

Thermal Characterization and Optimization in Platform FPGAs*

Priya Sundararajan, Aman Gayasen, N. Vijaykrishnan, T. Tuan
{psundara,gayasen,vijay}@cse.psu.edu, tim.tuan@xilinx.com

ABSTRACT

Increasing power densities in Field Programmable Gate Arrays (FPGAs) have made them susceptible to thermal problems. The advent of platform FPGAs has further exacerbated the problems by increasing the power density variations on the FPGA fabric. Therefore, we need to characterize the die temperature of platform FPGAs. In this paper, we first estimate the temperature distribution within a Virtex-4 FPGA by feeding the block power numbers in an architecture-level temperature simulator calibrated to reflect a real FPGA package. We analyze the impact of different hard-wired blocks on the temperature profile, and observe that they introduce intra-die variation in temperature of up to 20°C. Next, we evaluate the influence of placement on temperature. Our experiments indicate a decrease in peak temperature by changing the placement of hard blocks, especially the high-speed transceivers. We further propose an iterative placement technique to reduce the peak temperature, and apply it on real designs. Finally, we propose alternate organizations of the hard blocks in the FPGA fabric to reduce temperature.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aid—Placement and Routing;
B.8.0 [Performance and Reliability]: General

General Terms

Algorithms, thermal floorplan

Keywords

Platform FPGAs, placement, Virtex4, temperature, thermal

1. INTRODUCTION

Temperature is a growing concern in most integrated circuits. Improvements in fabrication technology, circuit design, architecture, and tools, have all contributed towards an increase in logic density as well as clock frequency. Increasing logic density and performance has in turn led to an increase in power densities, which manifests itself in the form of high temperatures. FPGAs are following a similar trend. Recent articles on thermal management from the leading FPGA manufacturers ([13, 2]) clearly indicate the growing importance of thermal issues in FPGA designs.

Die temperature must be controlled because it impacts the timing, leakage power, package design, and lifetime of the device. Circuits run slower when they are hot, and their lifetime reduces exponentially with

*This work was supported in part by grants from GSRC, NSF Career 0093085 and 0202007

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCAD'06, November 5-9, 2006, San Jose, CA

Copyright 2006 ACM 1-59593-389-1/06/0011 ...\$5.00.

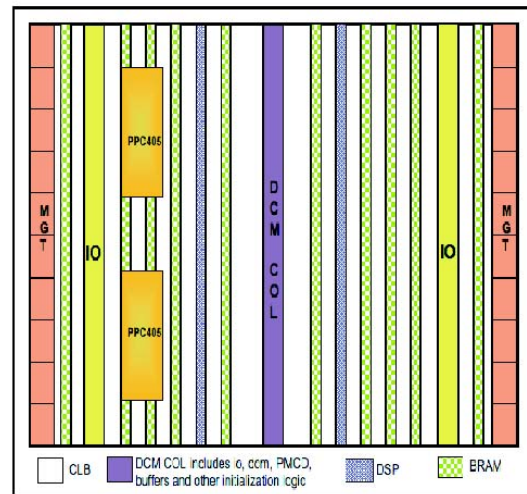


Figure 1: Virtex-4 FX100 device (not to scale)

increasing temperature. Furthermore, leakage power increases exponentially with temperature, which can cause a thermal runaway. All these factors have forced chip manufacturers to employ techniques to control the die temperature. These techniques can be divided into two categories, namely package level, and design level. Heat sinks, spreaders, and fans are the most common examples of package level techniques. These solutions try to efficiently *remove* the heat generated by the design. In contrast, design level techniques normally try to *reduce* the heat generated by the design.

The effect of thermal-aware placement or floorplan on an FPGA remains unexplored. The key reason for this is an absence of hotspots in the traditional FPGA fabric, which consisted of only identical CLBs (Configurable Logic Blocks). This is rapidly changing. FPGAs now act as platforms to build complete systems on chips. The most advanced FPGAs today contain several types of embedded circuit blocks, including high speed transceivers, multipliers, DLLs, and memories [16, 1]. The varying power densities across these blocks could lead to hotspots. The key objective of this work is to characterize the temperature distribution in an FPGA including these *hard* blocks, and observe the effect of placement on temperature.

