

Article

Towards Real-Time Monitoring of Thermal Peaks in Systems-on-Chip (SoC)

Aziz Oukaira ^{1,*} , Ahmad Hassan ¹ , Mohamed Ali ^{1,2} , Yvon Savaria ¹  and Ahmed Lakhssassi ³ ¹ Electrical Engineering Department, Polytechnique Montreal, Montreal, QC H3T 1J4, Canada² Microelectronics Department, Electronics Research Institute, Cairo 12622, Egypt³ Department of Engineering Computer Science, University of Québec in Outaouais, Gatineau, QC J8X 3X7, Canada

* Correspondence: aziz.oukaira@polymtl.ca

Abstract: This paper presents a method to monitor the thermal peaks that are major concerns when designing Integrated Circuits (ICs) in various advanced technologies. The method aims at detecting the thermal peak in Systems on Chip (SoC) using arrays of oscillators distributed over the area of the chip. Measured frequencies are mapped to local temperatures that are used to produce a chip thermal mapping. Then, an indication of the local temperature of a single heat source is obtained in real-time using the Gradient Direction Sensor (GDS) technique. The proposed technique does not require external sensors, and it provides a real-time monitoring of thermal peaks. This work is performed with Field-Programmable Gate Array (FPGA), which acts as a System-on-Chip, and the detected heat source is validated with a thermal camera. A maximum error of 0.3 °C is reported between thermal camera and FPGA measurements.

Keywords: Field-Programmable Gate Array (FPGA); Integrated Circuits (ICs); System on Chip (SoC); Gradient Direction Sensor (GDS); thermal camera; thermal monitoring; thermal peak



Citation: Oukaira, A.; Hassan, A.; Ali, M.; Savaria, Y.; Lakhssassi, A. Towards Real-Time Monitoring of Thermal Peaks in Systems-on-Chip (SoC). *Sensors* **2022**, *22*, 5904. <https://doi.org/10.3390/s22155904>

Academic Editors: Marco Lanuzza and Lionel Trojman

Received: 6 July 2022

Accepted: 4 August 2022

Published: 7 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The evolution of integrated circuits (ICs) has led to the design of increasingly dense circuits to allow implementing much more complex systems in a smaller silicon area [1]. This allows for reducing manufacturing costs and boosting systems' performance. However, chips with high integration density dissipate high power, which consequently induces overheating problems that can cause disastrous thermal peaks. Thus, an appropriate management of thermal dynamics including thermal monitoring is required to avoid performance degradation and lifetime reduction of ICs [2–4]. On-chip thermal behavior can be obtained by adopting an on-chip distributed oscillator network [5,6], where the frequency variations generated by the integrated oscillators indicate the thermal changes in the chip. Then, using the Gradient Direction Sensor (GDS) technique the authors could monitor the temperature source of the thermal sensors on the surface of the chip [7]. This source implies isotherms around it. The GDS method is a practical solution to locate the thermal peak in the simplest case of a single heat source [8,9], and the functional range of this GDS method is between −55 °C and 125 °C because, as the supply voltage decreases, the temperature dependence of the propagation delay can change from negative to positive for supply voltage levels between 0.7 V and 1.2 V [10,11]. However, the measured temperatures are not processed in real-time, which has an impact on the accuracy of the readings. In order to improve this accuracy, we propose in this work to collect and process the temperatures in real-time based on FPGAs. Therefore, FPGAs offer the possibility to perform real-time simulation and implementation with a time step of tens of nanoseconds in addition to the real-time reconfiguration that dynamically modifies the memory content of an FPGA [12].

In this paper, we propose a method that aims to detect thermal spikes in SoC using oscillator arrays distributed on the chip surface. An indication of the local temperature

of a single heat source is obtained using the GDS technique. The measured frequencies are mapped to local temperatures which are used to produce a thermal map of the chip. The proposed technique does not require external sensors.

This work is validated with a Field-Programmable Gate Array (FPGA) implementation and temperature measurements performed with a thermal camera. A maximum error of $0.3\text{ }^{\circ}\text{C}$ is reported between thermal camera and FPGA measurements. This is significant as the majority of designers of ICs and system-on-chips (SoCs) do not have effective means of predicting an internal thermal cartography in real-time.

This paper is organized as follows: Section 2 describes the theory of operation of the on-chip thermal peak detection unit, and the modeling analysis of the single heat source equation of the thermal peak detection unit. The obtained simulation and hardware implementation results, compared to a temperature prediction by the GDS method, are given in in Section 3. Section 4 concludes the paper by summarizing our main contributions and findings.

2. Theory of Operation of the Proposed on-Chip Thermal Peak Detection Unit

The heat generated by the operation of various circuits in an SoC creates heat sources at different points on the chip. Numerous point sources can be approximated as a single source (the point at that maximum temperature). Figure 1 illustrates the proposed architecture of a real-time thermal monitoring system based on the GDS method.

