UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Department of Computer Science
Faculty for Electrical Engineering, Computer Science and Mathematics
Computer Engineering Group

EVOLUTION OF HEAT FLOW PREDICTION MODELS FOR FPGA DEVICES

Masterthesis

submitted by

Hendrik Hangmann
Am Rippinger Weg 1c
33098 Paderborn
Matriculation number: 6533440

Paderborn, April 17, 2015

Gutachter____
Herr Dr. Paul Kaufmann

Gutachter____
Herr Jun-Prof. Dr. Christian Plessl

Hendrik Hangmann : *Evolution of Heat Flow Prediction Models for FPGA Devices* , © University of Paderborn, April 17, 2015

CONTENTS

1 Introduction 1
1.1 Motivation 1
1.2 Thesis Structure 2
2 Heat Modeling of Integrated Devices 3
2.1 Heat Models using physical device models 3
2.1.1 Discrete Cosine Transform Approach 3
2.1.2 HotSpot 3
2.2 Extension to On-Line Evolution of Heat Models 4
2.3 Temperature Measurement Methods 5
2.3.1 Built-In Temperature Sensor 5
2.3.2 Infrared Cameras 6
2.3.3 Ring Oscillators 7
2.3.4 Discussion on Accuracy and Calibration 10
2.4 Temperature Generation Methods 11
3 Online Evolution of Heat Models 17
3.1 Methodology 17
3.1.1 Temperature Model Definition 17
3.1.2 Learning and Optimizing Algorithms 17
4 Experiments and Evaluation 19
4.1 Experimental Setup 19
4.1.1 Architectural Setup 19
4.1.2 System Setup 19
4.2 RC-Network 19
4.2.1 Results for several Learning Algorithms 19
5 Conclusion 21
A Appendix 23

bibliography 25

LIST OF FIGURES

Figure 1 The temperature in the field region is derived from the power in the source region cf. [?] . . . 3
Figure 2 Stacked layers in a typical Ceramic Ball Grid Array (CBGA) package [?] . . . 4
Figure 3 RC-Network with two layers [?] . . . 4
Figure 4 Four Infrared (IR) images of the Field-Programmable Gate Array (FPGA) with temperatures from 80°C (white) to 95°C (black) cf. [?] . . . 7
Figure 5 A basic Ring Oscillator, consisting of an odd number of inverters . . . 8
Figure 6 Simplified schematic of a temperature sensor with capture counter cf. [?] . . . 9
Figure 7 A simplified overview of a Virtex-5 FPGA . . . 13
Figure 8 Pipeline consisting of Lookup Tables (LUTs) and Flip-Flops (FFs) . . . 14
Figure 9 Pipeline consisting of FFs . . . 14
Figure 10 FPGA-Implementation of a 1-level Ring Oscillator (RO) . . . 15
Figure 11 Architecture of the experimental setup . . . 19
Figure 12 FPGA experimental Setup . . . 20

LIST OF TABLES

Table 1 Advantages and Disadvantages of several temperature measurement techniques . . . 12

Table 2 The different experiment's heat patterns . . 20

ABBREVIATIONS

ADC	Analog-to-Digital Converter
BRAM	Block RAM
CBGA	Ceramic Ball Grid Array
CLB	Configurable Logic Block
CLK	Clock Signal
CMOS	Complementary Metal-Oxide-Semiconductor
DCM	Digital Clock Manager
DSP	Digital Signal Processor
FF	Flip-Flop
FIFO	First In First Out
FPGA	Field-Programmable Gate Array
IO	Input-Output
IR	Infrared
IC	Integrated Circuit
LUT	Lookup Table
PLL	Phase Locked Loop
RAM	Random Access Memory
RC	Resistance Capacitance
RO	Ring Oscillator
VHDL	Very High Speed Integrated Circuit Hardware Description Language
VLSI	Very Large Scale Integration
XOR	Exclusive OR

1

INTRODUCTION

1.1 motivation

Nowadays there is a large field of research in thermal effects on Very Large Scale Integration (VLSI) systems. Due to their shrinking device structures and the ever increasing power consumption, these state-of-the-art devices may cause dramatically increased thermal effects and high temperatures. This again could have consequences on the function and reliability of VLSI circuits. By the reason that the carrier mobility is degraded and the interconnect resistivity increases, the devices could suffer from decreased execution speed and longer interconnect delays [**?**], which is out of place in hard real-time systems and other today's fields of applications. Beyond that those timing errors can eventually lead to the premature occurrence of hard errors [**?**]. Hence local hot spots and high temperature gradients can lead to a shortened device life time and a shrunk package reliability.

