Titre: Title:	Accurate On-Chip Thermal Peak Detection Based on Heuristic Algorithms and Embedded Temperature Sensors
Auteurs: Authors:	Djallel Eddine Touati, Aziz Oukaira, Ahmad Hassan, Mohamed Ali, Ahmed Lakhssassi, & Yvon Savaria
Date:	2023
Type:	Article de revue / Article
Référence: Citation:	Touati, D. E., Oukaira, A., Hassan, A., Ali, M., Lakhssassi, A., & Savaria, Y. (2023). Accurate On-Chip Thermal Peak Detection Based on Heuristic Algorithms and Embedded Temperature Sensors. Electronics, 12(13), 18 pages. https://doi.org/10.3390/electronics12132978

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

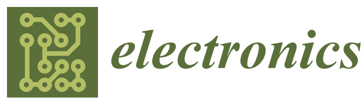
URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/54784/
Version:	Révisé par les pairs / Refereed
Conditions d'utilisation: Terms of Use:	CC BY

Document publié chez l'éditeur officiel
Document issued by the official publisher

Titre de la revue: Journal Title:	Electronics (vol. 12, no. 13)
Maison d'édition: Publisher:	MDPI
URL officiel: Official URL:	https://doi.org/10.3390/electronics12132978
Mention légale: Legal notice:	© 2023 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/).

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca



Article

Accurate On-Chip Thermal Peak Detection Based on Heuristic Algorithms and Embedded Temperature Sensors

Djallel Eddine Touati [1,*], Aziz Oukaira [2], Ahmad Hassan [2], Mohamed Ali [2], Ahmed Lakhssassi [1] and Yvon Savaria [2]

[1] Department of Engineering and Computer Science, Université du Québec en Outaouais, Gatineau, QC J9A 1L8, Canada; ahmed.lakhssassi@uqo.ca

[2] Electrical Engineering Department, Polytechnique Montréal, Montreal, QC H3T 1J4, Canada; aziz.oukaira@polymtl.ca (A.O.); ahmad.hassan@polymtl.ca (A.H.); mohamed.ali@polymtl.ca (M.A.); yvon.savaria@polymtl.ca (Y.S.)

* Correspondence: toud05@uqo.ca

Abstract: The reliability and lifetime of systems-on-chip (SoCs) are being seriously threatened by thermal issues. In modern SoCs, dynamic thermal management (DTM) uses the thermal data captured by thermal sensors to constantly track the hot spots and thermal peak locations in real time. Estimating peak temperatures and the location of these peaks can play a crucial role for DTM systems, as temperature underestimation can cause SoCs to fail and have shortened lifetime. In this paper, a novel sensor allocation algorithm (called thermal gradient tracker, TGT), based on the recursive elimination of regions that likely do not contain any thermal peaks, is proposed for determining regions that potentially contain thermal peaks. Then, based on an empirical source temperature detection technique called GDS (gradient direction sensor), a hybrid algorithm for detecting the position and temperature of thermal peaks is also proposed to increase the accuracy of temperature sensing while trying to keep the number of thermal sensors to a minimum. The essential parameters, H and R, of the GDS technique are determined using an automated search algorithm based on simulated annealing. The proposed algorithm has been applied in a system-on-chip (SoC) in which four heat sources are present, and for temperatures ranging between 45 °C and 115 °C, in a chip area equal to 25 mm^2. The simulation results show that our proposed sensor allocation scheme can detect on-chip peaks with a maximum error of 1.48 °C and an average maximum error of 0.49 °C by using 15 thermal sensors.

Keywords: finite element analysis (FEA); finite element method (FEM); GDS; heat transfer

Citation: Touati, D.E.; Oukaira, A.; Hassan, A.; Ali, M.; Lakhssassi, A.; Savaria, Y. Accurate On-Chip Thermal Peak Detection Based on Heuristic Algorithms and Embedded Temperature Sensors. *Electronics* **2023**, *12*, 2978. https://doi.org/10.3390/electronics12132978

Academic Editor: Domenico Caputo

Received: 25 May 2023
Revised: 30 June 2023
Accepted: 3 July 2023
Published: 6 July 2023

Copyright: © 2023 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

Due to the evolution of integration technologies with their ever-increasing power densities and clock frequencies, the performance, lifetime, and reliability of systems-on-chips (SoCs) have become more severely constrained by thermal issues than ever before. Failure mechanisms such as electromigration and thermomechanical stress, notably caused by the increased leakage power consumption [1–3], can increase the cooling cost and give rise to timing failures due to interconnect delays [4–7]. To tackle the thermal challenges associated with SoC implementation, designers commonly use thermal-aware floorplanning techniques to reduce thermal peaks and equally distribute temperature across a die. However, at the design stage, the workload profile and resource allocation are not fully known to designers, which makes the location of peaks and their values largely unpredictable [8].

