

A Sensor Distribution Algorithm for FPGAs with Minimal Dynamic Reconfiguration Overhead

Rajarshi Mukherjee, Somsubhra Mondal, and Seda Ogrençi Memik
Department of Electrical Engineering and Computer Science
Northwestern University, Evanston, IL 60208, USA

ABSTRACT

Thermal monitoring of a design plays a vital role to ensure safe and reliable thermal operating conditions. Thermal monitoring by employing thermal sensors is a popular technique for assessing thermal behavior and providing directives for run-time preventive measures. Field Programmable Gate Arrays (FPGAs) offer a wide range of functionalities and therefore static placement of thermal sensors during fabrication process is inefficient and inappropriate for such resource constrained devices. In such cases, dynamic reconfiguration is shown to be effective for thermal monitoring because the resources required to instantiate sensors and their exact placement might conflict with the placed and routed circuit. However, instantiating a large number of sensors will incur significant reconfiguration overhead. Our goal is to assist the dynamic reconfiguration based sensor placement approach such that it becomes more efficient. In this paper we present our Longest Edge Rectangular Bisection algorithm, that minimizes the number of required sensors and controls their distribution on the reconfigurable fabric. Our algorithm effectively monitors the design and maintains a given accuracy level in temperature sensing. For a set of hotspots distributed across different logic array sizes, our algorithm reduces the number of required sensors by 69.5% and 17.5% on an average, when compared to a uniform grid-based placement and a more efficient circular intersection based placement respectively.

Categories and Subject Descriptors

B.6.3 [Logic Design]: Design Aids – Optimization

J.6 [Computer-Aided Engineering]: CAD

General Terms

Algorithms, Measurement

Keywords

Dynamic Reconfiguration, Temperature, FPGA, Thermal Sensor

1. INTRODUCTION

CMOS devices have been scaled down for several years to achieve higher logic density and performance. FPGA devices have consistently kept up with this trend of shrinking technology sizes and increasing clock frequencies, and over the last decade they have made remarkable progress. Commercial FPGAs available today use 90 nm process technologies, and offer a wide range of functionalities. However, power efficiency of FPGAs has continually lagged behind these improved system capabilities, and is deteriorating with each generation of technology scaling.

An immediate outcome of this increased power dissipation is higher overall die temperature and/or larger temperature variation across the device. Application Notes from Xilinx [1] show that for a 90% device utilization, when a typical design is mapped onto Xilinx Virtex-E™, the die temperature could rise as much as 50°C above ambient. Higher on-chip temperatures have adverse effects on transistor switching speed and interconnect resistance. In addition, high die temperature can also lead to thermal stress that threatens system reliability by accelerating electromigration. Besides, increased leakage power consumption, higher packaging and cooling costs can also be attributed to high temperature. Hence, chip level thermal monitoring of FPGAs is particularly important for the following reasons. First, the high operating temperature of a device can exceed the critical die temperature, and for process technologies of 65 nm and below this could lead to severe thermal stresses. Second, leakage power increases exponentially with increase in temperature, and hence it can lead to an unstable positive feedback situation. Third, collecting thermal data from logic array is important to characterize an application, especially when used for rapid prototyping and emulation. An attempt to characterize the thermal behavior of a logic array by sensing the temperature of the device at a particular location will be inadequate and inaccurate.

Power and temperature characteristics of the logic array of an FPGA cannot be determined during the fabrication process because they are heavily application dependent. The flexibility offered by FPGAs makes them versatile for mapping different applications onto them, and hence depending on the usage of the programmable components power dissipation and thermal behavior of these devices can differ widely. Therefore, pre-assigning the number of required sensors and their locations to thermally monitor a design can be unsuitable and/or inefficient during design fabrication. In addition, modern FPGA devices also have embedded DSP blocks and microprocessor cores. Depending on whether a particular application utilizes them or not, the thermal sensing locations can change. A conservative static placement of sensors may evenly distribute sensors throughout the configurable logic block (CLB) array to monitor all the high temperature regions. On the other hand, a more practical approach is to identify possible hotspots for a given application, and then sensors can be embedded on certain locations using unutilized logic blocks at the pre-mapping stage [2]. However, this kind of static sensor placement takes up a lot of valuable FPGA resources, and this may not be desirable for mapping larger designs onto resource constrained devices like FPGAs. Besides, depending on the application and its placement and routing, it may not always be possible to find unused CLB(s) at the desired sensor location. Moreover, this can also incur significant communication overhead to read the sensor values periodically.