In this paper, we first show the temperature distribution in an FPGA that has only CLBs. Next, we show how this temperature profile changes with the addition of different kinds of hard blocks. Taking the Virtex-4 FPGA as an example, we particularly analyze those configurations that are found in Virtex-4 devices. Our results indicate that the high speed transceivers (MGTs), PMCDs (Phase-Matched Clock Divider), and DCMs (Digital Clock Managers) are significantly hotter than the rest of the fabric. After this characterization step, we evaluate the effect of placement on the peak temperature and observed that placement could reduce the temperature by up to 5.5%. We further propose an

Table 1: Power densities in 4VFX100 (Freq:500MHz)

Block Type		Power density (normalized to CLB)
DSP		0.78
CLB		1.00
PPC		1.32
IOB		2.33
BRAM	Dual Port	3.85
	Single Port	1.93
MGT	Transceiver	7.75
	Transmitter	4.22
	Receiver	4.11
PMCD		11.4
DCM	High Freq	11.46
	Low Freq	9.84

iterative thermal-driven placement approach and apply that to real designs. Finally, we suggest some alternate FPGA fabric organizations to reduce the temperature variations within the die.

2. RELATED WORK

Package designers have been considering thermal issues for a long time. Instead of considering variations in the temperatures on the die, they design the package to support the worst case specifications of the design. They typically provide the user with the thermal resistance (θ_{JA}) of the package, which is used to estimate the junction temperature (T_J) using

$$T_J = T_A + \theta_{JA} * Power, \quad (1)$$

where T_A is the ambient temperature, and $Power$ refers to the total power consumed by the chip.

As designing the package for the worst case junction temperature started becoming too expensive, researchers started looking at design level solutions to reduce the temperature. Commonly used techniques include Dynamic thermal management (DTM) [3] (e.g clock gating and voltage and frequency scaling) and Thermal-aware floorplanning ([8, 7],[5]).

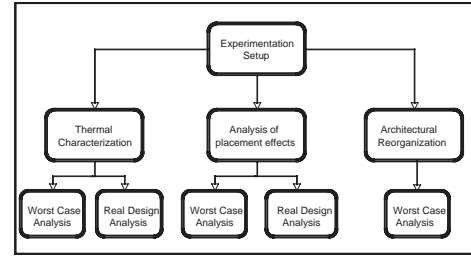
On the modeling front, several researchers have developed tools for estimating the die temperature (E.g. Hotspot ([11]), HS3D). HS3d [9] is a architecture level tool that performs only steady state temperature estimation, but is orders of magnitude faster than HotSpot. Since in this work we look at only steady state temperatures, we use HS3d.

Thermal issues in FPGAs are relatively unexplored. Some researchers have proposed the use of distributed sensors for monitoring temperatures in FPGAs [4, 14]. They, however, considered only CLBs in the fabric, and consequently, observed very little temperature variations across the die. [17] used the frequency of ring oscillators(distributed across FPGA fabric) to obtain real time thermal status of complex circuits implemented on the FPGA. This technique is useful for thermal monitoring at the board level. However, all testing was done using a Virtex board which had just CLBs. Our work is the first to thermally characterize a real platform FPGA.

3. TARGET FPGA

In this work we target advanced platform FPGAs from Xilinx and Altera [16, 1]. Platform FPGAs typically provide one or more embedded processors, several hard blocks (reconfigurable blocks with fixed functionality, such as multipliers), and abundant programmable logic. Figure 1 shows the organization of a Virtex-4 (or V4) FPGA device (XC4VFX100), which represents the most advanced FPGAs from Xilinx. This FPGA has 20 high speed (Multi Giga-Bit) transceivers (MGTs), 160 DSP blocks (perform multiply-accumulate functions), 6.768 Mbits of RAM (BRAM), 12 DCMs, 2 PowerPC cores (PPC), and 42,176 slices of programmable logic. All these blocks can operate at up to 500 MHz (except PPC, which operates at 450 MHz).

