

# On-Line Sensing for Healthier FPGA Systems

Kenneth M. Zick and John P. Hayes  
Computer Science & Engineering Division, EECS Dept.  
University of Michigan  
Ann Arbor, MI 48109 USA  
{kzick, jhayes}@umich.edu

## ABSTRACT

Electronic systems increasingly suffer from component variation, thermal hotspots, uneven wearout, and other subtle physical phenomena. Systems based on FPGAs have unique opportunities for adapting to such effects. Required, however, is a low-cost, fine-grained method for sensing physical parameters. This paper presents an approach to on-line sensing that includes a compact multi-use sensor implemented in reconfigurable logic, methods for instrumenting an application, and enhanced measurement procedures. The sensor utilizes a highly-efficient counter and improved ring oscillator, and requires just 8 LUTs. We describe how to measure variations in delay, static power, dynamic power, and temperature. We demonstrate the proposed approach with an experimental system based on a Virtex-5. The system is instrumented with over 100 sensors with a total overhead of only 1.3%. Results from thermally-controlled experiments provide some surprising insights and illustrate the power of the approach. On-line sensing can help open the door to physically-adaptive computing, including fine-grained power, reliability, and health management schemes for FPGA-based systems.

## Categories and Subject Descriptors

C.4 [Performance of Systems] – *Measurement techniques, Reliability, availability, and serviceability, Fault tolerance, Performance attributes*; B.7.3 [Integrated Circuits]: Reliability and Testing; B.8.1 [Performance and Reliability]: Reliability, Testing, and Fault-Tolerance

## General Terms

Measurement, Experimentation, Reliability, Performance, Design

## Keywords

FPGAs, on-line sensing, process variation, leakage, static power, dynamic power, reliability, temperature, ring oscillator, health management, physically-adaptive computing.

## 1. INTRODUCTION

As semiconductor technology advances, electronic systems are suffering from increasing physical difficulties. The amount of manufacturing variation in transistors and interconnect is growing dramatically [12], resulting in chips with many marginal regions and components. Combined with variations in application

activity, this is leading to excessive power consumption, thermal hotspots, and reduced performance and reliability. Given their fine-grained reconfigurability, FPGA-based systems have a unique potential for both measuring and managing physical effects. We refer to such a paradigm as physically-adaptive computing.

A traditional approach to handling physical variation is to estimate worst-case conditions and to add extra margin into a system. However, this is becoming increasingly sub-optimal as the amount of variation soars. Moreover, it is becoming harder to accurately model and simulate the resulting subtle and complex physical interactions. For example, simulations of thermal gradients across FPGA dies show considerable disagreement. One simulation-based study suggests a variation of 8.6°C [18], while a similar simulation sets the number at an alarming 20°C [24]. The limitations of simulation point to the need for accurate measurements in silicon.

One alternative approach is for the variations of each chip to be mapped, and for system implementations to be customized to the chip before deployment. A proposal for such “chip-wise” customization is presented in [5], using variation-aware place and route. However, manufacturers do not yet have the incentive to generate and provide detailed variation maps, and only in specialized cases can a system-maker shoulder the burden of chip-wise testing, CAD, and bitstream handling. Moreover, a pre-deployment approach cannot help with infant mortality, changes in workload, or other parameter shifts that occur in the field.

What is required, therefore, is a low-cost, *on-line* method for making fine-grained physical measurements in FPGA systems.

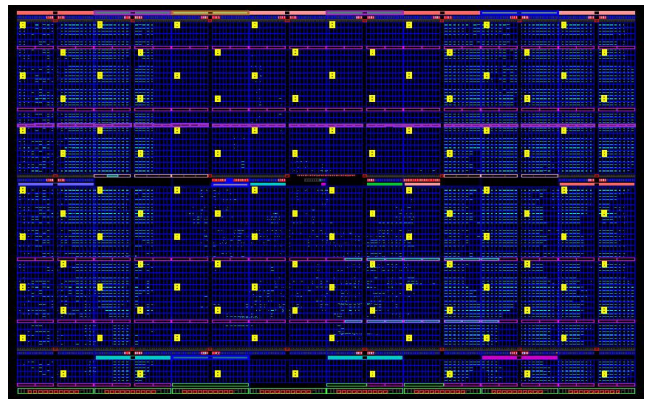


Figure 1. Floorplan of an FPGA-based system instrumented with an array of compact multi-use sensors (light-colored rectangles) implemented in reconfigurable logic.

© ACM, (2010). This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Proc. FPGA'10, <http://doi.acm.org/10.1145/1723112.1723153>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FPGA'10, February 21-23, 2010, Monterey, California, USA.

Copyright 2010 ACM 978-1-60558-911-4/10/02...\$10.00.

While some FPGA platforms include a single analog temperature and voltage sensor, they do not provide a means of measuring a variety of quantities at arbitrary locations. Thus the challenge is to use the standard digital logic in a reconfigurable fabric to measure as many key physical parameters as possible, and as accurately and unobtrusively as possible.

The key research questions that need to be addressed are the following. Which types of physical variations in FPGA systems can be measured? How can they be measured and with what overhead? How is such data useful? This paper addresses all of these questions, and makes the following contributions:

- The design of a flexible, compact, and easy-to-use sensor implemented in reconfigurable logic
- Improved procedures for measuring variations in delay, leakage power, dynamic power, and temperature
- A case study of a Virtex-5-based experimental system instrumented with over 100 sensors, including results from thermally-controlled experiments

The remainder of the paper is structured as follows. We discuss related work and motivating applications in Section 2. We then introduce the design of a new low-cost sensor and instrumentation in Section 3, and describe methods of on-line measurement in Section 4. We present experimental results in Section 5 and conclude in Section 6.

## 2. BACKGROUND

Physical phenomena are having a growing impact as semiconductor technology continues to scale. Phenomena of interest include inherent variations in components (transistors, interconnect) as well as non-uniformities in operating conditions such as temperature and supply voltage. Component variations result from the manufacturing process and also lifetime wearout. Variations within a die (*intra-die*) are in part spatially-correlated and in part stochastic from component to component. In this work we are largely concerned with spatially-correlated intra-die variation.

We will briefly discuss some of the work related to on-line sensing of physical effects. Reconfigurable platforms sometimes include an analog voltage sensor and temperature sensor connected to an A/D converter, such as with the Xilinx System Monitor [31]. However the sensor coverage is limited to a single fixed location. An early paper by Quénot *et al.* proposes the use of distributed ring oscillators to measure the voltage and temperature profile across an entire chip [20]. Works such as [14] describe associated measurement procedures. Regarding FPGAs in particular, there have been several attempts at on-line thermal sensing using ring oscillators, including [4][7][11][16]; unfortunately, these methods entail relatively high overhead, and are becoming less accurate since the temperature dependence of advanced reconfigurable logic ( $\leq 65\text{nm}$ ) increasingly is swamped by voltage noise. Additional research focuses on finding an appropriate arrangement and density of sensors across the reconfigurable fabric [19]. In this paper we will build upon the approaches mentioned above.

