Dynamically Inserting, Operating, and Eliminating Thermal Sensors of FPGA-Based Systems

Sergio Lopez-Buedo, *Member, IEEE*, Javier Garrido, *Member, IEEE*, and Eduardo I. Boemo

***Abstract*—In this paper, a new thermal monitoring strategy suitable for field programmable logic array (FPGA)-based systems is developed. The main idea is that a fully digital temperature transducer can be dynamically inserted, operated, and eliminated from the circuit under test using run-time reconfiguration. A ring-oscillator together with its auxiliary blocks (basically counting and control stages) is first placed in the design. After the actual temperature of the die is captured, the value is read back via the FPGA configuration port. Then, the sensor is eliminated from the chip in order to release programmable resources and avoid self-heating. All the hardware of the sensor is written in Java, using the JBits API provided by the chip manufacturer.**

The main advantage of the technique is that the sensor is completely stand-alone, no I/O pads are required, and no permanent use of any FPGA element is done. Additionally, the sensor is small enough to arrange an array of them along the chip. Thus, FPGAs became a new tool for researchers interested in the thermal aspects of integrated circuits.

***Index Terms*—FCCM, FPGA, ring-oscillator, run-time reconfiguration, thermal verification.**

I. Introduction

FIELD-programmable logic arrays (FPGAs) devices are nowadays one of the most important alternative to construct high-speed digital systems. This technology was marketed in the middle of the 1980s with a simple but strong argument: Its capability to be in-house erased and reconfigured in few milliseconds would allow the designers to correct errors or introduce last-minute modifications. This feature clearly distinguished FPGAs from other alternatives like standard cells or gate arrays, and guaranteed the success of the new devices.

However, the designers soon discovered that reprogrammability could also be utilized to create reconfigurable systems. That is, circuits that can modify or adapt its functionality to perform different tasks. This idea lead to new research lines like FPGA-based computer machines (FCCMs) or custom digital signal processors (CDSPs) [1], [2]. The next step was the incorporation of run-time reconfiguration. In this case, some blocks of the chip are modified on the fly to add a new function, meanwhile the other blocks continue working. This capability is present in some of the newest generation of FPGAs [3], [4]. Once again, a new feature puts under way innovative research lines like evolvable systems, and self-repaired or self-healing circuits [5].

Although the designers of electronic systems have proposed hundred of new applications for FPGA reconfigurability, to the best of our knowledge it has never been taken into account to check the thermal status of the chip itself. Temperature fits well with dynamic reconfiguration: unlike other applications, the transient times of thermal processes are several orders of magnitude greater than typical reconfiguration periods.

In a previous work [6], an FPGA was statically configured with thermal sensors to detect design errors that produce excessive dissipation, like bus contentions or short-circuits at the output pads. But dynamic reconfiguration makes possible new alternatives. Using the configuration port of the FPGA, the host can write the main configuration—for example, a custom signal processor core. In a given time, a second configuration with a set of thermal sensors can be inserted in the free space of the FPGA [Fig. 1(a)]. Then, the sensors are utilized to measure and store the temperatures of different parts of the die during the normal operation of the processor mapped on the FPGA. Finally, the results are read back by the host and the sensors are erased by a new run-time configuration that returns the chip resources to its initial state. No extra I/O pad has been used, and sensors neither influence the original circuit nor make use of permanent space.

A second application where run-time reconfiguration can be applied is the construction of a thermal map of the complete die. In this case [Fig. 1(b)], the circuit under test must be stopped and replaced by other configuration with an array of sensors. After that, the temperature in discrete points of the die can be measured and read back. For the chips utilized in our experiment, this operation takes near few milliseconds, so that the temperature variation respect to normal operation can be neglected. In this mode, reconfigurable FPGAs can be useful not only to construct digital system, but also to study the thermal details of them, like temperature gradients, hot spots, etc.