To reduce runtime temperature and alleviate thermomechanical stress, the acquired thermal data from embedded thermal sensors are used by dynamic thermal management (DTM) algorithms, in which thermal sensors are used to continuously track the hot spots and possibly adjust operating frequencies, driving strength, voltages, and fan speeds dynamically, which in turn can reduce the die and hot spot temperature [9–11].

To allow efficient DTM, peak detection must be accurate, as any underestimation of the acquired temperature could cause late triggering of the DTM, which could lead to early system failure [12,13]. The two most popular ways to determine the runtime temperature are through dynamic insertion of thermal sensors in a reconfigurable chip [14] and the physical embedding of on-chip sensors [15].

Thermal sensor allocation methods have been extensively studied in the literature in recent years. Several thermal model-based approaches to address thermal sensor allocation problems have been presented [16–20]. These techniques rely on the thermal characteristics of materials included in the system. Although thermal model-based approaches can accurately predict the temperature of the whole chip, it is necessary to acquire the temperature of the die at run time using thermal sensors, because the temperature profiles predicted by thermal methods may deviate very significantly from actual values due to wear-out mechanisms [19,21].

Given the harmful impact of hot spots and the variations in their values and locations, a number of thermal sensor allocation approaches that use integrated thermal sensors have been presented for measuring runtime thermal behavior and then assist DTM systems. The authors in [22] reported a grid-based interpolation scheme for multiprocessor chips. In this approach, hot spots are estimated through interpolation of their sensed neighbors' data collected at grid points. However, this approach requires a large number of sensors to detect a few hot spots, and the accuracy of this method is workload-dependent.

A non-uniform allocation-based k-means clustering algorithm was provided in [23] to compute the optimal positions of thermal sensors that best identify all hot spot locations. However, the accuracy of hot spot detection decreases, as the number of thermal sensors is far less than the number of hot spot positions.

The authors in [24] realized both static and runtime thermal monitoring by using a k-LSE (least square errors) method based on the 2D discrete cosine transform (DCT). The disadvantage of this work is the failure to take into account the effects of noisy sensors.

To overcome errors associated with sensor readings, the authors in [25] presented a technique that relies on simultaneous identification of the fine-grain power and thermal models from power consumption and the thermal data collected from thermal sensor TS measurements. Other important studies have been conducted on developing techniques that can reconstruct thermal fields using a thermal camera [26–29]; however, 3D-IC thermal measurements by IR camera lead to errors in estimating the real die temperature.

As mentioned earlier, thermal sensors can dynamically be allocated in reconfigurable systems for runtime thermal management. The authors in [14] verified the dynamic insertion, operation, and elimination of different system configurations comprising ring oscillator-based thermal sensors implemented in an FPGA. However, the obtained results are inaccurate due to the large number of implemented sensors and the delays in processing the sensed thermal data. A fuzzy clustering map was presented in [30] to measure the on-chip temperature by using 6×8 thermal array sensors. Many other articles deal with the problem of allocation of thermal sensors on FPGAs, which we quote [21,31–33]. However, the approaches proposed in the articles required prior knowledge, which makes them time-consuming. Moreover, the proposed approach can only find the right location when the workload is fixed, which is not practical. In [34,35], the gradient direction sensor (GDS) technique was used to detect overheating spots on FPGAs; however, the authors did not address the problem of multiple overheating spots.

To overcome the aforementioned limitations, we propose in this paper a hybrid sensor allocation algorithm based on the GDS and thermal gradient tracker (TGT) techniques. The TGT approach recursively eliminates all regions with no thermal peaks and tracks the locations of thermal peaks. The remaining parts of this paper are organized as follows. Section 2 provides an overview of the GDS algorithm and the proposed allocation scheme. Section 3 reports the results obtained using the proposed techniques. Finally, the conclusion draw in this paper is in Section 4.

2. Background

As mentioned in the introduction, reconfigurable systems enable instantaneous in situ thermal sensing to be useful when the thermal behavior of a chip is unpredictable. Thermal sensors embedded in FPGAs are typically based on ring oscillators composed of an odd number of inverters. However, hot spot locations can change, and a large number of thermal sensors may be required to cover all regions of a large chip. The GDS technique could be adapted to scan a region of interest to estimate the temperature of heat sources. In the following subsections, we introduce the means for implementing thermal sensors in FPGAs and the GDS technique.

2.1. Ring Oscillator-Based Thermal Sensor

A ring oscillator must comprise an odd number of delay stages connected as a ring. Its oscillation frequency is sensitive to temperature variations. As the temperature rises, the frequency of the ring oscillator typically decreases, as depicted in Equation (1) [36]. Thus, the temperature can be measured by calibrating the output of a ring oscillator drifts in Hz per °C. The advantage of this method is its ability to measure the junction temperature rather than the temperature of the package.

$$f_{RO} = \frac{1}{2 \cdot N \cdot T} \tag{1}$$

where N is the number of delay stages, f_{RO} is the frequency of the ring oscillator, and T is the total delay seen at the output of each stage. Figure 1 shows the block diagram of the ring oscillator with its auxiliary circuits embedded in the FPGA.