There has been very little work on on-line measurement of FPGA parameters other than temperature. An example of off-line characterization of FPGA component delays is the work by Sedcole and Cheung [22]. While the present paper is focused on *correlated* variation, an important and related issue is sensing stochastic variation. In [32] we address the sensing of stochastic

variations in component single event upset susceptibility. A recent paper proposes a method of characterizing extreme stochastic variations in threshold voltage for a projected 5nm technology [9]. There has been much work in the area of on-line testing for hard defects, which are in a sense highly-localized variations. As just one example, Rubin and DeHon introduce an approach to mapping and avoiding hard defects when a bitstream is loaded [21]. Finally, the general problem of how a system can effectively sense its own physical parameters is being studied in other contexts such as autonomic data centers [10] and self-modeling robotic systems [3].

A brief consideration of some relevant application domains helps to motivate fine-grained on-line sensing. Many embedded systems are required to exhibit low power, high reliability, and high autonomy. Examples include the FPGA-based systems deployed in spacecraft, UAVs, and similar systems, such as SEAKR's Virtex-4-based reconfigurable computer launched on the TacSat-3 mission [26]. Certain physical effects can be handled with coarse-grained control techniques (e.g. chip-wide voltage/frequency scaling, body bias, or power gating), but extreme variation calls for a finer-grained and more adaptive approach. Variation-aware reconfiguration can potentially improve embedded system efficiency and extend lifetimes.

Another relevant application domain is high-performance reconfigurable computing. As with embedded systems, there is a need for fine-grained sensing of power, temperature, and wearout. One difference is that high-performance systems may be more amenable to mitigation methods such as swapping out a failing FPGA, thermal-aware task scheduling [15], or on-line design optimization. Note that high-performance FPGA co-processors are increasingly created via high-level synthesis, meaning low-level physical effects such as hotspots may go untreated. In fact, there are proposals for performing lightweight circuit synthesis at run-time [27]. The trend towards high-level design increases the need for handling low-level effects at run-time.

## 3. INSTRUMENTATION

To enable effective on-line adaptation for physical effects, an FPGA-based system needs to be instrumented with the ability to measure physical parameters. The nature of spatial variations across an FPGA chip necessitates that measurements be relatively fine-grained, while system constraints dictate that the solution be low-cost and efficient. A key question is how instrumentation can be designed to achieve these conflicting goals.

One common type of on-chip digital sensor is based on a ring oscillator circuit which feeds a frequency counter. The ring oscillator acts as a test circuit sensitive to parameters such as temperature. The oscillator is enabled for some reference period and the number of pulses is counted. The measured frequency is then used to estimate the physical quantity of interest. The utility of such sensors for FPGA systems has been limited by their high sensitivity to supply voltage fluctuations, and by the associated hardware overhead.

We will now consider a much more efficient method of implementing the requisite frequency counters, as well as a compact implementation of a ring oscillator. We will then describe the proposed sensor and its methods of operation and access.

### 3.1 Efficient Counting

One approach to frequency counting is to implement a single centralized counter shared by multiple sensors [16]. Aside from routing congestion, this approach only allows one sensor to be enabled at a time, which prevents a snapshot of a spatial profile to be captured (a parameter like dynamic power can change dramatically in the several milliseconds needed for serialized measurements). Furthermore, certain parameters require a system to be paused during measurement, so serialization would cause a linear increase in the performance penalty. For these reasons we need a compact counter that can be instantiated in each sensor and operated in parallel.

The goals for the counter design include not just compactness but also ease of use by the software that reads and decodes the sensor data. Traditionally these two goals have been mutually opposing, forcing a trade off. A standard *binary counter* uses a binary representation for the count as the name implies; this format can be readily processed in software without special decoding. Unfortunately, a binary counter consumes a relatively large amount of hardware resources; a counter with period  $M$  requires  $\lceil \log_2 M \rceil$  LUTs and flip-flops. An alternative design is a linear feedback shift register counter (*LFSR counter*), which can be implemented very compactly using Xilinx's shift register LUTs (SRLs). For instance, a counter with period  $2^{15}-1$  can be built from an SRL, a flip-flop, and a LUT configured as an XOR [30]; this requires just 2 LUTs, compared to 15 for a binary counter. However, the count value is scrambled by the LFSR and must be recovered by solving the discrete logarithm problem [6]. The discrete log is normally difficult to solve, which is why it has uses in the field of cryptography. LFSRs have been used as event counters in cases where slow off-line decoding is acceptable, such as in silicon debug [6]. The required overhead can be reduced somewhat through the choice of the LFSR's characteristic polynomial, but remains non-trivial. A series of exponentiations must be performed (via polynomial multiplication), followed by lookups to residue tables that must be held in memory, followed by an application of the Chinese remainder theorem:

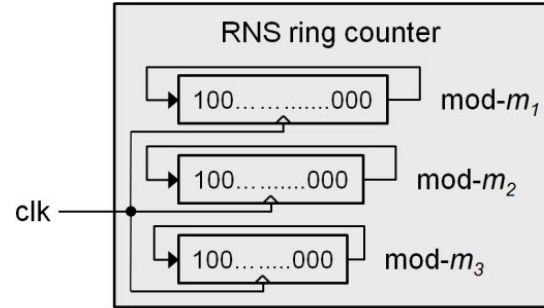
$$count = \sum_{i=1}^k r_i \left( \frac{M}{m_i} \right) v_i \bmod M \quad (1)$$

where  $k$  is the number of moduli,  $r_i$  is a residue,  $M$  is the counting period,  $m_i$  is a modulus, and  $v_i$  is a weight found with the extended Euclidean algorithm.

We propose a counter design that is both compact and relatively easy to decode: a residue number system (RNS) ring counter. An example is shown in Figure 2. This style of counter was proposed long ago for generating timing signals [8], but to our knowledge its potential as an extremely efficient event counter for FPGAs has not been recognized. The design is composed of multiple shift registers of varying lengths, each feeding back to itself. Each shift register acts as a modulo- $m_i$  "ring counter" and can be implemented with a  $m_i$ -bit SRL plus an optional flip-flop. The shift registers are initialized with a pattern such as a one-hot codeword. When the count is to be incremented the patterns shift to the right; the position of each "hot" bit represents the count value modulo  $m_i$ , in other words, a residue  $r_i$ . A modulo-3 ring simply cycles through the states of 100, 010, and 001. By implementing rings whose moduli are all pairwise relatively prime, the counter can reach a period of  $M = \prod m_i$ . Since the residue values are readily available simply by reading out the

individual shift registers, the RNS count value can be easily recovered just by applying the Chinese remainder theorem (1). Most of the processing overhead of an LFSR counter is avoided, and no memory is required for tables. Pseudocode for the two counter styles is shown in Figure 3.

