are all subsets of G. A minimum cardinality set S of grid cells must be found such that S contains at least one grid cell from the observable set of each point of interest. This guaranties the ability to satisfy the accuracy requirements since at any grid cell in the observable set the temperature of the point of interest can be sensed with desired accuracy. This problem is the minimum hitting set problem which is proven to be NP-complete. There are different heuristic algorithms for this problem. We use integer linear programming (ILP) to solve the minimum hitting set problem for minimizing the number of sensors. Let's suppose $x_a=1$ if a sensor is to be placed at grid cell a, otherwise $x_a=0$. In order to minimize the number of sensors, this cost function must be minimized:

$$\sum_{a \in G} x_a \qquad (9)$$

Since each observable set must have at least one sensor in it for the corresponding hotspot to be covered, this inequality must hold for each point of interest j:

$$\sum_{a \in Oj} x_a \geq 1 \qquad (10)$$

Based on the above, the sensor minimization problem can be formulated as an ILP problem as follows:

$$\begin{aligned} &\textit{minimize} \quad \sum_{a \in G} x_a \\ &\textit{subject to} \quad \sum_{a \in Oj} x_a \geq 1 \quad j = 1,\ldots,k \\ &x_a \in \{0,1\} \quad \forall a \in G \end{aligned} \qquad (11)$$

For each grid with $x_a=1$, a sensor is placed in the corresponding grid cell. To solve ILP problem, we use lp_solve [20] which is an integer linear programming solver freely available. lp_solve is based on the revised simplex and the branch-and-bound method for the integers.

5. EXPERIMENTAL RESULTS

In order to verify the effectiveness of the proposed technique, we applied it to a multi-processor SoC comprised of 6 *XScale*® cores [11]. The layout of the chip is shown in Figure 1. MiBench Ver 1.0 [12] programs are used as benchmarks for evaluation of the technique. MiBench is a free, commercially representative benchmark suite which is developed at the University of Michigan [13]. A set of programs from the automotive/industrial, network and telecommunications categories of MiBench are selected and run on datasets provided by [14]. These programs were used as the workload arriving for each core. To introduce idle intervals between the tasks, a Pareto distribution is used [15]. A timeout-based dynamic power management policy is applied in order to determine the active and low power states which each core experiences during running these workloads. We use the power values for active and sleep states reported in [11] to generate the power trace. HotSpot 3.0 [10] is used for thermal simulations. Parameters used for package are: convection capacitance 140.4 J/K, convection resistance 0.1 K/W, spreader thickness 10-3m, and initial temperature of 333°K.

The next three figures provide a real example which shows the case where benchmarks may not be able to generate the maximum temperature differences on the die since it may require very specific conditions. Figure 3 shows a simulation slice in which the temperature difference between points a and b has reached its highest value. The dotted line shows the maximum temperature difference estimated by our model. Although the temperature difference has reached its maximum value during the simulations, it is still lower than the maximum estimated by our model.

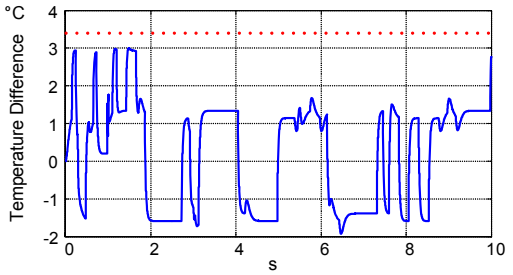

Figure 3 Temperature difference between points *a* and *b*

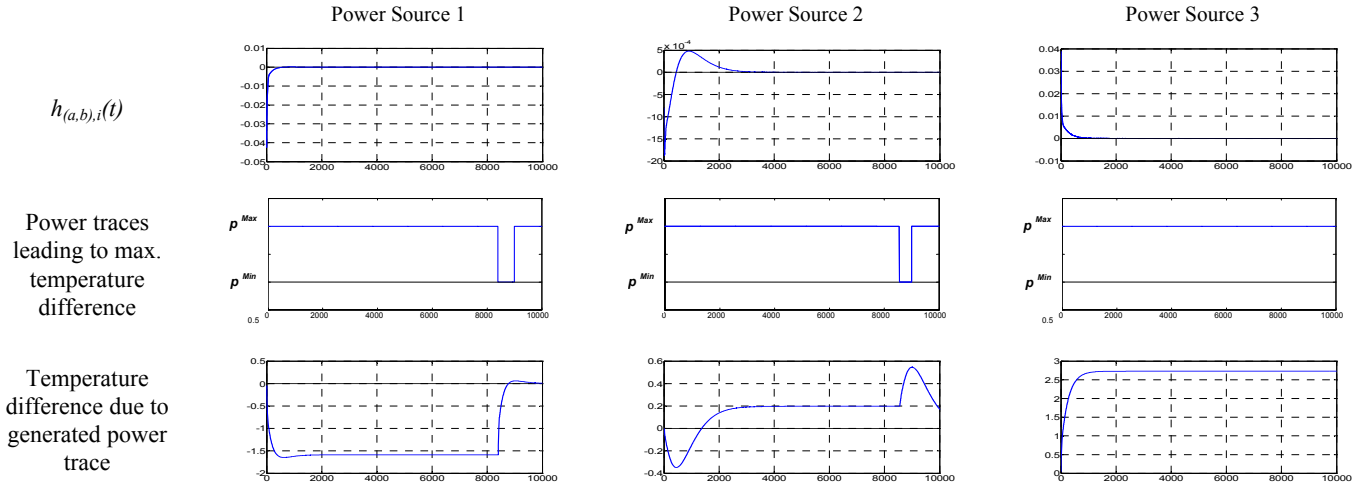

Figure 4 Constructing the power trace to generate the maximum temperature difference between points *a* and *b* by (time unit:10^{-4}s)