As a start it is highly recommended to monitor the on-chip temperature. For that reason many devices feature a built-in temperature sensor. But moreover it is important to predict the temperature at any place on the die at a given heat flow. By knowing the thermal conductivity of the die and its layers there are several ways to model the heat flow and to predict the die's temperature. In cases that the die's structure, e. g. the number of layers and its thermal conductivity is not known, for example in early-state VLSI design or FPGA based systems, reconfigurable devices may only require the on-line learning of a heat flow and temperature model.

In order to achieve this, FPGAs benefit from their reconfigurability. Thus it is possible to implement heat generating circuits on the one hand and a network of temperature sensors on the other hand. In combination with a Resistance Capacitance (RC) network, which is commonly used for heat flow prediction models, it is possible to generate and learn a temperature model suitable for FPGAs.

Unlike the heat models that use physical device information, which derive the RC network's parameters by the die's physical characteristics, FPGA based heat models need to learn this parameters on-line. For means of calibration and parameter optimization it is mandatory to heat up the die, both uniformly and by creating hot spots, i. e. high temperature spots on the die. Other approaches which are using the on-line learning model have the drawback that it may be infeasible to start with a fully random initial parameter set, as they start learning with an almost optimal solution. Furthermore, randomized hill climbing is used for the purpose of optimizing, which may also be improvable for fully random initial parameters [**?**], e. g. by using algorithms like Simulated Annealing or Paricle Swarm Optimization. Furthermore, improved heat generating circuits [**?**] and temperature sensors [**?**] may also enhance the on-line heat model.

1.2 thesis structure

Section 2 gives an overview of heat models, that are derivable from the die's physical characteristics. Also the RC networks will be introduced in Section 2.1.2. Building on those, Section 2.2 explains the basic extension to an online heat model. In the following the needed temperature sensors and heat generating circuits are described. Section 3 describes the online heat model in detail. In the following several optimization algorithms are introduced, whose results are presented and evaluated in Section 4. Section 5 summarizes and concludes this thesis and provides an outlook.

2

HEAT MODELING OF INTEGRATED DEVICES

2.1 heat models using physical device models

2.1.1 Discrete Cosine Transform Approach

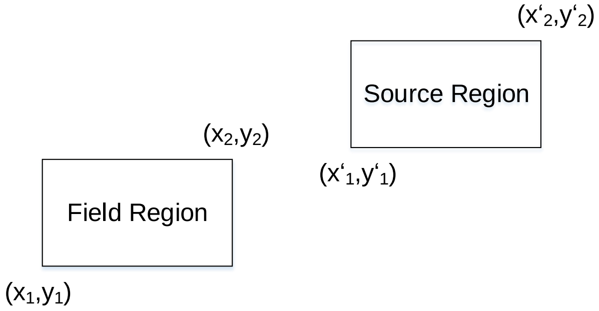

Figure 1: The temperature in the field region is derived from the power in the source region cf. [?]

$$\rho c_p \frac{\partial T(x,y,z,t)}{\partial t} = \nabla \cdot [k(x,y,z,T)\nabla T(x,y,z,t)] + g(x,y,z,t) \tag{2.1}$$

2.1.2 HotSpot

[?]

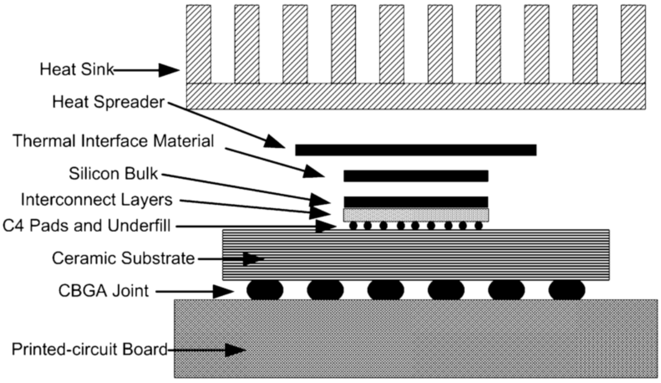

Figure 2: Stacked layers in a typical CBGA package [?]

