

Fine-Grain Thermal Profiling and Sensor Insertion for FPGAs

Somsubhra Mondal, Rajarshi Mukherjee, and Seda Ogrenci Memik
Department of Electrical Engineering & Computer Science
Northwestern University, Evanston, IL, USA

Abstract – *Increasing logic densities and clock frequencies on FPGAs lead to rapid increase in power density, which translates to higher on-chip temperature. In this paper, we investigate the thermal behavior of general applications on fine-grain reconfigurable fabrics and we introduce the pre-mapping sensor insertion problem for thermal monitoring. Our study shows that on average the maximum temperature on the chip is 19.5°C higher than the ambient temperature for a transition density of 0.5 at the primary inputs. For fine-grain reconfigurable devices targeted for general applications it is difficult to predict the locations of potential hotspots a priori. However, programmability presents a unique opportunity for effective thermal monitoring. It would allow us to perform a thermal simulation on a given design first and obtain the locations of potential points of interest in a design. Then, in the pre-mapping stage the design can be updated with insertion of thermal sensors. Given a set of expected hot spots in a design we aim to determine the minimum number of sensors and their locations in order to monitor these locations with a given sensitivity requirement. Since the thermal sensors are implemented using unused CLBs on the fabric it is essential to use the logic resources efficiently. We propose an efficient algorithm to solve the sensor placement problem addressing this optimization goal.*

I. INTRODUCTION

Over the last decade FPGAs have evolved at a rapid pace with improved performance and higher logic density, offering a wide range of functionalities. In order to achieve higher performance and logic density CMOS devices have been scaled down for several years, and 90nm technology FPGAs are already available in the market (e.g. Xilinx Virtex-4™). Several FPGA vendors have roadmaps to use 65nm technology in the near future¹. One obvious consequence of this trend is increased power consumption per unit silicon area, leading to a higher overall on-chip temperature and/or large temperature variations across the chip. Increasing on-chip temperatures can lead to thermal stress on the chip, increased leakage power consumption, higher cooling and packaging costs, and reduced system reliability. Reliability of a circuit depends on temperature and the Mean Time To Failure (MTTF) is adversely affected by higher operating temperatures. Degraded circuit performance and higher interconnect resistance can also be attributed to high on-chip temperature. Besides, even if a system does not reach very high temperatures to threaten safe operating conditions, it can accelerate electromigration, which can permanently damage the system in the long run. Hence, it is of utmost importance to consider temperature as a design parameter and to study the thermal characteristics of a chip in order to obtain a reliable and efficient design closure.

Regions on a chip that dissipate excessive amounts of heat are referred to as hotspots. In order to correctly understand the thermal characteristics of a design and to prevent circuit failure

it is important to detect such hotspots. In addition, real time monitoring of the events around such hotspots would be highly desirable. Power dissipation estimates, even at a fine granularity, are not sufficient to characterize the thermal behavior of the chip, since temperature mainly correlates with power density rather than absolute power dissipation, among other chip and packaging characteristics. Hence, thermal simulation and/or monitoring are critical to ensure reliable operating conditions.

Thermal monitoring of FPGA based systems have been proposed by Buedo et al. [1-3]. They proposed to instantiate ring oscillators as thermal sensors using reconfigurable logic, which were statically configured in FPGAs [2]. Dynamic insertion of sensors using run-time reconfiguration was also proposed [3]. Velusamy et al. [4] used the thermal simulator HotSpot for thermal modeling of FPGAs and also cross validated their results with the physical thermal sensor readings [2].

Both the works by Buedo et al. and Velusamy et al. mainly target SoCs implemented on an FPGA. SoCs typically have embedded microprocessor, microcontroller, miscellaneous glue logic, and a communication bus. Such applications contain several easy to predict candidate locations as hotspots. For example, as pointed out by Velusamy et al., it is expected for components such as the register file of a microprocessor core to become a potential hotspot. Sensor placement for such coarse grained functional units can be reasonably pre-determined [1, 4].

