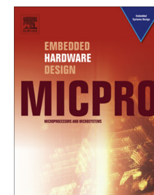
ARTICLE IN PRESS

Microprocessors and Microsystems xxx (2013) xxx–xxx

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro



Seven recipes for setting your FPGA on fire – A cookbook on heat generators

Andreas Agne [a,*], Hendrik Hangmann [a], Markus Happe [b], Marco Platzner [a], Christian Plessl [a]

[a] Department of Computer Science, University of Paderborn, 33098 Paderborn, Germany
[b] Communication Systems Group, ETH Zurich, 8092 Zurich, Switzerland

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
FPGA
Temperature
Heat
Heater
Heat-generating core
Measurement
Oscillator

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, these temperature simulations require a high computational effort if a detailed thermal model is used and their accuracies are often unclear.

In contrast to simulations, the use of synthetic heat sources allows for experimental evaluation of temperature management methods. In this paper we investigate the creation of significant rises in temperature on modern FPGAs to enable future evaluation of thermal management techniques based on experiments. To that end, we have developed seven different heat-generating cores that use different subsets of FPGA resources. Our experimental results show that, according to external temperature probes connected to the FPGA's heat sink, we can increase the temperature by an average of 81 °C. This corresponds to an average increase of 156.3 °C as measured by the built-in thermal diodes of our Virtex-5 FPGAs in less than 30 min by only utilizing about 21 percent of the slices.

© 2013 Elsevier B.V. All rights reserved.

1. 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 [2–4] to fully temperature-aware systems that are capable of learning their own thermal characteristics at 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.

In the area of thermal management, heat-generating circuits can be used to emulate active cores on a many-core system-on-chip architecture. Researchers can then migrate functionalities between homogeneous or heterogeneous cores in order to avoid high spatial temperature gradients on the chip [3]. Cores that generate a high temperature increase also consume much power, hence heat-generating cores could also be used for power management, such as dynamic frequency scaling. However, in this paper we do not study the power consumption of our heat cores. Furthermore, heat-generating cores have been utilized to experimentally evaluate the accuracy of FPGA-based temperature monitoring systems under stress. For instance, heat-generating circuits have been used to evaluate the quality of ring oscillators as local temperature sensors on modern FPGAs [12,14]. Heat generators can also be found in research areas that do not primarily focus on thermal or power

* Corresponding author. Tel.: +49 5251604348.
E-mail addresses: agne@upb.de (A. Agne), hhangman@mail.upb.de (H. Hangmann), markus.happe@tik.ee.ethz.ch (M. Happe), platzner@upb.de (M. Platzner), christian.plessl@uni-paderborn.de (C. Plessl).

0141-9331/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.micpro.2013.12.001

Please cite this article in press as: A. Agne et al., Seven recipes for setting your FPGA on fire – A cookbook on heat generators, Microprocess. Microsyst. (2013), http://dx.doi.org/10.1016/j.micpro.2013.12.001

management. For instance, high-power heat generators can mask the activity of a hardware encryption block in order to protect against side-channel attacks, i.e. power attacks [15]. In this example, heat generators improve the security of an FPGA-based circuit.

In summary it can been said that multiple research areas would profit from a cookbook such that researchers can follow well-defined recipes in order to build the set of heat-generators, which they require for their specific experiments. However, little consideration has been given to the design of such heat generating 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 such as LUTs, shift registers, RAM blocks, or multiply-accumulate blocks.

Our goal is to empirically and systematically investigate how the different on-chip resources of a modern FPGA can be utilized for heat generation and how much heat we can expect to generate with each resource. An analysis of the different causes of heat generation is beyond the scope of this paper and, beyond that, requires access to vendor proprietary design data. In previous work [16], we have presented 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. Compared to [16], in this paper we have re-evaluated our experiments using a more reliable external temperature probe, and we have investigated the impact of sample variation by comparing measurements on 4 different FPGAs. In [16] we used Xilinx' built-in thermal diode to quantify the temperature increases of our heat-generating cores over time. Since the accuracy of the built-in diode has been questioned in [12], we have repeated our experiments with an external temperature probe. The results of these measurements are also shown in this paper. Additionally, this paper discusses the sample variation between four different FPGAs of the same make and model. Finally, this paper gives an estimation of the spatial temperature gradients that can be achieved with our most efficient heat core. We hope that our results 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 2 presents related work on heat creation on FPGAs. We lay out the heat generator design space in Section 3. The methodology that we use to provoke high temperatures is presented in Section 3. In Section 4 we introduce our heat generator designs. Furthermore, we present dedicated heat-generating cores that utilize different sets of FPGA resources. In Section 5, we describe our experimental setup and evaluate temperature measurements of each dedicated heat-generating core. Finally, Section 6 concludes our paper and gives an outlook towards future developments.

2. Related work

This section presents related work on the creation of heat on FPGAs. Few years ago, most researchers relied on external heating blocks to generate heat [17,18]. In the last two years, more and more researchers investigated methods, where the FPGA itself generates heat due to processing using dedicated cores. This method is sometimes referred to as self-heating [14]. For instance, Ebi et al. [3] measured a spatial thermal gradient of 2 °C over 10 mm on a Xilinx Virtex II FPGA using an infrared camera. The heat was generated by 1000 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 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. For the sensor self-calibration, we have increased the chip temperature on the entire chip area evenly using dedicated heater circuits and have mapped the frequencies of the individual ring oscillators to the temperature readings of the built-in thermal diode. After this initial calibration phase each ring oscillator could translate its oscillation frequencies to its corresponding local temperature.