RC- Networks

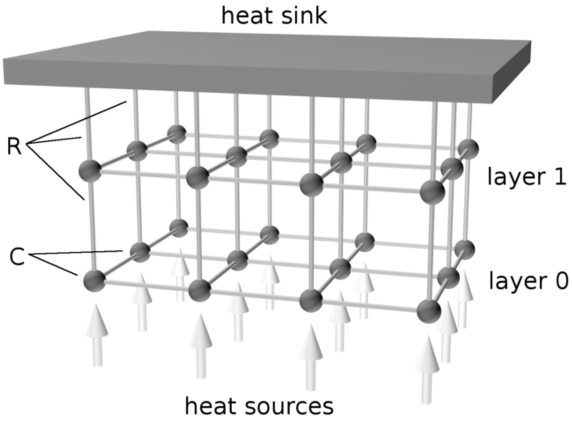

Figure 3: RC-Network with two layers [?]

2.2 extension to on-line evolution of heat models

- RCN
- heat up
- cool down
- measure temperature
- learn parameters
- evaluate

2.3 temperature measurement methods

2.3.1 Built-In Temperature Sensor

The most common way to measure temperatures on a VLSI chip is to implement a temperature sensor in Complementary Metal-Oxide-Semiconductor (CMOS). As stated in [?] the most effective way to achieve this goal is by using vertical bipolar transistors. This approach exploits the fact that the base-emitter voltage V_{be} of a bipolar transistor decreases approximately by $2\,mV°C$ and hence almost linearly with the temperature. Furthermore, the difference between two measured base-emitter voltages δV_{be} is almost linearly proportional to the absolute temperature T_{abs} (ptat) and can be expressed as depicted in Equation 2.2.

$$V_{ptat}(T_{abs}) = \frac{kT}{q} \cdot \ln(p) \tag{2.2}$$

The parameters for Equation 2.2 are the following:

- Boltzman's constant, $k = 1.38 \times 10^{-23}$
- Temperature in Kelvin, $T[°\,K]$
- Charge on an electron in coulomb, $q = 1.6 \times 10^{-19}\,C$
- Emission current density ratio p

This temperature sensor achieves an accuracy of $\pm 1°C$, by calibrating V_{be} at room temperature [?].

Also in modern FPGAs there is a trend of providing a pre-calibrated thermal diode, which are also provided at CMOS level. These devices can for example be found at the Virtex-5 FPGA by Xilinx, where the proportionality between the voltage and die temperature is as well exploited. Xilinx specifies this correlation with Equation 2.3 [?].

$$V_{ptat}(T_{abs}) = 10 \cdot \frac{kT_{abs}}{q} \cdot \ln(10) \tag{2.3}$$

Note that the emission current density ratio is here set to $p = 10$. Hence, the temperature coefficient of V_{ptat} is $-2\,mV/°C$.

Since the thermal diode is pre-calibrated, the temperature can be derived as depicted in Equation 2.4. To allow the further calculation it is mandatory to convert V_{ptat}, given as analog signal, into a digital number. The thermal diode on a Virtex-5 FPGA digitizes V_{abs} with the help of the built-in Analog-to-Digital Converter (ADC) and produces the 10 bit output ADC code $V_A DC$. The resulting maximum-measurement error of this on-chip temperature sensor is specified with °C [?].

$$T[^\circ C] = \frac{V_{ADC} \times 503.957}{1024} - 273.15 \qquad (2.4)$$

2.3.2 Infrared Cameras

Besides the above-named on-chip solutions of measuring the internal temperature, there is also the possibility to use IR cameras.

For instance, [?] measured a spatial thermal gradient of 2°C over 10 mm on a Xilinx Virtex II FPGA using an IR camera. Furthermore [?] used IR cameras to illustrate temperature gradients of 15°C, as depicted in Figure 4. The four IR camera pictures were taken in three second intervals. In each interval, temperature was generated in one of the chip's four corners. Inside the heated are the temperatures rose up to 95°C (depicted as black area), whereas the rest of the chip featured a temperature of 80°C.

Using an IR camera has the advantage of a high resolution. As stated in [?], IR cameras can achieve a spatial resolution of up to $30\mu m$ with a $0.5\times$ microscopy kit. Furthermore, if the camera is set up properly, i. e. a direct view on the silicon and knowledge of the emission values, this approach provides accurate temperature information. However, shortcomings of th IR camera approach are that one always needs direct view on the chip's silicon layer [?]. I.e. no heat sink, package or other material, where the emission values cannot be determined. Besides the high price for these devices, IR cameras cannot be employed in actual working conditions, because of their size [?].

