Eight Ways to put your FPGA on Fire – A Systematic Study of Heat Generators

Markus Happe, Hendrik Hangmann, Andreas Agne, Christian Plessl
Computer Engineering Group
University of Paderborn, Germany
Email: markus.happe@upb.de, hhangman@mail.upb.de, {agne, christian.plessl}@upb.de

***Abstract*—Due to the continuously shrinking device structures and increasing densities of FPGAs, thermal aspects have become the new focus for many research projects over the last years. Most researchers rely on temperature simulations to evaluate their novel thermal management techniques. However, the accuracy of the simulations is to some extent questionable and they require a high computational effort if a detailed thermal model is used.**

For experimental evaluation of real-world temperature management methods, often synthetic heat sources are employed. Therefore, in this paper we investigated the question if we can create significant rises in temperature on modern FPGAs to enable future evaluation of thermal management techniques based on experiments in contrast to simulations. Therefore, we have developed eight different heat-generating cores that use different subsets of the FPGA resources. Our experimental results show that, according to the built-in thermal diode of our Xilinx Virtex-5 FPGA, we can increase the chip temperature by 134 degree C in less than 12 minutes by only utilizing about 21% of the slices.

***Index Terms*—FPGA, temperature, heat, dedicated heat-generating core, measurement, oscillator**

I. Introduction

Temperature-aware reconfigurable systems have drawn a considerable amount of attention from researchers in recent years. Several approaches for temperature management have been proposed, ranging from dynamic voltage and frequency scaling [1] and temperature-driven thread scheduling and migration [2], [3], [4] to fully temperature-aware systems that are capable of learning their own thermal characteristics during run-time [5], [6].

With ever shrinking device structures and increasing densities, thermal management of FPGA-based systems will become more and more important in the foreseeable future [7], [8]. In order to evaluate thermal management techniques for FPGAs today, researchers usually take one of two different routes.

One approach is to perform a simulation involving a temperature model based on manufacturer specifications or real world measurements of the device's thermal characteristics. While this allows for great flexibility in the modeling of possible future devices such as stacked multi-layer FPGAs [9] or 3D-many-cores [10], the simulation approach has several drawbacks. The thermal characteristics of the simulated devices and circuits are often at least partly unknown and have to be estimated. Also, for larger systems—for instance reconfigurable system-on-chips (SoCs)—a full functional and temperature simulation may consume a prohibitively large amount of computation time.

Another approach is the implementation of the system under test on an actual FPGA, using external and internal devices to measure the on-chip temperature distribution, such as infrared cameras [3], ring oscillators [11], [12] and thermal diodes. When investigating thermal management techniques on real world systems, often the need arises to purposefully generate heat in specific regions of the FPGA [9], [13] using dedicated circuits.

Little consideration has been given to the design of such circuits in the past. Often, ad-hoc solutions involving flip flop (FF) pipelines are employed without further examination of possible alternatives. In reality there are many different resources on modern FPGA that may be integrated into a heat generator.

In this paper, we present a detailed examination of heat-generating cores, examining a variety of different approaches using different on-chip resources and heat generation techniques, such as ring oscillators, LUT-FF-pipelines, shift registers and DSP blocks. We hope that this will enable researchers to evaluate their designs using more efficient heating circuits, that can generate more heat per unit area, leading to greater rises in temperature.

The paper is structured as follows: Section II presents related work on heat creation on FPGAs. The methodology that we use to provoke high temperatures is presented in Section III. Furthermore, we present dedicated heat-generating cores that utilize different sets of FPGA resources. In Section IV, we discuss our experimental setup, before we present and evaluate temperature measurements of each dedicated heat-generating core. Finally, Section V concludes our paper and gives an outlook towards future developments.

II. Related Work

This section presents related work on the creation of heat on FPGAs. For instance, Ebi et al. [3] measured a spatial thermal gradient of 2 degree C over 10 mm on a Xilinx Virtex II FPGA using an infrared camera. The heat was generated by 1,000 toggling flip flops clocked at 100 MHz, where the heater is constrained to a rectangular area on the FPGA. Zhang et al. [9] used pipelines composed of look-up tables (LUTs) and FFs to generate heat on a Spartan 3E-250K FPGA. They could measure that, when utilizing 100% of the slices at a clock

978-1-4673-2921-7/12/$31.00

frequency of 100 MHz, the temperature of the FPGA rises to a steady state temperature of 55°C. Whereas the steady state temperature was 35°C when only 20% of the slices were used.

