Cross-CUT Interference Present in Timing Extraction

Timothy Linscott (Seattle University, Computer Engineering), *SUNFEST Fellow*

André DeHon, Implementation of Computing Group

***Abstract*— Built-In Self Tests such as those developed by Wong, Sedcole, et al. [1] and Gojman [2] for measuring the internal path delays of a reprogrammable chip require maximizing isolation to eliminate interference between their components. This work demonstrates that placing anything more than a single measurement circuit on a chip at a time can influence the results of a measurement. The load placed on the clock by the measurement circuits is explored as a possible cause along with the ways that the different clock quadrants can be exploited to reduce the clock's influence on the measurements. This work also begins characterizing the noise introduced by running Timing Extraction measurements in parallel and demonstrates how this noise can be minimized.**

***Index Terms*— FPGA, Timing, Self-Measurement, Component-Specific Mapping, On-Chip Delay Measurement**

I. Introduction

Field-Programmable Gate Arrays (FPGAs) are reconfigurable, general-purpose integrated circuits. They are divided up into Logic Array Blocks (LABs) that contain the basic components to create any given digital circuit. FPGAs are favored by many industries because the function of the hardware can be defined and then upgraded after it is installed. This eliminates the need for each new device to be custom-built in a lengthy and expensive fabrication process. FPGAs work by using only a fraction of the large numbers of paths in their LABs. As the number of components on an FPGA grows, the fitter software that maps the programmer's circuit to the FPGA does not have enough information to do so optimally. When the logic is fitted poorly on an FPGA, excess heat is generated, path delays increase and the lifetime of the chip declines [3]. Additionally, the voltage transients from recently-used, nearby components, internal transistor leakage and electromagnetic interference from rapidly switching wires all contribute to timing delays in ways that simulation cannot accurately predict [4]. In order to provide the fitter with all the information needed to wire an FPGA optimally, we need to be able to both test the path delays in an FPGA and understand the effects that adjacent components have on each other's speeds. Tools such as Timing Extraction help determine the effects of process variation and interference between circuits on an FPGA. However, these tools are themselves prone to interference since they are built from the very circuits they seek to measure. This work demonstrates the measurements taken in Timing Extraction are dependent on the number and relative location of timing circuits placed on a chip. It also describes configurations of multiple simultaneous experiments that have minimal effect on each other's measurements.

II. Background

A. Delay Built-in Self-Tests and Timing Extraction

Delay Built-In Self Tests (BIST) are techniques like those developed by Wong, Sedcole et al. [1] and further developed by Gojman [5] for finding delays on an FPGA. In Timing Extraction—the delay BIST developed by Gojman—the FPGA is decomposed into Discrete Units of Knowledge or DUKs—combinations of wires, logic and registers that make up paths on an FPGA. The Circuit Under Test (CUT) is a path defined by the chosen DUKs and is placed between a launch and a capture register driven by a clock signal. The components making up the path are then configured as buffers to allow the signal to pass unchanged, as shown in Figure 1.

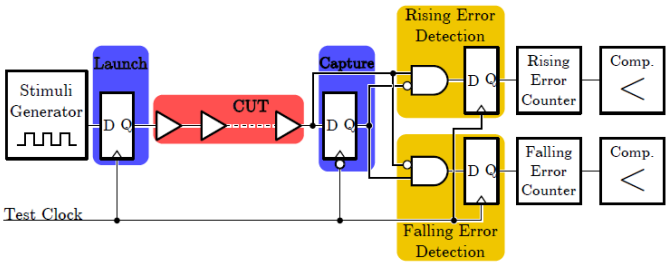

Figure 1. Diagram of a measurement circuit used in Timing Extraction.

When all of these components are fitted on the FPGA, they occupy three LABs in the Cyclone III architecture. When the BIST is run, the frequency of the clock is steadily increased until the contents of the launch and capture registers cease to match at the end of a clock tick half of the time. Half of the clock period—the amount of time the clock was at each voltage—is then said to be our path delay. In this way, we can find the path delay for both the rising clock transition and the falling transition [1]. When the path delays are known, they can used to solve for the delays over each DUK to give us the chip's delays at the finest possible granularity. To maximize precision, Timing Extraction is conducted in isolation. The FPGA controller is constrained to one side of the chip and the CUTs are kept as far apart as possible. Wong, Sedcole et al. ran the experiment with a 26x35 array of CUTs as well as with 52 sets of 16 CUTs [1] and Gojman placed 10 CUTs on the chip at a time [5]. Because measuring every path on an FPGA can be time consuming, Wong, Sedcole et al. and Gojman both recommended developing a parallel implementation of the BIST [1] [5].