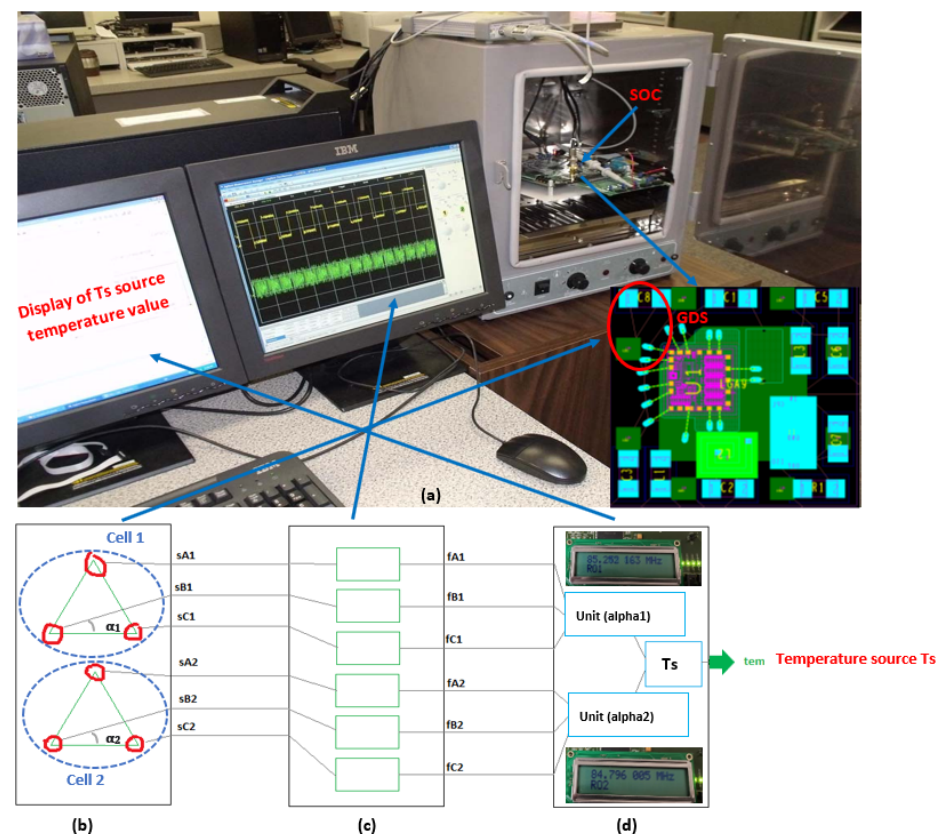


Figure 1. Proposed architecture of a real-time thermal monitoring system based on the GDS method: (a) operation of the on-chip thermal peak detection units, (b) RO units on an SoC; (c) the frequency counters, and (d) the computer to analyze the received data.

Figure 1a shows the three RO units for two cells on the SoC. Figure 1b shows the frequency counters connected to the RO units, and Figure 1c shows the computer used to analyze the received data that completes the thermal peak detection unit. In this system, six well placed thermal sensors transform temperatures into frequency signals. The obtained RO frequency values give information about the heat source T_s . The temperature

measurements are derived from signals sA1, sB1, and sC1 representing sensors A, B, and C, respectively located in cell 1 and from sA2, sB2, sC2 representing sensors A, B, and C, respectively located in cell 2. In Figure 1b, fA1, fB1, and fC1 represent frequencies obtained for A, B, and C, respectively located in cell 1 and fA2, fB2 and fC2 represent frequencies obtained for A, B, and C, respectively located in cell 2. In Figure 1c, the two angles $\tan(\alpha_1)$ and $\tan(\alpha_2)$ represent the deviation of the respective triangles of ROs from the heat source T_s . The two angles α_1 and α_2 will be evaluated subsequently based on Equations (1) and (2), in order to evaluate the performance of our method.

2.1. The Frequency Counter

As shown in Figure 1b, counters are employed to calculate the oscillation frequency of each RO. As shown in Figure 2, the counter counts until the reset signal is activated and restarts the counter. In the same figure, it counts up to 10 and then the reset signal goes to 1, which restarts the counter at 0.

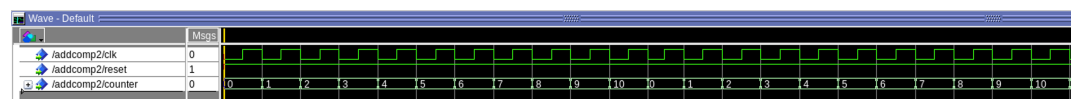


Figure 2. The counter counts until the reset signal is enabled.

Our frequency counter based on regression incurs almost no expense. This regression causes almost no overcharge. At run-time, the counter only has to calculate the temperature from a simple formula T_s (Equation (3)). This does not affect our performance unlike performance counters that required expensive online computations [13–15].