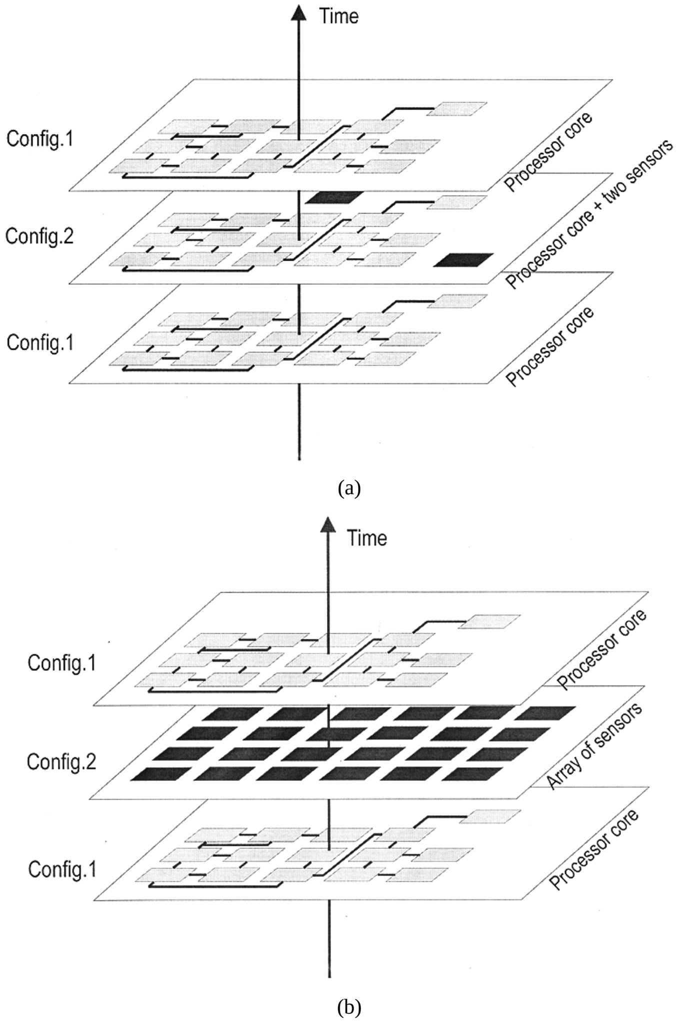
II. Description of the Sensor

Ring-oscillators can be utilized to measure die temperature [7]. They can be implemented on all FPGA architectures, and its fully digital output makes them a good alternative to the dedicated diode included in the newest FPGA families. But if the ring-oscillator is permanently situated in the circuit at design time, two I/O pads are required: one to measure the output frequency, and another one to disable the oscillator. In addition, the output signal could be measured internally using a counter,

Manuscript received November 1, 2001; revised April 4, 2002. This work was recommended for publication by Guest Editors C. J. M. Lasance and M. Rencz upon evaluation of the reviewers' comments. This work was supported by the Spanish Ministry of Science and Technology, under Contract TIC2001-2688-CO3-03, and by the FGUAM under Project 658001. This work was presented in part at the Seventh THERMINIC Workshop, Paris, France, 2001.

The authors are with the School of Computer Engineering, Universidad Autónoma de Madrid, Madrid 28049, Spain (e-mail: sergio.lopez-buedo@uam.es; javier.garrido@uam.es; eduardo.boemo@uam.es).

Digital Object Identifier 10.1109/TCAPT.2002.808011



(a)

(b)

Fig. 1. Two dynamic reconfiguration schemes on FPGA. (a) On-line thermal monitoring (left) and (b) temporary array of sensors to measure the die temperature distribution (right).

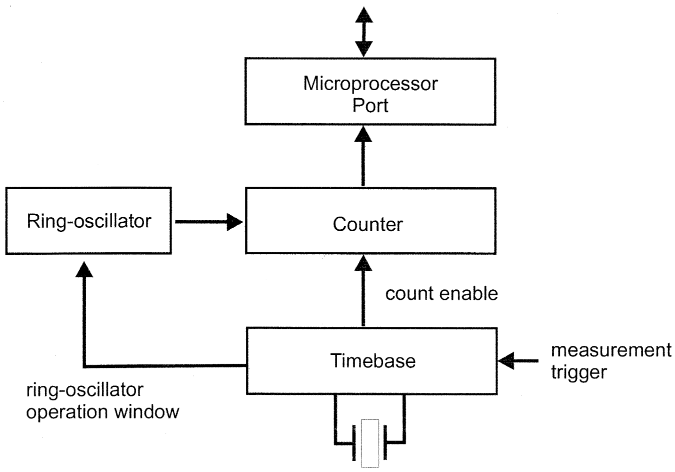


Fig. 2. Minimum circuitry to sense die temperature in FPGAs.

but this option would requires more FPGA resources, not only for counting and controlling, but also for implementing a port compatible with an external microprocessor (Fig. 2).