In previous work [13], we have used 10,000 toggling flip flops, clocked at 100 MHz, to generate spatial gradients of 6.5°C across the entire die of a Virtex-6 FPGA. We have measured the temperature distribution of the FPGA using a self-calibrated sensor grid where ring oscillators were used as temperature sensors which were calibrated against the built-in thermal diode.

Sayed and Jones [12] used LUT-FF pipelines to create different workloads on the FPGA to characterize a ring oscillator-based thermal sensor. They have toggled different amounts of the FPGA resources (20%,40%,60%,80%) at different frequencies, 50 MHz, 100 MHz, 150 MHz and 200 MHz. According to their observations, the Xilinx system monitor might measure the temperature incorrectly. Using an external thermometer they measured temperature differences of 12.7°C for a Virtex-5 LX50t FPGA and 20.3°C for a Xilinx Virtex-5 LX110t FPGAs between the temperature readings of the external thermal diode and the built-in thermal diode.

In this paper, we rely on the thermal measurements of the Xilinx system monitor whose error range is stated to be ±4°C by the vendor. In contrast to previous work, we perform a systematic study of heat-generating cores utilizing different on-chip resources and examine the impact of the clock frequency.

III. Methodology

In this section, we describe our general approach to generate heat on modern FPGAs and introduce our heat-generating cores (heaters).

A. Generating Heat on FPGAs

In this paper. we investigate which resources of an FPGA are best-suited to generate a significant rise in temperature. An FPGA is a semiconductor device that is based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. Nowadays, a CLB usually consists of multiple look-up tables and flip-flops. Furthermore, modern FPGAs contain additional components, such as random access memory blocks (BRAMs) as local memory and digital signal processor blocks (DSP blocks) as efficient computation units.

We developed dedicated cores which have the single purpose of generating heat. In our experiments we studied different resources (LUTs, FFs, BRAMs, DSPs) separately and, in the case of LUTs and FFs, in combination. In order to create maximum heat, we intend to provoke as much toggling of our signals and/or our storage elements as possible. For most of our designs we connected our resource elements in pipelines in order to minimize the interconnect inside the core. Since most of our cores are clock-controlled, we did not only evaluate the individual influence of each resource element but also the impact of the clock frequency on heat generation.

B. LUT and FF-based heaters

In this section, we present our heaters which are entirely based on LUTs and FFs.

1) LUT pipeline: Figure 1 depicts a LUT-based pipeline. We used 6-input LUTs (LUT6s) with inputs (I) and output (O) for our experiments. We use one of the input signals to enable the LUT. The other five input signals are connected to the preceding LUT in the pipeline. The LUT is configured in such a way that it inverts one of the five input signal whenever the heater is enabled. The first LUT in the pipeline receives a toggling input signal, i.e. from a system clock.

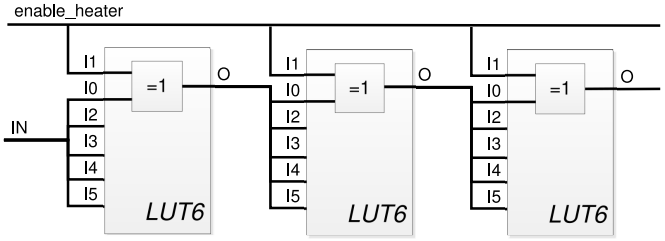


Fig. 1. LUT-based pipeline

2) LUT oscillator: LUTs can also be used as ring oscillators, where an odd number of inverters is connected to each other to form a ring. When the number of inverters is odd, the signal is unstable and toggles between '0' and '1'. The number of inverters and the delay of the interconnect define the toggling frequency. Since we intend to maximize the toggling frequency in order to create heat, we have used ring-oscillators with a single LUT, as depicted in Figure 2.