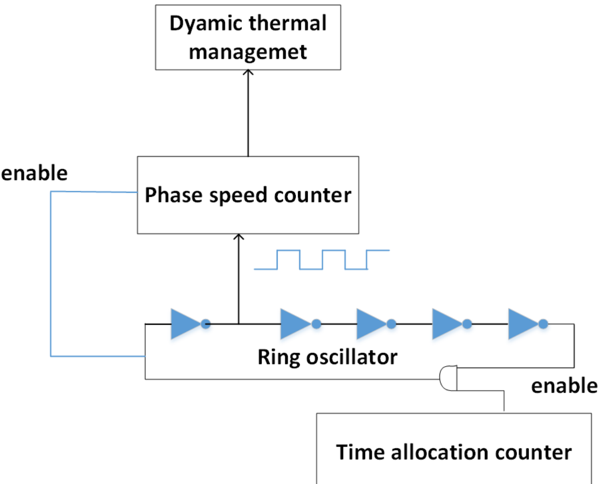


Figure 1. Thermal sensor architecture in an FPGA.

In this work, we use a five-stage ring oscillator as shown in Figure 1. Also, two counters are used, where the input of the first is connected to the ring oscillator that is used to measure the speed of the oscillation. As the temperature increases, the switching speed of the transistors and the oscillation slows down accordingly. The second counter is used to enable the ring oscillator when thermal sensing is needed. The data from the two counters are applied to the thermal management unit used for monitoring the chip.

2.2. Description of the GDS Technique

GDS is a thermal monitoring method for which the inverse heat transfer is solved problem [36]. It aims at detecting a single heat source on an integrated circuit surface based on the detection of isotherms. GDS was initially introduced in [37] to reduce the number of thermal sensors used to monitor the thermal profile of a heating surface.

The GDS technique is based on two thermal sensor cells separated by a distance H, as shown in Figure 2. In the case of a heat source located on the line connecting the two

sensors A and C, the temperature difference ΔT between the value measured by sensor A and that measured by the sensor C separated by a distance a from sensor A controls the values measured by these two sensors. In other words, the difference between their respective temperature measurements is proportional to the temperature difference only if the two sensors and the heat source are on the same line; otherwise, the value of the angle α must be taken into consideration for the appropriate calculation of T. For a heat source located at a distance r from sensor A, the temperature difference δT as in [36] is given by Equations (2) and (3), respectively.

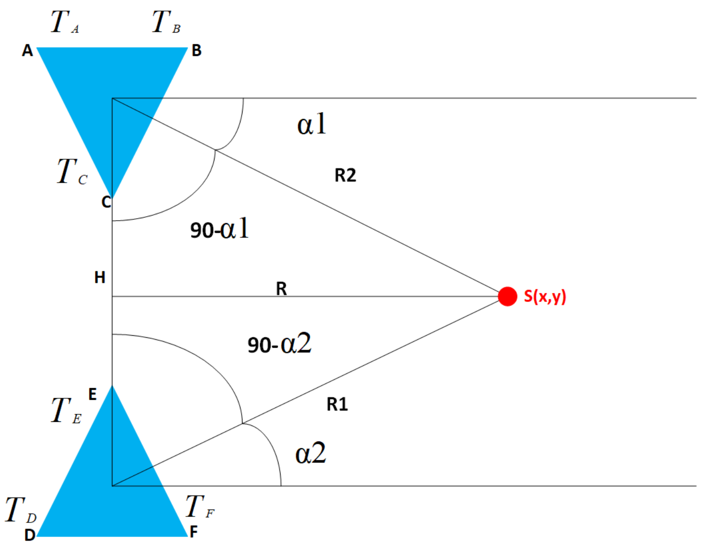


Figure 2. GDS configuration.

$$\frac{\Delta T_1}{R_1} = \frac{T_C - T_A}{\alpha_1 cos\alpha_1} \leftrightarrow \frac{V_C - V_A}{\alpha_1 cos\alpha_1} \tag{2}$$

$$\frac{\Delta T_2}{R_2} = \frac{T_D - T_F}{\alpha_2 cos\alpha_2} \leftrightarrow \frac{V_E - V_D}{\alpha_2 cos\alpha_2} \tag{3}$$

where α_1, α_2 is calculated by

$$tan\alpha_1 = \frac{2}{\sqrt{3}} \frac{V_B - V_A}{V_C - V_A} - \frac{1}{2} \tag{4}$$

$$tan\alpha_2 = \frac{2}{\sqrt{3}} \frac{V_F - V_D}{V_E - V_D} - \frac{1}{2} \tag{5}$$

The above equation allows us to deduce the value of the angle α_1, α_2. Once the position of the heat source is known, the next step is to calculate its temperature value.