In the approach proposed in this paper, no hardware overhead exists. Run-time reconfiguration is employed to insert a small sensor in the FPGA every time it is needed. Once the measurement has been made, the same configuration port is used to read back the temperature. Then, the circuit is eliminated from the FPGA configuration. The use of the standard configuration port avoids the construction of a dedicated I/O circuitry. In previous works [8], [9], this goal was achieved using the JTAG port. In FPGAs it is also possible to perform run-time reconfigurations using JTAG, but the main problem is that the speed that can be achieved employing this port is near one order of magnitude below those obtained using the dedicated configuration port.

III. Design Methodology

The schematic for the sensor utilized in this paper is shown in Fig. 3. A ring-oscillator was constructed using seven inverters, each one implemented in a four-input LUT (*look-up table*). The LUTs were placed separated, leaving a minimum space of a 1 CLB (*configurable logic block*) between them in order to increase the routing delay. Note that in this case, a high frequency output is not desirable: It implies both a higher counter size and extra self-heating. In the experimental measurements, a frequency output of 45.5 MHz at 25 °C was obtained using an XCV50PQ240-4 FPGA.

All the hardware of the sensor was written in Java, using the JBits Application Program Interface [10]. In short, JBits is a set of Java classes and methods that allow a low-level manipulation of the chip resources. This software can be used to create a circuit using structural descriptions. Only the Virtex FPGAs are supported in the version of the tool (2.7) utilized in this work.

The first step in JBits is to read an existing configuration bitstream, which will be used as a base for the design. To create the logic elements for the new circuit, the most common alternative is employing the library of parameterizable cores, but it is also possible to directly access to the low-level resources (look-up tables, multiplexers, etc.). Similarly, it is possible to implement the connections of the circuit using the router tool provided with JBits. Another alternative is to explicitly declare the routing resources to be used (local or global connection lines, PIPs, etc.). Once the bitstream for the new design is prepared, it can be downloaded into the FPGA.

The main advantage of JBits is its support for partial run-time reconfiguration [11]. Once a bitstream is written in the FPGA, in the following downloads only the parts that have changed are updated. This allows the user to perform very fast run-time reconfigurations, without having to interrupt the normal operation of the system. Finally, the contents of the FPGA can be read back from JBits, allowing the host processor to know the present state of each circuit.

In order to simplify the code, the two counters included in the sensor are identical, featuring a length of 14 b. The only external port is the clock input of the timebase counter, which must be connected to one of the main FPGA clock lines. Three enable signals, generated by four-input LUTs, are derived from the four most significant bits of this counter. The first, `RingEnable`, allows the ring-oscillator to start running. The second, `CaptEnable`, establishes the interval during which the ring-oscillator frequency output will be measured by the capture counter. Finally, `TimeEnable` is used to disable the