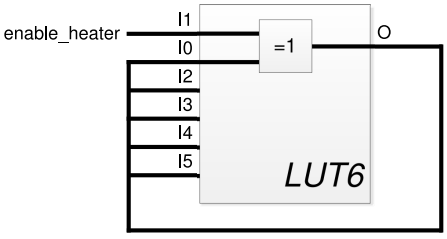


Fig. 2. LUT-based ring oscillator

3) SRL pipeline: Another possibility of using LUTs for the heat core is to use them as shift registers (SRLs). By cascading the SRLs it is possible to create one huge shift register, which is permanently shifting bits.

4) FF pipeline: One way to use FFs exclusively is to cascade them and build a shift register similar to the SRLs mentioned above. They have the same inputs, outputs and are clock-controlled with a clock enable signal. Beyond that, they have the same purpose, more precisely both pipelines are shift registers except that the FF can only store one bit compared to the 16 bit SRL in Virtex-5 FPGAs. Figure 3 shows an example for a FF-based heater with two elements.

5) LUT-FF pipeline: Another way to exploit the given hardware components is to use the whole slice, instead of solely LUTs or FFs. In order to combine LUTs with FFs, we

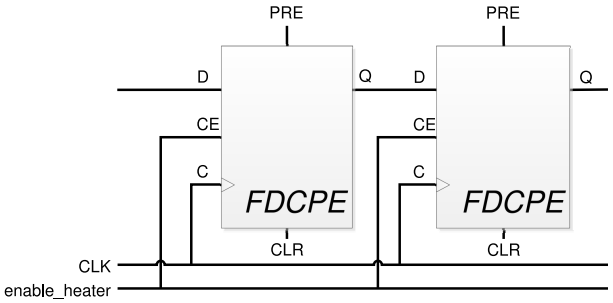


Fig. 3. FF-based pipeline

have designed a pipeline similar to the LUT pipeline (Figure 1) and the FF pipeline. The difference is, that between each pair of LUTs there is a FF interconnected, as can be seen Figure 4.

6) SRL-FF pipeline: An additional way to use LUTs and FFs is to make use of SRLs, instead of LUTs, and FFs. Therefore, each SRL's output signal is interconnected with the input signal of a FF. This creates a maximum capacity shift register for a given area.

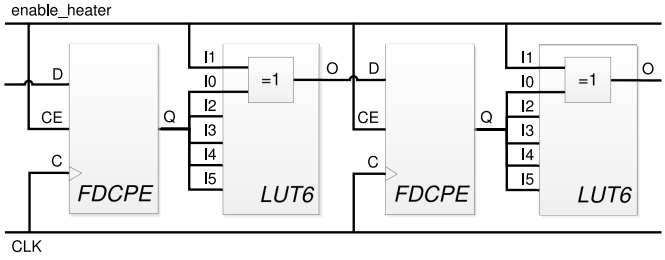


Fig. 4. LUT and FF-based pipeline

C. BRAM and DSP-based heaters

In addition to LUT and FF-based heaters, we have also studied the following heaters that focus on BRAMs and DSP blocks.

1) BRAM pipeline: The BRAM-based heater is composed in a pipeline as well, using the Xilinx primitive FIFO36. Therefore the data input bus and the data output bus (with a width of 32 bit) are interconnected to each other. Once the first BRAM is filled with random data, it passes the first 32-bit word in it to the next BRAM. Hence, all BRAMs are permanently changing their memory contents. The general structure of the BRAM-based pipeline can be seen in Figure 5.

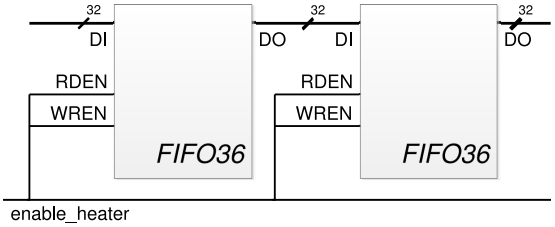


Fig. 5. BRAM-based pipeline

2) DSP pipeline: Finally, the DSP blocks are also cascaded and arranged in a pipeline. Therefore the the output signal is passed through all DSP blocks by interconnecting it with the successor's input signal C. All input signals A and B of the DSP blocks are connected to global signals, which change their values with each clock cycle. Figure 6 depicts two strongly simplified segments of the DSP pipeline.