2.2. Calculation of the Gradient Angle Units for the GDS Method

2.2.1. The Unit to Compute the Angle α

As we already described in Figure 1c, two units are used to calculate the angles α_1 and α_2 , respectively. Figure 3 illustrates the operating principle of the GDS method. This unit is used to evaluate the position and value of a single heat source on the surface of an SoC. It estimates the geometrical coordinates and temperature of the heat source. To obtain information on the parameters of a single heat source, we provide information on where the thermal peak comes from and the speed at which it evolves. To calculate the temperature of the heat source, a thermal peak detection unit was used. This unit receives the angles α_1 and α_2 (computed by the angle calculation unit), the distance h between cell 1 and cell 2, and the oscillation frequency of each RO. The factors requiring special attention during the development of this unit are the number of sensors, the spatial distribution of the heat source, and the network interconnections. This unit provides the angle α that is shown in Figure 3 by according the VHDL code developed in Section 3.

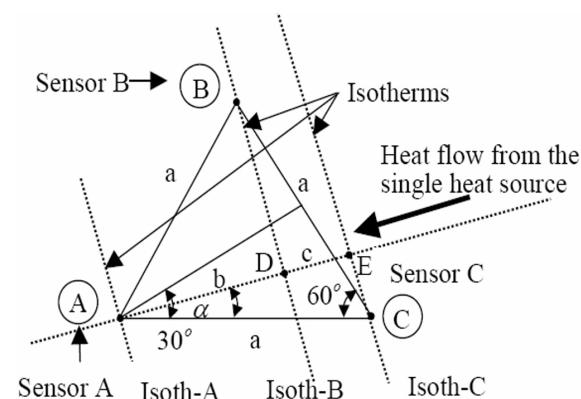


Figure 3. Placement of three sensors cell with $\alpha \in (0^\circ, 60^\circ)$.

The GDS method is based on the three embedded sensors adopted to form the first cell. Each cell is composed of three ROs that form a triangle. This unit was designed and tested using VHDL. Equations (1) and (2) define the main functionality of the angle calculation unit that have been simulated and implemented:

$$\tan(\alpha_1) = \frac{2(f_{B1} - f_{A1})}{\sqrt{3}(f_{C1} - f_{A1})} - \frac{1}{\sqrt{3}} \quad (1)$$

and

$$\tan(\alpha_2) = \frac{2(f_{B2} - f_{A2})}{\sqrt{3}(f_{C2} - f_{A2})} - \frac{1}{\sqrt{3}} \quad (2)$$

where α_1 and α_2 indicate the position relative to the heat source. Both formulas were coded in VHDL. As depicted in Figure 4, the angle calculation unit receives the generated frequencies f_A , f_B , and f_C from the three sensors A, B, and C, respectively, and calculates the tangents and angles (in radians) corresponding to these values.

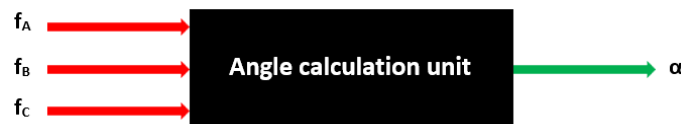


Figure 4. Angle calculation unit.

2.2.2. Source Temperature (T_s) Calculation

In order to obtain the temperature value of a single punctual heat source, we have to calculate the distance between the sensor and this source. Two sensor cells are required for this purpose as depicted in Figure 5. The cells are placed in a given distance (H), and each gives information about angle α (α_1 and α_2) in the direction of the heat source. Under the consideration that α is relatively small, we can assume that the heat source and the center of the cells form a triangle in which the length of one side and values of the angles adjacent to this side are known. This means that we can calculate the distances $R1$ and $R2$ between the heat source and the sensors. Now we can calculate the temperature gradient along the known distance. By adding distance to the temperature of the sensor, we obtain the temperature of the heat source. Two sensor cells A1, B1, C1 and A2, B2, and C2 are placed in two corners of a monitored layout at the distance H . Hence, the temperature of the heat source can be obtained by Equation (3). Figure 5 shows the description and distribution of the sensors cells:

$$\frac{\frac{H}{a}(f_{c1} - f_{A1})(\sqrt{3} + \tan \alpha_2)(1 + \tan^2 \alpha_1)}{\sqrt{3}(1 - \tan \alpha_1 \tan \alpha_2) - (\tan \alpha_1 + \tan \alpha_2)} + f_{A1} \rightarrow T_s \quad (3)$$

The effects on performance caused by temperature fluctuations are most often treated as linear scaling, but some sub-micron silicon processes require nonlinear calculations. For this reason, the oscillator network must be activated for a short period of time in order to avoid thermal variations involved by the oscillator itself (self heating).

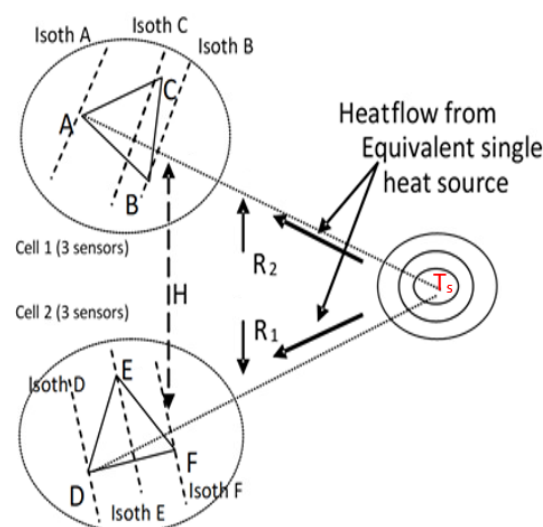


Figure 5. The description and distribution of the sensors' cells.