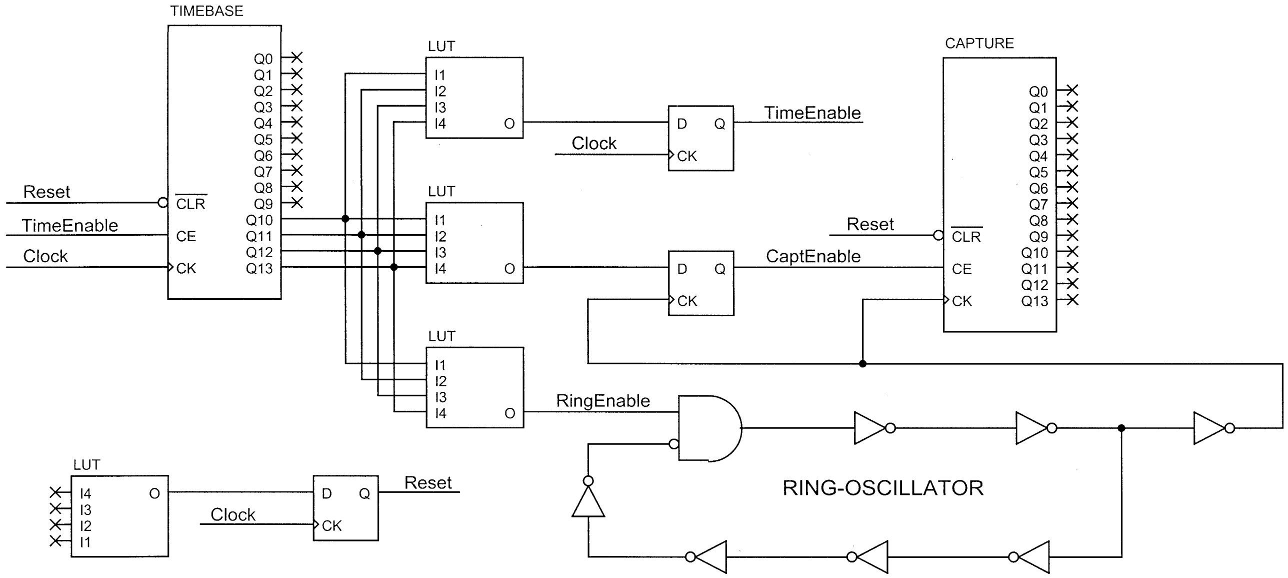


Fig. 3. Schematic of the reconfigurable temperature sensor.

timebase counter when it is no longer necessary. Consequently, when the measurement has finished all the circuit activity is disabled and any spurious power dissipation is avoided.

The above schema allows an arbitrary specification of the three enable times in steps of 1024 cycles (10-b count) of the main FPGA clock. For example, in the experiments the ring-oscillator runs 2048 cycles in each measurement. As a 3.6864 MHz clock was used, this corresponds to 555 μs. The capture counter is enabled during the second half of that time (277 μs), leaving the first 1024 cycles as a safe margin for oscillation start-up. Taking advantage of the flexibility of the timing, all these constants where empirically adjusted during the setup of the experiments.

Once the measurement has been made, the capture counter stores a value that is proportional to the frequency output of the ring-oscillator, and therefore, to the die temperature. This count is read back by the host processor, which calculates the actual temperature. Thus, the use of I/O pads to transfer the data is eliminated.

The sensor was created as an RTP (run-time parameterizable) core, so it can be easily instantiated in any JBits-based design. The only parameters that should be specified are the timings of the three enable signals, expressed as the contents of the LUTs that generate them. The placement of the different elements that compose the sensor is depicted in Fig. 4. It shows the function of each slice (half CLB, configurable logic block) utilized to construct the sensor. As it can be seen, the total size is 8 CLBs tall by 2 CLBs wide. That is, less than 5% of the total area of the smallest Virtex device, the XCV50. This fact makes the sensor suitable for mapping the thermal state of the FPGA in different points, especially if bigger devices are utilized. For example, in the XCV800 model, a sensor occupies around 0.3% of the device.

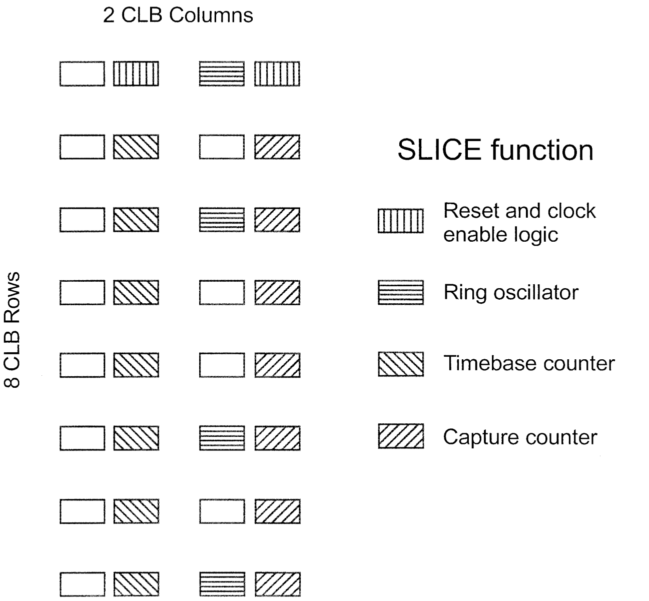


Fig. 4. Slice use of the sensor.