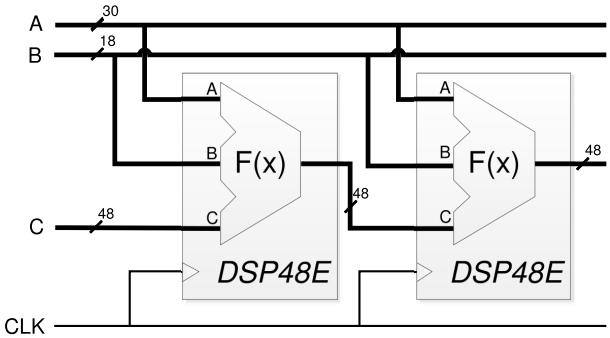


Fig. 6. DSP-based pipeline

IV. Experimental Results

In this section, we describe our experimental setup, present our temperature measurements for each dedicated heat-generating core and evaluate our experimental results.

A. Experimental Setup

For our experiments we used a Xilinx Virtex-5 XPUV5-LX110t FPGA (package: ff1136, speed grade: -1). The FPGA board features a default heat sink. For our LUT and FF-based heaters, we constrained the area of our heater to 61×61 slices which contain $14,884$ LUTs and $14,884$ FFs. This area was used by our LUT oscillator-based heater that contained 1,000 1-level ring oscillators.

As can be seen in Figure 8(b) this heater already heats up the FPGA to 195°C according to the built-in thermal diode. We restricted our experiments to 1,000 ring oscillators in order to prevent the destruction of the FPGA. To be able to compare the diverse LUT and FF-based heaters with each other, we have used the same amount of slices for each LUT and FF-based heater.

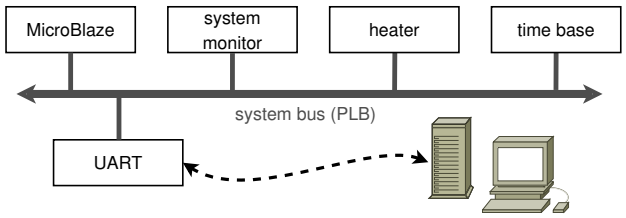


Fig. 7. Architecture of our experimental set-up

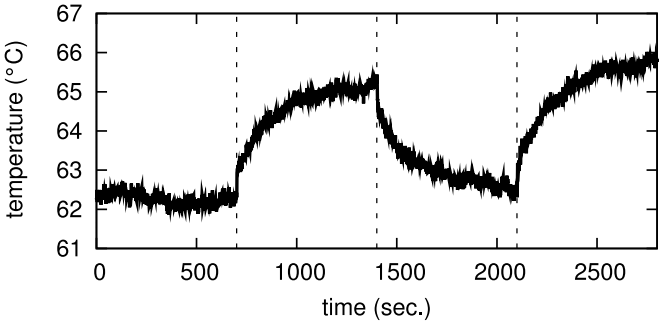
Our architecture is depicted in Figure 7. A MicroBlaze processor is connected to the Xilinx system monitor that accesses the sensor readings of the built-in thermal diode on the FPGA. Furthermore, our architecture contains a time base

and the heat-generating core under examination. The heater is enabled/disabled by a timer-driven program that runs on the MicroBlaze which also reads the temperature values. The temperatures readings are forwarded to a workstation using a UART interface.

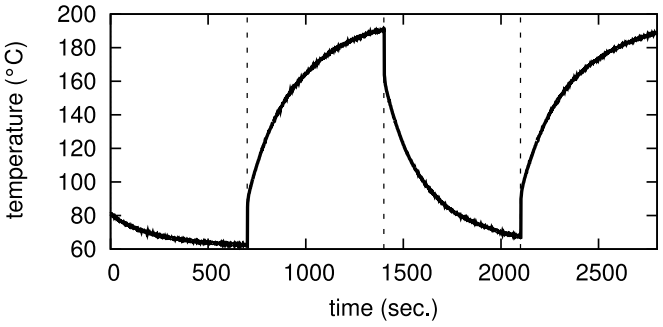
In all experiments we first wait for 700 seconds until the temperature of the FPGA is stable, before we enable the heater for 700 seconds. Next, we disable the heater for 700 seconds to see the fall in temperature for the cooling phase. Finally, we enable the heater again for another 700 seconds to confirm the repeatability of the experiment.