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. For the clock frequency of 100 MHz, they measured a temperature increase of +6 °C by utilizing 20% of the FPGA resources, +16 °C by utilizing 40% of the FPGA resources, +28.5 °C by utilizing 60% of the FPGA resources, and, finally, +37.0 °C by utilizing 80% of the FPGA resources on a Xilinx Virtex-5 LX110T FPGA, which contains 17,200 slices, by using the built-in thermal diode. For a Virtex-5 LX50T FPGA with only 7200 slices they measured a temperature increase of +4 °C by utilizing 20% of the FPGA resources, +8 °C by utilizing 40% of the FPGA resources, +12 °C by utilizing 60% of the FPGA resources, and, +17.5 °C by utilizing 80% of the FPGA resources (clocked at 100 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. For a Virtex-5 LX110T the junction-to-case thermal resistance θ_{rc} of the chip lies between 0.10 °C/W to 0.15 °C/W [19]. The maximum power consumed by the Virtex-5 LX110T was 5 W, which results in a maximum difference of 0.5–0.75 °C between the chip and the case. The difference of 20.3 °C between chip and case temperature can also not be explained by additionally taking into account the manufacturer's specification of a maximum measurement error of ±4 °C for the internal temperature sensor [20].

More recently, Tradowsky et al. [14] experimentally generated a spatial temperature gradient on a Xilinx Virtex-5 LX110T FPGA of +12 °C over the entire die by toggling 750 flip-flops (about 1% of the FPGA's flip-flops) at one corner of the FPGA. According to a net of four ring oscillator-based temperature sensors, this corner of the FPGA increased its temperature by +33 °C in about 28 min. The built-in thermal diode, which is assumed to be located in the center of the FPGA, measures a temperature increase of +21 °C in 28 min for the same experiment. The toggling frequency was not reported, but we assume that it was at least 400 MHz.

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.

3. Background and methodology

In the following section, we will first present a brief overview on heat generation in digital circuits and how clock frequency, voltage, and capacitance influence heat generation in general.

Please cite this article in press as: A. Agne et al., Seven recipes for setting your FPGA on fire – A cookbook on heat generators, Microprocess. Microsyst. (2013), http://dx.doi.org/10.1016/j.micpro.2013.12.001

Then, we present the different reconfigurable resources we examine as well as the methodology we follow in our experiments.

3.1. Heat sources

Sources of heat in semiconductor devices can be divided into two groups. Static heat generation is independent of switching activity and is caused by currents flowing through resistors. On highly integrated devices such as modern FPGAs, leakage currents inside transistors and between interconnects are major contributors to static heat generation. In contrast, dynamic heat is caused by cross-conduction losses as well as capacitive power losses due to switching activity. Given the capacitance C of a net, the supply voltage V, the toggle frequency f, the power loss P on that net can be calculated as

$$P = fCV^2$$

For instance, for a capacitance driven by a flip-flop with clock frequency f_{clk} that toggles on every cycle, the power loss is $P_{\mathrm{FF}} = \frac{1}{2} f_{\mathrm{clk}} CV^2$ because the toggle rate equals half the clock frequency.

On an FPGA there are a number of reconfigurable circuit elements that introduce power losses proportional to the toggle rate of the net they are part of. Since power losses are proportional to the toggle rate, it is possible to assign a characteristic energy E_i to each circuit element i that is released every time a switch occurs. The total power losses of a circuit can then be calculated as the sum:

$$P = f \cdot \sum_i E_i$$

The characteristic energies can be used to estimate a circuit's power consumption, as it is done by various automated power estimation tools. The spreadsheet-based Xilinx Power Estimator for instance, gives a characteristic energy of $1.641 \cdot 10^{-13}$ J for a flip-flop on a Virtex 5 device. Because of additional capacitance driven by the flip-flop, the real world power consumption may be higher.

In our investigation we focus on the following circuit components: slice elements (LUTs, FFs, Latches), on-chip block RAM, dedicated multiply accumulate units (DSP slices) and routing resources. On Virtex 5 devices, one in four LUTs may be used as shift registers or distributed RAM, hence we allow for different characteristic energies of the LUT components, dependent on the mode of operation.

In a synchronous circuit the toggle rate f can be adjusted through the clock frequency. When the aim is to increase heat generation, raising the clock frequency is of course an option. There is however a limit to this approach, since at some point setup and hold violations may occur between clocked components which may lead to aliassing effects and a decrease in overall switching activity, thereby lowering the total heat output.

3.2. Methodology

An FPGA is a semiconductor device that is based on a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. On a Xilinx Virtex-5 FPGA, a CLB contains multiple look-up tables and flip-flops. Besides CLBs, there are further components, such as random access memory blocks (BRAMs) and so called digital signal processing blocks (DSP blocks) that function as efficient multiply-accumultate units.