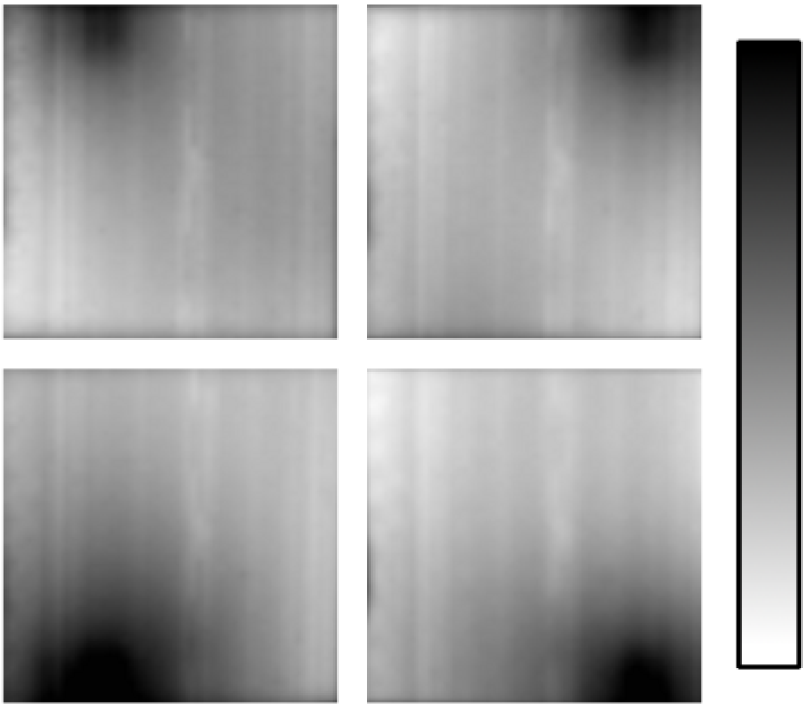

Figure 4: Four IR images of the FPGA with temperatures from 80°C (white) to 95°C (black) cf. [?]

2.3.3 Ring Oscillators

Nowadays, a widely spread technique to measure the internal on-chip temperatures of FPGA-based systems are based on ROs. These devices are composed of an odd number of inverters, which are connected in a chain. The endmost inverter's output is fed back to the first inverter. Figure 5 depicts such a basic RO. This leads to a device without a stable condition, causing each inverter's output to oscillate, i. e. the output Q toggles between 0 and 1 and maximum speed with frequency f_{osc}. A longer inverter chain leads to a lower frequency and in addition to less power consumption [?]. It is because of the fact that the RO's frequency f_{osc} is inversely proportinal to the on-chip temperature [?], that ROs can be used to measure the temperature on any location on the FPGA. The output frequency f_{osc} is dependant on the curcuit's delay. Furthermore, it is also known that ROs can be used in order to measure delay, leakage and dynamic power [?, ?].

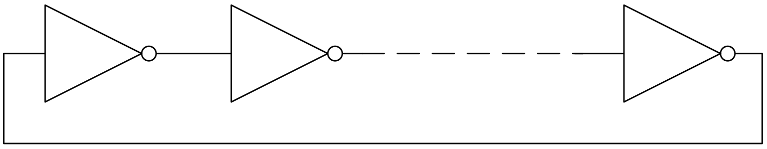

Figure 5: A basic Ring Oscillator, consisting of an odd number of inverters

Methodology

Measuring the internal on-chip temperature necessarily requires knowledge about the frequency f_{osc} of the RO. In order to estimate the frequency, several approaches [?, ?, ?, ?] make use of ROs combined with a capture counter. The counter, as depicted in Figure 6, is clocked with a system Clock Signal (CLK) and by the oscillating output Q of the RO. Q' toggles at the speed of Q and is sampled by CLK. Put simply, the capture counter samples the number S of oscillations at the signal Q' with the frequency clk. Afterwards the sampled number of oscillations can be derived to f_{osc}.

However, there are also differences in implementing the RO-based temperature sensor. These variable parameters are: number of inverters, system routing, length of measurement period t_m and the possible use of latches between the inverters. The use of latches was proposed in order to minimize the impact of routing [?]. For FPGAs designs, it is advisable to use latches instead of the additional wiring between the inverters.

The benefits of designing on an FPGA are the reconfigurability. This possible due to the Configurable Logic Blocks (CLBs) of which the FPGA is composed. The CLB however comprises slices, which contain LUTs and FFs. Hence, by using LUTs for the RO's inverters, the FFs can be used as latches without significant additional wiring.