Consider an example in which a counter of period  $2^{13}$  is required. A comparison of the three implementation styles is shown in Table 1. An RNS ring counter can be implemented with just 2 LUTs, using moduli of 33, 17, and 16. With the Virtex-5/6, one LUT acts as a 32-bit SRL and one acts as two 16-bit SRLs. This 2-LUT design is far more compact than a binary counter, and just as small as an LFSR counter. Moreover, the RNS ring counter requires no tables in memory, while the LFSR counter requires multiple tables totaling on the order of hundreds of bytes. Lastly, the RNS ring counter is almost as trivial to decode as a binary counter, comparing favorably to the dozens of polynomial multiplications needed for an LFSR counter.



**Figure 2. Compact counter design using the residue number system (RNS). All moduli are pairwise relatively prime. The period is  $M = m_1 \times m_2 \times m_3$ .**

```

LFSR Counter Decoding Algorithm
y ← polynomial contents of LFSR; // read counter
for i = 1 to k do
  for j = 1 to 2  $\lceil \log_2(M/m_i) \rceil$  do // exponentiations
     $y_i \leftarrow y_i \times y$ ; // polynomial multiplication
  end
   $r_i \leftarrow \text{TABLE}(y_i)$ ; // look up residue from table in memory
end
for i = 1 to k do // Chinese remainder theorem
   $count = count + r_i (M/m_i) v_i \bmod M$ ;
end
return count;

```

(a)

```

RNS Ring Counter Decoding Algorithm
r ← counter; // read k residues from counter
for i = 1 to k do // Chinese remainder theorem
   $count = count + r_i (M/m_i) v_i \bmod M$ ;
end
return count;

```

(b)

**Figure 3. Sketch of algorithms to recover the count for (a) an LFSR counter and (b) the proposed RNS ring counter.**

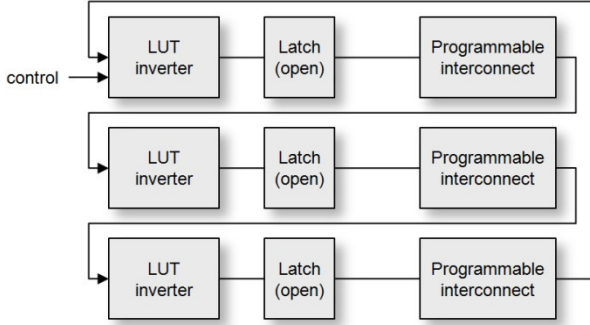


Figure 4. Ring oscillator design

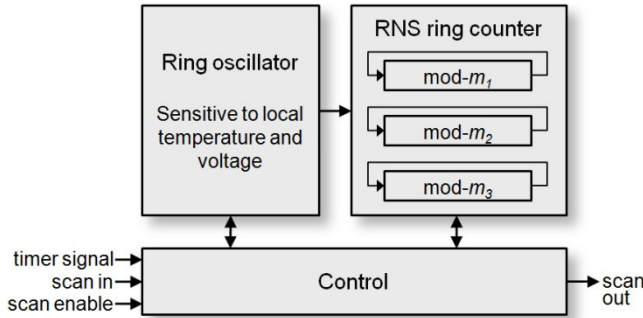


Figure 5. High-level diagram of proposed sensor

### 3.2 Ring Oscillator Design

The main goals for our ring oscillator are compactness, and sensitivity to relevant physical parameters. The basic design is shown in Figure 4. As with standard ring oscillators, it includes an odd number of inversions, and an on/off control switch. Two aspects of the design are non-standard. First, a latch is instantiated along with each LUT. The latch is held in the open state and acts as additional delay, increasing the fraction of ring oscillator delay that is due to transistors. Since the effect of temperature is stronger on transistors than wires, this increases the overall sensitivity to temperature. This improved sensitivity was validated by a hardware experiment comparing ring oscillators both with and without the latch (see Section 5.1). Other researchers have used latches in a ring oscillator [13], but the impact of latches on temperature sensitivity appears not to have been recognized.

The second benefit of the proposed ring oscillator is its compactness, using only 3 stages. Ring oscillators for thermal sensing have used various numbers of stages, including 4 [2], 48 [11], and 75 [4]. One rationale given for a large number of stages is to wash out stochastic variations, in the hopes that all ring oscillators will behave similarly. However, calibration would still be required due to correlated variations. A second reason is to minimize “self-heating,” since a slow and sparse implementation has lower heat density than a fast and dense one. We conducted a hardware experiment comparing a compact 3-stage ring oscillator to a 21-stage, sparsely-placed design and found no evidence of self-heating problems. We also tested the ring oscillator using enable periods ranging from nanoseconds to milliseconds and found highly consistent frequencies.

Table 1. Comparison of counter implementations assuming a maximum period of  $2^{13}$ .

Counter type	Hardware overhead		Ease of decoding
	Logic (LUTs)	Memory (Bytes)	
Binary	13	0	Trivial
LFSR using SRLs	2	Hundreds [6]	Dozens of polynomial multiplications; $\geq 2$ table lookups; $\geq 4$ integer ops
RNS ring using SRLs (proposed)	2	0	6 integer operations

Table 2. Comparison of reconfigurable-logic-based sensors

Sensor design	LUT count (target platform)	LUT count normalized to Virtex-5
Chen <i>et al.</i> 2007 [4]	140 (Altera ACEX)	$\geq 70$
Jones <i>et al.</i> 2007 [11]	100 (Virtex-4)	$\geq 50$
López-Buedo <i>et al.</i> 2004 [17]	34 (Virtex-1)	$\geq 17$
Design proposed here	8 (Virtex-5)	8

The main challenge in using reconfigurable logic-based ring oscillators to measure delay or temperature is their high sensitivity to the supply voltage. We propose leveraging this sensitivity to measure various types of voltage drop, and to subsequently infer useful physical parameters. The issue of voltage sensitivity will be discussed in Section 4.

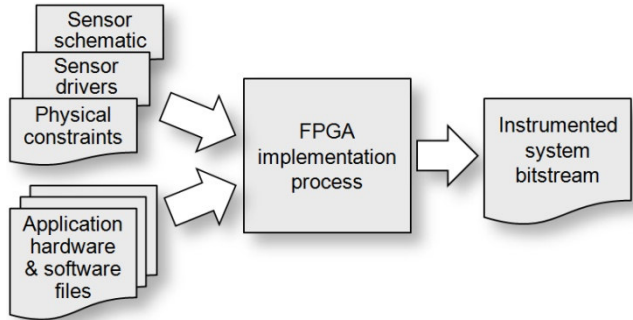
### 3.3 Sensor Design and Deployment

The proposed sensor design includes a ring oscillator, a frequency counter, and logic for control and access. A high-level diagram is shown in Figure 5. The sensor is enabled by a reference timer which is either on-chip or off-chip. A single timer can be used to enable multiple sensors simultaneously. The sensor resolution is dictated by the ring oscillator frequency and the length of the timer pulse. For instance, a frequency of 250MHz and a timer pulse of 40 $\mu$ s allow a resolution of 1 part in 10,000. After a measurement, the sensor data is read out via a scan output. An array of sensors can be connected via one or more scan chains. Embedded software controls the scan sequence and reads the sensor data. Note that there is no need to involve FPGA readback or reconfiguration.