We developed dedicated cores that 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, also in combination. In order to create maximum heat, we want to provoke as much toggling of signals and/or our storage elements as possible. Besides the mentioned FPGA resources, interconnect can play a huge role in the heat generation of a circuit. Unfortunately, unless the routing it is manually specified, interconnect utilization is largely unpredictable. Specifying the routing however is not a practical approach in many situations, since this means bypassing automated hardware synthesis and requires placing and routing circuit components by hand – a lengthy and error-prone process that has to be repeated for each new target architecture. We therefore aim at keeping the influence of routing to a minimum by manually placing circuit elements in a pipelined fashion, wherever possible. This reduces fan-out and unnecessary routing. Most of our cores are clock controlled, thus we did not only evaluate each type of resource, but also the impact of the clock frequency on heat generation.

The heat cores were specified as VHDL descriptions and automated tools were used for synthesis and implementation of the designs. Due to the nature of exploring the limits of the FPGA with respect to heat production, timing and design rules check (DRC) errors were expected for some of our heater designs. For some of our high frequency synchronous heaters, for instance, the routing algorithm was unable to generate designs that were guarantied to satisfy the inferred timing constraints. In these cases the tools were instructed to proceed and ignore timing violations.

3.3. External versus internal temperature sensors

We tested each heater design independently on 4 different Xilinx XUPV5 Boards. We obtained temperature measurements via the internal temperature diode as well as via an external temperature probe connected to the heat sink of the FPGA. Performing temperature measurements on an otherwise idle FPGA (no bitstream programmed) we noticed a significant deviation in the temperature as reported by the internal sensor between the individual boards. Fig. 1 depicts the idle temperature measurements, internal as well as external. While the temperatures obtained by the external probe have very little deviation (well below 1 °C), the internal sensors measure considerably different values (about 20 °C deviation). This finding contradicts the manufacturer's specification of a maximum measurement error of ± 4 °C for the internal temperature sensor [20]. In addition to the concerns about the temperature diode's accuracy raised in [12], it seems appropriate to focus on external temperature measurements in the remainder of this paper.

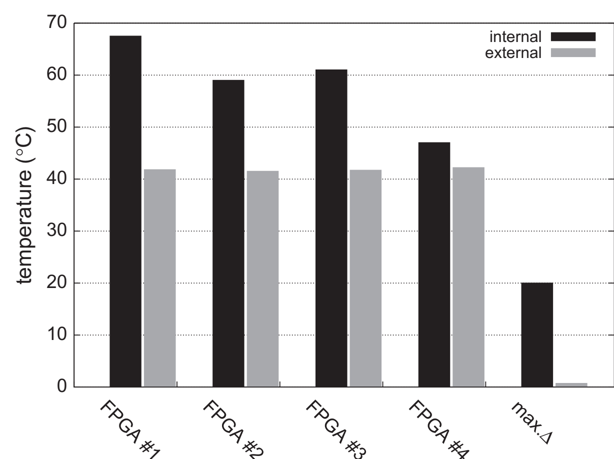


Fig. 1. Temperature measurements of idle FPGAs (at 25 °C ambient temperature). Temperature measurements by the external probe and internal sensors on an idle system.

Please cite this article in press as: A. Agne et al., Seven recipes for setting your FPGA on fire – A cookbook on heat generators, Microprocess. Microsyst. (2013), http://dx.doi.org/10.1016/j.micpro.2013.12.001

ARTICLE IN PRESS

4. Heat generators

In this section, we first present the heaters that are based on synchronous circuits. Since these heaters depend on a clock signal their heat outputs vary with the clock frequency. We then present the ring oscillator - a heat generating circuit based on a feedback loop, that does not use an external clock signal.

4.1. SRL pipeline

For our first heat core, we use the LUTs as shift registers (SRLs). By cascading the SRLs, it is possible to create one large shift register. By feeding an alternating sequence of 1s and 0s to the SRL pipeline's input, we make sure that the 16 bits within the SRLs undergo constant toggling.

4.2. 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. Fig. 2 shows an example for a FF-based heater with two elements.

4.3. 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 have designed a pipeline similar to the FF pipeline (Fig. 2). The difference is, that between each pair of FFs there is a LUT interconnected, as can be seen in Fig. 3. In this design all look-up tables are implemented as an exclusive OR (XOR) of the input signals *I*0 and *I*1.

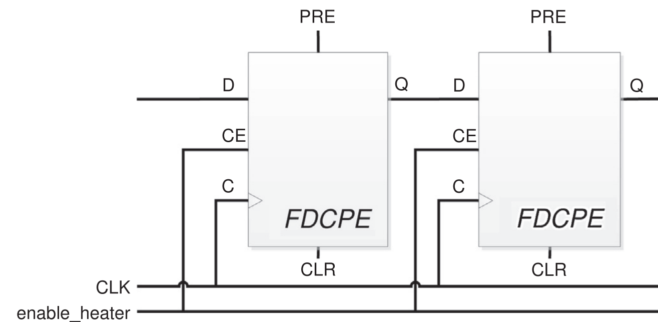


Fig. 2. FF-based pipeline.

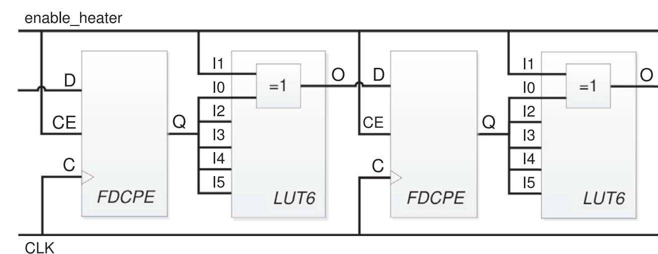


Fig. 3. LUT and FF-based pipeline.