In contrast to previous approaches, which used seven [?] or eleven inverters [?, ?] and no latches, a high-performance RO-based temperature sensor comprises 23 inverters and 24 latches [?]. The quality of the RO is derived by the sensor's resolution and noise/deviation. In addition to the utilization, the optimal measurement period t_m should not exceed the maximum length

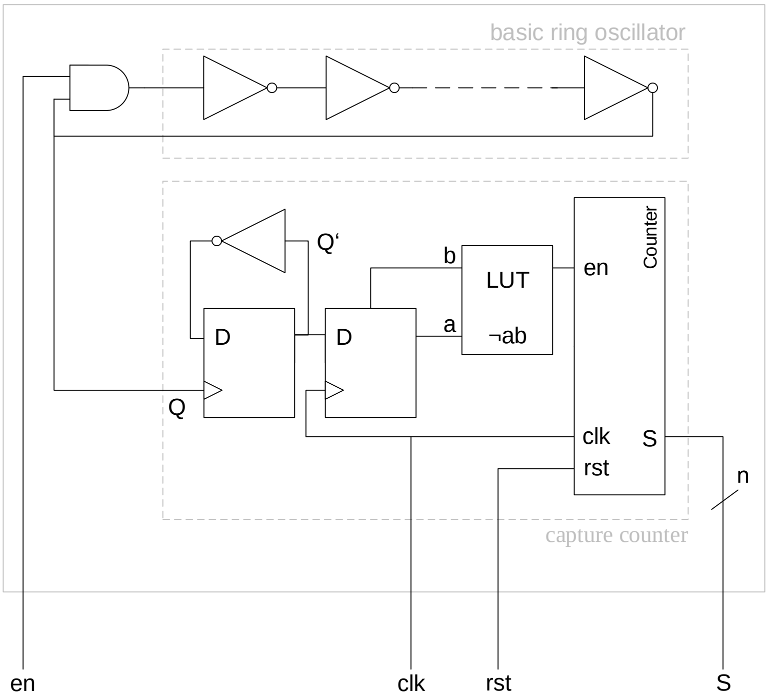

Figure 6: Simplified schematic of a temperature sensor with capture counter cf. [?]

of 2^{16} clock cycles, which is 655 µs, when the counter samples with 100 MHz. For longer measurement periods, there is a risk of self-heating, since ROs may lead to considerable temperature gradients [?].

As previously pointed out, the optimal measurement with the proposed temperature sensor includes the following steps [?]:

- Enable RO with 23 inverters and 24 latches
- Wait $2^{12} - 2^{16}$ clock cycles so that the RO can gain a constant frequency
- Sample Q' for t_m clock cycles
- Disable the RO
- Read out the counter value S.

Calibration Methods

Given the sensor count S of RO-based temperature sensors, it is not possible to predict a function that maps S to a temperature T. Instead, the sensors need to be calibrated with an pre-calibrated device, such as the built-in thermal diode, a temperature-controlled oven or an IR camera. While the device is heated up and cooled down afterwards, S is counted and the temperature is read in regular time intervals. For each sensor placed on the chip the linear mapping function is then determined by partial regression [**?**, **?**].

The following approaches request for a temporal temperature gradients equally distributed on the chip. Section 2.4 will give an overview of most common and useful methods for heating up the sensors, e.g. the RO-based temperature sensors.

As proposed in [**?**] the sensors were calibrated by an iron-constantan (Fe-CuNi) thermocouple which is placed in the centre of the package exactly measure the on-chip temperature T.

Section 2.3.2 already illustrated that IR cameras are able to exactly measure the on-chip temperature T [**?**], provided that the camera is pointing directly on the silicon and the emission value is known. Additionally of course, they can be used for calibrating the RO-based temperature sensors with high accuracy.

Also, temperature-controlled ovens might be used for calibrating the sensors. Because by setting and thereby knowing the surround temperature of the die, T and S can also be captured.

Another way to calibrate the sensors is to make use of the built-in thermal diode, which was presented in Section 2.3.1. While heating up the chip, the sensor data S and the diode temperature $T_{\mathtt{diode}}$ are captured. Note that this diode has ±4řC inaccuracy and is assumed to be in the centre of the FPGA [**?**].