In this paper, we investigate the thermal behavior of distributed applications mapped onto fine-grain reconfigurable logic. Speculating the locations of hotspots and inserting thermal sensors on a SoC based design is different than that of general applications mapped onto FPGAs. For general applications, having thermal sensors at pre-defined locations may not always be appropriate because thermal characteristics can vary widely across designs. This requires a pre-mapping sensor set generation and insertion stage in the design flow. We can exploit programmability of FPGAs to introduce an incremental change into the design before it is mapped onto the FPGA device. However, the task of embedding thermal sensors should be carried out under certain constraints. One approach to introduce thermal sensors into reconfigurable devices is to instantiate the sensors using vacant CLBs on the device [2]. However, we should note that the available CLBs for sensor instantiation are not unlimited. In order to circumvent this constraint dynamic reconfiguration has been proposed to insert and remove thermal sensors during run-time at alternating locations on the FPGA. This approach relieves the resource constraint. On the other hand, dynamic reconfiguration can bring performance penalties.

Therefore, optimizing the amount and locations of sensors presents a challenge for general applications mapped onto fine-grain reconfigurable fabrics. In this work, we introduce the thermal sensor insertion problem for general applications mapped onto fine-grain reconfigurable fabric aiming to address this challenge. Our specific contributions in this paper are as follows:

- We investigate the fine-grain thermal behavior of general applications on island style FPGAs,

¹ Xilinx and IBM have roadmaps to produce chips at 65nm. Lattice and Fujitsu are discussing the use of Fujitsu's forthcoming 65nm technology in future Lattice products.

- We formulate the sensor insertion problem for monitoring thermal behavior on such applications, and
- We propose an algorithm for the thermal sensor placement problem for reconfigurable fabrics.

The remainder of the paper is organized as follows. Section II presents our study of the thermal behavior of general applications on a reconfigurable fabric, and the underlying motivations for the sensor insertion problem based on the observations of this study. In Section III we discuss the thermal sensor insertion problem in detail, and present an efficient algorithm to solve this problem. Finally, Section IV summarizes our conclusions.

II. STUDY OF THERMAL BEHAVIOR

In this section we present a study of the thermal behavior of a set of MCNC benchmarks [5] on island-style architecture FPGAs. The following subsections illustrate our experimental methodology and setup for the study, and our observations, which motivate us for the subsequent sensor placement problem.

A. Methodology and Assumptions

Figure 1 depicts our experimental methodology for estimating the fine-grain thermal behavior of reconfigurable fabrics. We start with a technology-mapped netlist of look-up-tables (LUTs) and flip-flops (FFs) in blif format (Berkeley Logic Interchange Format) and pass it to the Activity Estimator tool [6] which determines the switching activity of each node in the circuit by applying the transition density model.

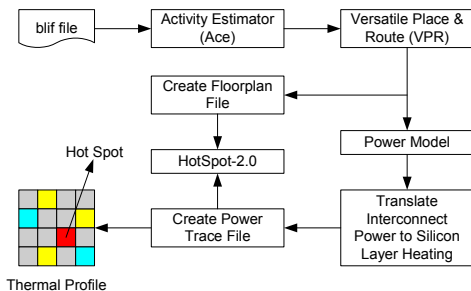


Figure 1. Experimental Methodology

The timing-driven packing tool T-Vpack [7] packs the LUTs and FFs to generate a netlist of CLBs. The switching information along with the netlist of CLBs is passed to Versatile Place & Route (VPR) [7] tool, which then places and routes the design. Power Model [6], an additional module within VPR calculates the power of each CLB and net, based on the switching activities at the nodes.

Power Model reports the power consumption of each individual CLB and power dissipated by each net. One of the main heat transfer paths from interconnect layers to the silicon layer is through vias. To incorporate the contribution of routing power dissipation into heating at the silicon layer we distribute the net power among the source and sink blocks along the respective net. Finally, we obtain the total power dissipation at each CLB that will eventually be evaluated for heat dissipation. The array of logic blocks along with their total power and the placement information (in the form of a floorplan) is then passed to HotSpot to obtain a thermal profile. We use HotSpot to generate a thermal profile of the CLBs. HotSpot was originally developed as an architecture level thermal modeling tool. Recent work has demonstrated the suitability of HotSpot for modeling thermal behavior of FPGA-based SoCs [4].