Dynamic reconfiguration is shown to be useful to alleviate this problem of heavy resource usage by statically placing sensors. In this approach sensors are instantiated by dynamic reconfiguration, temperature values are read, and then the resources used by the sensors are freed and allocated back to the application [3]. Although programmability of FPGAs offers a unique opportunity for embedding sensors during the pre-mapping stage, if the number of sensors instantiated is large, this will cause significant overhead. Therefore it is of utmost importance to reduce the number of required sensors and also place them judiciously in order to minimize dynamic reconfiguration overhead and read back delay of sensors. This necessitates a methodical and efficient approach to determine the minimum number of sensors and their locations such that the dynamic reconfiguration overhead is minimized, and at the same time an effective monitoring of the FPGA is achieved. We aim to accomplish this goal of effective sensor distribution in this paper.

Given a candidate hotspot map of an application, we propose an efficient *Longest Edge Rectangular*

Bisection algorithm to determine a minimum number of sensors to monitor the hotspots for the given application. Our algorithm also determines the physical locations of these sensors on the reconfigurable fabric. Our algorithm begins with creating the smallest rectangle that covers all the hotspots. This rectangle is then recursively bisected along its longest edge the resulting rectangles are squeezed and bisected again along their longest edges until these smaller rectangles are just small enough such that all the hotspots within can be effectively monitored by placing a sensor at the center of the rectangle. We show that our algorithm reduces the number of sensors by 69.5% on an average when compared to the standard grid based placement of sensors [4]. Also, a comparison with the circular intersection based sensor placement [5] shows that our algorithm yields a 17.5% savings in the number of required sensors.

Our specific contributions in this paper are as follows:

- Formulate the problem of sensor allocation to support dynamic reconfiguration of sensor resources into a given system,
- Formulate the problem of minimizing the required number of sensors to monitor a given hotspot map, and,
- Develop Longest Edge Rectangular Bisection algorithm to determine the minimum the number of required sensors and determine their placement.

The remainder of the paper is organized as follows: Related work is discussed in Section 2. Section 3.2 we present details of the sensor allocation and placement paradigm to minimize dynamic reconfiguration in reconfigurable logic. We discuss our Longest Edge Rectangular Bisection algorithm in Section 3.2.3. In Section 4 we present our results and make comparisons with existing work in literature. We summarize our conclusions in Section 5.

2. RELATED WORK

Buedo et al. [2-4] presented ring oscillator based thermal monitoring of FPGAs. Both static placement of sensors [2] and dynamic instantiation of sensors by run-time reconfiguration [3] is discussed.

An effective thermal simulator HotSpot is proposed by Skadron et al. [6]. A cross-validation approach by Velusamy et al. [7] compare sensors readings of statically configured ring oscillators with results generated by HotSpot [6]. However, Velusamy et al. specifically targeted FPGA implementation of SoCs with six sensors manually placed at predefined locations. In our work, we allocate a minimum number of sensors and determine their placement to

monitor a set of hotspots for a generic application mapped onto FPGAs.

Mondal et al. [5] proposed a circular intersection based sensor insertion approach for fine grain reconfigurable fabrics. The circular intersection based approach relies on placing the sensor at the intersection points of the hotspot’s range, whereas in this work we place sensors based on their coverage area.

Coverage problems in the domain of sensor networks bear some similarity with our problem, where coverage can be loosely defined as the quality of surveillance. These coverage problems aim to verify the successful deployment of sensors such that a set of certain objects or a certain area is covered by the said sensor deployment [8, 9]. On the other hand, our Longest Edge Rectangular Bisection algorithm determines the minimum number of sensors and their locations to monitor (cover) a given area (CLB array).

Another interesting coverage problem encountered in the sensor network domain is the Art Gallery Problem [10]. A solution to this problem aims to find the minimum number of observers required to obtain a complete coverage of the gallery visually. In this problem placing an observer is determined by a *line of sight*. On the other hand, for our sensor distribution problem to minimize reconfiguration overhead, placement of sensors is constrained by the error margin, which is determined by the range or coverage area of the sensor. Therefore, our problem also has a distant constraint in addition to establishing a line of sight.