IV. Operation of a Run-Time Configurable Sensor

Run-time reconfiguration dynamically changes the contents of the configuration memory of an FPGA. It allows the designer to modify only specific blocks of the circuit, while the remainder continues its normal operation. In Virtex FPGAs, the configuration bitstream is organized in frames, each corresponding approximately to a vertical portion of a column of CLBs. Partial reconfiguration is available on a frame-by-frame basis [3]. Taking into consideration that the frame size in the XCV50 is 48 bytes, and that the maximum configuration speed is 50 MB/s, the minimum configuration change will take roughly 1 μs. Fast enough to permit real-time temperature measurements.

The use of the thermal strategy proposed in this work requires a system composed by one or many FPGAs, and a host microprocessor running a Java Virtual Machine (JVM). During the initialization, the host microprocessor configures the FPGAs, and the system start working. Whenever the host checks the thermal status of the board, the following steps are executed.

1) The current bitstream is inspected to find a free space where to insert the sensor. This operation is usually made statically, by providing a free space to insert the sensor at design time. But it is also possible to make it dynamically, at run time: this alternative is useful in evolvable and adaptive systems, where the designer could ignore what circuit will be running in the FPGA at a certain time.
2) Once a possible location for the sensor is found, its elements are inserted and routed. Routing is dynamic, once again to avoid being limited to static designs. The possibility of a static routing also exists, but it would require to previously reserve a zone totally free of wiring.
3) The new bitstream (with the sensor embedded in it) is generated by JBits, and the FPGA is partially reconfigured: only the modified frames are updated. That is, the sensor is inserted in the FPGA meanwhile the system continues its normal operation.
4) As the FPGA is only partially reconfigured, the state of the registers of the sensor is unknown (the global reset signal will not be asserted to permit the normal operation of the circuit that is already running in the FPGA). To solve this problem, first the circuit will be downloaded with its reset signals activated, so that the count will be initialized at zero. Since the reset signal is enabled, no activity will be present. Immediately after that, the sensor is reconfigured again with the reset signals de-asserted (only one LUT needs to be changed). Thus, the sensor starts running.
5) Finally, once the measurement has been made, the state of the sensor is read back to obtain the frequency of the ring-oscillator. Then, the host microprocessor translates this count into the actual temperature, using a pre-calculated table.

The protocol proposed above solves one important problem in run-time reconfiguration: the management of partial reset signals. In all FPGA families, a global reset signal is asserted after a full reconfiguration in order to initialize all flip-flops to a known state. Partial run-time reconfiguration is incompatible with this mode of operation, because the activation of the global reset initializes all FPGA registers, not only the ones in the portion of the block being inserted. Therefore, the part of the circuit that has not changed will not continue its normal operation, because its memory will be cleared. But if the global reset is not asserted, the new logic inserted in FPGA will initialize at an unknown state, possibly causing a failure, especially if contains a counter or an state machine.

In order to verify the feasibility of the strategy, a Xilinx AFX PQ240-100 prototyping board with an XCV50PQ240-4 was utilized. A FPGA configuration data with the complete sensor described in Fig. 3 was written and read back using its `SelectMAP` port. It was connected via a custom interface to a host PC running JBits. This link was implemented through the parallel port of the PC. The option lead to a simple, yet full connectivity to the FPGA, at a moderate configuration speed (around hundreds of KB per second).

In order to calibrate the sensor, the FPGA board was placed in a temperature-controlled oven. An iron–constantan (Fe–CuNi) thermocouple was placed in the center of the package to measure chip temperature. For each temperature step, the chip was configured with the sensor and kept idle until it reached the thermal equilibrium. After that, the oscillator was enabled only during 555 μs, and the count value was read back. The procedure was repeated for different temperatures and power supply voltages.