Table 1 shows the power density numbers for the different types of blocks in Virtex-4, all normalized to the power density of a CLB tile. Power densities were calculated using the power numbers (obtained using the Virtex-4 power spreadsheet [15], assuming 12.5% toggle rates)

**Figure 2: Experimental Setup**

and dimensions of each block. Note that the power densities of DCMs and PMCDs are more than ten times that of the CLB. The MGT blocks also exhibit a high power density. DSP blocks, surprisingly, has low power density. BRAM power depends heavily on its configuration. The dual-port RAM consumes almost double the power of a single-port RAM. In our experiments, we configured BRAM as a dual-port RAM, with 50% write rate. The DCMs were operating in high-frequency mode. MGTs were configured as transceivers at 10.3Mbits/sec.

4. EXPERIMENTAL SETUP

We conducted three kinds of experiments (see Figure 2): thermal characterization, thermal placement, and thermal organization. While the first set focused on characterizing the temperature distribution in the FPGA, the other two explored ways to reduce the peak temperature. The next three sections will discuss the three sets of experiments in detail. Here, we explain the tools used (and developed) for these experiments.

Power Estimation

The method to estimate the power consumption varied based on the goal of the experiment. In cases where we wanted to model the worst-case scenario, we used the power numbers from the Virtex-4 power spreadsheet [15] (refer Section 3). In order to see the typical thermal profiles, we also experimented with several real designs. These designs were synthesized and implemented using Xilinx ISE 8.1i tools. We used the probabilistic flow of XPower to estimate their power consumption. In this mode, XPower estimates the switching frequencies of the nets in the design by propagating the switching information through the logic in the design. However, we observed that this propagation has several limitations, for example, it does not propagate the switching probabilities across flip flops (FFs). Therefore, we wrote a tool (*updateXml*) to augment the switching propagation in XPower. The output of *updateXml* is an XML file containing the switching frequencies of all the nets. XPower accepts this XML file as an input, and uses the switching frequencies in the file to calculate power consumption.

Temperature Estimation

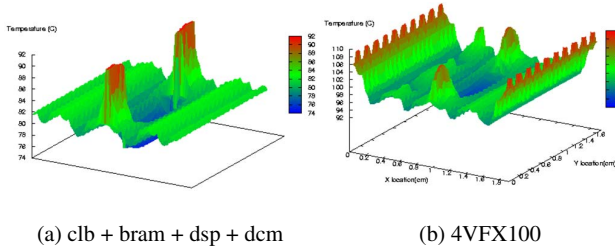
Both XPower and the power spreadsheet estimate the die (also called junction) temperature. They model the entire chip as one block, and use equation 1 to calculate the temperature. This model, however, assumes that power consumption is uniform across the die. This is true in the case of traditional FPGAs which has CLB based architectures, but with the introduction of platform FPGAs this scenario has changed. Since our interest lies in observing (and reducing) the temperature variations within the die, we cannot use any of these tools.

HS3d provides the flexibility to set several package and die parameters, such as the spreader thickness, package-to-air thermal resistance (r_{convec}), and substrate thickness. We first calibrated these parameters such that the temperature estimate is close to that from the power spreadsheet (for the same power numbers). The Silicon substrate was 500 μ m thick.

HS3d takes the floorplan of the device, and the distribution of power consumption within the die to generate the thermal profile. We created the floorplan from the layout of 4VFX100. While using the power