The temperature accuracy must be ensured with respect to a variation within $\pm 0.2\text{ }^{\circ}\text{C}$. The source temperature T_s is obtained at the output after the unit has processed the input data. The angle calculation unit was implemented and simulated in the ModelSim tool, where different values of f_A , f_B , and f_C are applied. The results obtained are compared with the calculated ones from the Excel tool. To allow obtaining a synthesizable VHDL code, the $\tan(\alpha)$ is calculated not as a real, but as an integer. A factor of 1000 is then used to obtain an integer result that equals 1000 times $\tan(\alpha)$. For simulation based validation, we used eight different parameter sets (combinations of f_A , f_B and f_C).

The results obtained through ModelSim are presented in Figure 6. Using the frequencies received from the frequency counter unit shown in Figure 1b, we can extract the results of the $\tan(\alpha)$ from this simulation presented in Figure 6. For example, the frequency collected from sensor A is $f_A = 101,000\text{ kHz}$, the frequency collected from sensor B is $f_B = 108,000\text{ kHz}$, and the frequency collected from sensor C is $f_C = 112,000\text{ kHz}$ given a value of $\tan \alpha = 0.157$ (see the part circled in red in Figure 6).

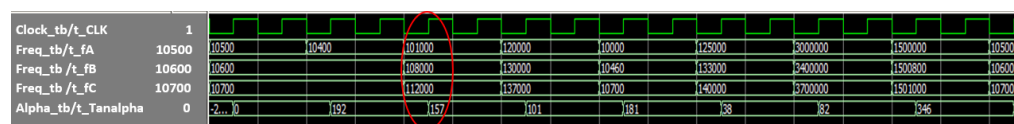


Figure 6. Simulation results of the angle calculation unit.

Moreover, the results shown in Figure 6 are compared with the results obtained Excel in Table 1.

Table 1. Comparison between results obtained with ModelSim and Excel.

f_a (kHz)	f_b (kHz)	f_c (kHz)	$\tan(\alpha)$ with Excel	$\tan(\alpha)$ with ModelSim
10,500	10,600	10,700	0	0
10,400	10,600	10,700	0.192	0.192
101,000	108,000	112,000	0.157	0.157
120,000	130,000	137,000	0.101	0.101
10,000	10,460	10,700	0.181	0.181
125,000	133,000	140,000	0.037	0.038
3,000,000	3,400,000	3,700,000	0.081	0.082
1,500,000	1,500,800	1,501,000	0.345	0.346

As shown in Table 1, the error is small between the Excel and Modelsim models (where line 6 shows the maximum difference of $\Delta = 0.001$ radians), which shows the efficiency of

our test bench code developed in VHDL. In addition, we have simulated the temperature calculation unit T_s (Equation (3)) using ModelSim. T_s is computed as an integer. A factor of 100 is then used to obtain an integer result that equals 10,000 multiplied by T_s .

The results obtained by ModelSim are presented in Figure 7. Using the received frequencies and the two units that are presented in Figure 1c, we can extract the results of the source temperature T_s from this simulation presented in Figure 7; for example, for the $\tan(\alpha_1) = 0.16$ and $\tan(\alpha_2) = 0.35$, give a value of source temperature $T_s = 10.1051$ °C (see the part circled in red in Figure 7).

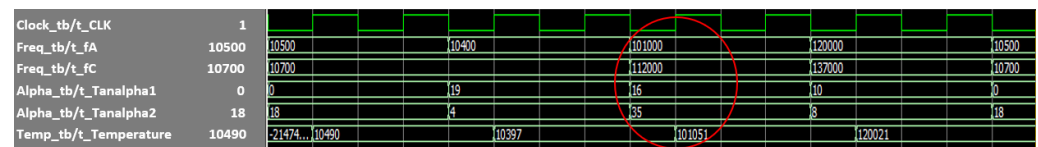


Figure 7. Simulation results of the temperature calculation unit.

To validate the results in Figure 7, we compare the results obtained with Excel and the following Table 2, which confirms the validity of the VHDL model.

Table 2. Results obtained with the ModelSim and Excel.

f_a (kHz)	f_c (kHz)	$\tan(\alpha_1)$	$\tan(\alpha_2)$	T_s with Excel	T_s with ModelSim
10500	10700	0	0.18	1.0486	1.0490
10,400	10,700	0.19	0.04	1.0391	1.0397
101,000	112,000	0.16	0.35	10.1046	10.1051
120,000	137,000	0.10	0.08	12.0020	12.0021

A good matching was achieved between the results obtained by Excel and ModelSim. The maximum difference is $\Delta = 0.06\%$ (as shown in line 2 of Table 2). This small difference is due to the approximations made with the integer model.