2.3.4 Discussion on Accuracy and Calibration

The above-named approaches all fit well in the application of measuring temperatures and - if the devices are pre-calibrated - calibration of other sensors. However, each has benefits and shortcomings, which are listed in Table 1. The thermal diode, which is built-in in many nowadays FPGAs benefits from being easily accessible via the system monitor. Using this temperature sensor requires no additional implementation or calibration. On the other hand the specified accuracy of ±4°C may be to much. Furthermore, this accuracy can possibly not be met in real life. As depicted in [**?**], the diode has measured incorrect temperatures compared to an external thermometer. On a Xilinx Virtex-5 FPGA the difference between both devices amounts to 20.3°C. Until now, it has not been resolved why this differences occurs. Another disadvantage of using the thermal diode is that it is not clear where on the FPGA it is placed, though it is assumed to be in the centre of the FPGA [**?**]. Hence the diode is not able to detect on-chip hot spots or the specific temperature of a certain instantiated circuit on the FPGA.

A much more accurate approach is the use of IR cameras. It is able to measure the on-chip temperatures with a high resolution, i. e. everywhere on the FPGA. This again leads to the possibility of detecting hot spots. On the other hand, IR cameras are not only very expensive but very bulky, which hinders the in-field application. Beyond that, you need to have direct vision onto the silicon, which requires unpacking of the die.

The most effective way to measure temperatures on reconfigurable devices is using RO-based temperature sensors. It has nearly the advantages compared to IR cameras, and is beyond that easy to implement in reconfigurable computing devices. However, these sensor network needs to be calibrated. As this work aims to reconfigurable devices, i. e. FPGAs, the only handy and imaginable approach is the thermal diode, which could be a disadvantage in occurrence of incorrect measurements.

2.4 temperature generation methods

As said earlier most sensors, e. g. RO-based, have to be calibrated before use. In order to do so accurately, the Integrated Circuit (IC) needs to be heated up to learn about the sensor output in combination with a pre-calibrated sensor temperature output. But besides that, it might be interesting to heat up devices for the sake of evaluating the functionality. This may include fault tolerance like timing or soft and hard errors under occurrence of heat, especially for scientific use.

Measurement Technique	**Pro**	**Contra**
Thermal diode	Built-in and pre-calibrated	Inaccuracy and no hot spot detection
IR camera	Very accurate, high resolution and hot spot detection	Very expensive, no usage in field, requires view on silicon and knowledge about emission values
RO-based sensor	High resolution and hot spot detection	Calibration

Table 1: Advantages and Disadvantages of several temperature measurement techniques

Several approaches are using a temperature-controlled oven in order to heat up the chip [**?**, **?**]. This way, the IC is heated up and simultaneously the ROs are calibrated. In the first instance, these approaches use heat generation for calibration. But as a matter of principle, temperature controlled ovens can be used to heat up ICs uniformly distributed on the die. Practically, on the other hand, it might also be interesting to create local hot spots and generate spatial gradients on the chip in order to learn about the chip's thermal properties. Temperature controlled ovens do not support this kind of researching, not least because of the needed bulky external device. A much handier way to calibrate sensors might be the laser trimming [**?**], which is out of place in the field of reconfigurable computing, for instance for designing on FPGAs.

A better way, which also perfectly matches the advantages of reconfigurable computing is the use of dedicated heat generating circuits, specified in Verilog or Very High Speed Integrated Circuit Hardware Description Language (VHDL), following the approach of self-heating. By instantiating the low level components of a FPGA, several approaches achieved temperature increases and local hot spots on the die. These components are mostly placed in the FPGA's CLB and comprise

- LUTs
  Arranged in the CLB's slices. LUTs can implement any *n*-

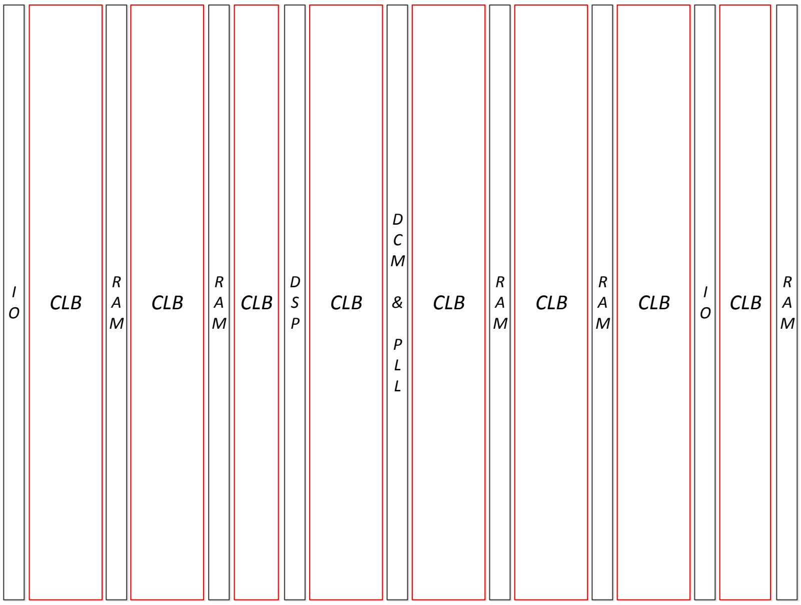

Figure 7: A simplified overview of a Virtex-5 FPGA

input logic function. The highest amount of input signals n is nowadays 6. Logical functions with a higher amount in input signals can be achieved by combining several LUTs