Figure 2 shows two thermal sensor cells separated by a distance H. The first cell includes the three sensors A, B, and C, which form the angle α_1 with the line connecting the heat source to the cell, and the second cell includes the sensors E, F, and D, which form the angle α_2. Using the second cell, we can calculate the temperature value of the source by calculating the distance between the source and the sensor cell. The heat source and the cells form a triangle in which the length of one side and the values of the angles between the heat source and the cells are known. This means that we can calculate the distances between the heat source and the sensors. Now we can calculate the temperature gradient along a known distance as given by [37] and shown in Equation (6). For more details about GDS, please refer to [34,35,37].

$$T_s \leftrightarrow V_s = \frac{H}{R}(V_{C1} - V_{A1}) \\ \frac{(tan^2\alpha_1 + 1)(\sqrt{3} + tan\alpha_2)}{\sqrt{3}(1 - tan\alpha_1 tan\alpha_2) - (tan\alpha_1 + tan\alpha_2)} + V_{A1} \tag{6}$$

The GDS method is based on the detection of isotherms using two cells, and it has been successfully applied in [34,35] to detect a single heat source in the IC. From a theoretical point of view, the authors do not discuss the influence of parameters H and R on the accuracy of the method. Also, another important question to ask is whether the two parameters H and R are a function of the T_s value. One of the contributions of this article is to study the influence of parameters H and R on the precision of thermal peak detection.

3. On-Chip Temperature Monitoring and Improved Sensor Allocation Scheme

Accuracy is important for the thermal monitoring of microelectronic devices, as overestimation of temperature negatively impacts performance and an excessively high temperature can drastically shorten the chip mean time to failure MTTF.

The simplest approach to accurately perform thermal monitoring is to embed a large number of thermal sensors in the monitored device, which increases the implementation costs. Another straightforward technique is to divide the chip into small grids and monitor each grid with thermal sensors, then the maximum estimated temperature over the whole grid is obtained.

To detect the overheating spots, we propose a new scheme to track the thermal gradient that allows for reducing the number of thermal sensors used. As shown in Figure 3, the thermal peaks are detected in two steps. First, the TGT algorithm is applied to locate the location of the thermal zones by following the thermal gradient. In the second step, GDS is applied to precisely determine the location and temperature of the heat source. The thermal parameters of the GDS are adjusted by an automatic search algorithm according to the accepted error. Further details are given in the rest of this section.

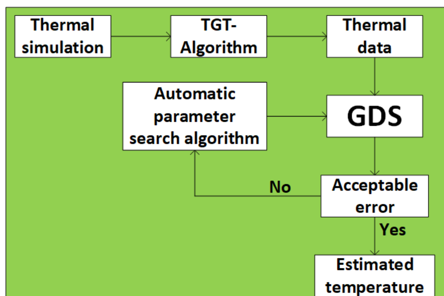


Figure 3. Proposed framework for thermal peak detection.

3.1. Thermal Gradient Tracker Scheme

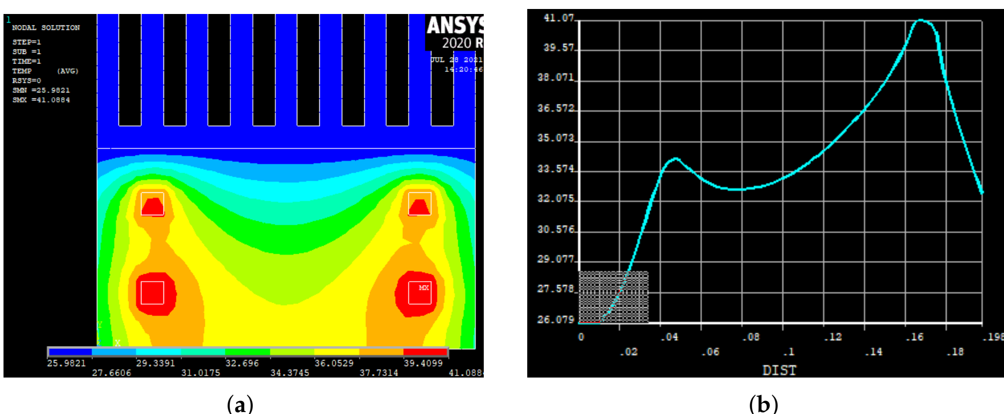
It is worth noting that the heat sources' location and intensity change as the workload and task mapping change. Our first goal is to systematically identify the potential location of heat sources.

To solve the multiple heat source detection problem, the first step is to model and simulate the spatiotemporal thermal distribution by the finite element method using the Ansys Multiphysics tool [38].

In this work, we chose to model and simulate the thermal behavior by FEM for its generality; i.e., it can accurately simulate the thermal behavior of complex structures.

To do so, we first built the geometry of a target IC with Ansys Multiphysics. Then, we injected four heat sources as shown in Figure 4a. The dimensions of the IC and its shape were taken from [15]. Figure 4a also shows an integrated circuit in which four heat sources continuously spread heat throughout the chip area. Figure 4b shows the spatial distribution of the temperature between two heat sources.

