

# Monitoring Temperature in FPGA based SoCs

Siva Velusamy<sup>†</sup>, Wei Huang<sup>‡</sup>, John Lach<sup>‡</sup>, Mircea Stan<sup>‡</sup> and Kevin Skadron<sup>†</sup>  
Departments of Computer Science<sup>†</sup>, and Electrical and Computer Engineering<sup>‡</sup>  
University of Virginia.  
{siva, skadron}@cs.virginia.edu, {whuang, jlach, mircea}@virginia.edu

## Abstract

*FPGA logic densities continue to increase at a tremendous rate. This has had the undesired consequence of increased power density, which manifests itself as higher on-die temperatures and local hotspots. Sophisticated packaging techniques have become essential to maintain the health of the chip. In addition to static techniques to reduce the temperature, dynamic thermal management techniques are essential. Such techniques rely on accurate on-chip temperature information. In this paper, we present the design of a system that monitors the temperatures at various locations on the FPGA. This system is composed of a controller interfacing to an array of temperature sensors that are implemented on the FPGA fabric. Such a system can be used to implement dynamic thermal management techniques. We cross validate the sensor readings with values obtained from HotSpot, a pre-RTL architectural level thermal modeling tool.*

## 1. Introduction

Continual improvements in process technology have enabled IC designers to increase logic density. However, this increase in logic density, coupled with increased frequency of operation have resulted in an exponential increase in power density. According to ITRS [1], microprocessor power density is projected to reach 100  $W/cm^2$  beyond the 50nm technology nodes. This exponentially increasing power density manifests itself as heat, which has to be continually removed from the die to ensure reliable operation. Another complicating factor is the presence of local hotspots, which arise due to varying levels of power dissipation across the chip.

Increased junction temperature and the presence of local hotspots has many deleterious effects. Increased temperature decreases the transistor switching speed, thereby directly reducing performance. Leakage current increases exponentially with temperature, causing a positive feedback loop between leakage power and temperature [2]. Hotspots are therefore a major determinant of packaging and cooling costs, even for chips with low operating temperature.

Sophisticated packaging techniques are now employed to handle increased temperatures. A spectrum of techniques from heatspreaders and heatsinks to active cooling are employed to spread the heat generated more efficiently. In high-performance microprocessors packaging costs rise by \$1 to \$3 per Watt of power dissipated [3]. To avoid designing packages for the worst case operating condition, current generation microprocessors employ some form of dynamic thermal management (DTM), e.g. clock gating, dynamic voltage scaling, etc. The DTM throttling is employed when operating temperature exceeds a certain predefined value.

As FPGA-based Systems-on-a-Chip (SoCs) are getting more popular, many of the issues regarding temperature need to be tackled. SoCs are employed in a wide variety of operating conditions, and as a result, a single packaging solution will be necessarily inefficient. Hence, implementing some form of dynamic thermal management becomes critical for cost effectiveness and ensuring reliability of operation.

In this paper, we present the design and empirical measurements for an FPGA system that continually monitors the temperature levels at various locations on the die, and activates a thermal response when a thermal emergency is detected. This system comprises of an array of temperature sensors implemented on the FPGA fabric. The main contributions of this paper are as follows:

- We describe the overall design of a system capable of implementing DTM in FPGA based SoCs.
- We discuss the issues involved in the design of a temperature sensor that can be implemented on an FPGA.
- We compare the sensor temperatures to values obtained from *HotSpot* [6] – an architectural level thermal simulator. This serves as a cross-validation of both *HotSpot* and our FPGA based system.
- We identify the spatial granularity at which thermal phenomena arise in FPGA based designs.

## 2. Design

We implemented our design on a Xilinx Virtex-2 Pro (*XC2VP7*) FPGA present in the *Insight Memec* board [4, 5]. However, the techniques used are equally applicable across other FPGA based systems with minor modifications.

### 2.1. System

A typical SoC contains a processor and peripherals implemented on the FPGA. This can be augmented to monitor temperature by adding the following components:

- An array of temperature sensors. Section 2.2 describes how temperature sensors can be implemented on the FPGA fabric.
- A controller that monitors the temperatures readings from the different sensors and activates the DTM mechanism when appropriate. The controller is described in Section 2.3.