Table 2: Temperatures for different configurations

S.No.	Config	All blocks consume max power					Varying power numbers					Hotspot
		Temperature (°C)				%	Temperature (°C)				%	
		Peak	Avg	Min	Variation		Peak	Avg	Min	Variation		
1	clb only	40.51	40.49	40.45	0.053	0.13	33.19	32.79	32.40	0.79	2.38	clb
2	clb+bram+dsp	78.64	76.40	72.95	5.69	7.23	53.41	50.82	48.75	4.66	8.73	bram
3	clb+bram+dsp+dcn	90.67	81.63	75.42	15.24	16.81	56.31	52.50	49.47	6.83	12.14	dcn
4	clb+bram+dsp+mgt	102.95	93.48	87.88	15.07	14.63	71.28	61.39	57.90	13.38	18.77	mgt
5	clb+bram+dsp+mgt+dcn	107.77	98.54	90.33	17.44	16.18	70.60	62.44	58.22	12.37	17.52	mgt
6	clb+bram+dsp+ppc+dcn	89.32	79.78	74.07	15.24	17.07	60.74	53.22	50.20	10.54	17.35	dcn
7	clb+bram+dsp+ppc+mgt	100.77	90.59	84.54	16.23	16.10	67.69	57.38	54.34	13.35	19.72	mgt
8	clb+bram+dsp+ppc+mgt+dcn	106.44	96.69	88.99	17.44	16.39	69.93	60.06	56.22	13.70	19.60	mgt
9	clb+bram+dsp+ppc+mgt+dcn+io	109.20	99.29	93.11	16.09	14.73	70.92	61.94	58.82	12.10	17.06	mgt

**Figure 3: Thermal profiles for various configurations**

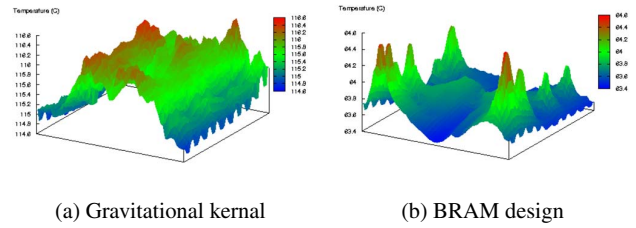
spreadsheet to obtain the power numbers, it is easy to estimate the power distribution because the spreadsheet merges the routing power for a tile with the logic power in every tile. However, it is not as easy while using XPower, as it reports logic and signal powers separately. Since we used XPower to estimate the power for real designs, we developed a utility, *updatePwr* to estimate the power distribution from XPower report. It picks power numbers from the XPower power report, and uses placement and routing information from Xilinx XDL file to distribute the power within the die. The XPower power report classifies power distribution into Logic and Signal power. For logic power, *updatePwr* parses the XDL to locate a logic block and adds the power consumed by that logic block to the tile where it is located. Note that *updatePwr* converts the logical coordinates in XDL to the physical location. For signals, *updatePwr* parses the XDL file to extract the pips (programmable interconnect points) used by every signal. It then distributes the signal power among the pips by assigning higher power to pips that drive longer routing segments. A constant ratio is assumed between the power of two resource types that toggle at the same rate. These ratios reflect the typical power scenarios.

5. THERMAL CHARACTERIZATION

In order to observe the effects of different types of blocks on the temperature distribution, we start with an FPGA containing only CLBs, and then progressively add the hard blocks. We do not include IO blocks when evaluating the effect of different hard blocks, because the power of IO blocks depends heavily on the standard used. However, as a final floorplan, which accurately models a Virtex-4 FX100 device, we include the IOBs with LVCMOS standard. We use this complete floorplan for placement and floorplanning experiments.

We keep r_{convec} at 1.0 K/W, which is the same as [12]. This r_{convec} reflects a package with a moderate heat sink, and gives peak temperatures around 100°C when all blocks on the FPGA are consuming power. The ambient temperature is 25°C for all.

The first half of Table 2 summarizes the various configurations we experimented with. For every configuration, all the blocks in the FPGA are used, and they all consume their maximum powers (as per the Virtex-4 power spreadsheet). We model 100% utilization case for this table because this allows us to observe the intrinsic variation in temperature caused by just the differences in power densities and location of the blocks, masking any design dependent differences. Here, since all CLBs are consuming the same power the temperature remains uniform

**Figure 4: Temperature profiles for real designs. $r_{convec} = 9.8$.**

throughout the die. The slight variations in temperature occur because the CLBs at the periphery see different thermal resistance compared with those at the center.

The second row in Table 2 shows the effect of adding BRAM and DSP blocks to the CLB-only fabric. Notice that the peak temperature increases to 78.65°C, mainly because of the much hotter BRAM blocks (see Table 1).