A distinguishing feature of the proposed sensor is its compactness. The entire design fits into a single Virtex-5 CLB. This includes logic for scan and for synchronizing the timer signal to the ring oscillator clock. As can be seen in Table 2, this is much smaller than previous designs that have been proposed for thermal sensing, even after normalizing to a single architecture.

The sensor design can be defined with a schematic or an HDL netlist. Sensor locations can be easily specified by using physical constraint statements such as Xilinx’s RLOC\_ORIGIN. Directed

routing statements are used to ensure consistent ring oscillator instances. Software drivers contain the code for operating the sensors and for accessing, decoding, and processing the data. By integrating the sensing infrastructure into a traditional system design flow, as shown in Figure 6, an instrumented application can be generated.



**Figure 6. Overview of the proposed design flow for FPGA system instrumentation**

An interesting question is: what is an appropriate spatial arrangement of sensors? Related works involving FPGA-based thermal sensors have either suggested a regular rectangular grid [17], or an irregular arrangement focused on expected application-specific hotspots [19]. Here we seek to perform systematic measurements of a variety of physical parameters, and thus employ a regular grid; specifically, we recommend a hexagonal tessellation due to its efficient area coverage [29]. An example of a hexagonal sensor array can be seen in Figure 1, where each sensor (shown as a light rectangle) has six nearest neighbors that are nearly equidistant.

A second question is: what is an appropriate density of sensors? In the past, this has been largely dictated by hardware overhead. Most work on FPGAs involves between 1 and 32 sensors per chip. One work for ICs suggests sensing at 96 locations [14]. With our proposed compact design there is the possibility of deploying over 100 sensors while incurring only 1% overhead. In principle, over 12,000 sensors could be instantiated in a high capacity Virtex-5 while using only half of the available slices. The actual density should match the granularity of the physical phenomena being sensed. One relevant measure is the distance at which areas of the chip are correlated (“correlation distance”). Sylvester *et al.* demonstrate that higher sensor densities enable increased accuracy, for instance, in predicting chip lifetimes [25].

#### 4. ON-LINE MEASUREMENT

In this section we address three general questions: which physical parameters can be measured in a reconfigurable fabric, how is such data useful, and what are the procedures for measurement? After defining some necessary concepts, we introduce enhanced procedures for measuring variations in four key parameters: delay, leakage, dynamic power, and temperature.

Ring oscillators have been widely used for sensing variations in delay. They have also been proposed for sensing temperature. The parameters of delay, temperature, and voltage are all closely related. For instance, a ring oscillator will generally slow down with higher temperature, but speed up with higher voltage. If two

of the three related quantities can be measured accurately, then the third can be inferred. This is typically how researchers have proposed using ring oscillators in reconfigurable fabrics. For instance, the ring oscillator delay is measured at a known voltage, and then a model is employed to estimate the temperature.

The traditional approach to ring oscillator-based sensing, for thermal sensing in particular, suffers from several limitations. One is that local supply voltages are often lower than intended, a phenomenon called *voltage drop* (also known as static IR drop [1]). Supply voltages can also undergo rapid, high-frequency swings after dramatic changes in switching activity. We will refer to these as *voltage transients* (also known as  $L^{di}/di$  events or voltage droops [14]). For many FPGA platforms the supply voltage can only be measured at a supply pin, using a single analog sensor; we will refer to this measured voltage as  $VCC_p$ .

#### 4.1 Delay

Every chip has fast areas and slow areas. Spatially-correlated variation in delay occurs due to factors such as imperfect lithography and transistor critical dimensions [22]. By mapping and adapting to delay variations, the system frequency can potentially be improved. Currently, the expected benefit is on the order of 10% [5][23], and rising. Note that a frequency gain of  $g\%$  can increase the price or utility of a system by much more than  $g\%$ , for applications that are performance-constrained.

Correlated delay variations can be mapped via controlled measurements of ring oscillator frequencies across a chip. A traditional procedure is to configure an FPGA fabric with a special test circuit containing ring oscillators. The chip is left in an idle state for an extended period (seconds to minutes) to eliminate voltage or temperature gradients caused by switching activity. Then the ring oscillators are sampled, either one at a time or (when feasible) simultaneously. The procedure can be repeated for a range of temperatures and supply voltages if necessary. We propose using a traditional procedure but with a key difference: the sensors are embedded in the application logic and the sensing is performed in the field. This reduces the testing bottleneck. Furthermore with an instrumented application there is no need for storing a separate test bitstream and for performing reconfigurations. The delay characterization can be performed occasionally during a system’s lifetime to track gradual shifts caused by wearout (e.g. due to negative-bias temperature instability).

#### 4.2 Leakage

Leakage is a phenomenon in which current flows through a transistor when it is not supposed to, like a leaky faucet. The amount of variation is much larger than it is for delay; the ratio of leakage from worst-case to best-case transistors on a single FPGA die has already passed 2X [28]. The main type of leakage, called sub-threshold leakage, is also very sensitive to temperature. Hot transistors leak more current and generate more heat, driving a positive feedback loop. It would be helpful to know a chip’s leakage profile so the problems of static power, thermal hotspots, and temperature-dependent reliability could be more effectively mitigated. One recent work finds that variation-aware adaptation could reduce leakage by 14%, even while simultaneously improving frequency [23].

To our knowledge there has been no low-cost method of sensing a leakage profile in FPGAs. The authors of [23] lament,

“Programming leakage sensors on the FPGA may not be feasible to employ due to their analog nature.” We propose a relatively straightforward method of characterization, using our digital sensors. First we will explain the physical mechanisms at work, and then describe the measurement procedure. Historically, ring oscillator frequencies have been modeled as linear with temperature, assuming a constant supply voltage. However, leakage current increases exponentially with temperature, and on modern chips such as the Virtex-5 this current is high enough to cause noticeable drops in the supply voltage and thus ring oscillator frequencies. Temperature-dependent voltage drop can be observed at the supply pins simply by reading the built-in analog voltage sensor at different ambient temperatures; we have found that this global drop increases by roughly 1mV from 75°C to 85°C on the Virtex-5 we tested. We also separately found evidence of *localized* on-chip voltage drops caused by local currents interacting with the on-chip power grid. At higher temperatures all ring oscillators will exhibit some slowdown due to the global effects (namely a decrease in carrier mobility, and global voltage drop due to both leakage and increased resistance in the power grid), but in addition there will be a variable slowdown primarily caused by leakage-induced local voltage drop. This slowdown provides evidence for the amount of leakage in the vicinity of the oscillator. While the absolute amounts of leakage currents across the chip are unknown, this method provides an estimate of the relative leakage profile.

The measurement procedure involves performing the ring oscillator delay characterization from Section 4.1 at two different die temperatures. [Systems must either leverage natural swings in ambient temperature, control the die temperature via a fan, or if necessary use a ‘heater’ configuration.] With these two measurements the effects of increased temperature on each region of the chip can be compared, and the leakage profile inferred.