B. Temperature Measurements

Figures 8-10 show the temperature measurements for the heaters, that have been introduced in Section III, on a Xilinx Virtex-5 LX110t FPGA.



(a) LUT pipelines



(b) LUT oscillators

Fig. 8. Measurements for LUT or FF-based heaters on a Xilinx Virtex-5 LX110t FPGA.

For the first heater that is based on LUT pipelines, we used 14 pipelines with 1,000 stages. The corresponding measurement is depicted in Figure 8(a). A 100 MHz clock signal is the input for all 14 pipelines. It can be seen that the FPGA temperature increases about 3°C in 700 seconds.

When we use 1,000 ring oscillators each implemented with a single LUT only instead we can heat up the FPGA to about 195°C which represents a temperature increase of about 135°C in 700 seconds, see Figure 8(b).

Both cores, toggle 1,000 signals. However, while the LUT-based ring-oscillators toggles its output signal at the highest possible frequency, the LUT pipelines toggle their signal at 100 MHz only. Another important difference is that the toggling signals of the oscillators are transfered to a central point in the heat core. Thus, the LUT oscillators heater utilizes more routing resources than the LUT pipelines. Because of the poor heat output, we have not experimented with different input frequencies for LUT pipelines.

Figure 9(a) shows the measurements for the SRLs. The SRL heater has shown superior results to the LUT heater when considering that it utilizes only a fraction of the LUTs, because SRLs can only be mapped to specific slices of the FPGA. Therefore, we could only map 41 SRL pipelines with 100 stages into our constrained area. Thus, the heater only consumed 4,100 LUTs instead of 14,000 LUTs that are used for the LUT pipelines.

In our experiments, we were not able to clock the SRL heaters beyond 300 MHz. Higher frequencies resulted in lower rises in temperature compared to our results with 300 MHz. Other LUT and FF-based heaters have shown a similar behavior for frequencies higher than 400 MHz.

We could observe that for these cases our heaters do not toggle their signals reliably anymore. Timing checks of the routed design have been disabled in order to allow higher clock rates, such as 400 MHz for the LUT-FF pipelines.

For our FF heater, we again used 14 pipelines with 1,000 stages each. Similar to the experiments for the SRL pipelines, see Figure 9(a), an increase in the clocking frequency also increases the heat on the FPGA. The FF heater could increase the FPGA temperature between 5°C and 22°C depending on the clock rate, as can be seen in Figure 9(b).