B. Ring Oscillators

The ring oscillator is conceptually similar to the launch-and-capture model of Timing Extraction except that the ring oscillator follows a path of inverters rather buffers. Thus, it creates its own clock rather than needing to be supplied with

one. The oscillator is simply an odd number of inverters connected in a loop so that the output toggles between high and low at a predictable frequency. As with Timing Extraction, it can measure delays through internal logic and wires, but ring oscillators cannot measure over registers. These can be used as temperature sensors when the oscillator is primarily composed of transistors instead of wires. As the temperature rises, the transistor delay increases as given by Equation 2 and the frequency of the oscillator slows. Detection circuits can pick up the frequency changes and infer the temperature increase [4]. Because they are often implemented with a high number of inverters, the oscillator spans multiple LABs and makes for a good test of the overall temperature of an FPGA [6].

C. *Self-Heating Effects*

Whenever a path in an FPGA is used, it generates heat. The energy dissipated in a toggling wire is given by the equation:

$$E = \frac{\alpha_k}{2} V_{dd}^2 L C_k \quad (1)$$

Where α_k is the toggle rate, L is the wire length and C_k is the capacitance of the channel. We expect that this heat will spread over the chip proportionate to $1/r^2$ in accordance with basic physics. The current though the drain of a transistor I_d is proportionate to $e^{1/T}$ where T is the temperature in Kelvin. Since the drain current decreases with temperature, charge flows across the transistor slower. Because the transistor has a capacitance C, reaching a new voltage level V has a time delay T_d given by the following:

$$T_d = CV/I_d \quad (2)$$

Use of the wires and transistors in the FPGA generates heat which in turn decreases the amount of current that can flow through the transistors. Because less current can flow, the delays through the circuit increase since more time is needed to transition between voltage levels.

D. *Thermal-Aware CAD*

Thermal-Aware CAD attempts to incorporate these ideas of self-heating-induced delays into the routing and placement of components on FPGAs. Because using longer wires increases energy use, fitting software tends to place components as close together as possible. Because of the proximity, the components heat up and delays increase. Thermal-Aware CAD looks for a golden mean between energy use and heat control. Recent Thermal-Aware CAD has been able to reduce on-chip temperatures by 10-14°C using only mathematical approximations of the heat generation in a component [7].

III. Set-up and Process

A. *Cyclone III Architecture*

The experiments were conducted on a set of fourteen Arrow BeMicro FPGA Evaluation Kits with Cyclone III FPGAs model EP3C16F256C8N. The Cyclone III architecture uses a 65nm process technology and is optimized to minimize power consumption. The chip is laid out as a 40 by 28 grid of LABs with two columns reserved as memory and another two as multipliers. Each LAB has 16 Logic Elements made of up of a 4-input Look-up Table and a register. At each of the corners of the FPGA is a Phase-Locked Loop (PLL) which is used to drive the clocks in the experiment. Because of the granularity of the PLLs, the resolution of the clocks is limited to ±1.6ps. Figure 2 shows a diagram of a Cyclone III FPGA.

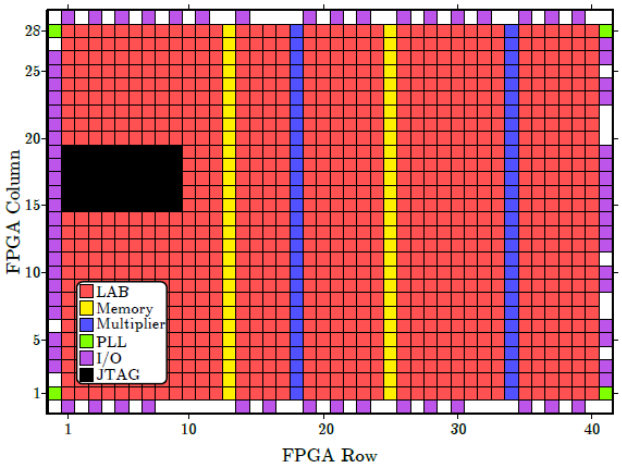

Figure 2. Resource diagram of the Cyclone III model EP3C16F256C8N FPGA.