### 4.3 Dynamic Power

Dynamic power can have an uneven spatial distribution due to uneven application activity. This leads to thermal hotspots, reliability issues, and extra power consumption. For example, the power supply may have to be set to a high voltage to offset the worst-case local voltage drop. Dynamic power profiles can usually be estimated at design time, or in some cases with performance counters. Nevertheless certain systems may have an unknown profile, such as those with autonomously-generated configurations [27]. Sensing the profile in these cases can allow improved configurations to be found. We now describe a simple method of inferring a dynamic power profile. The basic approach is to sample ring oscillator frequencies as the system is running, briefly pause the system until no switching occurs, and then immediately sample the frequencies again. A change in frequency indicates the size of local voltage drop due to switching current. Temperature and leakage effects remain nearly constant since the time between readings is much less than the thermal time constant. Very similar approaches have been used for sensing voltage variations [7][14], but here we exploit the fact that voltage drops can be used to infer dynamic power. Furthermore, we propose the following enhancement to previous methods. Voltage transients can introduce large errors in ring oscillator data. We add a simple check for such events; instead of taking a single reading while the application is running we propose taking multiple consecutive readings. Each reading is spaced out by the length of the problematic transients (on the order of 1ms). A quick check of data consistency can determine whether a voltage

transient event occurred. If not, the procedure can move forward and the system can be paused; otherwise new readings are required. This method is most practical for applications with a relatively steady-state power profile or activity phases much longer than milliseconds. The proposed procedure is as follows:

1. Sample ring oscillator frequencies twice while application is running
2. Check consistency; if a voltage transient is detected repeat 1
3. Pause the application
4. Wait for voltage transients (caused by pause) to dissipate
5. Sample ring oscillator frequencies again
6. Resume the application
7. Compare the frequency shifts of ring oscillators

### 4.4 Temperature

Thermal hotspots lead to early wearout, lower operating frequencies, higher static power, and extra cooling costs. On-line sensing of a thermal profile can account for physical realities (e.g. variations in leakage or packaging) and allow for enhanced mitigation schemes. Unfortunately, thermal profiles are very difficult to measure with standard reconfigurable logic. The main problem is that the logic on some modern FPGAs is not very sensitive to temperature. With older technologies ring oscillator frequencies would shift by 20% over the full temperature range [2]; with the Virtex-5 we find the shift to be only 2.5%.

The dependence on voltage, however, is quite strong. Local voltages cannot be easily controlled or measured, adding noise to temperature measurements. Previous works tend to assume that by pausing system activity, the supply voltage very quickly becomes spatially uniform and thus temperature can be cleanly estimated. This is no longer valid in general, since leakage variations (greatly amplified by temperature variations) cause non-uniform drops in the supply voltage. A second problem is that the supply voltage at the pins ( $VCC_p$ ) may not be measurable with fine resolution. For instance, the Xilinx System Monitor [31] reports  $VCC_p$  with only 3mV resolution, which could introduce errors of 10-20°C into temperature estimates. A third problem is that voltage drop causes non-linearity in the delay, temperature, and  $VCC_p$  relationships, so the conventional linear models are becoming less accurate. Temperature sensing could be improved if additional built-in analog sensors were available on FPGAs, or if leakage current could be selectively disabled via power gating. Absent those, there is a need for new models and methods.

We propose the following procedure for estimating a thermal profile. First, ring oscillator measurements are made across the range of voltages and temperatures. A model of temperature as a function of delay and voltage is built, for instance via surface fitting in MATLAB. This characterization and modeling can be performed off-line and need only be done once for a given ring oscillator design and target platform type. An example will be detailed in the next section. When a system is deployed into the field, variation-aware calibration is performed to find the coefficients of the temperature model for each sensor. The data from delay and leakage characterization can be used for this calibration. At run-time, the chip region being sensed must be paused and left idle for long enough to allow any voltage transients to dissipate. The ring oscillator frequencies are then measured, and the supply voltage at the pins is sampled (an average of multiple readings can improve the limited resolution). This data is then plugged into the empirical model and the local temperature is estimated.

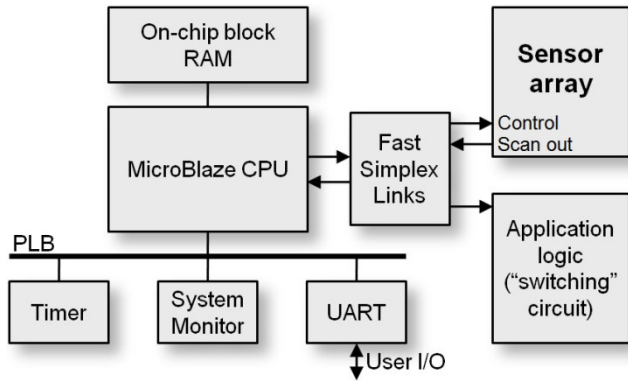


Figure 7. Block diagram of the experimental system implemented on a Virtex-5 FPGA

Table 3. Sensor specifications

Item	Specification
Hardware resources	8 LUTs per sensor (896 total)
Data size	82 bits (= 49 + 17 + 16)
Measurement period	40 $\mu$ s
Resolution	1 part in 10,000 @ 40 $\mu$ s
Sensor array readout time	$\sim$ 100 $\mu$ s

## 5. EXPERIMENTAL RESULTS

We describe our experimental setup and then present the results of sensing experiments involving delay, leakage, dynamic power, and temperature.

### 5.1 Setup

We designed an experimental FPGA-based system and instrumented it with 112 of the proposed sensors. A block diagram of the design is shown in Figure 7. The design was instantiated on each of two Xilinx Virtex-5 FPGAs residing on XUPV5-LX110T boards. The reconfigurable fabric is composed of 160x54 CLBs, corresponding to 160x108 Virtex-5 slices. The sensors are arranged on a hexagonal grid of size 16x7 to fit the dimensions of the die. Sensors are nearly equidistant to each of their six neighbors. The layout can be seen in Figure 1, rotated right by 90 degrees. The sensor frequency counter was designed with moduli of 49, 17 and 16 for a counting period of 13,328. Key sensor specifications are shown in Table 3. Note that the readout time for the entire sensor array is about 100 $\mu$ s, which is much faster than thermal time constants.

The system contains a MicroBlaze 7.10.a CPU and three peripherals connected to the processor local bus (PLB): an XPS Timer for enabling the sensors, a core for interfacing to the System Monitor, and a UART16550 for user input and output. All of these operate at 100MHz. In addition, application logic was included for experimental purposes. The application logic acts as a switching/heater circuit with regions that can be independently enabled. It consists of 28,200 LUTs and flip-flops which can toggle at high speed. Both the array of sensors and the

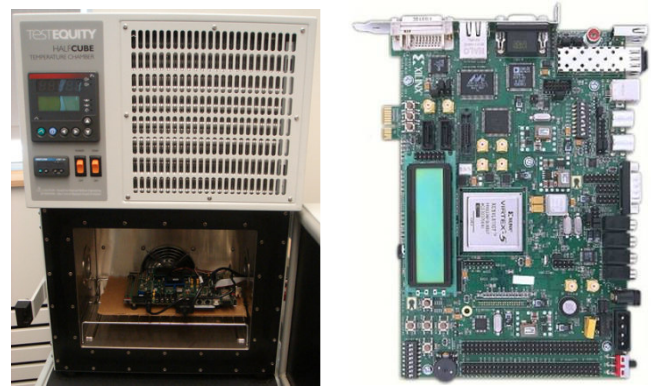


Figure 8. Picture of the experimental setup in the thermal chamber (left). Picture of an XUPV5 circuit board (right).