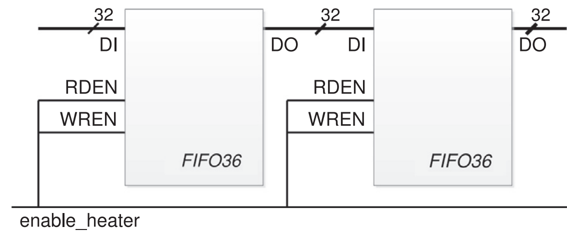


Fig. 4. BRAM-based pipeline.

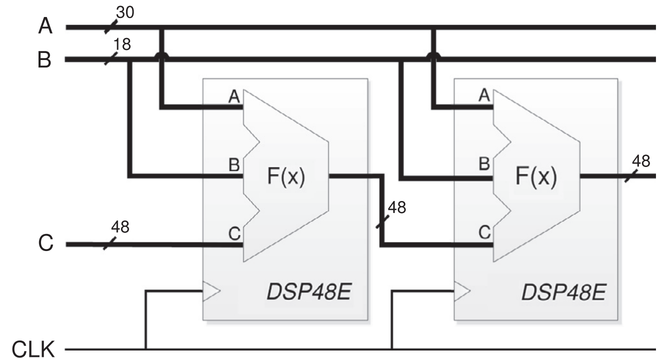


Fig. 5. DSP-based pipeline.

4.4. 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 connected to the input signal of a FF. This creates the most area efficient shift register for the available area.

4.5. BRAM pipeline

The BRAM-based heater also uses a pipeline design, built on 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 have their memory contents changed constantly. The general structure of the BRAM-based pipeline can be seen in Fig. 4.

4.6. DSP pipeline

DSP blocks are also cascaded and arranged in a pipeline. Therefore 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 are toggled at every clock cycle. Fig. 5 depicts two strongly simplified segments of the DSP pipeline.

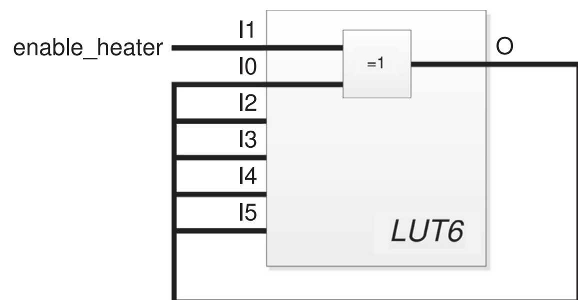


Fig. 6. LUT-based ring oscillator.

Please cite this article in press as: A. Agne et al., Seven recipes for setting your FPGA on fire – A cookbook on heat generators, Microprocess. Microsyst. (2013), http://dx.doi.org/10.1016/j.micpro.2013.12.001

4.7. The LUT-based ring oscillator

LUTs can also be used as ring oscillators, where an odd number of inverters are 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 Fig. 6.

5. 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.

5.1. 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. An external Voltcraft K204 thermometer has been attached to the heat sink of the FPGA in order to measure the outside temperature of the device. The manufacturer specifies the measurement accuracy of the thermometer as 0.1 °C for temperatures up to 200 °C. 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.

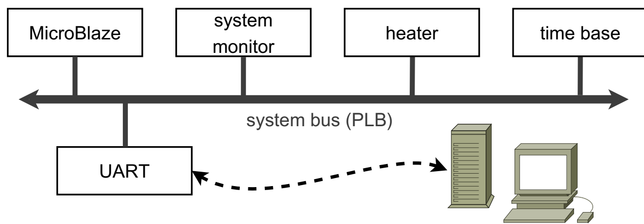


Fig. 7. Architecture of our experimental setup [16].

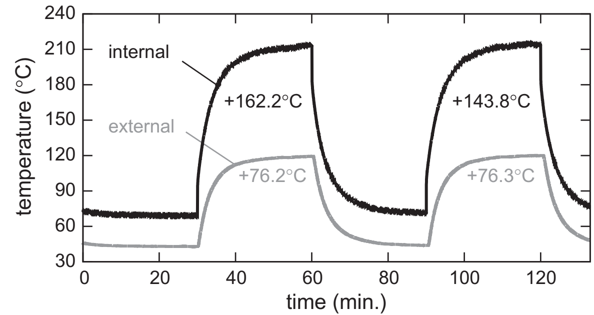


Fig. 9. Measurements of the internal thermal diode and the external temperature probe for the LUT oscillators heat core on a Xilinx Virtex-5 LX110T FPGA.

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.

Our architecture is depicted in Fig. 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 temperature readings are forwarded to a workstation using a UART interface.

In all experiments we first wait for 30 min until the temperature of the FPGA is stable, before we enable the heater for 30 min. Next, we disable the heater for 30 min to see the fall in temperature for the cooling phase. Finally, we enable the heater again for another 30 min to confirm the repeatability of the experiment.