Note from this figure that the temperature is high close to the first source and it reduces as the distance grows. Somewhat before the midpoint between the two heat sources, the temperature starts to increase again and keeps increasing until reaching the location of the second heat source, and it then begins to decrease again.

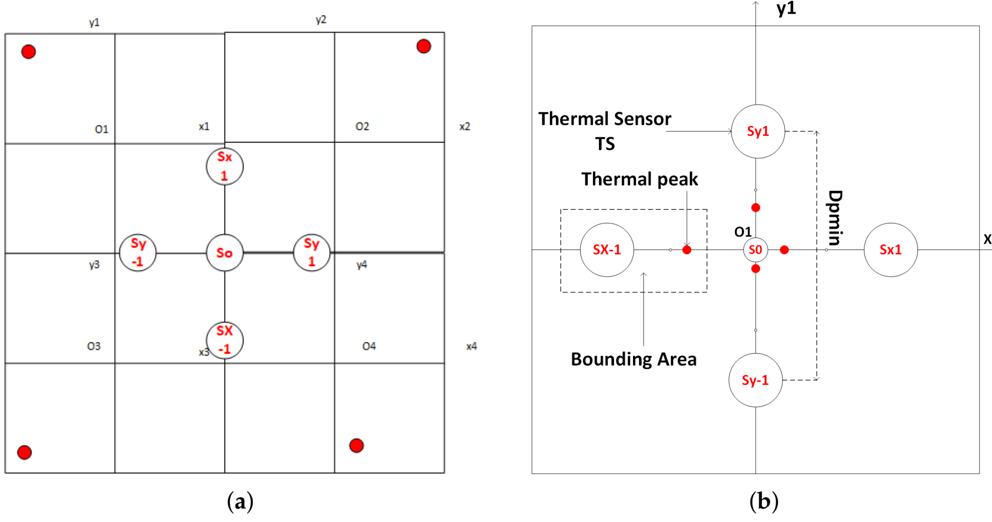


(**a**) (**b**)

Figure 4. Simulation of heat sources in a modeled IC. (**a**) Transient thermal cartography. (**b**) Spatial distribution of the temperature between two sources.

The proposed algorithm is based on the fact that the points near a heat source are hotter than those far from the source. Also, the algorithm aims at eliminating from consideration those areas that do not represent any increasing thermal gradient.

In our experimental study, we limit ourselves to an IC with four heat sources, without loss of generality. The first step is to divide the area of the IC into four grids, each of which is likely to include an intense heat source, and each plane is marked by a Cartesian plane whose origin is at the center of the grid, as shown in Figure 5a.



(**a**) (**b**)

Figure 5. Sensor placement method. (**a**) IC grid. (**b**) Illustration of temperature measurement modes.

After dividing the areas of the IC, the next step is to take thermal measurements at the center of each grid: on its right and on its left. The temperature measurements are made on the x-axis and on the y-axis. The temperature measurements are taken from five points, as shown in Figure 5b. The temperature readings on the selected points determine the direction of the next scan and the removal of areas that do not need to be thermally scanned. Therefore, the area that probably contains the thermal peak can be identified. We concentrate on our explanation of the first plane, but we can generalize it for the other planes.

To determine the next zone targeted by the thermal scan, we compare the values read on the sensors S_0, S_{x1}, S_{x-1}, S_{y1}, S_{y-1}. When heat diffuses from a single heat source, the points closer to the source are hotter. In fact, with five temperature readings, we can distinguish 32 possibilities. However, there are two distinct cases on which we build our thermal peak detection algorithm.

The two cases are when the temperature of a sensor S_0 is the highest compared to the temperatures of the other sensors S_{x-1}, S_{x1}, S_{y-1}, S_{y1}, and in the second case, the temperature read on S_0 is lower than the temperature of other sensors. When S_0 is greater, the likelihood of a thermal peak in this zone is very high. Thus, in this case, it is assumed that the thermal peak is inside the bounding area, bounded by S_{x-1}, S_{x1}, S_{y-1}, S_{y1}. In Figure 6a. We illustrate the principle of the first case. The bounding area that supposedly has a thermal peak is colored in red, and the remaining zone is assumed not to contain the hottest point and is colored in blue.

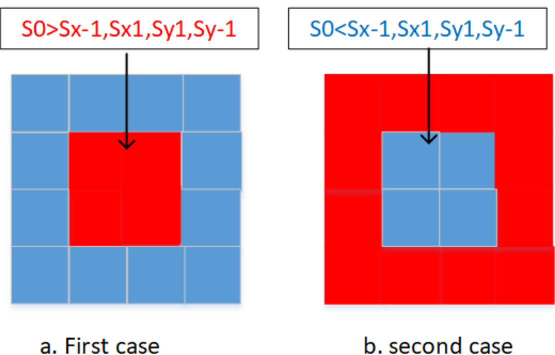


Figure 6. Limiting the thermal peak area.