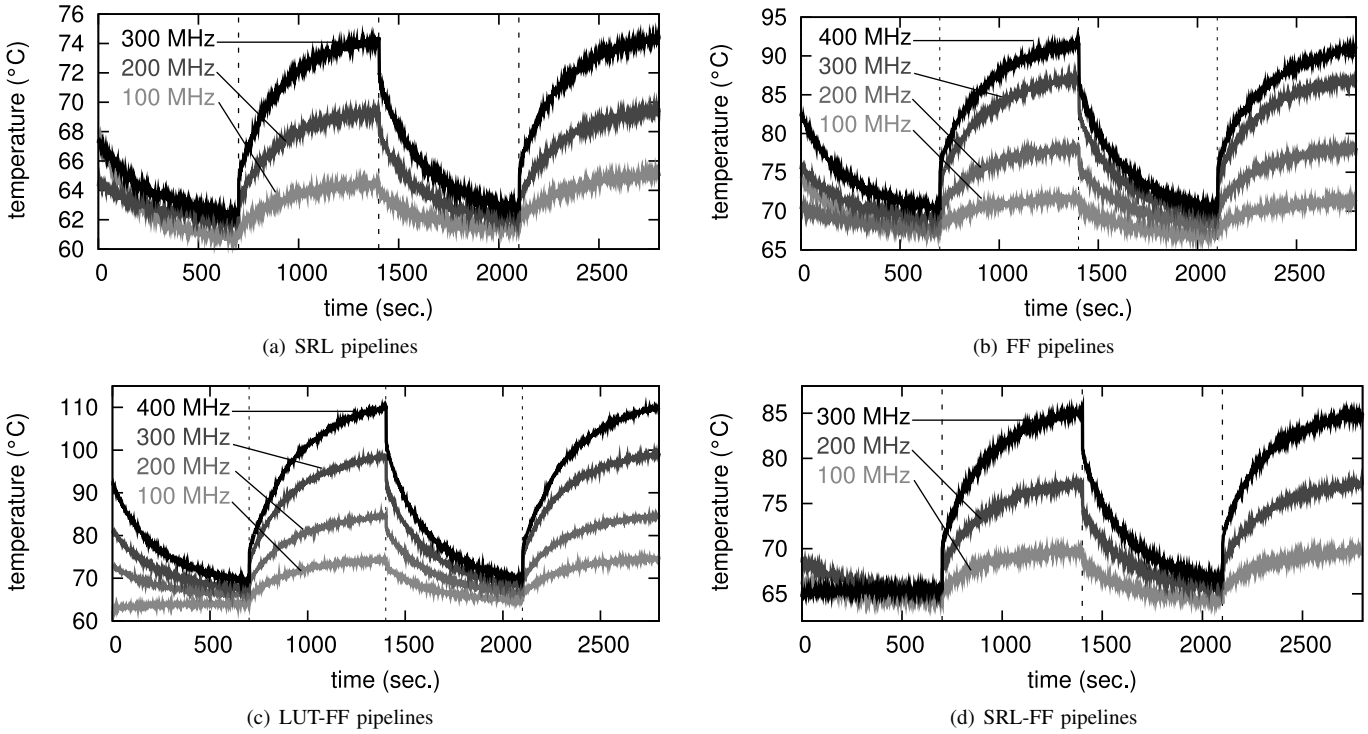
Hybrid heaters that use both, LUTs and FFs, clearly outperform the heaters which either use LUTs and FFs. This can be seen in Figure 9(c) for the LUT-FF pipelines. The LUT-FF heater contains 14 pipelines with 1,000 stages where each stage contains a LUT and a FF. The result is similar for the SRL-FF heat core which is depicted in Figure 9(d). Note that again the SRL-FF heat core uses 41 pipelines with 100 stages and, thus, significantly less resources than the LUT-FF heater.

Figure 10(a) shows the measurements from the BRAM heater. Here, we were able to clock the heater up to 500 MHz. The BRAM heater contains 130 BRAMs and can heat up the FPGA from 5°C up to 25°C depending on the clocking frequency. Finally, the DSP heater contains a single pipeline of 38 DSPs. This time we were able to clock the heater up to 550 MHz; the results can be seen in Figure 10(b).

For the DSP blocks, the temperature increase in 700 seconds again depends on the clocking frequency. Here, the increase was 2°C for 100 MHz and 14°C for 550 MHz. However, this pipeline additionally introduced a high number of LUTs and FFs and it might be the case that most of the temperature is generated by these resources. This seems plausible if the measurement results for the heaters that are purely based on LUTs and FFs are taken into account.

C. Resource Utilization

Table I lists the resource utilization of the different heat-generating cores. Note that each heater includes a bus attachment to the processor local bus (PLB) which utilizes a few



(a) SRL pipelines

(b) FF pipelines

(c) LUT-FF pipelines

(d) SRL-FF pipelines

Fig. 9. Measurements hybrid LUT-FF-based heaters on a Xilinx Virtex-5 LX110t FPGA.

LUTs and FFs. This can be seen, i.e., for the LUT pipelines that consume 248 additional LUTs and 376 FFs. The LUT oscillators required additional logic to ensure that the 1-level ring oscillators are connected to the system, so that they were not trimmed by the place and route tools. Finally, the DSP pipelines use additional LUTs and FFs for auxiliary logic.