Figure 4 shows the situation in which the actual temperature difference found by our method exceeds the maximum temperature difference reported by simulations using standard benchmarks. The first row shows $h_{(a,b),i}(t)$ which is the response of the temperature difference between *a* and *b* to the impulse applied at input *i*. $h_{(a,b),i}(t)$ is calculated by differentiating the response of the temperature difference to the step input *i* (since it is in discrete time, this is done by differencing the consecutive samples of step response.

To keep the example easy to follow, only three power sources are considered and the rest of power sources are off. Based on the $h_{(a,b),i}(t)$, the power traces generating the maximum temperature differences are calculated as explained in section 3. Applying each of these power traces leads to the corresponding temperature differences shown at third row.

The overall temperature difference due to all power sources which provides the maximum difference at time 0.9s (time unit 9000) is shown in Figure 5. As this example shows, the maximum temperature difference occurs under very specific conditions which may be difficult to see with standard benchmarks, but can happen in real working conditions. This accentuates the necessity of a model such as ours that has a proven ability to provide the maximum temperature difference on the die. The difference between simulations and model in this example is not large because of low power consumption of *XScale*® cores, the fact that we looked at only 3 cores with observation points in proximity of each other. The error can be much more significant in modern processors with higher powered devices. Depending on which of the two points has higher temperature at any time instance, the temperature difference between them could be positive or negative. The model provides the upper bound for both positive and negative directions. Table 1 shows that the errors as high as 9ºC happen in estimates of temperature differences when relying only on simulations. Even using combinations of different benchmarks does not resolve the problem. Such errors can cause significant functional and reliability issues. For example, if the maximum temperature difference between a sensor and a hotspot is underestimated, it may cause late activation of DTM which may result in serious reliability problems.

The simulation overhead of our method is minimal. The largest overhead occurs if we used simulation instead of analytical techniques to calculate the impulse responses. It involves simulating one step response for each power source compared to simulating the whole set of benchmarks when standard simulation is used.

Table 1 Errors in temperature differences in simulations

	Positive			Negative		
	Mean	**Std. Dev.**	**Max.**	**Mean**	**Std. Dev.**	**Max.**
Automotive	1.09	0.78	4.34	1.01	0.72	3.77
Network	6.89	2.60	9.39	6.96	0.55	9.55
Telecommunication	6.02	2.53	9.16	6.36	2.47	9.29
Mixed benchmarks	1.15	1.15	8.05	0.59	0.55	7.08

We also compare our sensor placement with previous techniques. Previous techniques such as [17] and [18] depend on calculation of the range of the hotspot which is the maximum distance *r* from the hotspot that a sensor can be placed while still maintaining the intended accuracy. This range has a circular form and is centered at the point of interest which limits the accuracy and the efficiency of such techniques. To be able to guaranty a desired accuracy, this circle must be centered at the point of interest and be completely within the observability area of the point of interest with the same accuracy. Some points which meet the accuracy requirements might be missed, as demonstrated on SoC with 6 *XScale*® cores shown in Figure 6. The two red x's represent the points of interest to be monitored. The observability areas of the points of interest are shown by solid green and blue lines. Circular ranges of the hotspots are shown by dotted circles. Our sensor placement technique considers the observability area of a point of interest instead of its circular range as the potential location of a sensor. Therefore it can identify the point marked by * at the overlapping part of the observability areas to place a single sensor to monitor both points of interest.

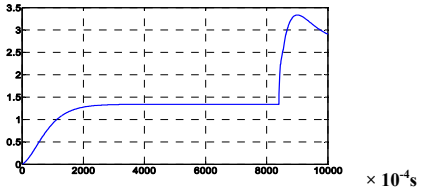

Figure 5 Overall temperature difference of example in Figure 4

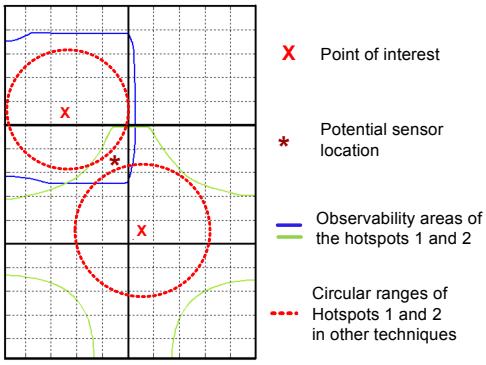

Figure 6 Using observability area vs. circular range

When using the circular ranges, this sensor location would not be identified since the ranges do not overlap; and therefore two separate sensors would be required.