- FFs
  Also stored in slices next to LUTs. FFs can be initialized with a start value.
- Digital Signal Processors (DSPs)
  DSP blocks are compact and high-speed circuits and fulfil the special purpose of huge arithmetical and logical operations. Usually there is a special area between some CLBs
- Block RAMs (BRAMs)
  Also arranged between several CLB. BRAMs can for instance be used as First In First Out (FIFO), single or dual port Random Access Memory (RAM).

Figure 7 depicts the arrangement of these components on Virtex-5 FPGA. Additionally Input-Output (IO) blocks, Digital Clock Manager (DCM) and Phase Locked Loop (PLL) are added.

It is easy to see that the most effective way to generate temperature increases and especially local hot spots or spatial temperature gradients is to use the CLBs, i. e. LUTs and FFs. The designer is thereby almost not restricted to an area that can be heated. As illustrated in [?] the used FPGA increased its temperature up to 37°C by utilizing 80% of the CLBs. More precisely, the LUTs and FFs were arranged in a pipeline as depicted in Figure 8, clocked at 100 MHz. The output of each LUT was wired with the input of another FF, whose output was on the contrary was wired to another LUT and so on.

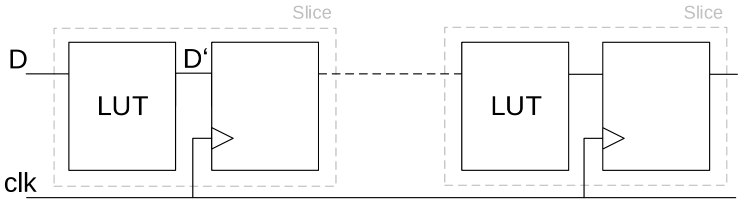

Figure 8: Pipeline consisting of LUTs and FFs

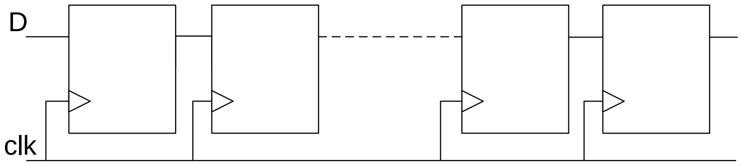

Figure 9: Pipeline consisting of FFs

Local hotspots were for instance created by using a FF pipeline as illustrated in Figure 9. As [?] depicted, it is possible to create a spatial gradient of 6.5°C by utilizing 10.000 FFs on the FPGA, which were clocked at 100%.

As a systematic study of possible heat generating circuits yielded, there are even more temperature increases and spatial gradients possible [?]. A average overall temperature rise of 81.2°C was achieved by implementing 1.000 single-level oscillators on the FPGA. As Figure 10 depicts, this oscillators were realized with LUTs, which fed back the outputs to the input *I0*. The logical function that was stored in this LUT is a simple Exclusive OR (XOR), which function as an inverter, when the enabling signal is activated. This slim and effective heat generating circuit also achieved a spatial temperature gradient of up to 10°C.

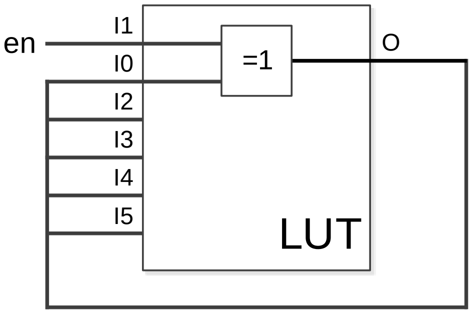

Figure 10: FPGA-Implementation of a 1-level RO

3

ONLINE EVOLUTION OF HEAT MODELS

3.1 methodology

3.1.1 Temperature Model Definition

$$mse(P) = \frac{1}{|N||M|} \cdot \sum_{i \in M} \sum_{j \in N} (T_s(P, t_i, j) - T_m(t_i, j))^2 \qquad (3.1)$$