TABLE I
RESOURCE UTILIZATION FOR THE HEATERS

heater	LUTs	FFs	BRAMs	DSPs
LUT pipelines	14,248	376	-	-
LUT oscillators	14,608	375	-	-
SRL pipelines	4,364	376	-	-
FF pipelines	233	14,376	-	-
LUT-FF pipelines	14,276	14,376	-	-
SRL-FF pipelines	4,408	4,476	-	-
BRAM pipelines	223	311	130	-
DSP pipelines	2,252	4,310	-	38
Virtex-5 LX110t	69,120	69,120	256	64

D. Summary

Table II summarizes the temperature increases for all heaters. The two highest temperature increases are generated by the LUT oscillators and the LUT-FF heater at 400 MHz. The routing of the heaters appears to have a major impact on the heat generation. The LUT oscillators heater utilizes a lot of routing resources and outperforms the other (pipelined) heaters that have limited routing.

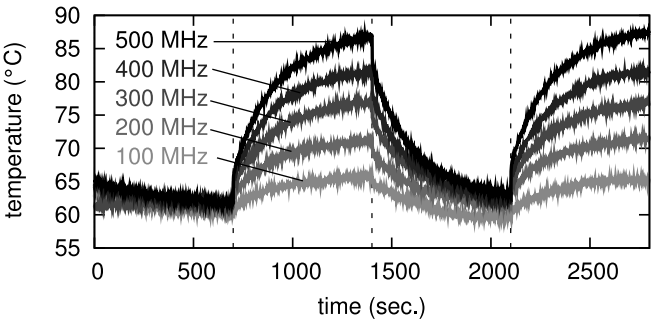
In general, the advantage of heaters, which are based on LUTs or FFs is their high flexibility. They can have arbitrary size and can be placed almost anywhere on the chip.

TABLE II
TEMPERATURE INCREASE OF THE HEATERS OVER 700 SECONDS

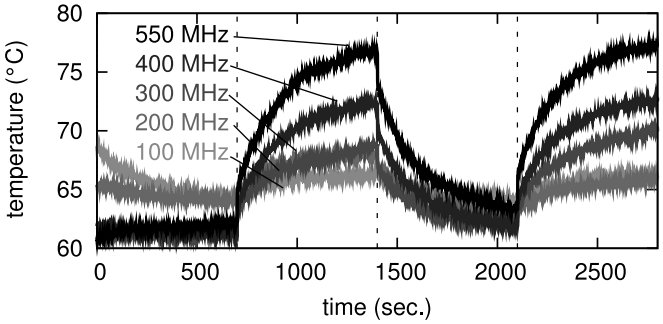
heater	temperature rise				
LUT	+3°C				
LUT osc.	+134°C				
heater	**temperature rise with different frequencies [MHz]**				
	100	**200**	**300**	**400**	**500/550**
SRL	+4°C	+7°C	+12°C	-	-
FF	+5°C	+11°C	+18°C	+22°C	-
LUT-FF	+10°C	+15°C	+31°C	+41°C	-
SRL-FF	+6°C	+12°C	+20°C	-	-
BRAM	+5°C	+10°C	+14°C	+20°C	+25°C
DSP	+2°C	+4°C	+7°C	+11°C	+14°C

In contrast to LUTs getting their input from a system clock, SRLs show better results for heat creation if resource utilization is taken into account. However, the amount of LUTs that can implement SRLs is limited on the Virtex-5. Hence, for the same rectangular area constraint, less heat can be be generated for SRLs.

DSPs blocks show poor results for heat creation, since a significant amount of generated heat is likely to come from the surrounding logic. In contrast, the BRAMs can heat up the FPGA on their own.



(a) BRAM pipelines



(b) DSP pipelines

Fig. 10. Measurements BRAM or DSP-based heaters on a Xilinx Virtex-5 LX110t FPGA.

V. CONCLUSION

In this paper, we propose eight dedicated cores that utilize a different subset of the FPGA resources in order to create heat. We focused on look-up tables, flip flops, digital signal processor blocks and RAM blocks. For our clocked heat-generating cores we additionally analyzed the influence of clock frequency on heat generation.