Our proposed Longest Edge Rectangular Bisection algorithm is based on bisection as the name suggests. Similar techniques are extensively used in physical design for placement [11]. In rectangular dissection based placement algorithms, the primary focus is to minimize the net cut costs, and elements can be moved across partitions. Our approach is a more constrained case, where dissections are only allowed at the midpoint of the longest edge.

Using our algorithm we first create the smallest rectangle that covers all the hotspots. This rectangle is bisected recursively into smaller equal rectangles and squeezed to form smallest rectangles to cover all hotspots within. The algorithm terminates when all bisected rectangles are smaller than the coverage of the sensor and sensors are placed at the center of such rectangles.

3. SENSOR DISTRIBUTION

Dynamic reconfiguration in reconfigurable fabrics offers a unique opportunity for adjusting the placement of thermal sensors to monitor its thermal characteristics. In this section we present our pre-

mapping thermal sensor distribution paradigm. First, we give a brief overview of the structure of a thermal sensor and how it is implemented on a reconfigurable architecture. Next, we discuss in detail our thermal sensor allocation and placement algorithm.

3.1 Implementation of Thermal Sensors

Ring oscillators are extensively used for implementing thermal sensors on reconfigurable fabrics. The relationship between transistor switching speed and temperature is exploited to measure temperature by sensing the output frequency of the ring oscillator. Figure 1 shows a schematic of a ring oscillator based thermal sensor. It consists of an odd number of inverters linked together in a chain like structure, and the capture counter is clocked by the output of the ring oscillator, which specifies the oscillation frequency.

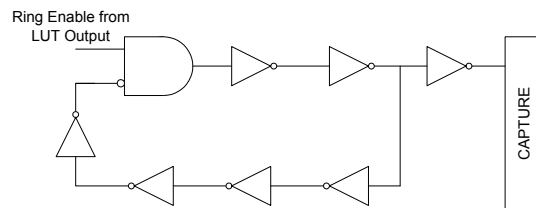


Figure 1. Schematic of a thermal sensor

These ring oscillator based thermal sensors can be implemented on FPGAs using Look-Up Tables (LUTs) [7]. Using configuration port capabilities of Xilinx technology, these ring oscillator based thermal sensors can be inserted, operated, and eliminated dynamically from the circuit by full or run-time reconfiguration [3].

3.2 Sensor Allocation and Placement

Regions on a chip that dissipate excessive amounts of heat are referred to as hotspots. Given a set of hotspots on a chip, our proposed algorithm aims to achieve the following objectives:

- Minimize the number of sensors required to detect these hotspots, and
- Determine the placement of the sensors to monitor these hotspots.

3.2.1 Key Terms and Problem Overview

Before we go into the details of our sensor distribution paradigm, we briefly discuss some of the key terms used throughout this paper and present a brief overview of the problem in this section.

Grid-Based Placement: Uniform placement of sensors (irrespective of locations of the hotspots) within a fixed grid across the entire CLB array of the FPGA is referred to as grid-based placement. Figure

2 shows a grid-based placement of sensors, where the entire CLB array is partitioned into grids, and a sensor is placed at the center of each grid.

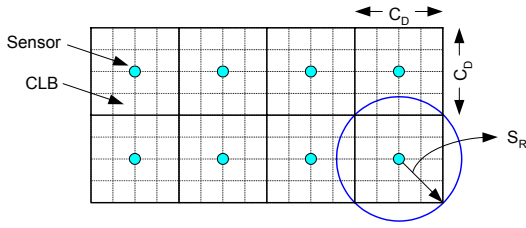


Figure 2. Grid-based Placement of Sensors

Range of Sensor: A sensor can only be placed at a certain maximum distance from a hotspot that it is intended to monitor within a given error margin. This maximum distance from a sensor at which the approximation error in measuring temperature is below a pre-defined level is defined as the range of the sensor, denoted by S_R .