The clock network of the Cyclone III is driven by a set of 20 global clocks. [8] These clocks drive the local clocks in the chip's four quadrants which in turn drive clocks that connect to the LABs along each row. Although the exact schematics of the clock network are not disseminated by Altera, experimental evidence implies that the four quadrants are four rectangles at the chip's corners with vertices at the chip's center. When the first component is added to a clock domain, a new local clock needs to be activated. When a new clock is activated, the activity in the local clock network increases because the local clock driver needs to expend additional energy on more clocks.

B. *Experimental Process*

The experiments conducted for this report were primarily studies into the relative locations of the CUTs. By changing the arrangement of the measurement circuits during Timing Extraction, we sought to influence the outcome in predictable ways. Thus, additional CUTs could act as both data collection devices and white noise generators. The complexity of the measurement circuit gave us several forms of noise, including clock loading, rapidly toggling wires and heat.

Paths were measured in three different ways: in isolation, in serial and in parallel. First, paths were isolated by forgoing placing other CUTs on the chip and measured to give a baseline for the path delay. As with Wong, Sedcole et al. and Gojman, the paths were measured in serial with other CUTs placed but only one active at a time. Paths were also measured in parallel so that multiple CUTs were active and generating noise all at once.

IV. Experimental Results

A. Interaction between CUTs

Running experiments in parallel is desirable because of the increase in speed, but doing so will also generate additional activity on the chip. This activity will lead to self-heating which will in turn slow down the paths tested. The magnitude of this effect will determine whether parallel experiments produce worthwhile results. To test the viability of running experiments in parallel, a set of paths were chosen and tested over a set of experiments. In one trial 15 CUTs were placed on the FPGA per experiment, then 7 and 8 CUTs, 5 CUTs and 3 CUTs. These same configurations were run both in serial and in parallel, and the objective path delays are given in Figure 3.

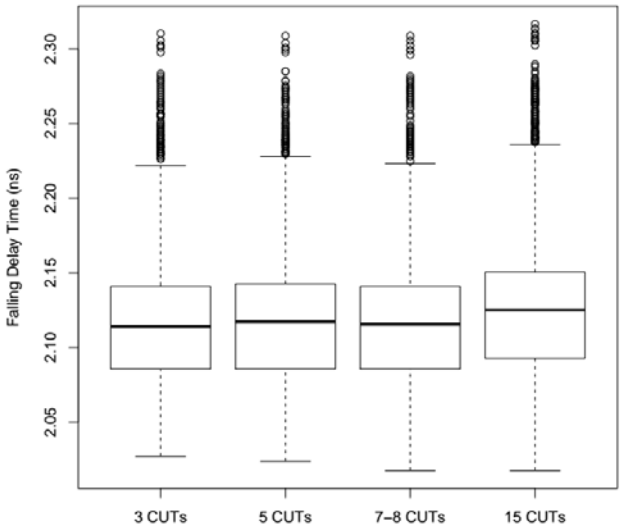

Figure 3. Box plot showing the delays recorded in experiments with different numbers of CUTs on the chip.

The mean rising delay for the 5 CUT and 7-8 CUT cases is ~1ps higher than that of the 3 CUT case. But the 15 CUT data is on average 11ps higher than the 3 CUT data, meaning that parallel activation of this many CUTs causes a measurable slowdown in the tested paths.

Because the distance between the CUTs varied along with the number of CUTs placed, a set of paths were chosen from the top right corner of the FPGA by the only unused PLL and a noise generating CUT was placed near the path. The CUTs were run in parallel so that the second CUT would be generating noise. Distances between 2 and 12 LABs away were chosen and the noise generator was placed both on the same row and on the same column, while keeping both CUTs in the same clock quadrant. The path delays with the noise generator in each position were correlated and are given in Figure 4. Cases where the noise generator was on the row are shown in blue and cases where it was on the column are in red.

When the noise generator is placed on the column, its distance to the CUT has little effect. But when it is placed on the same row, the effects are Gaussian tending towards higher delays when the noise generator is placed 12 LABs away. Rather than demonstrating the assumed behavior where added heat causes delay increases proportional to $1/r^2$, the results depended on some other factor.