The power values obtained from Power Model are fairly accurate for 180nm technology based on the architectural parameters available with VPR. We chose to estimate the values for the 130 nm technology considering it to be more practical, though the architecture specific process parameters were not available. We use the results presented by De et al. [8] to

calculate the power-scaling factor when migrating from 180nm node to 130nm node. We have synthesized a set of MCNC benchmarks onto a Xilinx Virtex-II device, which is manufactured at 130 nm. We then used XPower [9] for power estimation of the programmable logic array and verified our empirically determined power-scaling factor. For our experiments, the power-scaling factor is determined to be 5x. The static power dissipation of Xilinx Virtex-II family is found to be within 5-20% of the total power [10]. For our experiments we assume that 12% of the total power is static power, and unused logic blocks only dissipate static power. Hence, unused CLBs have been annotated with the corresponding static power values. The instantaneous power of the logic blocks is then provided to HotSpot as a trace file. We have used a similar HotSpot configuration for heat spreader and sink as suggested by Velusamy et al. [4].

B. Observations

In this section we present experimental results of our study of thermal behavior of a set of MCNC benchmarks mapped onto island-style FPGAs. For our experiments we packed 10 LUTs per cluster, and each cluster has 22 external inputs. This configuration has been determined to be the one of the most efficient configurations in terms of delay, area, and routability [7].

TABLE I. Maximum temperature difference for benchmarks

Benchmark	Array Size	Freq. (MHz)	Maximum Temperature Difference (°C)		
			TD 0.25	TD 0.50	TD 0.75
ex5p	13	25.28	18.33	19.36	20.40
tseng	13	24.69	18.53	19.26	19.92
apex4	14	23.68	17.96	18.81	19.67
misex3	14	27.83	18.82	20.63	22.33
alu4	15	28.10	18.80	20.59	22.30
diffeq	15	26.01	17.94	18.65	19.26
apex2	16	25.67	18.71	20.43	22.11
s298	16	13.16	18.22	18.64	19.05
seq	16	24.95	18.60	20.18	21.76
frisc	21	11.60	17.52	17.85	18.13
elliptic	22	16.25	18.27	19.13	19.88
spla	22	20.42	17.94	18.86	19.77
ex1010	24	17.51	17.80	18.48	19.09
pdc	24	14.79	17.81	18.57	19.30
s38584.1	28	25.65	19.71	20.83	21.88
bigkey	29	45.80	19.77	21.69	23.26
dsip	29	34.76	19.25	20.26	21.03
Maximum	29	45.80	19.77	21.69	23.26
Average	20	23.89	18.47	19.54	20.54

Table I shows the temperature difference between maximum steady state temperature of the FPGA and ambient temperature for our benchmark set. In our experimental setup we have used an ambient temperature of 23°C. Array size represents the dimension of the smallest square array of CLBs in which the particular benchmark could successfully be placed and routed. The next three columns represent the maximum temperature difference when the transition densities (TD) of the primary inputs are 0.25, 0.5, and 0.75 respectively. The clock frequency of a circuit is bounded by its critical path delay determined by VPR [7]. As depicted in Table I the on-chip temperature can rise by a significant amount, by as high as 23°C. Thermal monitoring of the chip can be beneficial to investigate the causes for this behavior and potentially to develop design modifications.

Figure 2 shows the power distribution of the bigkey benchmark with a transition density of 0.5 at the primary inputs. The CLBs, which are used by the design, consume both static and dynamic power, and hence have higher values of power. On the other hand, unused CLBs consume only static

power, and hence, have very low power values. The input-output blocks (IOBs) at the periphery have high power values compared to the unused CLBs, and hence we observe the slightly elevated region along the boundary of the array.

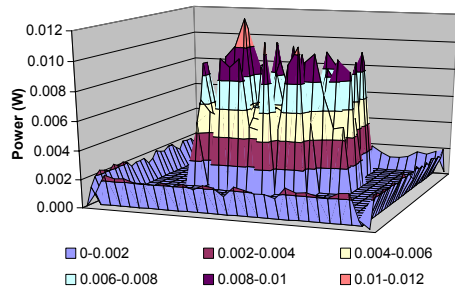


Figure 2. Power distribution across the CLB array for the *bigkey* benchmark.