The second case is illustrated in Figure 6b. The blue area represents the area deemed unnecessary to scan for hot spot detection; however, the red area must be thermally scanned. The red zone shown in Figure 6b includes the following possibilities: $S_{x1} > S_0 > S_{x-1}$, $S_{x1} < S_0 < S_{x-1}$, $S_{y1} < S_0 < S_{x1}$, and finally $S_{y1} > S_0 > S_{y-1}$.

In fact, the first case is used to surround the thermal peak area. In this case, we can distinguish four possibilities, as shown in Figure 7a. In all these possibilities, the temperature source is bounded by the first part of the segments, connecting point S_0, and the other four points S_{x-1}, S_{x1}, S_{y-1}, S_{y1}.

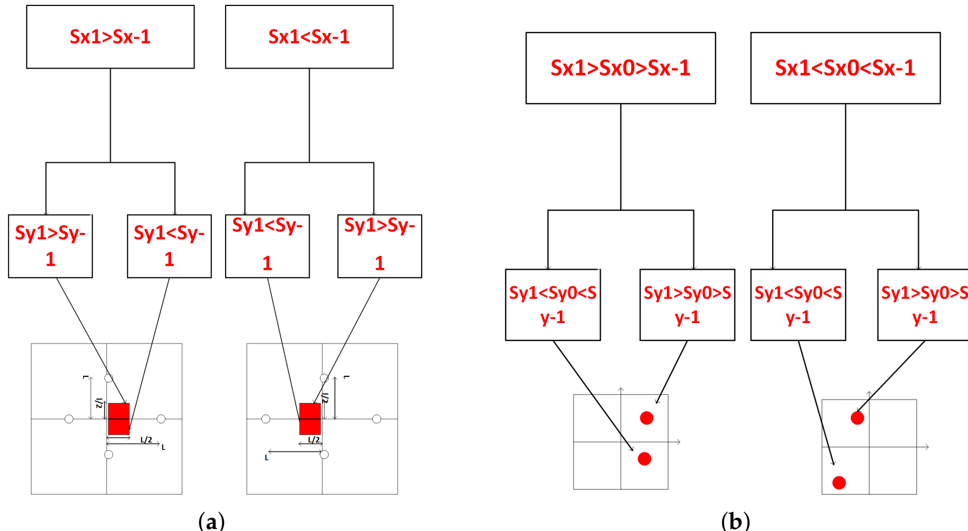


Figure 7. Thermal gradient tracker. (**a**) Location of thermal peak area. (**b**) Scan area selection procedure.

Therefore, the area containing the heat source can be determined by reading the other points, as can be seen in Figure 7a. Now, we consider the second case. According to the temperature of the measured points, we decide on the next zone that is subject to the next temperature measurement, as shown in Figure 7b. Once a zone is selected, the next step is to calculate its geometric center to take the temperatures of the center and its neighbors according to the diagram shown in Figure 7.

Assume that the area A colored in red shown in Figure 8a is selected for the next thermal scan, i.e., $S_{x1} > S_0 > S_{x-1}$, and $S_{y1} > S_0 > S_{y-1}$. The next step is to find the middle of this zone, and the goal of taking the thermal measurements in the middle of zone A is to decrease the number of sensors needed to find the thermal peaks. In other words, we only scan areas that potentially contain a thermal peak to facilitate thermal peak point identification.

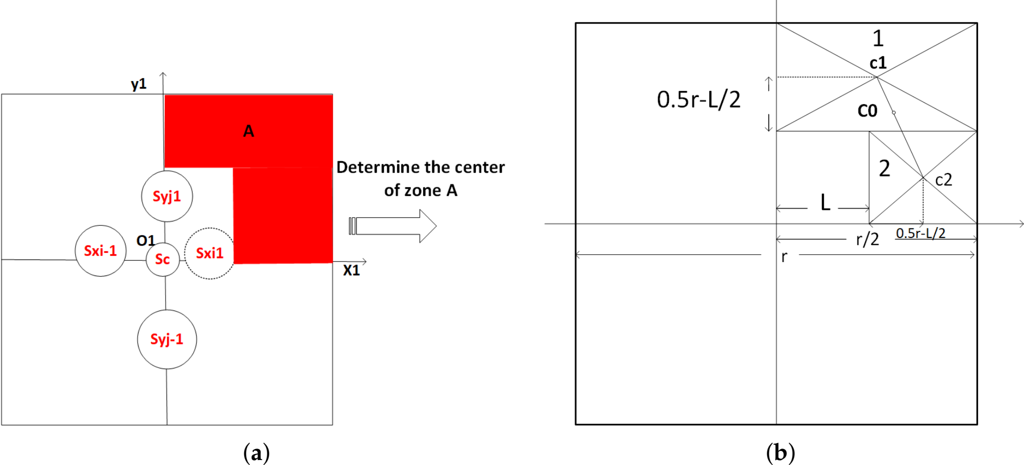


Figure 8. Computing of the peak zone geometry. (**a**) Selection of the next area to be scanned. (**b**) Calculation of the geometric center of the area to be swept.

