Generating Adjustable Temperature Gradients on modern FPGAs

Hendrik Hangmann

February 27, 2012

1 Motivation

Nowadays there is large field of research in thermal effects. The shrinking device structure leads to dramatically increased thermal effects and high temperatures. This could have the consequence of soft (timing) errors, and contribute to the premature occurrence of hard errors [1]. Especially on modern FPGAs there were several studies on thermal management. But most of these thermal management techniques were studied using thermal simulations.
As a basis for further researches, which work on real-world systems, it is mandatory to be able to generate spatial thermal gradients. This could lead to the research of thermal imbalances and local hot spots on the chip. Therefore it is important to use dedicated heat-generating circuits.

2 Formulation of problem

There are several different ways to develop dedicated heat-generating circuits on a FPGA. The FPGA consists of three main parts: Slices, Block RAMs (BRAMs) and Digital Signal Processors (DSPs). These components get wired dynamically, depending on the desired functionality. The functionality is described in Hardware Description Languages such as VHDL.

Slices consists of several Look-Up Tables (LUTs) und Flip-Flops (FFs) and can be combined to any desired circuit. DSPs are hard-coded circuits for special uses, e.ġ. calculations. By the reason that e.ġ. multipliers would take a high amount of space if they were realized by LUTs and FFs, these tasks are mapped to special processors for that purpose. Because of their special purpose, they are highly space-saving. BRAMs are memory units.

It is now possible to instantiate Slices (LUTs and FFs), DSPs or BRAMs and use them specifically for heat-generating circuits. During the research it's eligible to identify the part that generates the most heat and if the mixture of parts outperforms that solution. Furthermore the influence of frequencies of these parts is varied to get most heat.

3 Objective target

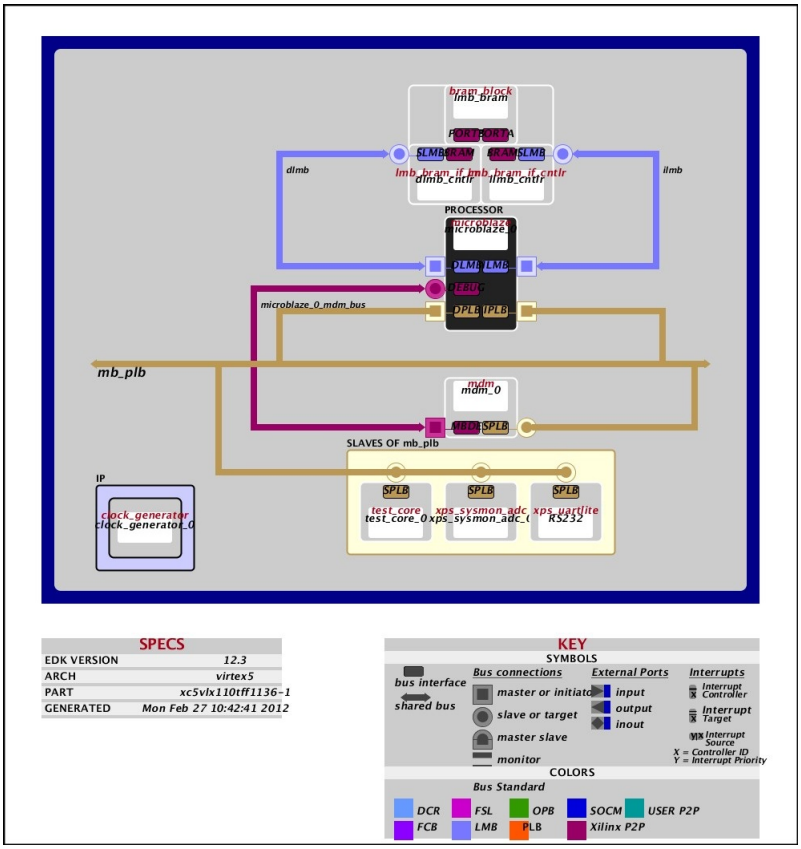

Figure 1: MicroBlaze System

The heat-generating cores are implemented as an embedded system into an existing system on the FPGA called MicroBlaze. MicroBlaze is a microcontroller used by *Xilinx*, with the characteristic that it is also given in VHDL. Figure 1 depicts this system with the MicroBlaze Processor. Via the PLB bus (*mb_plb*) several cores can be embedded to the system: e.ġ. heat-generating cores and the system monitor, from which the temperature can be read.

In order to read the temperature as correctly as possible it is mandatory to decouple the heat-generating cores from the MicoBlaze architecture. Therefore the MicroBlaze architecture and the heat-generating core are given each a defined range on the FPGA. I.ė. each core uses a defined set of Slices, DSPs and BRAMs.

After that, LUTs, FFs, DSPs and BRAMs should be individually investigated to identify the highest heat potential. It is eligible to study afterwards how these components can be combined to reach even more heat. These tasks are marked as milestones in the time schedule given in Figure 2.
Each of these cores will additionally work with diverse frequencies to be able to measure the influence of clocking. Optionally, I am going to examine the Xilinx Core Generator, to examine the quality of the generated cores.

4 Time management

Nr.	Process type	Beginning	Finishing	Predecessor
1	**Literature and related work**	**Do 01.03.12**	**Mi 01.08.12**	
2	Search Literature	Do 01.03.12	Fr 30.03.12	
3	Study literature	Mo 02.04.12	Mi 01.08.12	2
4	Write up thesis	Di 01.05.12	Mi 01.08.12	
5	**Research**	**Do 01.03.12**	**Di 31.07.12**	
6	Core generating tools	Do 01.03.12	Fr 01.06.12	
7	LUTs	Do 01.03.12	Fr 30.03.12	
8	Flip-Flops	Mo 02.04.12	Mi 02.05.12	7
9	DSPs	Do 03.05.12	Fr 01.06.12	8
10	BRAMs	Mo 04.06.12	Fr 29.06.12	9
11	Hybrids	Do 03.05.12	Fr 20.07.12	
12	Diverse clocking	Fr 16.03.12	Fr 20.07.12	
13	Evaluation	Do 05.04.12	Di 31.07.12	
14	**Submission**	**Mi 25.07.12**	**Mi 01.08.12**	

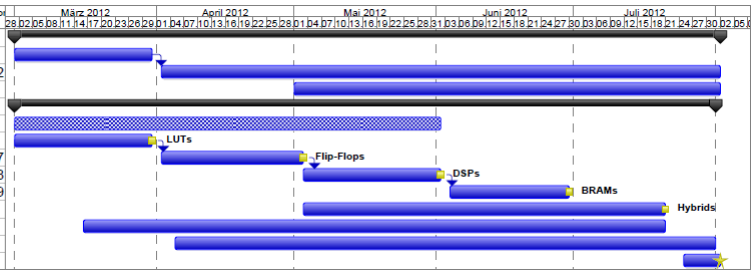

Figure 2: Time schedule

References

[1] Borkar Shekhar. Designing reliable systems with unreliable components. *Micro, IEEE*, 2006.