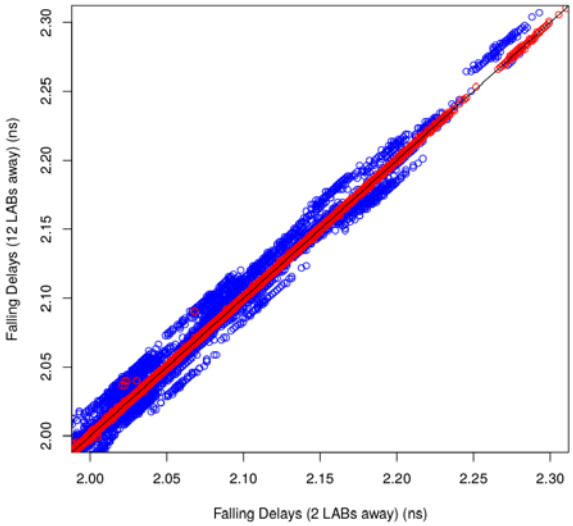

Figure 4. Relative timing delays when a noise generator is placed 12 LABs away versus 2 LABs away.

B. Clock Loading Effects

In order to reduce the effects that CUTs have on each other, the relative location between two CUTs was changed over a series experiments so that the delay of one could be found based on the position of another. The CUT to be tested was placed in the center of a clock quadrant at (30, 9) and the second was moved to positions within and surrounding the quadrant. These experiments were run both in serial and parallel. The falling delays of the CUT as a function of the second CUT's position are given in Figure 5 for the parallel experiment and in Figure 6 for the serial trial.

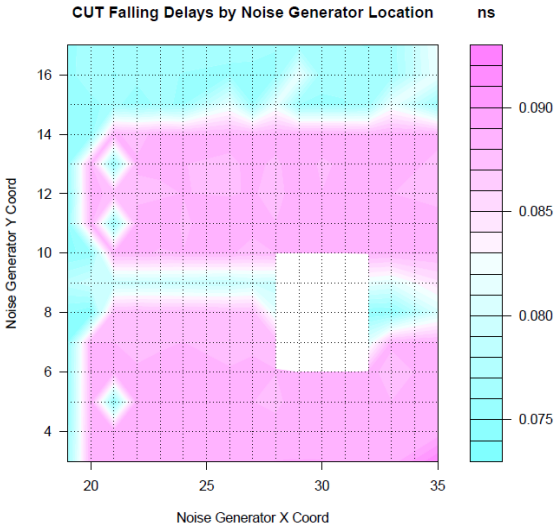

Figure 5. Falling delays over a CUT at (30,8) as a function of the position of a noise generator running in parallel, in ns. Note that there are no data points at x=25 or x=34 because of the memory and multiplier columns at those positions.

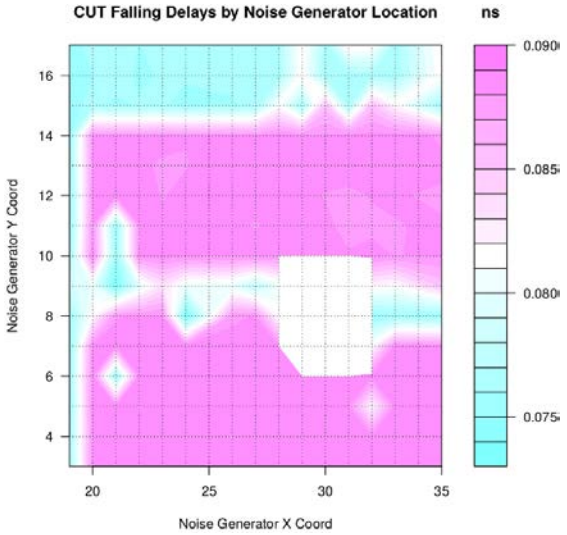

Figure 6. Falling delays over a CUT as a function of a noise generator running in series.

When the two CUTs are within the same quadrant, but occupying different rows, the path delays tended to be higher. When the CUTs are in different quadrants, the path delay drops. Note that although activating new clocks in different quadrants loads the global clock, it will not increase the activity on the clock local to the first CUT. However, when the two CUTs share a row or nearly share a row, the delays are almost as low as the out-of-quadrant case. If no new clock needs to be activated by this placement, the activity in the local clock remains roughly constant and little additional energy is introduced.

C. Viability of Parallelism

If the delays are lowest when the clock is not loaded by additional CUTs, can accurate results be taken in parallel so long as the CUTs are in different quadrants? Or does the addition of even a single CUT running in serial influence the measurement's accuracy?