DCMs are even hotter than BRAMs, and all of them are clustered in the same column. Hence, they cause the temperature to climb much higher, to 90.67°C. Since the CLBs are still relatively cool, temperature varies by more than 16% across the die. Figure 3(a) shows the temperature profile.

MGTs are the second hottest blocks, after DCMs, but they affect the temperature variation slightly differently because they are placed at both the ends of the fabric and are larger in number. Since table 2 models the 100% utilization case, the MGTs heat the two ends of the fabric so severely that the minimum temperature in the CLBs also rises. Hence, we observe that temperature variation reduces to 14.64%, compared to 16.81% for the DCM case.

Adding both DCMs and MGTs to the fabric further raises the temperature to 107°C. The hotspot in this case occurs close to the DCMs, since they have the highest power densities.

The PowerPC (PPC) is relatively cool. Hence, adding the two PPC cores to the fabric reduces the peak temperature slightly. The final row in Table 2 shows the results for the complete 4VFX100 device. The hotspot occurs near the MGTs and DCMs (see Figure 3(b)).

After the worst case scenario, we went on to investigate a more common case, one in which power also differs among the blocks of the same type. In order to model this, without being restricted to the features of any specific design, we varied the power numbers for every block between zero and the maximum (for that block type) randomly (using Perl's *rand()* function). The right half of Table 2 summarizes the results for this setup. We observe that the peak temperatures for all configurations have reduced (compared with those for max power), but the percent variation in temperature for the 4VFX100 device has increased from 14.73% to 17.06%.

The resource utilization of a design can cause its temperature profile to deviate from the expected. Figure 4 shows two such examples. Considering the resource utilization for the gravitational kernel design (refer Table 3), from Table 1, we would expect the IOBs to be the hotspots. Surprisingly, the peak temperature lies in the middle of

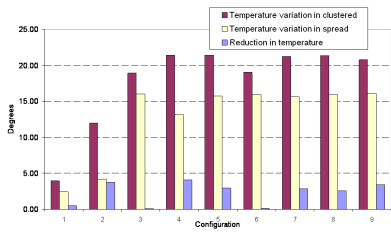


Figure 5: Effect of spreading the blocks. X-axis label is the S.No. in Table 2.

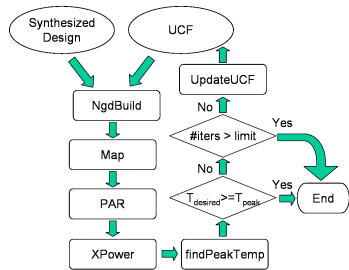


Figure 6: Iterative thermal placement.

the fabric at a CLB. This is because, while almost all the CLBs are used, only about 10% of the IOBs are occupied. Although one CLB does not consume too much power, the combined power density of a clustered group of CLBs is larger than the power density at a group of IO tiles. Another interesting case is the FFT-coregen design (see Figure 4(b)), which uses many BRAM blocks. However, the enable rate of these BRAMs is low, and hence, contrary to our expectations, they do not form hotspots. Instead, the IOBs are the hottest blocks in this design. For the Aurora design, the thermal profile shows high peaks at the MGTs (see Figure 7) as expected. Previous studies have shown that the typical utilization of an FPGA is only about 60% [6, 10]. Hence, to model the typical (instead of worst) case, we reduced the utilization of the fabric to 60% for all the configurations, and again estimated the temperatures for two different placements: first with all the blocks clustered together, and second with the blocks spread out uniformly. All blocks utilized consume maximum power. Figure 5 plots the temperature variations for the two placements for various configurations. It also shows the difference between the peak temperatures for the two cases. Note that spreading out the blocks reduces the temperatures for all configurations. However, when the fabric contains only CLBs, the difference between the two placements is only about 0.5°C. In contrast, for one of the configurations, it is more than 4°C. This motivates us to develop a thermal-driven placement algorithm, explained in the next section.

The above analysis illustrates the thermal issues that arise because power consumption varies among the different block types. A large intra-die temperature variation is undesirable because the lifetime of the chip is determined by the hottest block (since lifetime reduces exponentially with temperature). Furthermore, the increase in circuit delay with temperature can cause timing failures (and possibly malfunction). Considering the exponential dependence of leakage and lifetime on temperature, even small reductions in the peak temperature can be crucial.