As expected, we could observe that a higher clock frequency results in higher temperatures and that the most heat can be generated using excessive routing combined with high frequencies. Hence, you can save FPGA resources using higher clock frequencies to achieve similar temperatures. In our experimental results, we were able to increase the device temperature by 134°C using a large number of 1-level ring oscillators on a Virtex-5 LX110t FPGA in a time interval of about 12 minutes.

Thus, we can create high temperatures on today's FPGAs. For our LUT and FF-based heaters, which include the two hottest ones, we only utilized about 21% of the available slices. Hence, we believe that our heaters can also create significant spatial temperature gradients on an FPGA. The presented results might be useful for researchers that intend to evaluate their novel FPGA-based thermal management techniques with real measurements.

In future work, we plan to generate different temperature distributions on modern FPGAs based on our heat-generating cores. We will also investigate the effectiveness of proposed thermal management techniques which have so far only been evaluated using simulations.

ACKNOWLEDGMENT

This work was partly supported by the German Research Foundation (DFG) as part of the priority program "Dependable Embedded Systems" (SPP 1500) and by the International Graduate School of Dynamic Intelligent Systems. The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement n°257906.

REFERENCES

[1] M. Salehi, M. Samadi, M. Najibi, A. Afzali-Kusha, M. Pedram, and S. Fakhraie, "Dynamic Voltage and Frequency Scheduling for Embedded Processors Considering Power/Performance Tradeoffs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 10, pp. 1931 –1935, 2011.

[2] F. Mulas, D. Atienza, A. Acquaviva, S. Carta, L. Benini, and G. DeMicheli, "Thermal Balancing Policy for Multiprocessor Stream Computing Platforms," *IEEE Transactions on CAD of Integrated Circuits and Systems*, vol. 28, no. 12, pp. 1870–1882, 2009.

[3] T. Ebi, D. Kramer, W. Karl, and J. Henkel, "Economic Learning for Thermal-aware Power Budgeting in Many-core Architectures," in *IEEE International Conference on Hardware-Software Codesign and System Synthesis (CODES+ISSS)*, 2011.

[4] A. Gupte and P. Jones, "Hotspot Mitigation using Dynamic Partial Reconfiguration for Improved Performance," in *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, 2009.

[5] E. Kursun and C.-Y. Cher, "Temperature Variation Characterization and Thermal Management of Multicore Architectures," *IEEE MICRO*, pp. 116–126, Jan./Feb. 2009.

[6] Y. Ge, P. Malani, and Q. Qiu, "Distributed Task Migration for Thermal Management in Many-core Systems," in *Proceedings of the 47th Design Automation Conference (DAC)*. ACM, 2010.

[7] S. Borkar, "Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation," *IEEE MICRO*, pp. 10–16, Nov./Dec. 2005.

[8] Seyab and S. Hamdioui, "NBTI Modeling in the Framework of Temperature Variation," in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*. European Design and Automation Association, 2010, pp. 283–286.

[9] C. Zhang, R. Kallam, A. Deceuster, A. Dasu, and L. Li, "A Thermal-mechanical Coupled Finite Element Model with Experimental Temperature Verification for Vertically Stacked FPGAs," *Microelectronic Engineering*, vol. 91, pp. 24–32, March 2012.

[10] T. Ebi, H. Rauchfuss, A. Herkersdorf, and J. Henkel, "Agent-based Thermal Management using Real-time I/O Communication Relocation for 3D Many-cores," in *International Conference on Integrated Circuit and System Design: Power and Timing Modeling, Optimization, and Simulation (PATMOS)*. Springer, 2011, pp. 112–121.

[11] C. Rüthing, A. Agne, M. Happe, and C. Plessl, "Exploration of Ring Oscillator Design Space for Temperature Measurements on FPGAs," in *IEEE International Conference on Field Programmable Logic and Applications (FPL)*, 2012.

[12] M. Sayed and P. Jones, "Characterizing Non-ideal Impacts of Reconfigurable Hardware Workloads on Ring Oscillator-Based Thermometers," in *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, 2011.

[13] M. Happe, A. Agne, and C. Plessl, "Measuring and Predicting Temperature Distributions on FPGAs at Run-Time," in *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, 2011.