3. Implementation and Results

For the experiment, we used a DE1 Altera FPGA board and the Quartus Prime software. The FPGA acts as an SoCs with the ability to locate and implement the required ROs for our sensor cells, whereas Quartus Prime provides us with a frequency counter. The ring oscillators are implemented by default in the Altera source code by considering six LookUp Table (LUT) or six delays by default. In addition, Altera's Quartus Prime configuration defaults to an ambient temperature of 25 °C. This can be seen in Figure 8.

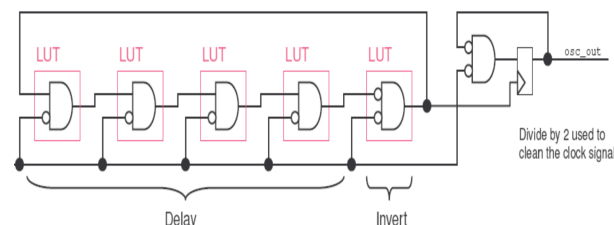


Figure 8. Configuration of the ring oscillator provided by Altera.

In this work, we did not focus on the optimization of the delay cell circuits, which is a task left for future work. Based on Figure 8, our experimental design divided into three main parts: simulation, synthesis, and implementation of the VHDL code. Firstly, we developed the VHDL code, and then the Register Transfer Level (RTL) version of the model was generated as shown in Figure 9.

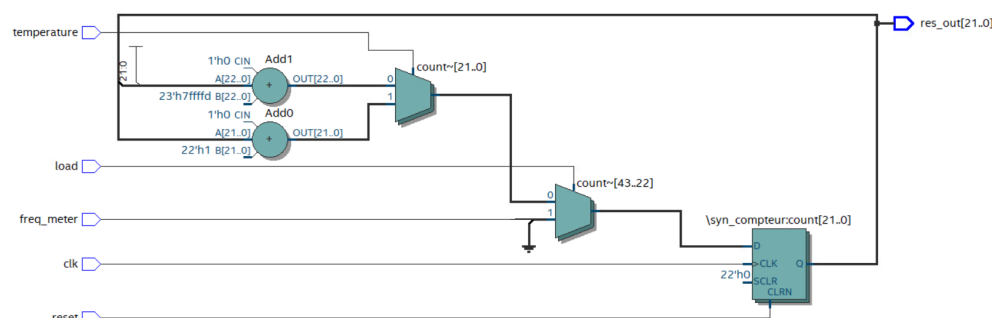


Figure 9. Register Transfer Level structure of the temperature monitoring unit with Quartus Prime.

The logical simulation of the developed VHDL codes is shown in Figure 10, where three correct values of the frequency as a function of the LUT number were obtained (see the part circled in red in Figure 10).

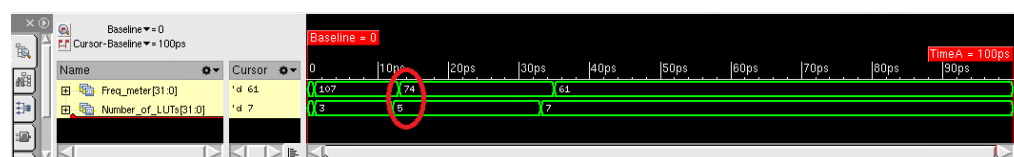


Figure 10. Simulation of the temperature monitoring unit with NClaunch tool.

Among the advantages of VHDL coding is the use of the test bench, which allows for verifying the capability of our algorithm to operate the GDS method according to the initial specifications. Test vectors were subsequently created to ensure specific coverage by optimizing the test time, or to minimize performance degradation.

We report that, in this work, we have a conversion time that is around 641 μ s, and a conversion rate of 7.4 kHz.

3.1. Implementation on the FPGA Board

It is worth noting that the Quartus Prime software allows a manual placement of the circuit on the FPGA. Figure 11 shows the physical location of the six ROs (The RO occupies a 0.39 μ m²) on the FPGA.

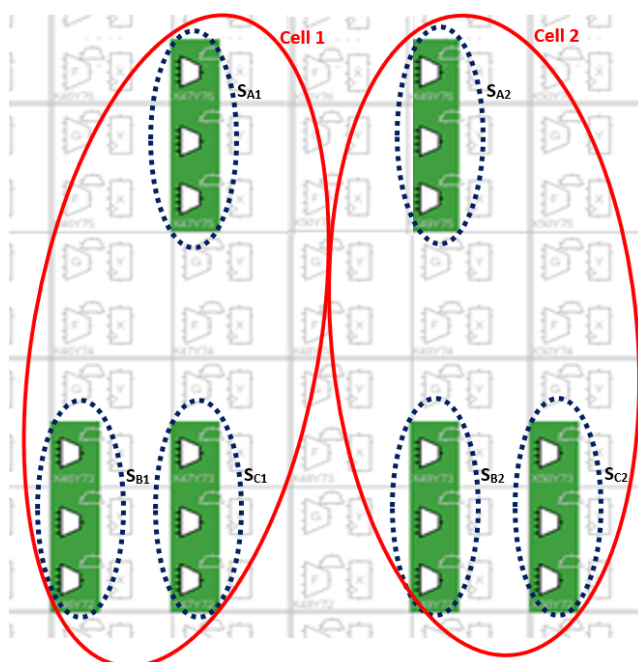


Figure 11. Physical location of the six ROs sensors on the FPGA.