6. THERMAL-AWARE PLACEMENT

The previous section showed that changing the placement of blocks can affect the peak temperature. To experiment with real designs, we propose an iterative thermal placement technique.

Iterative Thermal Placement

The key motivation for developing this placement technique was its practicality. Our intention was to use existing tools to experiment with

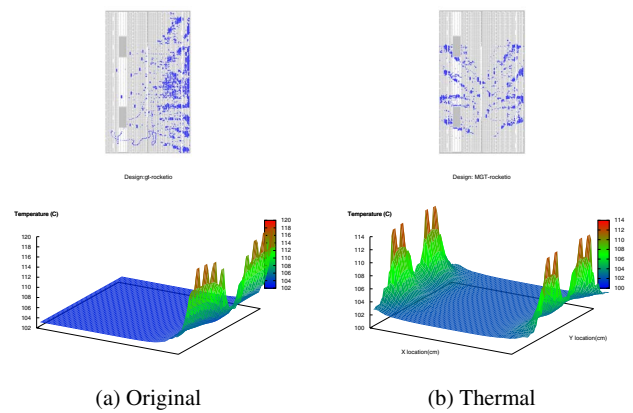


Figure 7: Effect of MGT placement in a real design. $r_convec = 9.8$

real designs and demonstrate the immediate benefits that we can obtain. We also made sure that our placement changes did not have any repercussions on timing and routability.

Figure 6 shows the iterative thermal placement technique. The flow starts with a synthesized netlist that is run through Xilinx implementation tools (*Ngdbuild*, *Map*, *PAR*). After every run of *PAR*, power is estimated using *XPower* and *updateXml* (as described in Section 4). Power numbers obtained from *XPower* are distributed across an empty 4VFX100 floorplan using *updatePwr*. The updated floorplan is then fed to *HS3d* to obtain the peak temperature. If the peak temperature is above the desired value, we add placement constraints to reduce hotspots, and re-run the implementation tools. The maximum number of iterations for this flow can be fixed by the user.

The placement constraints we introduce depend on which resource type has the peak temperature. Once a constraint is introduced it is carried over to all subsequent iterations. In the case of a CLB, we introduce location constraints (*LOC*) to retain placement of logic within the hot CLB and we prohibit placement (*PROHIBIT*) of logic in the surrounding slices. These empty slices would isolate the hotspots and allow the heat spreader to reduce the peak temperature in the subsequent runs. In case an IOB is the hottest block, we move the flip flop from the IOB tile to the slice fabric to reduce logic power consumed by IO tile. In addition to this, except for the hot IO tiles, we introduce location constraints to retain IO placement. Placement is prohibited in the hot IO tiles. These constraints would ensure that in the next iteration the placer retains placement for all IO sites except for the IOs that were hotspots. The logic previously placed in the now prohibited tiles need to be placed in a new site which, based on its location, would have an effect on the peak temperature of the design. The IO placements are restricted by the requirements of the board layout. However, with this algorithm we can determine which IOs would need to be moved and how much benefit we can get in terms of temperature reduction. For black boxes like *DSP*, *MGT*, *BRAM* the constraints we introduce are similar to that for a CLB. If the hotspot occurs at a hard block, we fix it to its existing location and prohibit its surrounding sites (both CLB and hard block tiles). The *DCM* and *PMCD* blocks were distributed within the *DCM* column to reduce peak temperature.

Results of Thermal Placement

We experimented with a suite of 5 real designs, most of them were generated using Xilinx tools. For our experiments we used an r_convec value of 9.8([16]) and an ambient temperature of 45°C. For the first design in Table 3, *triple_des*, the variation in temperature across the die is 4.3%. This is because most of the device is unused. Since the hotspots in this design are the IOs, the IO placement is changed to obtain the decrease in peak temperature by 2.2°C.