Sensor Coverage Area: In Figure 2 the circle centered at the right bottom sensor with radius S_R defines its region of coverage. However, for grid-based placement, this coverage area is rectangular and the coverage area of a sensor placed at the center of a grid is the CLBs or the chip area within a grid. For example in Figure 2, there are 8 sensors (2×4 array) and the coverage area of each is 16 CLBs (4×4 array). We refer to that as each sensor *covers* a 4×4 array of CLBs.

Coverage Area Dimension: The maximum dimension of a rectangular bounding box, such that the distance from the center of the rectangle to any of its corners (half of the diagonal) is less than or equal to S_R is defined as the coverage area dimension, denoted by C_D . In general, considering a rectangular coverage area with height h and width w , we have $\{h, w\} \leq C_D$. Thus, C_D equals $\sqrt{2} \cdot S_R$.

In a grid-based placement of sensors, with the increase in CLB array size, there is a proportional increase in the number of required sensors. Such heavy deployment of sensors will incur significant overhead during dynamic reconfiguration and read back. To address this issue of increasing sensor array size, we propose an effective method to reduce the number of sensors required to monitor a given set of hotspots, and also determine their physical location on the reconfigurable fabric.

3.2.2 Problem Formulation

Given,

- a $M \times N$ array of configurable logic blocks,
- a set of hotspots $H = \{h_1(x_{h1}, y_{h1}), h_2(x_{h2}, y_{h2}), \dots, h_k(x_{hk}, y_{hk})\}$; (x_{hi}, y_{hi}) being the coordinates of hotspot h_i , and

- a coverage area dimension C_D of the sensors used
- Find a minimum set of sensors $S = \{s_1, s_2, \dots, s_n\}$ and location $(x_{si}, y_{si}) \forall s_i \in S$ such that:
- each hotspot h_i is covered by some sensor s_j , given by a one-to-many relationship $s_j \rightarrow H_{s_j}\{h_u, \dots, h_v\}$, such that the dimensions of the minimum enclosing rectangle for each H_{s_j} does not exceed C_D .

3.2.3 Sensor Distribution Using Longest Edge Rectangular Bisection

For a given design, a map of potential hotspots can be obtained by thermal profiling. Effective monitoring of these hotspots with minimal resource overhead calls for a judicious allotment of sensors and to determine their physical locations on the reconfigurable fabric. Our Longest Edge Rectangular Bisection algorithm aims to solve this sensor allocation and placement problem by recursively bisecting the CLB array along its longest edge to form *covering rectangles* around hotspots. Finally, sensors are instantiated at the center of each such rectangle.

Given a set of hotspots in the $M \times N$ CLB array, our algorithm aims to minimize the number of required sensors to effectively monitor all the hotspots. The key step of our approach is to recursively create covering rectangles by bisection along the longest edge until both dimensions of the covering rectangle are less than or equal to C_D .

The algorithm starts out with a single covering rectangle CR_1 , which is basically the smallest rectangle in the entire $M \times N$ CLB array that covers all the hotspots. A *Bisection Tree* is maintained that keeps track of the bisections (horizontal or vertical). CR_1 is bisected along its longest edge and the edges of the bisected rectangles are *squeezed* to form *tight* covering rectangles such that they occupy minimum area to cover all hotspots within their boundary. Finally, when no more bisection is possible, the number of leaf nodes of the Bisection Tree equals to the number of sensors required and sensors are placed at the center tight covering rectangles corresponding to the leaf nodes. The pseudo code of our algorithm is shown in Figure 3.

An illustration of our algorithm is shown in Figure 4. First our algorithm creates tight covering rectangle CR_1 , which in this example is the entire $M \times N$ CLB array. In Figure 4, each bisection along the longest edge is marked with an H or V, indicating whether it is a horizontal or a vertical bisection. CR_1 is bisected vertically to obtain tight covering rectangles CR_2 and CR_3 . In consecutive bisections, both CR_2 and CR_3 are horizontally bisected to form CR_4 though CR_7 , as shown in the figure. Assuming that C_D spans the length of 4 CLBs, CR_4 , CR_5 , and

CR_7 cannot be bisected any further, because their dimensions are less than C_D . But, CR_6 can still be vertically bisected to yield CR_8 and CR_9 . So finally, when no more bisection is possible the algorithm terminates, and the number of required sensors equals to the number of leaf nodes in the Bisection Tree.