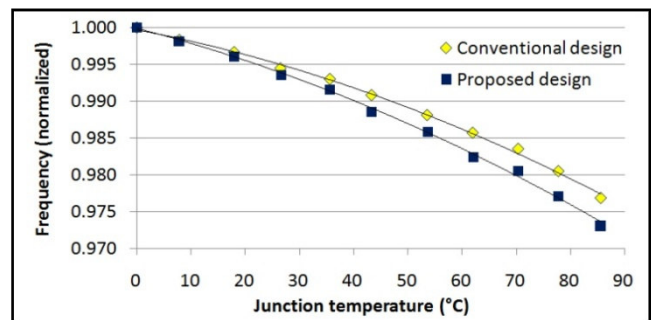


Figure 9. Frequency dependence on temperature for proposed ring oscillator design and conventional design.

application logic are connected to the MicroBlaze via a shared set of fast simplex links of minimum size. The design was implemented with the Xilinx ISE and EDK 10.1 toolset. The experimental control software was written as a standalone application in C for the MicroBlaze, and fits into 32KB of block RAM. Thermally-controlled experiments were performed using a TestEquity 105A thermal chamber, as pictured in Figure 8.

We first tested the proposed ring oscillator design, and compared its temperature sensitivity to a conventional design. Oscillator frequencies were measured at a range of temperatures from 0 to 85°C, while the supply voltage at the pins was maintained at a constant 1.009V. Results validate that the proposed design has a stronger temperature dependence than the conventional design, as seen in Figure 9. The relative improvement in the slope is 17%.

### 5.2 Delay

The first experiment using the proposed on-line measurement method is a characterization of delay variations. This type of measurement is required for calibrating the ring oscillators. The ambient temperature was set to 25°C and the system was left in the idle state until it reached thermal equilibrium. The junction temperature was 35°C as reported by the System Monitor. Ten consecutive readings of the oscillators were taken. We found that the readings at individual sensors are highly consistent; the standard deviation for each set of 10 readings is approximately 0.02% of the mean. In other words, the random measurement error for an oscillator with mean frequency 250MHz is just .05MHz.

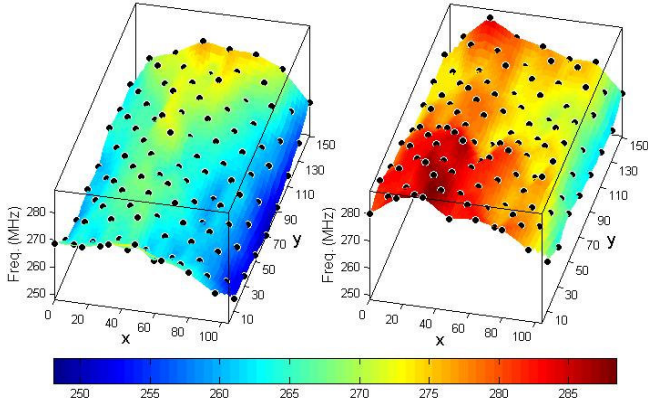


Figure 10. Frequency profile for chip 1 (left) and chip 2 (right) in the idle state at  $T_j = 35^\circ\text{C}$ .

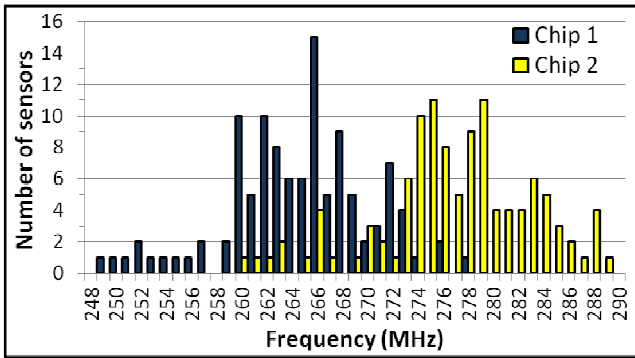


Figure 11. Histogram of ring oscillator frequencies for chip 1 (dark) and chip 2 (light) in the idle state at  $T_j = 35^\circ\text{C}$ .

The frequency profiles of the two FPGA chips are shown in Figure 10. The  $x$  and  $y$  coordinates represent “slice” locations. Dots indicate the average frequencies measured at the locations of the sensors. The frequencies at locations between sensors are estimated using linear interpolation. Several observations can be made. Chip 2 is noticeably faster than chip 1, due to inter-die variation and board-level differences (slightly different supply voltages and heat sinks). Here we are more concerned with *intra-die* variation. In both cases, the right side of the chip is the slowest region. The fastest region is near the top of the grid for chip 1 but near the bottom for chip 2. The distribution of measured frequencies is shown in Figure 11. The amount of spatial variation can be expressed via a coefficient of variation, which is the standard deviation across all sensor frequencies divided by the mean. Both chips exhibit a coefficient of variation of approximately 2.3%, which corresponds to a  $3\sigma$  variation of approximately 7%.

### 5.3 Leakage

The second experiment is a characterization of a leakage profile. We measured all ring oscillator frequencies while the system was idle, but this time at two different temperatures. The ambient temperature was controlled by the thermal chamber. The measured leakage variations are visualized in Figure 12, which shows the relative frequency shifts that occur when the steady-state junction temperature is changed from  $35^\circ\text{C}$  to  $85^\circ\text{C}$ . Larger

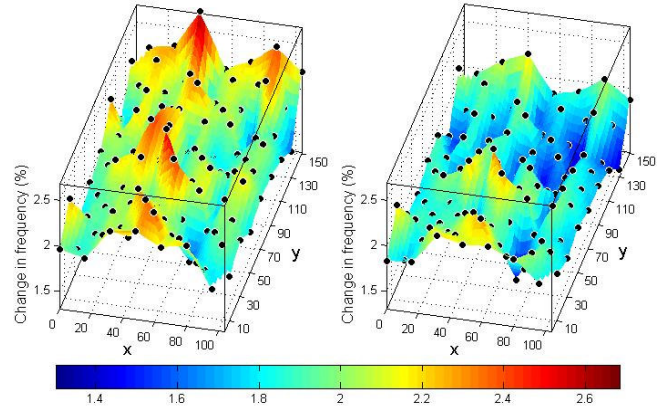


Figure 12. Map of leakage current variations for chip 1 (left) and chip 2 (right).