For experimental purposes, the FPGA core was powered by an Agilent E3631A DC power supply. This lets us measure the power consumed by the logic on the FPGA core.

### 2.2. Temperature Sensor

Our temperature sensor is based on the design proposed by [7, 8]. This utilizes the fact that the transistor switching speed is directly proportional to the temperature. In such a sensor, a ring oscillator is implemented on the die (the FPGA fabric in this instance), and the frequency of oscillation is used as a proxy for the temperature. We calibrated the sensor by measuring the ring oscillator frequencies at various temperatures and found that a change of 380 kHz in ring oscillator frequency to correspond to a degree change in temperature. For detailed explanation regarding the design of the ring oscillator and our calibration tests, please refer to [16].

In a notable deviation with previous results published in [8], we also see a stronger dependence of the frequency with voltage. We attribute this to shrinking supply voltages - our experiments are on a Virtex-2 Pro device with a core supply voltage of 1.3V, while the previous generation of FPGAs used a core voltage of 1.8 Volts or 2.5 Volts. As the core supply voltage keeps getting lower, the impact of supply voltage noise on the sensor frequency increases. Hence special care has to be taken while measuring the temperature using such a sensor in the presence of voltage variation or noise. Further investigation of the impact and mitigation of voltage noise is warranted.

In our experiments we found each sensor to consume approximately 1 mW of power which is less than 1% of the power consumed by the FPGA core. To keep the power consumed by the sensors low, and to avoid local heating, they are enabled only for a short time. The sensor's capture counter has enough precision to store a change of less than 0.1 °C at this measurement period.

### 2.3. Controller

The controller determines when the sensors have to be enabled, and performs the steps required to enable the sensors and read back the values from the sensors. Once the controller knows the temperatures at various locations on the die, it can decide to activate some thermal management technique if needed.

In our implementation, we use the embedded PowerPC processor present in the Virtex-2 Pro device as a controller. The sensors are connected to the processor via the On-chip Peripheral Bus (OPB) bus [10]. Each of the sensors interface to the bus using the General Purpose I/O (GPIO) core. This provides a simple interface for the processor to set or reset signals directly in the sensor.

To obtain a reading from the sensor, the controller initially resets all the counters present in the sensor. The sensor is then enabled thereby activating the ring oscillator for *Enable Time + Capture Time*. The Capture Counter is enabled after *Enable time* has elapsed. Both the ring oscillator and the counter are disabled after capture time. The value from the capture counter is then read back and the frequency value is mapped to temperature based on data from initial calibrations.

## 3. Comparison with HotSpot

In this section, we compare the temperatures obtained using simulation with *HotSpot* – an architectural level thermal simulator [6]. *HotSpot* has been validated against a test chip and against finite element simulation [11]. Hence such a comparison serves as a cross-validation of both *HotSpot* and the sensor architecture.

*HotSpot* takes in a number of parameters that affect the thermal properties of the system. These include the properties of the silicon, thermal interface material, heat spreader, and heat sink. Table 1 lists all the configuration changes we made to *HotSpot* to reflect the setup of the FPGA based system used in the experiments.

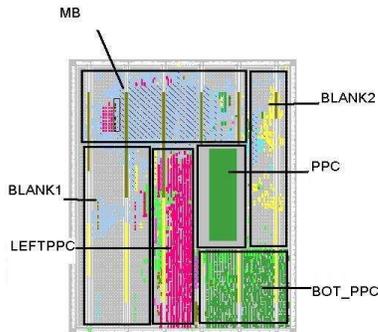
To compare *HotSpot* with our system, we create a design with 6 distinct areas listed in Table 2. The area marked *PPC* contains the PowerPC processor and *MB* contains the MicroBlaze processor. The two areas marked *left\_ppc* and *bott\_ppc* contain miscellaneous glue logic necessary for measuring the temperature, and logic necessary for clocking and communication from the system respectively. The power dissipation in each of the units is obtained as follows. The current supply to the FPGA core is provided through an Agilent E3631A constant power supply source that indicates the current to the core at each instant. This gives us the total power dissipation by the FPGA core. From the Virtex-II Pro datasheet [5], we know that the PowerPC core itself consumes 0.9 mW/MHz, and since it is being run at 50 Mhz, the average power dissipation is 45 mW. For the MicroBlaze core power, we created a design with a MicroB-