Longest Edge Rectangular Bisection Algorithm	
Inputs:	
[1]	Dimension (Rows, Columns) of the CLB Array
[2]	Set of hotspots H and their coordinates
[3]	Coverage area dimension C_D
Outputs:	
[1]	Number of required sensors n
[2]	Locations of these n sensors
<ol style="list-style-type: none"> 1. Create initial tight covering rectangle (TCR) CR_1 for all the hotspots 2. Create Bisection Tree with root CR_1 3. $TCR_{current} = CR_1$ 4. If (dimensions(CR_1) $\geq C_D$) Push back CR_1 to list $L_{Can-Be-Bisected}$ 5. Else Push back CR_1 to list $L_{Cannot-Be-Bisected}$ 6. List iterator $L_{iterator} = L_{Can-Be-Bisected}.begin()$ 7. While ($L_{iterator} \neq L_{Can-Be-Bisected}.end()$) <ol style="list-style-type: none"> a. $TCR_{current} = \text{TCR pointed by } L_{iterator}$ b. Bisect $TCR_{current}$ along its longest edge c. Create $TCR_{Left-Child}$ and $TCR_{Right-Child}$ d. Delete $TCR_{current}$ from $L_{Can-Be-Bisected}$ e. If (dimensions($TCR_{Left-Child}$) $\geq C_D$) Push-back $TCR_{Left-Child}$ to $L_{Can-Be-Bisected}$ f. Else Push-back $TCR_{Left-Child}$ to $L_{Cannot-Be-Bisected}$ g. If (dimensions($TCR_{Right-Child}$) $\geq C_D$) Push-back $TCR_{Right-Child}$ to $L_{Can-Be-Bisected}$ h. Else Push-back $TCR_{Right-Child}$ to $L_{Cannot-Be-Bisected}$ i. Advance $L_{iterator}$ j. $TCR_{current} = \text{TCR pointed by } L_{iterator}$ 8. End-While 9. $n = L_{Cannot-Be-Bisected}.size()$ 10. Place sensors at the center of each TCR in list $L_{Cannot-Be-Bisected}$ (leaf nodes of the Bisection Tree) 	

Figure 3. Pseudo Code of the proposed Longest Edge Rectangular Bisection Algorithm

From Figure 4 we see that 5 sensors will be required in this case, and the sensors should be placed at the centers of tight covering rectangles CR_4 , CR_5 , CR_7 , CR_8 , and CR_9 . The final placement of sensors is shown in Figure 5.

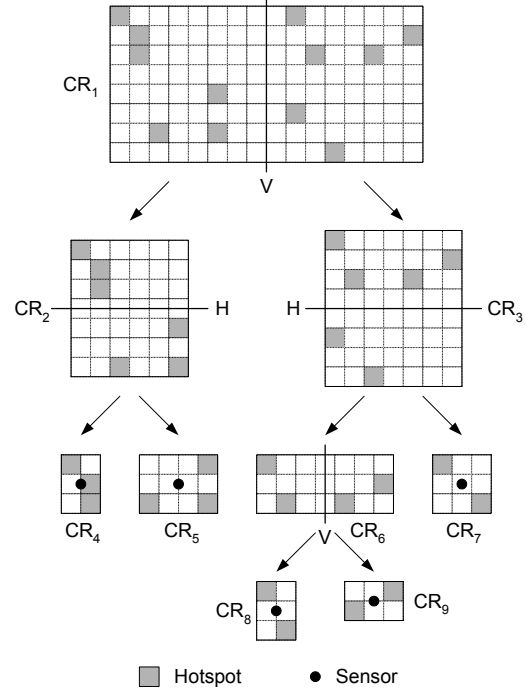


Figure 4. Bisection Tree of Longest Edge Rectangular Bisection Algorithm

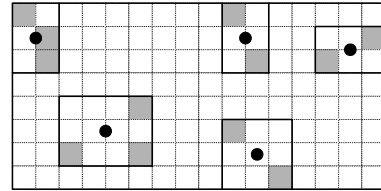


Figure 5. Allocation and Placement of Sensors given by our Longest Edge Rectangular Bisection Algorithm

For the same hotspot distribution as shown in Figure 5, we observe from Figure 2, that for grid-based placement the required number of sensors will be 8. On the other hand, only 5 sensors are required to monitor the same hotspot map by our Longest Edge Rectangular Bisection algorithm.