Figure 3 shows the temperature profile of the corresponding power distribution obtained by thermal simulation using HotSpot, where the maximum steady state temperature occurs around the central zone of the FPGA. Different designs will have different regions of localized heating. Such hotspots can be reasonably pre-determined for SoC based designs mapped onto FPGAs. On the other hand, for general applications it is not possible to make such predictions and hence the sensor locations cannot be determined a priori. However, we can then use flexibility of reconfigurable architectures to insert thermal sensors at the best possible locations by utilizing hints from thermal simulation.

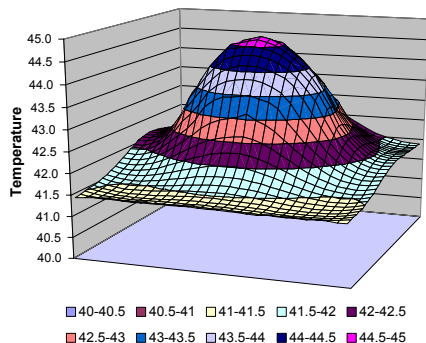


Figure 3. Temperature profile for the *bigkey* benchmark.

Our experiments revealed that the difference between maximum and minimum temperature in the logic block array is 3.34°C (average across all benchmarks). This means that, although the average temperature across the chip can rise significantly, the thermal gradient across the chip can be very steep. However, even if the thermal gradient is not very steep for a design, it can still be important to detect even a relatively small temperature differential. Such localized heating may signal local faults and defects. Thermal sensors are sensitive to measure minute temperature differences, as low as 0.6°C [3]. Optimal use and distribution of thermal sensors for capturing such events would present an additional challenge.

Velusamy et al. [4] used embedded PowerPC in Virtex-II Pro as a microcontroller to control such an array of thermal sensors connected by On-chip Peripheral Bus (OPB) [11]. We argued earlier that for a general and distributed application the sensor selection is more challenging and might require a higher number of sensors than in the case of monitoring a SoC implementation. However, even for thermal monitoring of SoC style designs mapped onto FPGAs, the minimization and judicious placement of sensors can simplify microcontroller and peripheral design associated with reading the temperature measurements from the sensors.

These facts motivate the need for optimal management and insertion of thermal sensors on FPGAs. We aim to achieve this goal in this work by formulating the thermal sensor insertion problem and proposing an efficient solution.

III. THERMAL SENSOR INSERTION

Reconfigurable fabric presents a unique opportunity to adjust the amount and distribution of sensing throughout a given design in the pre-mapping stage. At first the temperature of the reconfigurable logic can be profiled using a thermal modeling tool as presented in Section II. Then, certain localized high temperature zones can be identified, which need to be monitored using sensors. Lee et al. [12] presented an analytical model that describes maximum temperature differential between a hotspot and a region of interest for microprocessors. The temperature of the hotspot decays exponentially with the distance from a hotspot. A sensor can only be placed at a certain maximum distance from the hotspot that it is intended to monitor with a given sensitivity. We refer to this as the *Range* of the hotspot. At the same time, if multiple hotspots are in the vicinity of each other, a single sensor may be used to sense the highest temperature of the localized heating regions. In the next subsections we formally define the thermal sensor insertion problem and propose a solution to this problem.

A. Problem Formulation

Given,

- a $p \times q$ array of logic blocks,
- a set of hotspots $H = \{h_1, h_2, \dots, h_k\}$,
- a set of corresponding temperatures $T = \{t_1, t_2, \dots, t_k\}$,
- a set of ranges $R = \{r_1, r_2, \dots, r_k\}$, such that $r_i = f(t_i) \forall r_i \in R, \forall h_i \in H$, and $\forall t_i \in T$,

our goal is to determine a set of sensors $S = \{s_1, s_2, \dots, s_n\}$, and the position p_i of each $s_i \in S$, such that,

- each element s_i covers a subset of H which we denote by the one to many relation $s_i \rightarrow H_{s_i} \{h_u, \dots, h_v\}$ where $1 \leq u \leq v \leq k$,
- the distance $d(s_i, h_j) < r_i \forall h_j \in H_{s_i}$, and $\forall r_j \in R_{s_i}$
- the number of sensors, n , is minimized