Figure 11 represents the implementation of the six ROs on the FPGA board. After downloading the VHDL code, the program was running and the outputs were displayed in Figure 12.

The clock is set to 50 MHz, and a dryer is used to increase the temperature of the DE1 (Development and Education) FPGA board. Figure 12 shows the obtained temperature results.

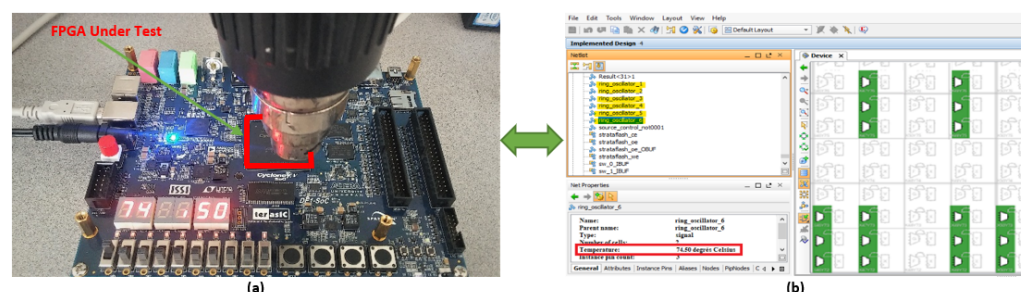


Figure 12. Read temperature by: (a) the FPGA board and (b) the Quartus Prime tool.

Figure 12a shows the temperature displayed by FPGA under test. Figure 12b shows all the information pertinence's on the six RO sensors by the Quartus Prime tool. Our experimental results show that the detected temperature from the six ROs is 74.50 °C.

As mentioned in the Introduction, the purpose is to detect thermal peaks using the GDS method, and we tried to vary the temperature to see its influence on the frequency. The obtained values are reported in the Table 3 below:

Table 3. Results of the oscillation frequency versus temperature obtained.

Temperature (°C)	Frequency (MHz)
25	99.88
30	98.21
40	96.12
50	94.18
60	92.08
70	90.20
80	88.10
90	86.04
100	84.17
110	82.12
120	80.02

Table 3 shows that, as the temperature increases, the frequency decreases by a constant value of 2.07 MHz for each 10 °C difference.

In order to validate the results found in Figure 12 and show the ability of the proposed GDS method to monitor the temperature in real-time, we then proceeded with thermal camera measurements as reported in the next section.

3.2. Experimental Measurements by the Thermal Camera

In this work, a JENOPTIK infrared camera [16] is used to capture the board's temperatures. It offers a spectral range of 7.5 to 13 μm with a frequency of 61 MHz and an image resolution of 320×240 pixels. It also offers an accuracy of ± 0.2 °C.

The software used for this camera is the IRT analyzer. It is installed on a computer on which we make the acquisition and extraction of thermal measurements.

Figure 13 shows the whole setup used for measurement by infrared thermography.



Figure 13. Thermal measurement system exploiting an infrared camera.

To extract the temperature values from the thermal camera, we used an image processing tool (IRT analyzer) that allows us to analyze the raw data collected by the infrared camera and convert them into digital data that can be used in the validation process.

Figure 14 shows an overview of the IRT analyzer tools. The use of the high precision IRT tool greatly increases the sensitivity and quality of the data received by the thermal camera, as well as the ability to analyze the experimental measurements.

The results displayed in Figure 14 are obtained using the generation of the library integrated under the IRT tool and the measurements that are performed via the thermal camera.

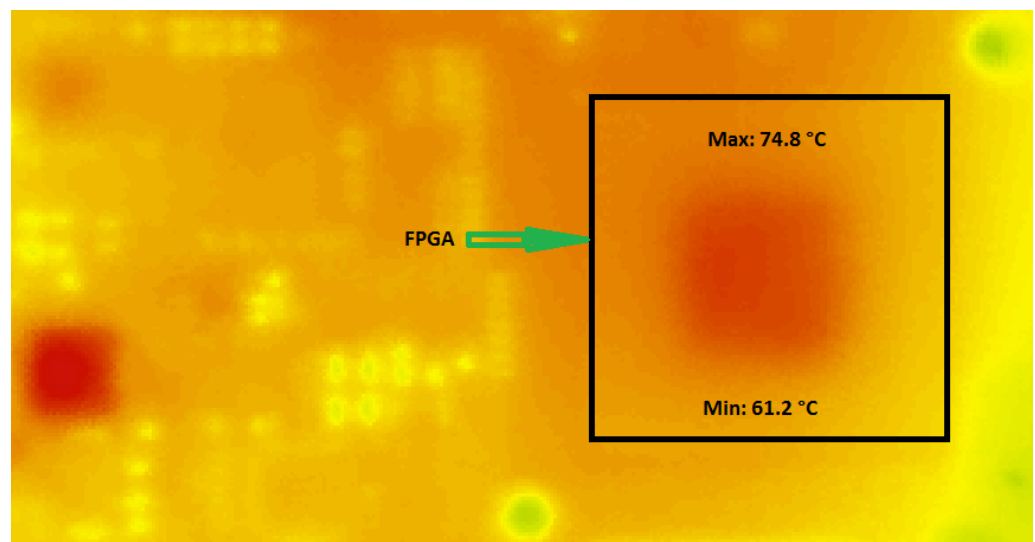


Figure 14. Temperature measurement of the FPGA board by thermal camera.