5.2. Temperature measurements

Fig. 8 shows the temperature progressions for the heaters, which have been introduced in Section 4, on a single Xilinx Vir-

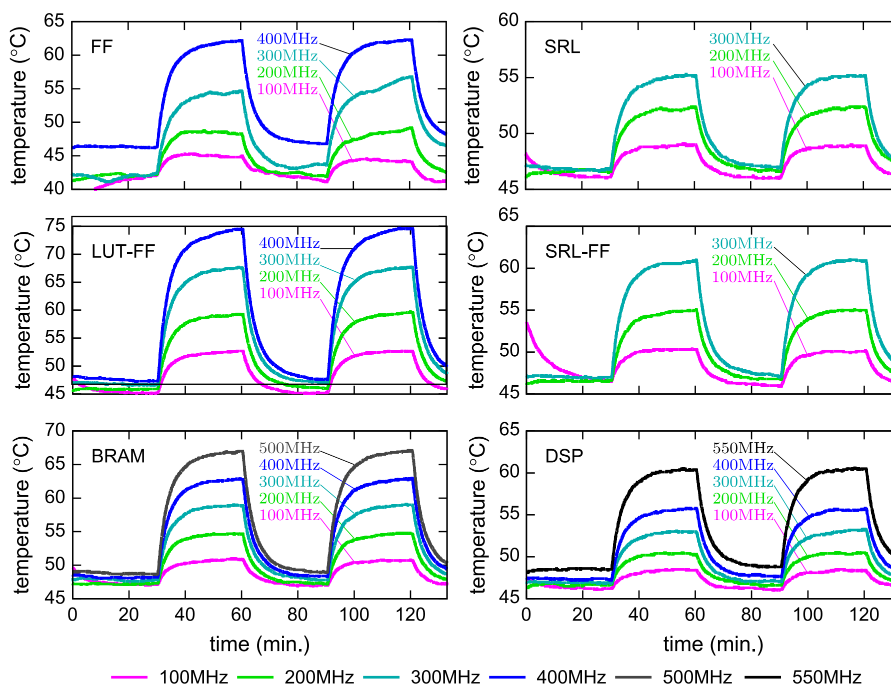


Fig. 8. Measurements for six heat cores on a Xilinx Virtex-5 LX110T FPGA using external temperature probe.

Please cite this article in press as: A. Agne et al., Seven recipes for setting your FPGA on fire – A cookbook on heat generators, Microprocess. Microsyst. (2013), http://dx.doi.org/10.1016/j.micpro.2013.12.001

Table 1
Temperature increase of the heaters over 30 min: the highest temperature increases per heater are highlighted.

Heater	Temperature rise (°C)				
LUT osc.	+81.2				
	With different frequencies [MHz]				
	100 (°C)	200 (°C)	300 (°C)	400 (°C)	500/550 (°C)
SRL	+2.9	+5.5	+8.4	–	–
FF	+3.4	+7.7	+12.2	+15.6	–
LUT-FF	+7.2	+13.1	+20.2	+26.3	–
SRL-FF	+4.5	+8.5	+13.5	–	–
BRAM	+3.9	+7.6	+11.2	+14.5	+18.1
DSP	+2.4	+3.7	+6.1	+8.5	+11.9

Table 2
Resource utilization for the heaters.

Heater	LUTs	FFs	BRAMs	DSPs
LUT osc.	14,608	375	–	–
SRL	4364	376	–	–
FF	233	14,376	–	–
LUT-FF	14,276	14,376	–	–
SRL-FF	4408	4476	–	–
Constrained area	14,884	14,884	–	–
BRAM	223	311	130	–
DSP	2252	4310	–	38
Virtex-5 LX110T	69,120	69,120	256	64

tex-5 LX110T FPGA. Unless noted otherwise, all temperatures mentioned in the text refer to values averaged over 4 different Virtex-5 LX110T devices as obtained by a temperature probe attached to the FPGA's heat sink. Temperature increases caused by the heat generators are reported with respect to the idle temperature of the FPGA. In our experiments the FPGA is considered idle, when it is powered up and configured, but the heat generators are not yet active.

The least effective heat generator turned out to be SRL based heater. We implemented the heater with 41 SRL pipelines each consisting of 100 SRL-LUTs, giving a total of 4,100 LUTs used for heat generation. In Xilinx Virtex-5 FPGAs SLICEM slices can implement shift registers or distributed memory using look-up tables whereas SLICEL slices cannot [21]. The ratio between SLICEM slices and SLICEL slices on the used Virtex-5 LX100T FPGA is about 1:3. Because only 1 in 4 LUTs can be configured as a SRL, we could only use a quarter of the LUTs in the constrained area. The temperature raise induced by the heater starts with 4.5 °C at 100 MHz and reached its maximum with 13.5 °C at 300 MHz. We were not able to increase the heat output further by increasing the clock frequency. Instead, the heat production decreases for operating frequencies beyond 300 MHz due to an overall decrease in switching activity incurred by timing violations. We could confirm this by sampling the output signals of the 41 SRL pipelines.

For our FF heater, we used 14 pipelines with 1,000 stages each. As expected, raising the clocking frequency lead to an increase in the FPGA's temperature. The FF heater managed to heat up the FPGA to temperatures ranging from 3.4 °C at 100 MHz up to 15.6 °C at 400 MHz. Going beyond 400 MHz did not yield a higher heat output.