To evaluate our sensor placement technique, we used different values for the desired accuracies of points of interest and compared our sensor placement technique with the techniques such as [17] which use circular ranges. Table 2 shows the number of sensors required to monitor 8 hotspots of the chip with specified accuracy which is the maximum tolerable error between the temperature sensor and the actual temperature at the point of interest. As Table 2 shows, our sensor placement technique needs fewer sensors to monitor the same hotspots with the same accuracy.

Table 2 Number of required sensors to monitor 8 hotspots of the chip in the proposed technique and range-based techniques

Desired Accuracy	1	2	3	4	5	6	7
Our sensor placement technique	7	7	6	5	5	5	4
Techniques using circular range	7	7	7	7	6	6	6

6. CONCLUSION

Having a good estimate of maximum temperature variations across the die is necessary for a number of applications such as placement of thermal sensors and reliability analysis. Normally extensive simulations are used in order to determine these variations, thus incurring significant overhead. The model proposed in this paper removes the simulation overhead by providing a model for accurate estimation of maximum temperature differences over various points across the die. It also identifies the conditions under which the maximum temperature differences and variations occur. This aspect of the model is helpful for generation of test data for stress tests and augmenting the benchmarks to check when maximum temperature differences occur in real life situations. Our experiments show that when using simulations, maximum temperature difference underestimations can be as high as 9ºC. In this work we also show how our method for obtaining maximum temperature differences can be used at the heart of a new sensor placement technique. As compared to previously proposed methods, we can reduce the number of sensors needed by 16% on average while guaranteeing the specified sensor accuracy.

7. ACKNOWLEDGEMENT

This work has been funded by Sun Microsystems, and the University of California MICRO grant 06-198.

8. 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 Transactions on Very Large Scale Integration (VLSI) Systems, 14(5):501-513, May 2006.

[2] M. Pedram, S. Nazarian, "Thermal modeling, analysis, and management in VLSI circuits: Principles and Methods," In Proceedings of IEEE, special issue on Thermal Analysis of ULSI, 2006.

[3] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," In Proceedings of the 40th Design Automation Conference, pp. 338–342, June 2003.

[4] C. J. Lasance. Thermally driven reliability issues in microelectronic systems: status quo and challenges. Microelectronics Reliability, 43:1969–1974, 2003.

[5] A. K. Coskun, T. Rosing, K. Mihic, Y. Leblebici and G. De Micheli. "Analysis and Optimization of MPSoC Reliability." In Journal of Low Power Electronics (JOLPE), April 2006.

[6] A. K. Coskun, T. Rosing and K. Whisnant. "Temperature Aware Task Scheduling in MPSoCs." In Proceedings of Design Automation and Test in Europe (DATE), 2007.

[7] A, Naveh, E. Rotem, E. A. Mendelson, S. Gochman, S. R. Chabukswar, K. Krishnan, A. Kumar, "Power and Thermal Management in the Intel® Core™ Duo Processor" Intel Technology Journal. May 2006.

[8] W. Huang, M. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," in Proceedings of Design Automation Conference, 2004, pp. 878–883.

[9] M. Barcella, W. Huang, M. Stan, and K. Skadron. "Architecture-Level Compact Thermal R-C Modeling." Tech Report CS-2002-20, Univ. of Virginia Dept. of Computer Science, July 2002.

[10] HotSpot, http://lava.cs.virginia.edu/HotSpot/

[11] Intel PXA270 processor, electrical, mechanical and thermal specification data sheet. http://www.intel.com.

[12] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite." IEEE 4th Annual Workshop on Workload Characterization, Austin, TX, December 2001.

[13] MiBench Version 1.0 Webpage, http://www.eecs.umich.edu/mibench/

[14] G. Fursin, J. Cavazos, M. O'Boyle and O. Temam. "MiDataSets: Creating The Conditions For A More Realistic Evaluation of Iterative Optimization." *Proceedings of the International Conference on High Performance Embedded Architectures and Compilers (HiPEAC 2007)*, Ghent, Belgium, January 2007

[15] T. Simunic, L. Benini, P. Glynn, G. De Micheli, "Event-Driven Power Management," IEEE Transactions on Computer-Aided Design, July 2001.

[16] R. Mukherjee, S. O. Memik, "Systematic temperature sensor allocation and placement for microprocessors." Proceedings of the ACM/IEEE Design Automation Conference (DAC '06).

[17] K.-J. Lee and K. Skadron. "Analytical Model for Sensor Placement on Microprocessors." In Proceedings of the IEEE International Conference on Computer Design (ICCD), Oct. 2005, pp. 24-27.

[18] S. Mondal, R. Mukherjee, and S.O. Memik. "Fine-Grain Thermal Profiling and Sensor Insertion for FPGAs." Proceedings of IEEE International Symposium on Circuits and Systems 2006.

[19] M. Cho, S. Ahmedtt, D. Z. Pan, "TACO: temperature aware clock-tree optimization," Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design, p.582-587, November 06-10, 2005.

[20] LP-solve. http://www.geocities.com/lpsolve/