Figure 14 shows the areas where the highest temperature was observed (74.80 °C) on the DE1 map. The realization of a temperature measurement system by infrared thermography allowed us to extract the thermal experimental values with an emissivity equal to 1.

To validate the proposed method, we proceed to a comparison between thermal camera and FPGA measurement where the following Table 4 summarizes the obtained results.

Table 4. Comparison between thermal camera and FPGA measurement results.

Measurements	by Thermal Camera	by GDS	Error Rates (°C)
Temperature (°C)	74.80	74.50	0.3

This table clearly shows that the temperature values are almost the same for both results. The maximum error is about 0.3 °C, which indicates a good result for the validation of our method. Table 5 summarizes the GDS method on integrated circuits compared to similar works.

Table 5. Comparison of the performance of the GDS method on integrated circuits compared to similar works.

References	Proposed	[17]	[18]	[19]	[20]	[21]
Method	GDS ^(a)	STS ^(b)	FEM ^(c)	DTM ^(d)	TSERO ^(e)	DTS ^(f)
Temperature (°C)	110 ^(*)	75	80	100	120	125
Error (°C)	0.3	0.8	3.7	2.8	2.9	0.6
Real-time Monitoring	Yes	No	No	No	No	No

^(a) Gradient Direction Sensor (GDS); ^(b) Smart Temperature Sensor (STS); ^(c) Finite Element Method (FEM); ^(d) Dynamic Thermal Management (DTM); ^(e) Temperature Sensor Employs two Ring Oscillators (TSERO); ^(f) Digital Temperature Sensor (DTS). ^(*) Maximum measured temperature.

The performance of the proposed method based on ROs is reported in Table 5. Our results are compared to those of other methods' solutions presented in [17–21] (see Table 5). We can deduce that our work has significant potential, especially in terms of error and real-time data processing, which is not presented anywhere else. This performance includes the speed of obtaining the information that will allow us to intervene in real-time, especially since the majority of high-throughput SoCs so far do not have effective ways to predict thermal peaks and assess temperature in real-time.

4. Conclusions

This paper proposes a new thermal monitoring method that exploits in-situ temperature measurements derived from ring oscillator (RO) frequency measurements for thermal peak detection. A thermal peak detection unit was designed. Frequencies are converted to temperatures using the proposed models. In order to verify the presented method, we implemented three RO sensors for each cell on an FPGA board and temperature measurements were validated with a thermal infrared camera. A maximum error of 0.3 °C was observed between measured and validated temperatures.

The objective of this research is to obtain information about thermal peaks in real-time, which helps designers to react timely to possible hazards caused by thermal hot spots. The paper also provides other benefits such as helping to characterize and locate thermal peaks. Through simulations and experiments, it was shown that ring oscillators (ROs) are capable of providing in-situ temperature measurements.

This work offers a solution that allows for identifying thermal induced stress and local overheating of integrated systems that are major concerns for integrated circuits' designers.

A limitation of our proposal is that the integrated sensors are designed in CMOS technology, which is limited by the maximum operation temperature (around 120 °C). In addition, the sensitivity of the integrated sensors must be improved by improving the circuitry of ring oscillators. Our future work will focus on improving the sensitivity of integrated sensors and optimizing the accuracy of the proposed technique.

Author Contributions: A.O. developed the first version of the proposed thermal monitoring system and wrote a first version of the paper, A.H. completed the work and improved the monitoring system, M.A. drove deep revision of the paper to bring it in its present form, Y.S. advised the two first authors in all steps of the work, from early conceptualization, paper writing, and revising to get it in its present form, and he also manages the project that funds this work, and finally A.L. is a senior author and thermal modeling specialist, who supervised all steps of the work, from conceptualization to paper writing and revision. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Sciences and Engineering Research Council of Canada and the Regroupement Stratégique pour la Microélectronique du Québec.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors do not declare any conflict of interest.