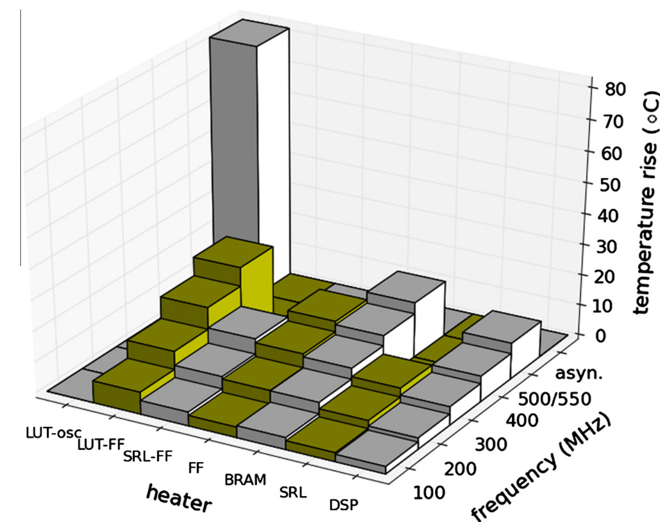


Fig. 10. Graphical representation of the temperature rises of all individual heaters over 30 min. Note that the LUT osc. heater is not clocked.

Table 3
Sample variation for selected experiments.

Heater	Board number				Delta
	1	2	3	4	
LUT osc.	*76.3*	76.8	82.1	**89.7**	13.4
SRL	8.0	**9.3**	*7.8*	8.5	1.5
FF	*14.7*	**16.4**	15.1	16.0	1.7
LUT-FF	26.6	*25.4*	25.9	**27.4**	2.0
SRL-FF	*13.1*	13.2	13.6	**14.0**	0.9
BRAM	*17.6*	18.3	18.0	**18.4**	0.8
DSP	*11.3*	11.6	**12.7**	12.1	1.4

Hybrid heaters that use both, LUTs and FFs, clearly outperform the heaters which either use LUTs and FFs. 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. Note that again the SRL-FF heat core uses 41 pipelines with 100 stages and, thus, significantly less resources than the LUT-FF heater.

For the BRAM heater, we were able to clock the heater up to 500 MHz. The BRAM heater contains 130 BRAMs and can heat up the FPGA from 3.9 °C up to 18.1 °C depending on the clocking frequency. Finally, the DSP heater contains a single pipeline of 38 DSPs. We were able to clock the heater up to 550 MHz.

For the DSP blocks, the temperature increase in 30 min again depends on the clocking frequency. Here, the increase was 2.4 °C for 100 MHz and 11.9 °C for 550 MHz. However, this pipeline additionally introduced a high number of LUTs and FFs and it is possible 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.

While synchronous heating circuits are ultimately constrained by signal timing with respect to a fixed clock signal, the ring oscillator suffers no such restriction. Here, the oscillation frequency is a direct result of the signal delay, which means toggling occurs as fast as physically possible. When we use 1,000 ring oscillators each implemented with a single LUT, we can increase the FPGA's external temperature by 81.2 °C on average.

We have performed our experiments on four Xilinx Virtex-5 XUPV5 boards. Fig. 9 shows how the internal an external temperatures change over time for one specific FPGA board (FPGA #1). The external temperature rises from 43.2 °C to 119.4 °C in the first heating phase and from 43.8 °C to 120.1 °C in the second heating phase, the internal temperature rise measured by the built-in thermal diode is even higher.

At these heat sink temperatures we can only speculate about the precise junction temperature of the FPGA, which at this point clearly exceeds recommended operating conditions. Measurements of the internal thermal diode quickly surpass 125 °C, exceeding the measurement capabilities of the device as specified in [20].

Please cite this article in press as: A. Agne et al., Seven recipes for setting your FPGA on fire – A cookbook on heat generators, Microprocess. Microsyst. (2013), http://dx.doi.org/10.1016/j.micpro.2013.12.001

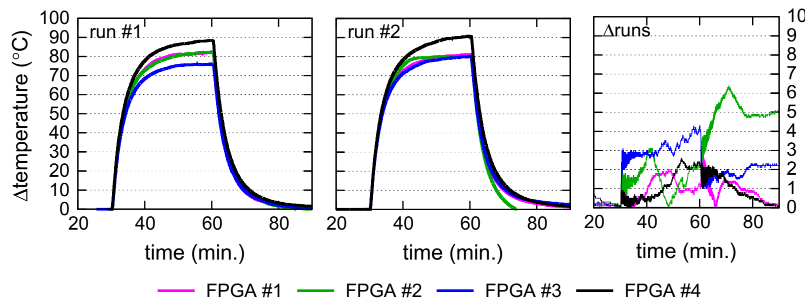


Fig. 11. Temperature development for the LUT oscillator heater for all four FPGAs for two different test runs (left, middle) and the absolute temperature differences between both runs (right)

Table 1 and Fig. 10 summarize the temperature increases of all heat cores measured by the external temperature probe over a period of 30 min. It can be seen that the LUT oscillator heater generates most heat, followed by the LUT-FF heat core, which is clocked at 400 MHz.

5.2.1. Resource utilization

Table 2 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 LUTs and 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.

5.3. Sample variation

We performed all our experiments on four different Xilinx XUPV5 boards that are identical in construction and feature a Virtex5-LX110T FPGA. All FPGAs were configured with the same configuration bitstream. While our goal is not to present a comprehensive study on sample variation, our results can be used to give an indication on what to expect when the heat cores are transferred to other FPGAs within the same device family. Table 3 shows the temperature increases for the seven heat cores running at their respective frequency of maximum temperature yield. Highlighted are the lowest (italic) as well as the highest (bold) temperature gains achieved. The column 'Delta' denotes the difference between minimum and maximum temperatures.