All remaining designs in Table 3 were created with Xilinx Coregen. The second design, *gravitational_kernel*, has a variation in temperature of 1.5% across die. The peak temperature in this case is on the

Table 3: Effect of placement on temperature of real designs

Design	Resource Utilization	Fmax (MHz)	Peak Temperature (°C)		Hotspot
			Before	After	
triple_des	Slice: 27%; IOB: 52%	206	79.22	77.05	IOB
gravitational kernal	Slice: 87%; IOB: 10%; DSP: 77%;	200	116.49	117.71	CLB
fft_coregen	Slice: 33%; IOB: 90%; DSP: 52%; BRAM: 94%	100	64.56	64.25	IOB
cordic	Slice: 1%; IOB: 20%; DCM: 8%; PMCD: 12%	208	52	51.18	DCM
MGT-aurora	Slice: 6%; IOB: 3%; MGT: 40%	149.1	119.12	113.7	MGT

Table 4: Effect of change in FPGA organization

Organization	DCM	Peak Temp	Avg Temp	Min Temp
original	yes	109.20	99.29	93.11
original	no	104.04	93.71	89.51
IOB \leftrightarrow MGT	yes	107.05	99.20	92.80
IOB \leftrightarrow MGT	no	99.76	93.62	89.33
MGT + DSP	yes	107.70	98.50	92.90
MGT + DSP	no	97.30	92.90	89.45
PPC moved	yes	107.70	98.50	92.90
PPC moved	no	97.30	92.90	89.41
DCM + IOB	yes	103.05	98.50	92.89

CLB blocks instead of the IO blocks, in spite of the IO blocks having higher power density. Placement changes did not have an impact on the peak temperature. This is because the hotspots kept moving across the fabric but did not have an impact on temperature. This is an expected behavior as this design falls into the category of a traditional FPGA.

The third design, *fft-coregen* has a temperature variation of 1.7%. The hotspot in this case is an IOB and not the Block RAM because the toggle rate of the Block RAM is low.

The fourth design, *cordic*, is a relatively small design. Temperature variation across die is 3.7%. Not placing the DCM block and PMCD block close to each other reduced the peak temperature by 0.82°C.

The fifth design, *MGT-aurora*, has a temperature variation of 13.3% across the die. In this case, our placement changes reduce the peak temperature by 5.58°C. Figure 7 shows the change in temperature as MGT placement is changed for the MGT-aurora design, which uses 8 MGTs. In Figure 7 (a), all the MGTs are placed in one column on the FPGA. This is the default placement by the ISE 8.1i tools for the design created by Xilinx Coregen. Figure 7 (b) instead spreads out the MGTs in the two MGT columns present on the FPGA. This reduces the peak temperature by nearly 5°C, without violating the timing constraints specified by Coregen. Hence, we conclude that MGT placement has a significant impact on peak temperature.

All the five designs discussed above were able to meet timing in spite of the placement constraints introduced.

7. THERMAL-AWARE FPGA ORGANIZATION

In the final set of experiments, we propose alternate FPGA organizations to reduce the temperature variations within the die. We demonstrate their effectiveness by showing the die temperature for a 100% utilized FPGA. Note that since all blocks are used, and they all consume near-maximum power, thermal-aware placement does not reduce the temperature in this case. However, even in this worst case scenario, some changes in the FPGA organization reduce temperature.

Table 4 shows the temperatures for different FPGA organizations. The changes are progressive, i.e., row 3 includes the changes of row 2, and so on. The first step towards reducing temperature was moving the

MGTs. As a first step, we interchange the MGT and IOB columns, and observe that the peak temperature reduces by 2.15°C. This happens because the Silicon can better spread the heat from the MGTs when they are placed near the center. This change in floorplan can be implemented easily with the column-based modular architecture of Virtex-4 (ASMBL) [16].

Since DSP blocks are relatively cool, the next organization mixes the MGTs with DSPs in one column. This increases the width of the chip as the MGTs are wider than the DSP blocks. Now, one DSP-MGT column contains 5 MGTs and 10 DSPs. In the case when DCMs are used, the hotspot occurs at DCMs, and therefore, we do not see any reduction in peak temperature. However, when DCMs are not used, MGTs are the hotspots. Hence, in that case, the peak temperature reduces by 2.46°C.