3.1.2 Learning and Optimizing Algorithms

Simulated Annealing

Evolutionary Algorithms

Metropolis Algorithms

Gradient Descent Algorithms

Particle Swarm Optimizations

Hill Climbing

Evolutionary Algorithms

4

EXPERIMENTS AND EVALUATION

4.1 experimental setup

4.1.1 Architectural Setup

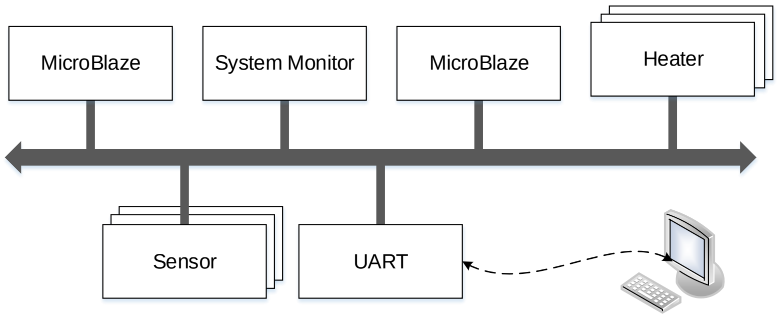

Figure 11: Architecture of the experimental setup

4.1.2 System Setup

4.2 rc-network

4.2.1 Results for several Learning Algorithms

- Discussion of Convergence
- Discussion of Accuracy
- Discussion of On-Line Suitability
- Discussion of Embedded Implementation

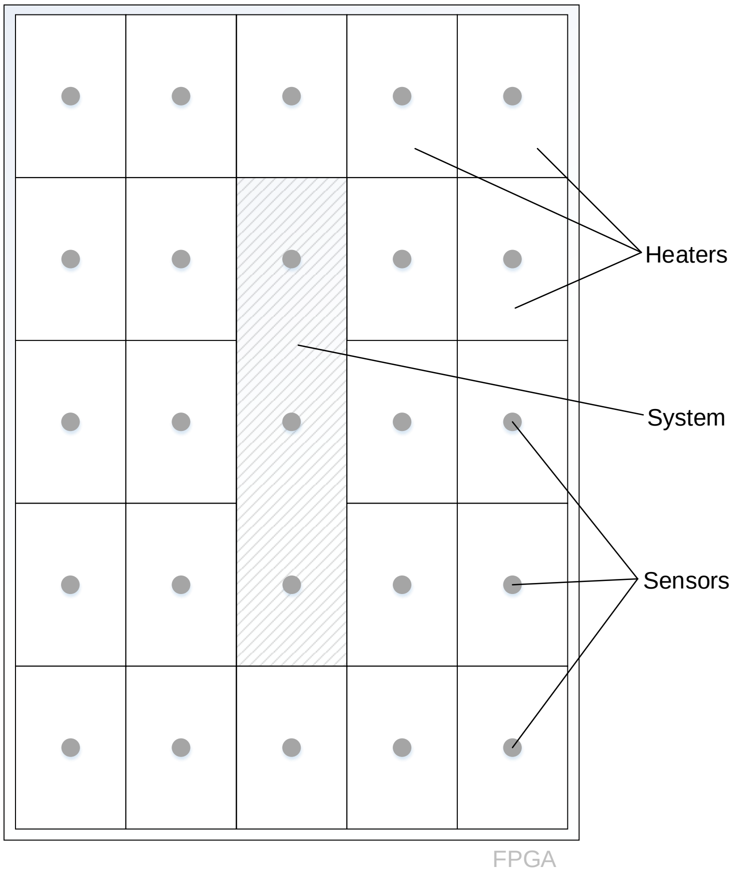

Figure 12: FPGA experimental Setup

Time [min]	**Heat Pattern**
0 – 10	All heaters activated
10 – 15	Five upper heaters activated
15 – 20	Five lower heaters activated
20 – 25	Five leftmost heaters activated
25 – 30	Five rightmost heaters activated
30 – 40	Cool-down. No heaters activated

Table 2: The different experiment's heat patterns

5

CONCLUSION

In this thesis I have presented crazy shit!

A

APPENDIX

The implementation of the temperature measurement system and all other files can be found on the appended CD. The folder structure is as follows.

EIDESSTATTLICHE ERKLÄRUNG

Hiermit versichere ich, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Paderborn, April 17, 2015

Hendrik Hangmann