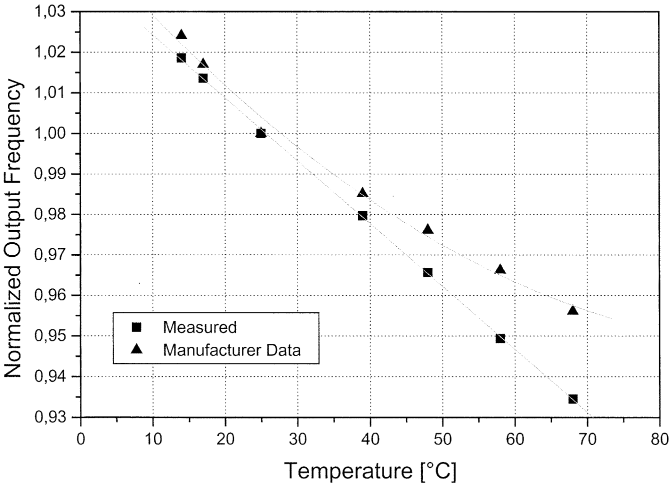


Fig. 5. Normalized ring-oscillator frequency response versus die temperature.

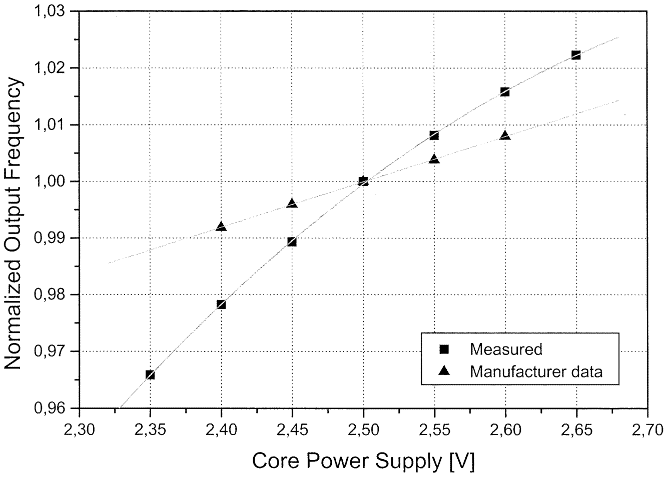


Fig. 6. Normalized ring-oscillator frequency response versus power supply voltage.

Fig. 5 shows the output of the sensor versus the die temperature, and Fig. 6 its dependence to power supply variations. The two graphics are normalized at 25 °C and 2.5 V. Ring-oscillators exhibit a linear response in the normal range of temperature operation. Their outputs are situated in a band of frequencies centered in 45.5 MHz. Both the experimental measurements and the ones predicted by the prorating option of the timing analysis tools are displayed. The graphics show that the variation of the oscillation frequency with the temperature is large enough to make this technique feasible. It could be also inferred that large increments of the chip temperature (such as those caused by serious circuit errors) cannot be masked by moderate (±5%) power supply variations. Finally, there is a significant difference between the actual results and the ones predicted by the tool provided by the chip manufacturer. This makes advisable a previous calibration of the sfensor if accurate measurements are needed.