Figure 8b shows the center C_0 of the zone A. Area A consists of the two rectangles 1 and 2 as shown in Figure 8b, and it is L-shaped. Let C_1 and a_1 be the center and the area of rectangle 1, respectively, and C_2 and a_2 the center of rectangle 2, respectively. The center of zone A is therefore the barycenter of (C_1, a_1) (C_2, a_2), which is located between C_1 and C_2 and that verifies Equation (7):

$$C_1C_0 = \frac{a_2}{a_1 + a_2} \cdot C_1C_2 \tag{7}$$

3.1.1. Stopping Criteria

The aim of the proposed algorithm is to estimate the thermal peaks as precisely as possible using a reduced number of sensors; the stopping criteria depend on these constraints. The first stopping criterion is to fall in the first case shown in Figure 6a. In the second stopping criterion, we are satisfied with determining a zone smaller than the one that can be experimentally determined. In fact, the value of the area at which we stop our algorithm depends on the following factors: (1) precision, (2) the distance between the sensor and the source, (3) the accuracy of the sensors used to make thermal measurements, and (4) the technique used to estimate the temperature of the source—GDS or a conventional technique such as fuzzy-based clustering [30].

3.1.2. Corrective Scheme

To facilitate the development of our algorithm, we built it on the assumption that the heat source is outside the rectangle formed by the points on which we place the temperature sensors if the measured temperature of the points S_{x1}, S_{x-1}, S_{y1}, S_{y-1} exceeds the measured temperature at point S_0; otherwise, it is inside the rectangle formed by these points if the measured temperature of point S_0 is greater than that at the other points.

Based on these assumptions, a correction algorithm is applied. In the scenario shown in Figure 9, the heat source is at the second segment midpoint that connects point S_0 and one of the four points S_{x1}, S_{x-1}, S_{y1}, or S_{y-1}. In each of the four scenarios shown in Figure 9, when one of the temperatures at points S_{x1}, S_{x-1}, S_{y1}, S_{y-1} is greater than the temperature at point S_0, the TGT algorithm assumes that the heat source is not in the currently scanned area, and it continues the thermal scan incorrectly while missing the true peak.

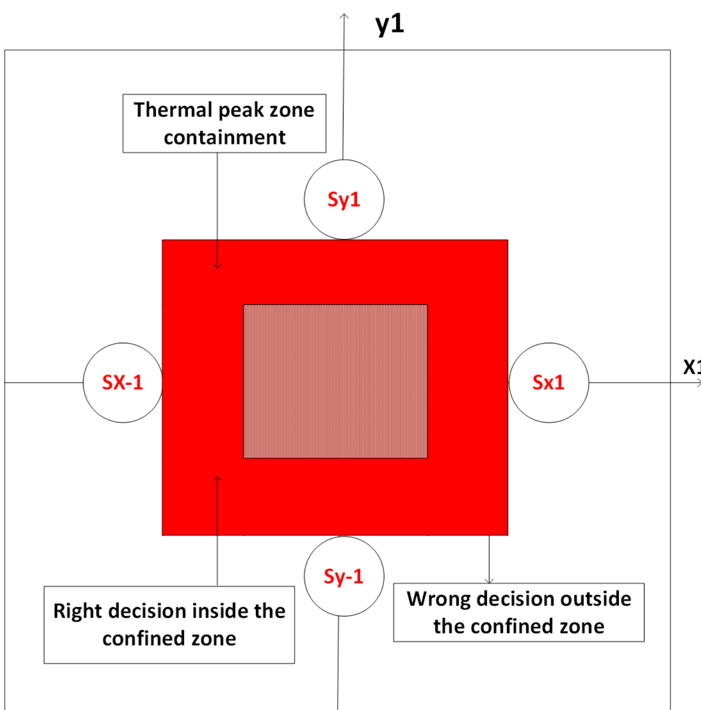


Figure 9. Illustration of a wrong decision.

Based on this erroneous decision, the algorithm decides that the next area to be scanned is outside the rectangle; then, the thermal peak estimate is incorrect and the estimated error is very high. To avoid this scenario, a decision for correction is made.

When making a decision on the area to be scanned in the near future and determining the center of this area, the maximum temperature measured in this area is compared to that of the previously scanned area, and if this temperature is higher, correction is mandatory. Hence, the algorithm goes back to the previous zone and determines the bounding area by comparing the temperatures between the sensors S_{x1}, S_{x-1}, S_{y1}, S_{y-1}.

The pseudo-code of the thermal zone detection algorithm is given in Figure 10. This algorithm does not determine the real location of thermal peaks and their temperatures; it only tracks the thermal gradient and gives the highest temperature points close to the heat sources. This can lead to significant errors. In fact, it is not easy to optimize the distance between the location of the sensors and the thermal peak [15].