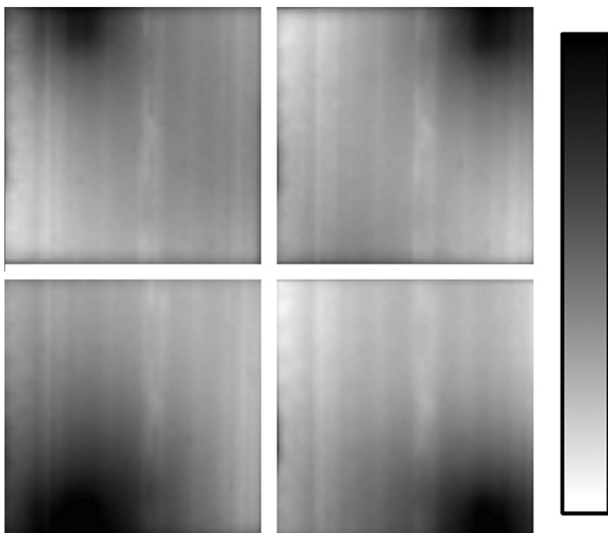


Fig. 12. Four infrared images of the FPGA captured in three second intervals. The scale ranges from 80 °C (white) to 95 °C (black).

As can be seen in the table above, the sample variation seems especially pronounced for the LUT oscillator. While we do not know the cause for this variation, likely causes are process variation and variations in the FPGA's power supply. To confirm that these results do indeed reflect sample variation as opposed to variations in the experimental setup (exact position of the temperature probe, ambient conditions, etc.) we repeated the tests for the LUT oscillator on all four FPGAs.

The results of this can be seen in Fig. 11, which shows the temperature progression for the LUT oscillator in two independent test runs (left versus right subfigure). While this is no replacement for a more rigorous treatment of errors in the experimental setup, the comparison demonstrates that sample variation as opposed to experimental variation is the main cause for the observed temperature differences between the four different FPGAs.

5.4. Spatial temperature gradients

While the results presented above show that it is clearly possible to generate significant temperature increases from within the reconfigurable fabric of a modern FPGA, a question that remains is, how the heat cores affect the spatial temperature distribution of the chip. More specifically, we want to quantify the maximum spatial temperature gradient that we can generate using our most effective heater. Spatial temperature gradients are of particular interest for researchers in the field of thermal thread management as they call for sophisticated spatial thread mapping strategies and the balancing of computational activity on the chip at runtime. In order to demonstrate that such gradients can indeed be achieved we stripped the package off of one of our Virtex5 LX110T devices and captured the temperature distribution on the backside of the die (the side that is up in a flip chip package) with the help of a thermal imaging device.

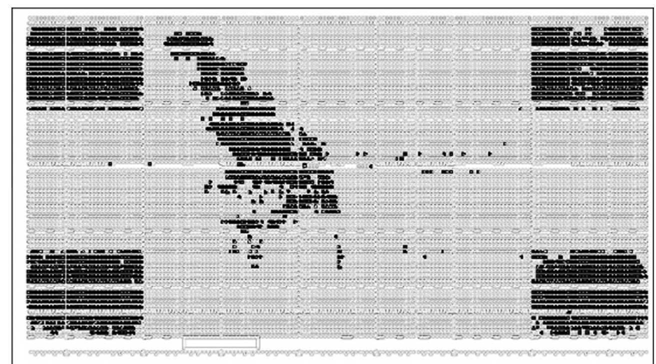


Fig. 13. Floorplan of a Virtex5 LX110T device with four LUT oscillator heat cores (one heat core per corner) and the other components of the experimental setup (central structure)

Please cite this article in press as: A. Agne et al., Seven recipes for setting your FPGA on fire – A cookbook on heat generators, Microprocess. Microsyst. (2013), http://dx.doi.org/10.1016/j.micpro.2013.12.001

Fig. 12 shows the temperature profile of four of our LUT oscillator heat cores at the corners of the FPGA. The cores were activated for three seconds each, in clockwise order.

Fig. 13 shows the spatial location of the circuit as seen in the Xilinx FPGA Editor. Due to the low thermal resistance of the die, the die acts as a spatial low-pass filter (heat source is blurred) for the observed temperature distribution. A significant temperature gradient of up to 10 °C over the entire die can still be observed across the device. This result reinforces the need for future temperature-aware configuration strategies.

6. Conclusions

In this paper, we propose seven dedicated cores that utilize different subsets of the FPGA resources in order to create heat. We focus on look-up tables, flip-flops, digital signal processor blocks and RAM blocks. For our clocked heat-generating cores we additionally analyze the influence of clock frequency on heat generation. During our experiments, we were able to increase the device temperature by more than 81 °C using a large number of minimal ring oscillators on a Virtex-5 LX110T FPGA in a time interval of 30 min. For our LUT and FF-based heaters, which include the two hottest ones, we only utilized about 21% of the available slices. For the most efficient heater, the LUT oscillator heater, we could generate spatial temperature gradients of up to 10 °C over the entire die in only three seconds.

In summary, we successfully demonstrated the generation of high temperatures on today's FPGAs. As expected, we could observe that a higher clock frequency results in higher temperatures and that most heat can be generated using excessive routing combined with high frequencies. To achieve a certain temperature goal, a designer can thus trade FPGA resources for the clock frequency. This paper—a cookbook on heat generators—is intended to serve as a guideline for setting up experiments for heat generation on FPGAs. Such experiments will be required 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 that have so far only been evaluated using simulations.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme under Grant Agreement No. 257906 and the Collaborative Research Centre "On-The-Fly Computing" (SFB 901).