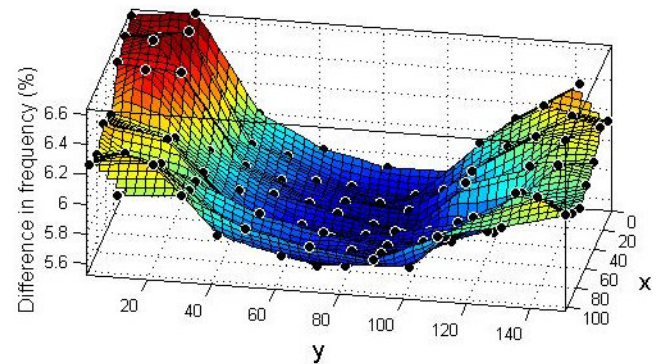
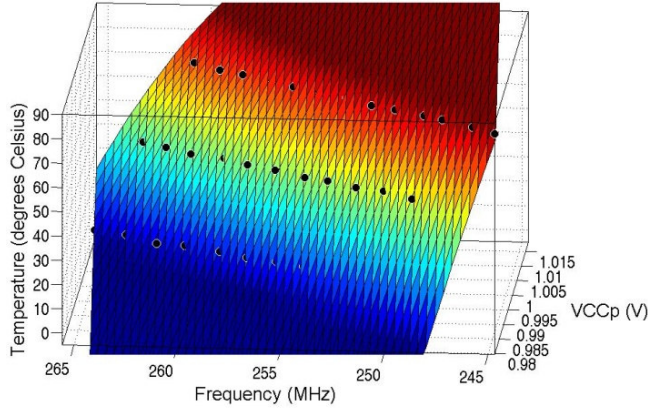


Figure 13. Map of frequency changes due to switching activity.

shifts are evidence of higher leakage. Certain ring oscillators on chip 1 slow down by 1.6%, while others slow down by 2.7%. For chip 2 the range is 1.3% to 2.4%. Ring oscillator frequency and leakage both depend on threshold voltage, so one might expect the two parameters to be highly correlated. In other words, fast regions might tend to be leaky regions. Analysis of the intra-die experimental data shows that there is only a small correlation; the correlation coefficient of these two parameters is 0.19 for both chip 1 and chip 2.

### 5.4 Dynamic Power

We next test the ability to sense a dynamic power profile. A demonstration application was set up to generate high switching activity near the top and bottom of the reconfigurable fabric. The middle portion was left idle so that clean measurements could be taken and validated against the System Monitor, which sits near the center of the die. Specifically, 4,500 slices toggled in the upper third of the die with  $y$  coordinates  $\geq 110$ , and 2,550 slices toggled in the bottom portion of the die. The proposed procedure was applied, which isolates the effect of switching activity. Measurements indicate that ring oscillator shifts are indeed largest in the two regions with switching activity, as shown in Figure 13. The image is rotated to illustrate the changes along the  $y$  axis. This type of measurement provides important information about the spatial extent of switching effects. For instance, while the application was set up with a dramatic discontinuity in activity at



**Figure 14. Measured relationship between temperature, sensor frequency, and supply voltage for the sensor at (48, 81).**

$y = 110$ , the measured frequency shifts are surprisingly continuous, showing a relatively even slope that begins around  $y = 100$  and extends all the way to the top of the die.

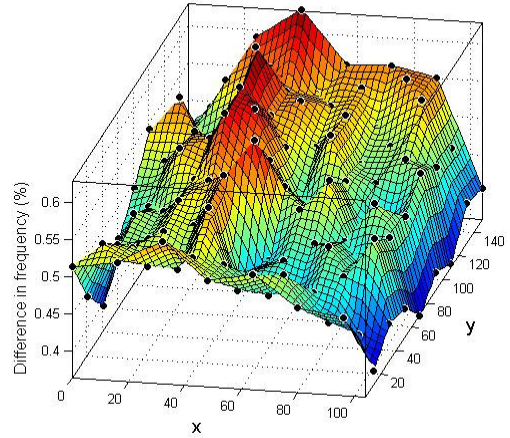
## 5.5 Temperature

In this final experiment we consider temperature variations. We first build an empirical model of frequency, temperature and voltage. The sensor array is placed such that one sensor is immediately adjacent to the System Monitor at  $(x,y) = (48,81)$ , allowing temperature estimates for that location to be validated against the built-in analog sensor. We want to know what type of model fits the data and with what accuracy. We measured frequencies over a range of temperatures from 0 to 85°C, and over a 40mV range of voltages. The supply voltage  $VCC_p$  was modified by enabling various amount of application activity, from zero activity all the way to 100% application activity in 10% steps (equivalent to about 600 slices of application logic). In total 34 data points were collected, with 10 readings at each point. Various models were fitted to the data using MATLAB, in order to find an effective temperature estimation function  $T(x,y) = f(Freq(x,y), VCC_p)$ , where  $T(x,y)$  is the temperature at die location  $(x,y)$ , and  $f$  is a function of frequency  $Freq(x,y)$  and voltage  $VCC_p$ . We found that a traditional model assuming linearity with both frequency and voltage provides a root mean square error of 5.7°C. In contrast we found that a 2<sup>nd</sup> order polynomial model provides a better fit and lower error. The form of the model is as follows, with coefficients  $c_i$ :

$$T(x,y) = c_1 Freq(x,y)^2 + c_2 VCC_p^2 + c_3 Freq(x,y)VCC_p + c_4 Freq(x,y) + c_5 VCC_p + c_6 \quad (2)$$

This model reduces the error to 3.5°C. Plots of the surface fit can be seen in Figure 14; the surface is curved and twisted rather than planar. The error with this model is comparable to the accuracy of analog sensors such as the one built in to the System Monitor, which is rated to  $\pm 4^\circ\text{C}$ .

In addition to the validated measurements at location (48,81), we sensed the thermal effects across the entire fabric for the application mentioned above. Ring oscillator frequencies were measured immediately after the application was paused, and compared to the steady-state idle case. The resulting shifts in



**Figure 15. Map of frequency changes due to temperature effects**

frequencies are illustrated in Figure 15. This profile includes not only the direct effect of temperature, but also the indirect effect caused by temperature-dependent leakage.

## 6. CONCLUSIONS

We have introduced an approach to on-line sensing that includes a compact multi-use sensor, methods of instrumenting an application, and enhanced procedures for measuring physical parameters. Novel use of a residue number system counter yields sensors that are surprisingly compact, while an enhanced ring oscillator enables a 17% improvement in temperature sensitivity. The proposed sensor fits into just 8 LUTs on a Virtex-5. We have described procedures for measuring the profiles of four parameters of key interest to system designers: delay, dynamic power, leakage power, and temperature. The work suggests possibilities for future research involving additional physical parameters, measurement procedures, sensor array arrangements, and adaptive algorithms. As demonstrated with our experimental system, the proposed approach is suitable for low-cost sensing of a variety of parameters, and is accurate enough to estimate spatial profiles that can be used for a wide range of on-line adaptations. We see a growing need for this type of on-line sensing in support of physically-adaptive computing, with the goal of more efficient, more reliable, and healthier reconfigurable systems.