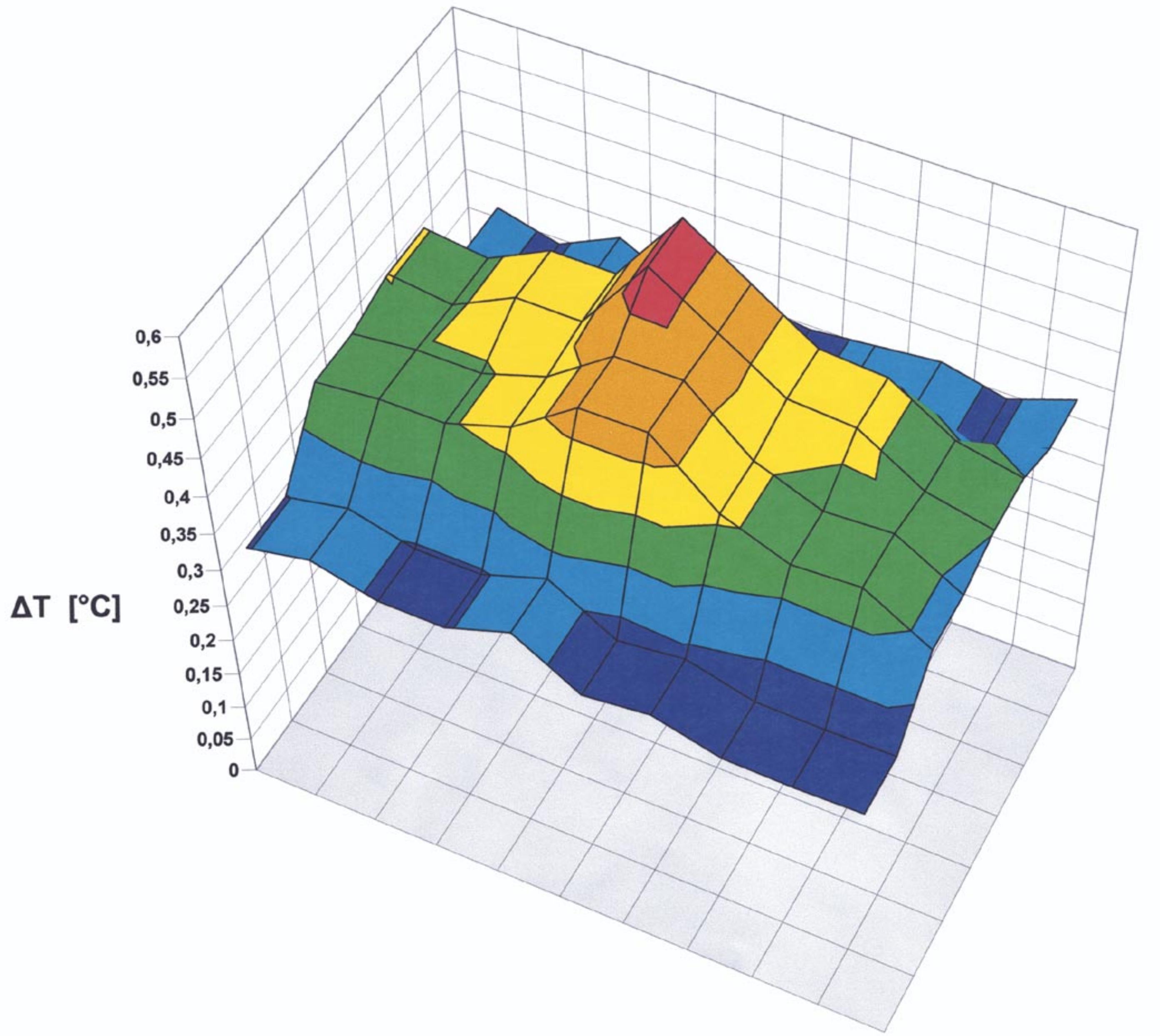


Fig. 7. Map of the temperature increments in the die as effect of the activation of a hot-spot.

Other previous results also reinforce the importance of an accurate calibration: topologically identical FPGA-based oscillators have the same temperature variation (near 0.20% per °C), but each of them runs at a different frequency for a given temperature [6].

As a final example of the proposed technique, Fig. 7 shows a map of the temperature increments in a bigger device (XCV800PQ240-4) after a hot-spot has been activated in the middle of the die. This source of heat is composed by 32 flip-flops and four-input look-up tables (LUTs) switching at 50 MHz, arranged in a 4 × 4 CLBs square (that is, about 0.3% of the die). When active, it dissipates 25 mW, causing an increment of temperature around its location. This hot-spot is detected by constructing a thermal map of the device, using an array of 4 by 10 sensors.

V. Conclusion

In this paper, a new method for the thermal testing of FPGA-based systems using run-time reconfiguration has been presented. This technique allows the measurement of the die temperature of the chips without permanent use of any programmable resources. The temperature sensor is completely stand-alone; that is, no A/D converters or other external devices are needed. Moreover, it is not necessary to make any PCB modifications to implement this idea (provided that run-time reconfiguration is already supported) as no additional I/O pads are needed. The technique is useful in FPGA-based systems to check if the devices are operating above the specified temperature range, or to detect errors that cause excessive power dissipation, like bus contention, unconnected inputs, timing errors in RAM modules, accidental activation of the JTAG port, etc.

The functionality of the JBits technology has been successfully tested, proving to be useful for thermal investigations. A variety of fast and low-cost experiments can be done taking advantages of the short design cycle and reprogrammability of the devices. This technique can even be used to create a thermal map of the device, without the requiring any external equipment. In this way, FPGAs became a complementary tool for those technologists interested in the thermal aspects of the packaging of integrated circuits.

ACKNOWLEDGMENT

The authors wish to thank E. Ruiz and P. Rodríguez for their valuable help during the setup of the experiments.