Parameter	Value	Comments
t <sub>chip</sub>	0.8 mm	Thickness of the chip
r <sub>convec</sub>	13.9 K/W	Convection resistance, from [12]
s <sub>sink</sub>	0.011 m	Heatsink width, set to simulate no heatsink
t <sub>sink</sub>	0.1 mm	Heatsink thickness, set to simulate no heatsink
t <sub>interface</sub>	0.025 mm	Thickness of interface material
die size	6.57 mm x 7.04 mm	Scaled to 130 nm technology from the value in [14]

**Table 1. HotSpot parameters to match the test system**

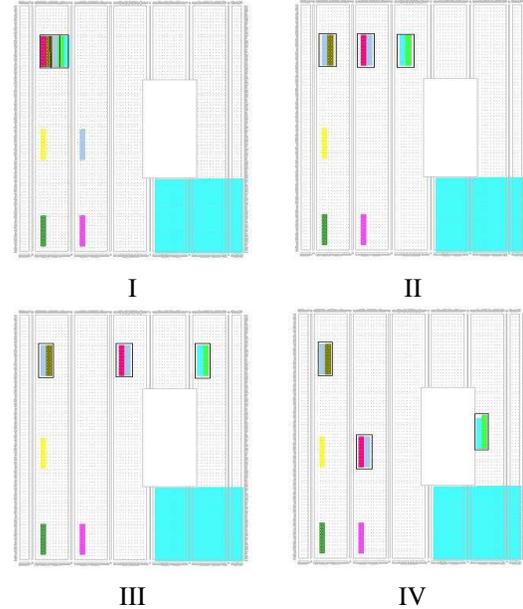
laze core and measured the current when the core is active versus when it is clock gated. The difference gives the average power dissipation in the MicroBlaze core. For the two blank areas, we assign only the static power dissipation as obtained using the Xilinx Power Estimator (XPower) [15]. The remaining power consumed by the core is split proportional to the areas of the two remaining blocks.

Using the floorplan shown in Figure 1, we compare the outputs obtained using *HotSpot* with the specified parameters and the values obtained from the sensors. For each unit, the sensor is placed as close to the center as possible. For PowerPC the sensor is placed at the edge of the PowerPC block. The results, measured as deviation from ambient temperature (23 °C) are recorded in Table 2.



**Figure 1. Floorplan used for comparison with HotSpot**

We see that on average the difference in temperatures predicted by *HotSpot* and those obtained from the sensors differ by less than 0.2 °C. The temperature differentials from the ambient are low because of the lower power density of in each of the units. When a circuit is implemented on an FPGA, the place and route tools tend to place only the critical logic close together and spread out other logic. Further, an FPGA implementation is also sparse since majority of the interconnect is unused. As a result is hard to obtain higher power densities. In our experiments, we noticed that the highest power density was obtained in the MicroBlaze core which was written using hand-placed structural VHDL.



**Figure 2. Placement of 3 hot units: (I) - close, (II) - medium, (III) - large, (IV) - distant. The 3 units are shown bounded with a black box.**

#### 4. Granularity of Temperature Variations

Power density levels across a chip vary significantly. This is because units like the register file and the issue queues in a superscalar processor, are small in area, but are accessed frequently. This results in local hotspots across the chip.

In this section, we find out experimentally if local hotspots can be reduced by spreading the high power density units over a broader area. Our objective is to find if small hotspots can be mitigated if they are split and spread among other cold units.

Figure 2 shows the four different configurations for the placement of a single high power density unit. In the first configuration, the block is packed together within a narrow region. In the remaining three configurations, the block is split into three sub-blocks, and spread out over varying distances. In all cases, the total power consumed by the blocks

Unit	Power (mW)	Sensor Temperature	HotSpot Temperature
blank1	0.1	3.4	3.37
left_ppc	75	3.5	3.69
bott_ppc	75	3.4	3.67
ppc	45	3.5	3.66
mb	313	4.1	3.96
blank2	0.1	3.4	3.38

**Table 2. Temperature readings obtained from HotSpot and FPGA sensors**

is identical, and the power density across each block is also identical.