Next, we move one PPC to the other horizontal half of the device. This, however, does not change the peak temperature (because DCMs, which reside at the center of the fabric, remain hotspots).

Finally, we distribute the DCM blocks into three columns, mixing them with IO blocks. This is a reasonable choice because even the current architecture mixes DCMs with IOBs in a column. This reduced the peak temperature to 103.05°C, which is 6.15°C below the original temperature.

8. CONCLUSION

Our work demonstrates temperature variations of up to 20°C within the die of a 90nm platform FPGA. We observed that the high speed transceivers and DCMs are the hottest blocks on the FPGA. We further proposed an iterative placement technique to reduce the peak temperature, and demonstrated that the peak temperature reduced by 5.5°C after the change in placement of MGTs in a real design. A change in placement, however, is incapable to reduce the peak temperature if most of the MGTs (or DCMs) are used. Therefore, we developed alternate organizations of the hard blocks to reduce the peak temperature by almost 6.15°C even when all resources are utilized.

9. REFERENCES

- [1] "Altera Products Documentation," <http://www.altera.com/literature>.
- [2] "Thermal Management for 90-nm FPGAs," Application Note 358, Altera Corporation.
- [3] D. Brooks and M. Martonosi, "Dynamic Thermal Management for High-Performance Microprocessors," In Proceedings of the 7th International Symposium on High-Performance Computer Architecture, 2001.
- [4] S. Lopez-Buedo, J. Garrido, and E. Boemo, "Dynamically Inserting, Operating, and Eliminating Thermal Sensors of FPGA-based Systems," IEEE Transactions on Components and Packaging Technologies (CPM), Vol.25, No.4, pp.561-566, December 2002.
- [5] G. Chen and S. Sapatnekar, "Partition-driven standard cell thermal placement," In Proceedings of the International Symposium on Physical Design, 2003.
- [6] A. Gayasen et al., "Reducing leakage energy in FPGAs using region-constrained placement," In Proceedings of International Symposium on Field-programmable gate arrays, 2004.
- [7] Y. Han, I. Koren, and C. A. Moritz, "Temperature aware floorplanning," in Second Workshop on Temperature-Aware Computer Systems (TACS-2), held in conjunction with ISCA-32, June 2005.
- [8] K. Sankaranarayanan, S. Velusamy, M. Stan, and K. Skadron, "A Case for Thermal-Aware Floorplanning at the Microarchitectural Level," In the Journal of Instruction-Level Parallelism, Vol. 7, Oct 2005 (<http://www.jilp.org/vol7>).
- [9] G. M. Link and N. Vijaykrishnan, "Thermal Trends in Emerging Technologies," in International Symposium on Quality Electronic Design (ISQED), 2006.
- [10] P. Sundararajan, C. Patterson, C. Carmichael, S. McMillan, and B. Blodget, "Estimation of Single Event Upset Probability Impact of FPGA Designs," in Proceedings of MAPLD, 2003.
- [11] K. Skadron, M. R. Stan, et al., "Temperature-Aware Microarchitecture," In Proceedings of the 30th International Symposium on Computer Architecture (ISCA), 2003.
- [12] K. Skadron, M. R. Stan, et al., "Temperature-Aware Microarchitecture: Modeling and Implementation," In ACM Transactions on Architecture and Code Optimization, Vol. 1, No. 1, Mar 2004, Pages 94-125.
- [13] A. Telikepalli, "Designing for Power Budgets and Effective Thermal Management," In Xcell Journal, Issue 56, 2006. (http://www.xilinx.com/publications/xcellonline/xcell_56)
- [14] S. Velusamy et al., "Monitoring Temperature in FPGA based SoCs," In Proceedings of the International Conference on Computer Design (ICCD), 2005.
- [15] "Xilinx Power Tools," <http://www.xilinx.com/power>.
- [16] "Xilinx Products Documentation," <http://www.xilinx.com/literature>.
- [17] S. Lopez-Buedo and J. Garrido, "Making Visible the Thermal Behavior of Embedded Microprocessors on FPGAs: a Progress Report," In Proceedings of International Symposium on Field-programmable gate arrays, 2004.