REFERENCES

[1] D. Buell, J. Arnold, and W. Kleinfelder, *Splash 2. FPGAs in a Custom Computing Machine*. New York: IEEE Press, May 1996.

[2] G. Goslin, "A guide to using field programmable gate arrays (FPGAs) for application-specific digital signal processing performance," Tech. Rep., Xilinx Inc., San Jose, 1995.

[3] Xilinx, Inc., "Virtex configuration architecture advanced user guide," Tech. Rep., Application Note XAPP151, 2000.

[4] J. Faura, C. Horton, P. Duong, J. Madrenas, M. Aguirre, and J. Insenser, "A novel mixed signal programmable device with on-chip microprocessor," in *Proceedings of the 1997 Custom Integrated Circuits Conference (CICC'97)*. New York: IEEE Press, 1997, pp. 103–106.

[5] M. Sipper, D. Mange, and E. Sanchez, "Quo vadis evolvable hardware," *Commun. ACM*, vol. 42, no. 4, pp. 50–56, Apr. 1999.

[6] S. Lopez-Buedo, J. Garrido, and E. Boemo, "Thermal testing on reconfigurable computers," *IEEE Design Test Comput.*, pp. 84–90, Jan.–Mar. 2000.

[7] G. Quenot, N. Paris, and B. Zavidovique, "A temperature and voltage measurement cell for VLSI circuits," in *Proceedings of the 1991 EURO ASIC Conference*. New York: IEEE Press, 1991, pp. 334–338.

[8] V. Szekely, C. Marta, Z. Kohari, and M. Rencz, "CMOS sensors for on-line thermal monitoring of VLSI circuits," *IEEE Trans. VLSI Syst.*, vol. 5, pp. 270–276, Sept. 1997.

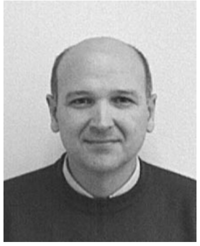
[9] V. Szekely, M. Rencz, A. Pahi, and B. Courtois, "Thermal monitoring and testing of electronic systems," *IEEE Trans. Comp. Packag. Technol.*, vol. 22, pp. 231–237, June 1999.

[10] Xilinx, Inc., "JBits Xilinx reconfigurable computing platform," Tech. Rep., San Jose, CA, JBits 2.7 Tutorial, 2000.

[11] S. McMillan and S. Guccione, "Partial run-time reconfiguration using JRTR," in *Proceedings of the FPL2000 Workshop*, ser. Lectures Notes in Computer Science 1896. Berlin, Germany: Springer-Verlag, 2000.

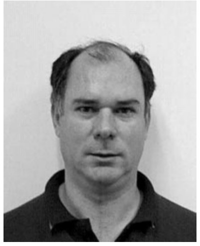
Sergio Lopez-Buedo (M'95) received the M.S. degree in telecommunication engineering from the Universidad Politecnica de Madrid, Spain, in 1997 and is currently pursuing the Ph.D. degree at the Universidad Autonoma de Madrid, Spain.

He is an Associate Professor of computer architecture with the Universidad Autonoma de Madrid. His current research interests include thermal testing, high-speed and low-power FPGA-based systems, computer arithmetics, and reconfigurable systems.



Javier Garrido (M'96) received the B.Sc., M.Sc., and Ph.D. degrees in physics from the Universidad Autonoma de Madrid, Spain, in 1974, 1976, and 1984, respectively.

He is currently a Titular Professor of computer architecture at the School of Computer Engineering, Universidad Autonoma de Madrid, Spain. His current research interests are in the areas of semiconductor technologies (such as dielectrics layers and III–V etching). He is also involved in the design of FPGA-based systems and low-power techniques.



Eduardo I. Boemo received the electrical engineering degree from the Universidad Nacional de Mar del Plata, Argentine, in 1985 and the Ph.D. degree in telecommunication engineering from the Universidad Politécnica de Madrid, Spain in 1996.

He is Titular Professor of ASIC design and Vice Director of Research at the School of Computer Engineering, Universidad Autónoma de Madrid, Spain. Current research interests include the design of FPGA-based systems, low-power techniques, computer arithmetics, thermal testing, self-timed circuits, and E.E. education.