References

[1] M. Salehi, M. Samadi, M. Najibi, A. Afzali-Kusha, M. Pedram, S. Fakhraie, Dynamic voltage and frequency scheduling for embedded processors considering power/performance tradeoffs, IEEE Trans. Very Large Scale Integrat. (VLSI) Syst. 19 (10) (2011) 1931–1935.

[2] F. Mulas, D. Atienza, A. Acquaviva, S. Carta, L. Benini, G. DeMicheli, Thermal balancing policy for multiprocessor stream computing platforms, IEEE Trans. CAD Integrat. Circ. Syst. 28 (12) (2009) 1870–1882.

[3] T. Ebi, D. Kramer, W. Karl, 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, P. Jones, Hotspot Mitigation using dynamic partial reconfiguration for improved performance, in: International Conference on Reconfigurable Computing and FPGAs (ReConFig), 2009.

[5] E. Kursun, C.-Y. Cher, Temperature variation characterization and thermal management of multicore architectures, IEEE MICRO (2009) 116–126.

[6] Y. Ge, P. Malani, 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 (2005) 10–16.

[8] Seyab, 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, L. Li, A thermal–mechanical coupled finite element model with experimental temperature verification for vertically stacked FPGAs, Microelectron. Eng. 91 (2012) 24–32, http://dx.doi.org/10.1016/j.mee.2011.11.011.

[10] T. Ebi, H. Rauchfuss, A. Herkersdorf, 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, 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, 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, C. Plessl, Measuring and predicting temperature distributions on FPGAs at run-time, in: International Conference on Reconfigurable Computing and FPGAs (ReConFig), 2011.

[14] C. Tradowsky, E. Cordero, T. Deuser, M. Hübner, J. Becker, Determination of on-chip temperature gradients on reconfigurable hardware, in: International Conference on Reconfigurable Computing and FPGAs (ReConFig), 2012.

[15] N. Kamoun, L. Bossuet, A. Ghazel, Correlated power noise generator as a low cost dpa countermeasures to secure hardware aes cipher, in: International Conference on Signals, Circuits and Systems (SCS), 2009.

[16] M. Happe, H. Hangmann, A. Agne, C. Plessl, Eight Ways to put your FPGA on fire – a systematic study of heat generators, in: International Conference on Reconfigurable Computing and FPGAs (ReConFig), 2012.

[17] T. Kumar, T. Al Saleh, S.R.S. Prabaharan, H.A.F. Mohamed, An FPGA based temperature controller for differential thermal analyzer, in: ICROS-SICE International Joint Conference, 2009.

[18] T. Kumar, H.A.F. Mohamed, B.A.C.M. Naleem, V. Ganeish, An FPGA based real-time remote temperature measurement system, in: International Conference on Electronic Devices, Systems and Applications (ICEDSA), 2010.

[19] Virtex-5 FPGA Packaging and Pinout Specification, June 2012. <http://www.xilinx.com/support/documentation/user_guides/ug195.pdf>.

[20] Virtex-5 FPGA System Monitor User Guide, February 2011. <http://www.xilinx.com/support/documentation/user_guides/ug192.pdf>.

[21] Virtex-5 FPGA User Guide, March 2012. <http://www.xilinx.com/support/documentation/user_guides/ug190.pdf>.

Andreas Agne is currently a PhD student at the Computer Engineering Group at the University of Paderborn, Germany. He holds a diploma degree in Computer Science (University of Paderborn, 2010). His research interests include operating systems and self-adaptation strategies for reconfigurable heterogeneous multi-core architectures.

Hendrik Hangmann is a Computer Science student at the University of Paderborn, Germany. He earned his Bachelor degree in 2012 and is currently working on his Master degree. He is currently employed as a student research assistant at the Computer Engineering Group in Paderborn, where he investgates the thermal characteristics of reconfigurable devices.

Markus Happe is a PhD student in the field of Computer Engineering at the International Graduate School Dynamic Intelligent Systems in Paderborn, Germany. He holds a master degree in computer science (University of Paderborn, 2008). Since 2013 he works as a research assistant in the Communication Systems Group at the ETH Zurich. His research interests include heterogeneous multi-core architectures, embedded operating systems, and self-adaptation strategies.

Marco Platzner is Professor for Computer Engineering at the University of Paderborn. Previously, he held research positions at the Computer Engineering and Networks Lab at ETH Zurich, Switzerland, the Computer Systems Lab at Stanford University, USA, the GMD – Research Center for Information Technology (now Fraunhofer IAIS) in Sankt Augustin, Germany, and the Graz University of Technology, Austria. He holds diploma and PhD degrees in Telematics (Graz University of Technology, 1991 and 1996), and a Habilitation degree for the area hardware-software codesign (ETH Zurich, 2002). His research interests include reconfigurable computing, hardware-software codesign, and parallel architectures.

Christian Plessl earned a PhD degree in Computer Engineering from ETH Zurich in 2006, and a Dipl.-Ing. degree in Electrical Engineering in 2001, also from ETH Zurich. Since 2011, he has been assistant professor for Custom Computing at the University of Paderborn, Germany. His current research interests include parallel and reconfigurable computer architectures, hardware-software codesign and self-aware adaptive computing systems.