## 7. ACKNOWLEDGMENTS

We would like to thank: Yiqian Min and David Blaauw for assistance; Andrea Pellegrini, Joseph Greathouse, Prabal Dutta and Robert Dick for helpful comments; Xilinx and Sun Microsystems for equipment donations. This work was supported in part by a NASA Langley Research Center GSRP Fellowship and by the National Science Foundation under grant CCF-0702276.

## 8. REFERENCES

- [1] A.H. Ajami, K. Banerjee, A. Mehrotra and M. Pedram, "Analysis of IR-drop scaling with implications for deep submicron P/G network designs," *Int'l Symp. Quality Electronic Design*, pp. 35-40, Mar. 2003.
- [2] E. Boemo and S. López-Buedo, "Thermal monitoring on FPGAs using ring-oscillators," *Int'l Workshop on Field-Programmable Logic and Applications*, pp. 69-78, 1997.

- [3] J. Bongard, V. Zykov and H. Lipson, "Resilient machines through continuous self-modeling", *Science*, 314(5802): 1118-1121, 2006.
- [4] P. Chen, M. Shie, Z.-Y. Zheng, Z.-F. Zheng and C. Chu, "A fully digital time-domain smart temperature sensor realized with 140 FPGA logic elements," *IEEE Trans. Circuits and Systems I*, 54(12): 2661-2668, Dec. 2007.
- [5] L. Cheng, J. Xiong, L. He and M. Hutton, "FPGA performance optimization via chipwise placement considering process variations," *Int'l Conf. Field Programmable Logic and Applications*, pp. 1-6, Aug. 2006.
- [6] D. Clark and L. Weng, "Maximal and near-maximal shift register sequences: efficient event counters and easy discrete logarithms," *IEEE Trans. Computers*, 43(5): 560-568, May 1994.
- [7] R. Conn Jr., "Method and apparatus for measuring localized temperatures on integrated circuits," U.S. Patent 6067508, May 23, 2000.
- [8] J. Fletcher, M. Perlman, W. Rousey and A. Messner, "System for generating timing and control signals," U.S. Patent 3866022, Feb. 11, 1975.
- [9] B. Gojman and A. DeHon, "VMATCH: using logical variation to counteract physical variation in bottom-up, nanoscale systems," *Int'l Conf. Field-Programmable Technology*, Dec. 2009.
- [10] N. Jiang and M. Parashar, "Enabling autonomic power-aware management of instrumented data centers," *Int'l Symp. Parallel & Distributed Processing*, May 2009.
- [11] P.H. Jones *et al.*, "Adaptive thermoregulation for applications on reconfigurable devices," *Int'l Conf. on Field-Programmable Logic and Applications*, pp. 246-253, Aug. 2007.
- [12] International Technology Roadmap for Semiconductors, <<http://www.itrs.net>>, 2009.
- [13] P. Kohlbrenner and K. Gaj, "An embedded true random number generator for FPGAs," *Int'l Symp. Field Programmable Gate Arrays (FPGA'04)*, pp. 71-78, 2004.
- [14] A. Krishnamoorthy and A. Detofsky, "Mapping variations in local temperature and local power supply voltage that are present during operation of an integrated circuit," U.S. Patent 7233163, Jun. 19, 2007.
- [15] E. Kursun and C. Cher, "Variation-aware thermal characterization and management of multi-core architectures," *Int'l Conf. Computer Design*, pp. 280-285, Oct. 2008.
- [16] S. López-Buedo and E. Boemo, "A method for temperature measurement on reconfigurable systems," *Design of Circuit and Integrated Systems Conf.*, pp. 727-730, Nov. 1997.
- [17] S. López-Buedo and E. Boemo, "Making visible the thermal behaviour of embedded microprocessors on FPGAs: a progress report," *Int'l Symp. Field Programmable Gate Arrays (FPGA'04)*, pp. 79-86, 2004.
- [18] P. Mangalagiri, S. Bae, R. Krishnan, Y. Xie and V. Narayanan, "Thermal-aware reliability analysis for platform FPGAs," *Int'l Conf. Computer Aided Design*, pp. 722-727, 2008.
- [19] S.O. Memik, R. Mukherjee, M. Ni and J. Long, "Optimizing thermal sensor allocation for microprocessors," *IEEE Trans. Computer-Aided Design of ICs and Systems*, 27(3): 516-527, Mar. 2008.
- [20] G.M. Quénot, N. Paris and B. Zavidovique, "A temperature and voltage measurement cell for VLSI circuits," *Euro ASIC '91*, pp. 334-338, 1991.
- [21] R. Rubin and A. DeHon, "Choose-your-own-adventure routing: lightweight load-time defect avoidance," *Int'l Symp. Field Programmable Gate Arrays (FPGA'09)*, pp. 23-32, 2009.
- [22] P. Sedcole and P. Cheung, "Within-die delay variability in 90nm FPGAs and beyond," *Int'l Conf. Field Programmable Technology*, pp. 97-104, 2006.
- [23] S. Srinivasan and V. Narayanan, "Variation aware placement for FPGAs," *Emerging VLSI Technology and Architectures*, March 2006.
- [24] P. Sundarajan, A. Gayasen, N. Vijaykrishnan and T. Tuan, "Thermal characterization and optimization in platform FPGAs," *Int'l Conf. Computer Aided Design*, pp. 443-447, 2006.
- [25] D. Sylvester, D. Blaauw and E. Karl, "ElastIC: An adaptive self-healing architecture for unpredictable silicon," *IEEE Design & Test of Computers*, 23(6): 484-490, June 2006.
- [26] I.A. Troxel, M. Fehring and M.T. Chenoweth, "Achieving multipurpose space imaging with the ARTEMIS reconfigurable payload processor," *Aerospace Conf.*, pp. 1-8, Mar. 2008.
- [27] F. Vahid, G. Stitt, and R. Lysecky, "Warp processing: dynamic translation of binaries to FPGA circuits," *IEEE Computer*, 41(7): 40-46, July 2008.
- [28] H.Y. Wong, L. Cheng, Y. Lin and L. He, "FPGA device and architecture evaluation considering process variations," *Int'l Conf. Computer-Aided Design*, pp. 19-24, Nov. 2005.
- [29] C. Wu and D. Verma, "A sensor placement algorithm for redundant covering based on Riesz energy minimization," *Int'l Symp. Circuits and Systems*, pp. 2074-2077, May 2008.
- [30] Xilinx, Inc., "Linear Feedback Shift Registers in Virtex Devices," App. note XAPP210 (v1.3), Apr. 2007.
- [31] Xilinx, Inc., "Xilinx UG192 Virtex-5 FPGA System Monitor User Guide."
- [32] K.M. Zick and J.P. Hayes, "On-line characterization and reconfiguration for single event upset variations," *Int'l On-Line Testing Symposium*, pp. 243-248, June 2009.