Assigning level 0 to the root node, the maximum number of bisections (or in other words, the number of times the *while* loop in Figure 3 is executed) for each level l is 2^l . For n hotspots, the height of the bisection tree can be at most $\lg n$, and hence excluding the leaf nodes the maximum value of l is $\lg n - 1$. Therefore, the number of bisections is given by $O(\sum 2^l)$, where $0 \leq l \leq \lg n - 1$, which can be simplified as $O(n)$. Each bisection takes constant time. Thus running time of our algorithm is linear with the number of hotspots.

4. EXPERIMENTAL RESULTS

We performed experiments to determine a minimal set of sensors and their physical locations for a wide variety of logic array sizes and different hotspot maps. We present our experimental results and make comparisons with grid-based placement by Buedo et al. [4] and circular intersection based placement by Mondal et al. [5]. In the next subsection we first discuss our experimental framework and methodology, and then we present our experimental results.

4.1 Experimental Methodology

In this section we present our experimental framework and methodology. Figure 6 shows our sensor allocation and placement methodology.

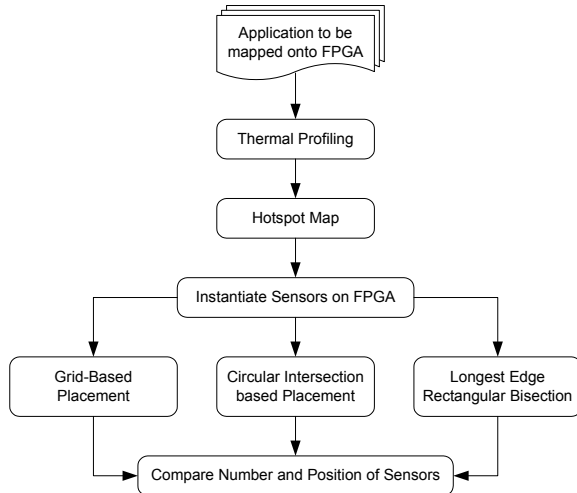


Figure 6. Sensor Placement Methodology

We start with a set of applications that will be mapped onto an FPGA, and then perform thermal simulation on the logic array. Effective thermal simulators have recently been proposed [6, 12]. Based on this thermal simulation, a map of hotspots for each application is generated and coverage area of thermal sensor is estimated. Previous work on grid-based placement of sensors [4] assume that a sensor is placed at the center of a 14×11 CLB array. Our sensor coverage area dimension CD is derived from this placement in order to make an even comparison with existing work. Approximating this CLB array size to its nearest square we have CD as 12, or in other words, we assume that the largest CLB array size a sensor can cover is 12×12 . Using our Longest Edge Rectangular Bisection algorithm the required number of sensors and their locations are determined. The sensors can then be placed using dynamic reconfiguration or embedded statically if the CLB utilization of the design has enough slack. For

comparison with the circular intersection based placement by Mondal et al. [5], we have chosen its radius such that the converge area of the sensor is same in both approaches. In the next section we present our experimental results and make comparisons with existing work on sensor placement for thermal monitoring.

4.2 Results

Input maps of hotspots are created by a random assignment of 5 to 75 hotspots for a given CLB array size. For each CLB array size, 5 to 75 hotspots were assigned randomly in increments of 5 hotspots. For our experiments we have different array sizes. They are 64×42 , 96×64 , 128×86 , and 160×110 . Different chips from the Xilinx Virtex4 FPGA family motivated the choice of such CLB array sizes. The purpose for using all such different configurations – array sizes and hotspot distribution, is to ensure that we evaluate our proposed technique on a wide variety of inputs and design characteristics. Next we present our results in Table I.

Table I. Number of sensors obtained using Longest Edge Rectangular Bisection for different number of hotspots assigned to different CLB array sizes.