B. Sensor Insertion Algorithm

In this section we discuss our thermal sensor insertion algorithm. We assume that a set of localized high temperature regions H is determined by prior thermal modeling of a particular design to be mapped onto the array of configurable logic blocks. Range $r_i \in R$ can be calculated using the following formula [12],

$$r_i = 0.5 \cdot K \cdot \ln\left(\frac{T(h_i)}{T(h_i) - \Delta T}\right),$$

where constant K depends on the packaging characteristics of the chip, and ΔT is the sensitivity of the sensor. We create a set of circles C where each circle $c_i \in C$ is centered around the hotspot h_i and has a radius $r_i \in R$. We reduce the radius to factor in a margin of safety to approximate each c_i with the CLBs that lie fully within the circle. For hotspot $h_i \in H$, let the range be denoted as $r_i \in R$ and the corresponding circle as c_i . A sensor $s_j \in S$ (to be determined) intended to monitor h_i can be instantiated in any $CLB \in c_i$. In other words, each $CLB \in c_i$ is said to cover hotspot h_i . For each CLB in c_i , we add h_i in their respective list of covered hotspots denoted as *Listcov_hotspots*. For each hotspot $h_i \in H$, we similarly determine the CLBs that cover them.

Figure 4 shows the outline of our algorithm. Our algorithm works as follows. The CLB , which covers the maximum number of hotspots, is chosen for the sensor placement. Identifying such a CLB ensures that a sensor s_j placed in such a CLB can accurately sense temperature of maximal number of hotspots $h_i \in H$ for which $d(h_i, s_j) < r_i$. Then, the *Listcov_hotspots* is updated for the remaining $CLBs$ to reflect the hotspots that are already covered. The algorithm iteratively allocates $CLBs$ for

sensors, such that maximal number of hotspots is covered with minimum sensors.

Sensor Insertion Algorithm	
Input:	CLB array after place & route, and power estimation
Output:	Number & Location of sensors
1.	Determine set of hotspots to be monitored
2.	Determine range of hotspots $r_i = f(t_i) \forall h_i \in H$
3.	For each CLB in such a range determine Listcov_hotspots
4.	Initialize Set_sensors and Set_cov_hotspots to null
5.	While not all hotspots are covered
6.	Place a sensor in CLB which covers maximum hotspots
7.	Add hotspots covered to Set_cov_hotspots and add the sensor to Set_sensors
8.	Set the position of the sensor as the position of the CLB
9.	Update Listcov_hotspots for rest of CLBs
10.	Output Set_sensors and p_i for each s_i

Figure 4. Thermal sensor minimizing and placement algorithm.

Figure 5 demonstrates how our algorithm works on a logic array. The circles around hotspots represent their respective ranges. The shaded CLBs in circles represent feasible sensor positions that cover the respective hotspot. Using our algorithm we have determined sensor s_1 covers hotspots $\{h_1, h_2, h_3\}$ and sensor s_2 covers $\{h_4, h_5\}$. This shows we can use fewer sensors than hotspots and monitor the thermal behavior of the system.

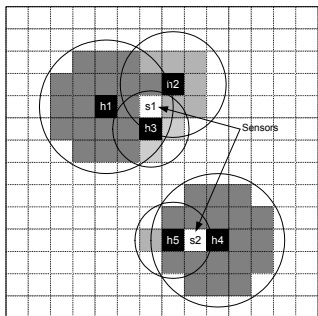


Figure 5. Example of our thermal sensor insertion.

Figure 5 demonstrates how our algorithm works on a logic array. The circles around hotspots represent their respective ranges. The shaded CLBs in circles represent feasible sensor positions that cover the respective hotspot. Using our algorithm we have determined sensor s_1 covers hotspots $\{h_1, h_2, h_3\}$ and sensor s_2 covers $\{h_4, h_5\}$. This shows we can use fewer sensors than hotspots and monitor the thermal behavior of the system.

We assume that thermal sensors can be implemented by a single CLB. Our algorithm can be easily extended if multiple CLBs are required to implement a sensor. During placement of sensors we anticipate that an unused CLB will be available at the location determined by our algorithm. In case such a CLB is not available, we expect either of two cases to happen: (1) to find an unused CLB in close proximity; such a location will still cover the hotspots since we have factored in a safety margin, or (2) local remapping of the design can be done to create an unused CLB for sensor placement.