Pairs of CUTs were placed on the FPGA so that in each experiment one CUT was placed on the lower half of the chip and the second was placed in a similar position in the same column in the quadrant above the first. In this way, a path from every LAB was measured both in serial and parallel. A baseline was created by measuring each path in complete isolation twice. The delta percent delays (DPD) were found for each path by finding the percentage that the isolated measurement increased or decreased when noise was introduced by the serial and parallel trials. These were compared with a control group consisting of the two identical isolated trials compared to each other. Figure 7 presents this data.

The DPDs are +0.04% on average and tend to stay between 0 and 0.08% in both trials, but may range up to ±1.5%. For an average path, a 0.08% change is 1.6ps, which is the size of a single clock step. Placing a second CUT in a different quadrant has a modest impact on performance. Turning this CUT on to run the experiment in parallel has no further impact on the outcome of the measurement.

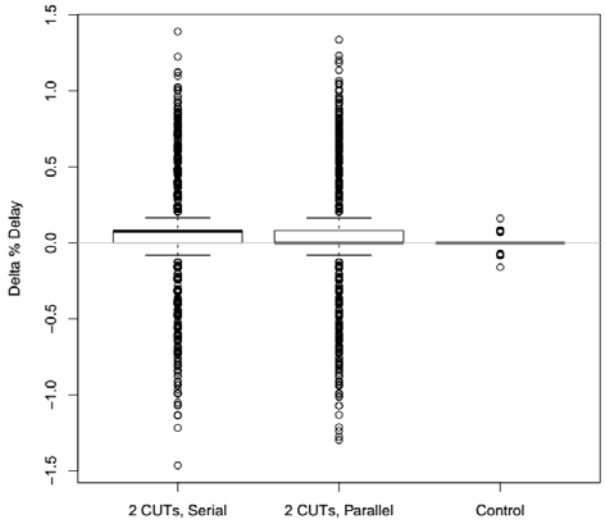

Figure 7. Delta Percent Delays comparing an isolated measurement to measurements in parallel and serial with a second CUT in a different quadrant.

To demonstrate the effects of testing multiple CUTs in the same quadrant, pairs of CUTs with equal relative distances were chosen, avoiding pairs that fell along the same row. Likewise, sets of twelve CUTs were chosen, with each set occupying a three-by-eleven block of LABs. These sets were run in serial and in parallel and the DPD was found with respect to the isolated case. Figure 8 compares these data.

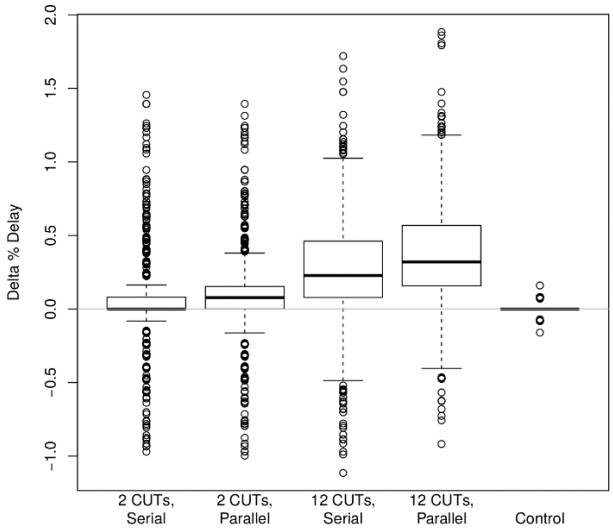

Figure 8. Delta Percent Delay for sets of two and twelve CUTs in the same quadrant both in serial and parallel.

Adding a second CUT even in serial usually has a modest effect, increasing the path delays by a small percentage. Turning this CUT on increases the delay by an average of 0.08%. Placing 12 CUTs rather than one increases the delays by 0.24% when they are run in serial and 0.38% when they are run in parallel, up to a maximum measured increase of 1.88%.

These experiments were continued by placing 4, 6, 12 and 24 CUTs into a single clock quadrant in one experiment. The mean delta percent delay was found when the CUTs were run in serial and parallel as compared to the isolated case. Figure 9 shows these delay increases as a function of the number of CUTs placed on the FPGA.