CLB	64×42			96×64			128×86			160×110		
Hotspots	grid	circ-int	rect-bisc	grid	circ-int	rect-bisc	grid	circ-int	rect-bisc	grid	circ-int	rect-bisc
5	18	4	3	40	4	5	72	5	5	114	5	4
10	18	6	6	40	9	7	72	8	9	114	9	9
15	18	8	7	40	11	10	72	11	11	114	15	14
20	18	10	7	40	15	10	72	15	13	114	17	16
25	18	12	7	40	17	14	72	16	15	114	17	18
30	18	12	7	40	17	14	72	20	17	114	22	21
35	18	14	8	40	21	15	72	22	18	114	25	22
40	18	16	8	40	24	18	72	25	19	114	31	24
45	18	15	8	40	23	18	72	27	20	114	32	27
50	18	15	8	40	27	20	72	25	22	114	32	31
55	18	16	12	40	26	21	72	27	26	114	35	31
60	18	18	12	40	29	22	72	29	27	114	40	33
65	18	18	12	40	28	23	72	32	28	114	41	36
70	18	18	12	40	26	23	72	30	28	114	42	38
75	18	20	12	40	32	23	72	33	28	114	48	39
Avg.	18	11.6	8.6	40	17.6	13.8	72	18.3	15.9	114	21.8	19.7

We have shown the number of sensors required to monitor the hotspots for different array sizes

varying from a minimum of 64×42 to a maximum of 160×110. Grid based placement requires constant number of sensors per array size. For example the number of sensors required in grid based placement is 18 for 64×42 array and 114 for 160×110 array. Next we compare Longest Edge Rectangular Bisection with circular intersection based placement. Longest Edge Rectangular Bisection offers a 26.1% and 9.58% decrease in number of sensors for arrays 64×42 and 160×110 respectively. For the smaller array hotspots are more densely distributed which offers more opportunities for our technique to place sensors such that they can cover higher number of hotspots. Both Rectangular Bisection and circular intersection based placement techniques rely on the availability of the thermal profile for an application suite. Our technique leads to an overall 17.5% reduction in number of sensors compared to circular intersection based placement technique. Also, Rectangular Bisection achieves 69.5% decrease in number of sensors over grid-based placement when profiling data for an application suite is available. The run time for determining sensors for grid based placement is constant time. In comparison our algorithm takes $O(n)$ time to allocate and place sensors. The circular intersection based approach relies on placing the sensor at the intersection points of the hotspot's range and has higher computational complexity.

5. CONCLUSIONS

Monitoring temperature by employing thermal sensors is a widely used technique which is shown to be useful in evaluating thermal characteristics of designs and improving system reliability. Programmability of FPGAs presents the opportunity to instantiate sensors by dynamic reconfiguration. In this paper, we have presented a novel Longest Edge Rectangular Bisection algorithm for allocation and placement of thermal sensors in FPGAs that minimizes the dynamic reconfiguration overhead associated with instantiating sensors on the CLB array. Our algorithm is based on a recursive bisection technique that bisects a bounding box, which covers all the hotspots in the beginning. We demonstrate that for a set of benchmarks, our methodology leads to 69.5% decrease in number of sensors over grid-based placement and 17.5% reduction in number of sensors compared to circular intersection based placement technique.

6. ACKNOWLEDGMENT

This research is supported in part by National Science Foundation Career Award 0546305.

7. REFERENCES

- [1] A. Lesea and M. Alexander, "Powering Xilinx FPGAs," 2002.
- [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. 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.
- [5] S. Mondal, R. Mukherjee, and S. O. Memik, "Fine-Grain Thermal Profiling and Sensor Insertion for FPGAs," presented at International Symposium on Circuits and Systems, 2006.
- [6] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-Aware Microarchitecture," presented at International Symposium on Computer Architecture, 2003.
- [7] 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.
- [8] C. F. Huang and Y. C. Tseng, "The Coverage Problem in a Wireless Sensor Network," presented at Workshop on Wireless Sensor Networks and Applications, 2003.
- [9] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," presented at INFOCOM 2001.
- [10] V. Chvatal, "A Combinatorial Theorem in Plane Geometry," *Journal of Combinatorial Theory* vol. 18, pp. 39-41, 1975.
- [11] N. A. Sherwani, *Algorithms for VLSI Physical Design Automation*. Norwell, MA: Kluwer Academic Publisher, 1995.
- [12] Y. Yang, Z. G. C. Zhu, L. Shang, and R. P. Dick, "Adaptive Chip-Package Thermal Analysis for Synthesis and Design," presented at Conference on Design, Automation, and Test in Europe, 2006.