We repeat the same experiment varying the size of the block. The results are tabulated in Tables 3 and 4.

block size	Config I	Config II	Config III	Config IV
2 slices	2.81	2.71	2.66	2.58
4 slices	4.15	3.40	3.06	3.02
8 slices	6.82	6.86	6.62	6.9

**Table 3. Average temperature differential from ambient across various methods of splitting the hot unit**

block size	Config I	Config II	Config III	Config IV
2 slices	3.02	2.87	2.81	2.71
4 slices	4.46	3.67	3.26	3.21
8 slices	7.21	7.24	6.96	7.1

**Table 4. Maximum temperature differential from ambient across various methods of splitting the hot unit**

In Tables 3 and 4 the temperature levels are different since the total power consumed is different. However within each row, the total power density is maintained constant.

We can note that when the hot unit is of size 12 slices, splitting it and spreading it around into 3 blocks of size 4 slices each contributes to a much higher temperature reduction, than if the block size is 8 slices. When the sub block size is just 2 slices, there are no significant hotspots that can benefit from spreading the heat around. These effects of spreading are predicted by *HotSpot* and also by simple thermodynamics. This experiment proves that we can measure such effects.

## 5. Conclusion

In this paper, we have developed an FPGA thermal monitoring technique. The technique is configurable such that temperature sensors can be placed anywhere on the system and be easily accessed by a controller. We have shown that the temperatures obtained from the sensors correlate well with temperatures predicted by *HotSpot*. This serves

as a cross-validation for both *HotSpot* and the sensor architecture. Finally, we have experimentally shown that local hotspots can be avoided by splitting up a large hot region into multiple smaller units and interspersing them along with cooler units.

## 6. Acknowledgements

This work is supported in part by NSF grants CCR-0103364 (CA-REER), CCF-0429765, Army Research Office grant W911NF-04-1-0288, a Univ. of Virginia FEST Award, two research grants from Intel MRL, and an IBM Faculty Partnership Award. We would like to thank Tim Tuan from Xilinx Labs and other members of the *comp.arch.fpga* community for their help. We would also like to thank Russ Joseph and the anonymous reviewers for their insightful comments.

## References

- [1] The International Technology Roadmap For Semiconductors (ITRS), 2003.
- [2] S. Heo, K. Barr and K. Asanovic. Reducing Power density through Activity Migration. In *Proc. of ISLPED*, Aug 2003.
- [3] S. Borkar. Design challenges of Technology Scaling. In *IEEE Micro*, Jul-Aug 1999.
- [4] Insight Memec board documentation. <http://www.insight.memec.com>
- [5] Xilinx Virtex-2 Pro User Guide. <http://direct.xilinx.com/bvdocs/publications/ds083.pdf>
- [6] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *Proc. ISCA-30*, June 2003.
- [7] S. Lopez-Buedo and E. Boemo. Making Visible the Thermal Behavior of Embedded Microprocessors on FPGAs - A Progress Report. In *Proc. FPGA*, Feb 2004.
- [8] S. Lopez-Buedo, P. Pernas and E. Boemo. Thermal Testing on Reconfigurable Computers. In *Proc. of IEEE Design and Test of Computers*, Jan-Mar 2000.
- [9] Xilinx FPGA Editor. <http://toolbox.xilinx.com/docsan/xilinx6/books/manuals.pdf>
- [10] IBM On Chip Peripheral Bus (OPB). <http://www-03.ibm.com/chips/products/coreconnect/>
- [11] W. Huang, M. Stan, K. Skadron, K. Sankaranarayanan and S. Ghosh. Compact Thermal Modeling for Temperature-Aware Design. In *Proc. of DAC*, June 2004.
- [12] Device Packaging and Thermal Characteristics. <http://direct.xilinx.com/bvdocs/userguides/ug112.pdf>
- [13] D. Brooks and M. Martonosi. Dynamic Thermal Management for High-Performance Microprocessors. In *Proc. of HPCA*, 2001.
- [14] C. Yui, G. Swift and C. Carmichael. Single Event Susceptibility Testing of the Xilinx Virtex II FPGA. In *Proc. of MAPLD*, 2001.
- [15] Xilinx XPower, <http://www.xilinx.com/xpower>
- [16] S. Velusamy, W. Huang, J. Lach, M. Stan and K. Skadron. Monitoring Temperature in FPGA based SoCs. *University of Virginia Technical Report, CS-2004-39*.