C. Experimental Results

We performed experiments to determine a minimal set of sensors for a variety of logic arrays and different hotspot distributions. Our algorithm determines the minimum number and placement of the sensors such that all the hotspots can be sensed with a given sensitivity. Figure 6 shows the number of sensors for different number of hotspots for a $n \times n$ logic array where $n = 50, 100, 150,$ and 200 . Our experiments show that, the chances of a sensor covering multiple hotspots are greater for a higher number of hotspots in a given logic block array. For example, for a 150×150 array the number of sensors is 19 for both 30 and 35 hotspots. As seen from Figure 6, as the number of hotspots increases, the increment in the number of required

sensors gradually decreases (and eventually leads to saturation). Because of the small array dimension this trend is best observed in the 50×50 array.

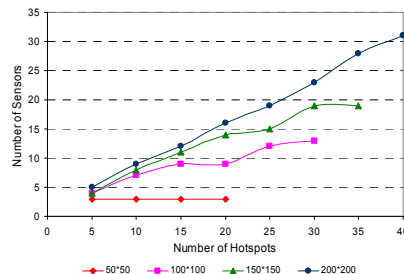


Figure 6. Number of sensors vs. number of hotspots for different CLB array sizes.

IV. CONCLUSIONS

With advancing process technologies in FPGAs, temperature will play a vital role in all future designs. In this paper we present a study of the thermal behavior of a set of MCNC benchmarks on homogeneous island-style FPGA architecture. Our results show that there is a considerable increase in the temperature of some logic blocks. This can lead to higher leakage power consumption, increased packaging and cooling costs, and reduced system reliability in the long run. A flexible methodology is presented in this paper for determining the amount and positions of thermal sensors to be deployed on a FPGA for an arbitrary design that uses distributed fine-grain reconfigurable logic. The eventual power dissipation and thermal profile of a FPGA is determined by the by application rather than by the FPGA vendor [13]. Hence, we envision the sensor placement algorithm presented in this work will be particularly beneficial.

V. ACKNOWLEDGMENT

This research is supported in part by NSF Career Award 0546305.

REFERENCES

- [1] S. Lopez-Buedo and E. Boemo, "Making Visible the Thermal Behaviour of Embedded Microprocessors on FPGAs: a Progress Report," presented at *International Symposium on Field Programmable Gate Arrays*, 2004.
- [2] S. Lopez-Buedo, J. Garrido, and E. I. Boemo, "Thermal Testing on Reconfigurable Computers," *IEEE Design and Test of Computers*, vol. 17, pp. 84-91, 2000.
- [3] S. Lopez-Buedo, J. Garrido, and E. I. Boemo, "Dynamically Inserting, Operating, and Eliminating Thermal Sensors of FPGA-based Systems," *IEEE Transactions on Components and Packaging Technologies*, vol. 25, pp. 561- 566, 2002.
- [4] S. Velusamy, W. Huang, J. Lach, M. R. Stan, and K. Skadron, "Monitoring Temperature in FPGA based SoCs," presented at *International Conference on Computer Design*, 2005.
- [5] S. Yang, "Logic Synthesis and Optimization Benchmarks," 1991: Microelectronics Center of North Carolina.
- [6] K. K. Poon, "Power Estimation for Field Programmable Gate Arrays," MS Thesis in Dept. of Electrical and Computer Engg, University of British Columbia, 1999.
- [7] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*, 1999.
- [8] V. De and S. Borkar, "Low Power and High Performance Design Challenges in Future Technologies," presented at *Great Lake Symposium on VLSI*, 2000.
- [9] Xilinx XPower, www.xilinx.com/xpower
- [10] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic Power Consumption in Virtex-II FPGA Family," presented at *International Symposium on Field Programmable Gate Arrays*, 2002.
- [11] IBM CoreConnect Bus Architecture, www.chips.ibm.com
- [12] K.-J. Lee, K. Skadron, and W. Huang, "Analytical Model for Sensor Placement on Microprocessors," presented at *International Conference on Computer Design*, 2005.
- [13] Virtex-II Device Package User Guide, www.xilinx.com