References

1. Ali, M.; Hassan, A.; Honarparvar, M.; Nabavi, M.; Audet, Y.; Sawan, M.; Savaria, Y. A Versatile SoC/SiP Sensor Interface for Industrial Applications: Implementation Challenges. *IEEE Access* **2022**, *10*, 24540–24555. [\[CrossRef\]](#)
2. Salvi, S.S.; Jain, A. A Review of Recent Research on Heat Transfer in Three-Dimensional Integrated Circuits (3D ICs). *IEEE Trans. Compon. Packag. Manuf. Technol.* **2021**, *11*, 802–821. [\[CrossRef\]](#)
3. Iradukunda, A.-C.; Huitink, D.R.; Luo, F. A review of advanced thermal management solutions and the implications for integration in high-voltage packages. *IEEE J. Emerg. Sel. Top. Power Electron.* **2019**, *8*, 256–271. [\[CrossRef\]](#)
4. Li, X.; Li, Z.; Zhou, W.; Duan, Z. Accurate on-chip temperature sensing for multicore processors using embedded thermal sensors. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2020**, *28*, 2328–2341. [\[CrossRef\]](#)
5. Shor, J. Compact thermal sensors for dense cpu thermal monitoring and regulation: A review. *IEEE Sens. J.* **2020**, *21*, 12774–12788. [\[CrossRef\]](#)
6. Rahmanikia, N.; Amiri, A.; Noori, H.; Mehdipour, F. Performance evaluation metrics for ring-oscillator-based temperature sensors on FPGAs: A quality factor. *Integration* **2017**, *57*, 81–100. [\[CrossRef\]](#)
7. Wójciak, W.; Napieralski, A. Thermal monitoring of a single heat source in semiconductor devices—The first approach. *Microelectron. J.* **2021**, *28*, 313–316. [\[CrossRef\]](#)
8. Lakhssassi, A.; Bougataya, M.; Boustany, C.; Massicotte, D. Thermal stress monitoring using gradient direction sensors. In Proceedings of the IEEE International Northeast Workshop on Circuits and Systems and TAISA Conference, Montreal, QC, Canada, 22–25 June 2008; pp. 177–180. [\[CrossRef\]](#)
9. Oukaira, A.; Ettahri, O.; Lakhssassi, A. Modeling and FPGA implementation of a thermal peak detection unit for complex system design. *IJACSA Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 307–312. [\[CrossRef\]](#)
10. Xie, S. The Design Considerations and Challenges in MOS-Based Temperature Sensors: A Review. *Electronics* **2022**, *11*, 1019. [\[CrossRef\]](#)
11. Zambrano, B.; Garzón, E.; Strangio, S.; Crupi, F.; Lanuzza, M. A 0.05 mm², 350 mV, 14 nW Fully-Integrated Temperature Sensor in 180-nm CMOS. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *69*, 749–753. [\[CrossRef\]](#)
12. Lopez-Buedo, S.; Garrido, J.; Boemo, E.I. Dynamically inserting, operating, and eliminating thermal sensors of FPGA-based systems. *IEEE Trans. Compon. Packag. Technol.* **2002**, *25*, 561–566. [\[CrossRef\]](#)
13. Weissel, A.; Bellosa, F. Dynamic thermal management for distributed systems. In Proceedings of the 1st Workshop on Temperature-Aware Computer Systems, Munich, Germany, 19–23 June 2004.
14. Bellosa, F.; Weissel, A.; Waitz, M.; Kellner, S. Event-driven energy accounting for dynamic thermal management. In Proceedings of the Workshop on Compilers and Operating Systems for Low Power (COLP'03), New Orleans, LA, USA, 27 September 2003; Volume 22.
15. Lee, K.-J.; Skadron, K. Using performance counters for runtime temperature sensing in high-performance processors. In Proceedings of the IEEE International Parallel and Distributed Processing Symposium, Denver, CO, USA, 4–8 April 2005; Volume 25, pp. 8–16. [\[CrossRef\]](#)
16. Rangel, J.; Soldan, S.; Kroll, A. 3D thermal imaging: Fusion of thermography and depth cameras. In Proceedings of the International Conference on Quantitative InfraRed Thermography, Bordeaux, France, 7–11 July 2014; Volume 3. [\[CrossRef\]](#)
17. Chen, P.; Shie, M.-C.; Zheng, Z.-Y.; Zheng, Z.-F.; Chu, C.-Y. A fully digital time-domain smart temperature sensor realized with 140 FPGA logic elements. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2007**, *54*, 2661–2668. [\[CrossRef\]](#)
18. Ziabari, A.; Park, J.-H.; Ardestani, E.K.; Renau, J.; Kang, S.-M.; Shakouri, A. Power blurring: Fast static and transient thermal analysis method for packaged integrated circuits and power devices. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2014**, *22*, 2366–2379. [\[CrossRef\]](#)

19. Zambrano, B.; Garzon, E.; Strangio, S.; Iannaccone, G.; Lanuzza, M. A 0.6 V–1.8 V Compact Temperature Sensor with 0.24° C Resolution, $\pm 1.4^\circ$ C Inaccuracy and 1.06 nJ per Conversion. *IEEE Sens. J.* **2022**, *22*, 11480–11488. [[CrossRef](#)]
20. Huang, Q.; Joo, H.; Kim, J.; Zhan, C.; Burm, J.. An energy-efficient frequency-domain CMOS temperature sensor with switched vernier time-to-digital conversion. *IEEE Sens. J.* **2017**, *17*, 3001–3011. [[CrossRef](#)]
21. Tang, Z.; Fang, Y.; Shi, Z.; Yu, X.-P.; Tan, N.N.; Pan, W. A 1770- μm^2 Leakage-Based Digital Temperature Sensor With Supply Sensitivity Suppression in 55-nm CMOS. *IEEE J. Solid State Circuits* **2020**, *55*, 781–793. [[CrossRef](#)]