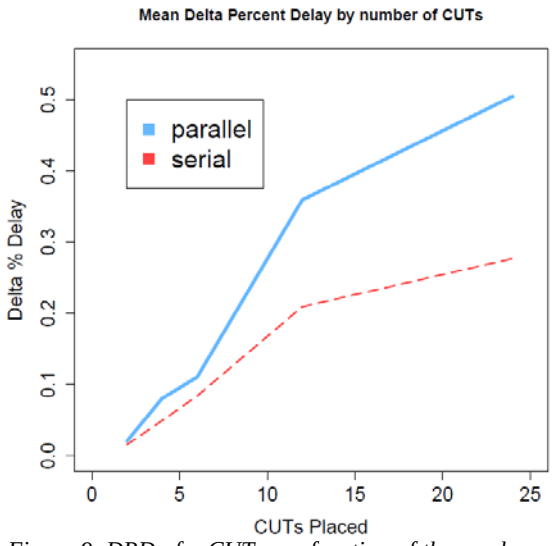

Figure 9. DPD of a CUT as a function of the number of other CUTs placed during the experiment. Serial trials are in orange and parallel trials are in blue.

As additional CUTs are placed into the experiment, the paths they measure will be slowed by each other's presence. However, when these CUTs are activated and the experiment is run in parallel, the effect they have increases by between 32% and 82%. While the effects of placing CUTs is lower when they are run in serial, the delays they induce can become substantial when too many of them are placed in a single quadrant.

V. Discussion and Conclusion

The measurement circuits used in Timing Extraction are prone to variation. Changing the configuration of the experiment by adding additional CUTs—even inactive CUTs in a different clock quadrant—changes the measured path delays by as much as 1.5% to 2%. Choosing the location of the CUTs by placing them in different quadrants or possibly in the same rows as each other can reduce, but not eliminate, this variation. Running the CUTs in parallel will always increase the noise on the measurements and often increase the path delays above the serial and isolated cases.

Whether these variations of 1.5% to 2% are a concern to the user depends on the application. In most practical applications, no path is run in maximum isolation, but will experience noise from nearby LABs. Further, running a complete timing analysis on the Cyclone III would require 2,736,556 distinct bitstreams to test each path individually and in complete isolation. [5] Such a process would be very time consuming, and may not be worth the increased accuracy of measurement.

VI. Further Questions

This work focused only on the interaction between CUTs used in Timing Extraction and depended on them for noise generation. To increase the precision in choosing clock resources, more finely controlled circuits could be developed to selectively activate individual clocks. Thus the effects of turning on new row-specific clocks could be studied. Likewise, because the CUTs are tied to the clocks, other effects including self-heating and crosstalk from toggling wires were not explored. These effects certainly can influence delays and could be characterized in further research.

The advantages to placing two CUTs in the same row have not been characterized. Doing so does mitigate clock loading, but less so than placing the CUT in a separate quadrant. Whether this can be used to increase parallelism is unclear.

Acknowledgment

The authors would like to acknowledge the support of the National Science Foundation, through NSF REU grant no. 1062672.

References

[1] J. S. J. Wong, P. Sedcole and P. Y. K. Cheung, "Self-Measurement of Combinatorial Circuit Delays in FPGAs," ACM Transactions on Reconfigurable Technology and Systems (TRETS), vol. 2, no. 2, June 2009.

[2] B. Gojman and A. DeHon, "GROK-INT: Generating Real On-Chip Knowledge for Interconnect Delays Using Timing Extraction," 2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), pp. 88-95, 11-13 May 2014.

[3] E. Stott, J. Wong, P. Sedcole and P. Cheung, "Degradation in FPGAs: measurement and modelling," FPGA '10 Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays, pp. 229-238, February 2010.

[4] K. Zick and J. Hayes, "Low-Cost Sensing with Ring Oscillator Arrays for Healthier Reconfigurable Systems," ACM Transactions on Reconfigurable Technology and Systems, vol. 5, no. 1, March 2012.

[5] B. Gojman, "GROK-FPGA: Generating Real On-Chip Knowledge for FPGA Fine-Grain Delays using Timing Extraction," 2014.

[6] J. J. L. Franco, E. Boemo, E. Castillo and L. Parrilla, "Ring oscillators as thermal sensors in FPGAs: Experiments in low voltage," Programmable Logic Conference (SPL), 2010 VI Southern, pp. 133,137, 24-26 March 2010.

[7] S. Bhoj, "Thermal aware FPGA architectures and CAD," International Conference on Field Programmable Logic and Applications, pp. 701-702, 8-10 September 2008.

[8] Altera Corporation, Cyclone III Device Handbook, 12 ed., vol. 1, San Jose, CA, 2012.