1-Int: r (chip dimension), n=4 (number of peaks),
2- Recursively divide the chip into four grids.
3- Measure the temperature of the center and its neighbors for each grid.
4- **If** case one return the position and value of peaks.
5- **Else if not stop creteria**
Decide the next thermal scan zone and compute its center.
6-**Else** return the peak's position and its value.
7-**End.**

Figure 10. Pseudo-code of the proposed TGT algorithm.

To solve this problem and to improve the accuracy, we propose using a hybrid technique based on the TGT algorithm, to locate the area where peak temperatures are expected, and the GDS, to interpolate the reading of sensors and to accurately estimate the temperature of the heat source and its location. This will be explained in the next section.

3.2. Thermal Gradient Tracker Scheme Based-GDS

As previously explained, the GDS technique may not be directly applicable, and it is necessary to quantify the relationship between the two parameters H and R on one side and the position and the source temperature on the other side. To do so, we first propose an automatic search algorithm based on simulated annealing in order to find the best position of the sensors with respect to the heat source on one side and on the other side to know if the position of the sensors changes with the change of the value of the source. Once the automatic search algorithm finds the best parameters (H, R), the thermal sensors remain fixed, and the temperature of the source is interpolated using the GDS technique.

3.2.1. Automatic Parameter Search Algorithm

In order to facilitate the study of the GDS technique before implementing it on an FPGA board for runtime thermal monitoring, we propose an automatic search algorithm to find the best pair (H, R) in which the temperature can be precisely estimated from the source.

The search for pair (H, R) is formulated here as a combinatorial optimization problem. The optimization problem consists in minimizing, as much as possible, the cost function (the energy of the system), defined as the quadratic error between the source temperature and the temperature estimated by the GDS technique.

In addition to finding the best pair that can minimize the cost function, we are interested in keeping R as high as possible. The greater the value of R, the greater the thermal coverage area, which means fewer temperature sensors are needed for real-time accurate monitoring of thermal profiles in integrated circuits.

The objective function or the cost function is given as follows in Equation (8):

$$F_o = (T_{GDS}(H, R) - T(s)_D)^2 subjectto \frac{R[i]}{R[i-1]} - 1 > 0 \quad (8)$$

where $T_{GDS}(H, R)$ is the temperature estimated by the GDS technique for each pair, $T(s)_D$ is the desired temperature that is obtained by simulation, and finally F is the error between the desired and estimated temperatures. The condition on R is to ensure a maximum thermal coverage area that can be accurately scanned by GDS. The pseudo-code of this procedure is shown in Figure 11.

```
1-Choice of an initial solution  Fo(H0,R0)
 an initial temperature, cooling constant, and the final temperature t-f
2-Repeat steps 3 to 14 until t <t_f
3-Repeat steps 4 to 17 r times
4-Generate a random number p, 0 <p <1
5- Calculate the new cost function Fo for a new configuration H, R.
6-Evaluate the energy difference between the current cost function and the previous
one.
7-If   exp(-ΔE/t)≥P   then go to 9
8- If  exp^(-ΔE/t) ≤ P  then go to 2
9-Accept the new configuration.
10-If 6 is violated go to 3.
11-If E_min <the current energy goes to step 2.
12-If 6 is not violated: E_min ← the current energy and keep the current
configuration (H, R).
13-If E_min is the desired error go to 15.
14- t ← t × A.
15- Get best configuration, E_min and end the program
```

Figure 11. Pseudo-code to find the parameters that best estimate the peak.

3.2.2. Hybrid Algorithm TGT Based-GDS

The TGT thermal sweep algorithm proposed in Section 2 allows thermal sweeps to be performed only in areas probably containing thermal peaks. It was concluded that the accuracy of estimation and localization of thermal peaks by this algorithm can be improved by increasing the cost in terms of the number of temperature sensors used. Thus, the

precision depends on key parameters of the algorithm, such as the distance between the temperature sensors and the criterion for stopping the algorithm.

The accuracy of the method also depends on that of the sensors used. According to the experimental results produced by the GDS technique, the minimum distance between the sensors and the stopping criterion of the TGT algorithm can be adjusted while increasing the minimum distance between the sensors and the stop zone without unduly degrading the accuracy of peak temperature estimation results and hot spot location. Consequently, we can achieve our objective of precisely detecting thermal peaks using a reduced number of temperature sensors through the application of a combination of GDS and TGT.

In the combined TGT-GDS method, based on the TGT method, the areas where the GDS technique needs to be applied are determined based on thresholds that can be experimentally obtained. The results obtained by the GDS technique show that the minimum area that can be covered is larger than that imposed by TGT for the same precision. Therefore, the minimum coverage area is the first criterion for stopping the proposed TGT-GDS technique. In fact, the second criterion for stopping the TGT-GDS technique depends on the temperature and the remaining area of the IC to be swept.

The problem with this second stopping criterion is that it depends on the temperature of the remaining area that we do not know. An empirical temperature estimation method was reported in [39]. The model is not sufficiently accurate [20], but it can be taken as an indication of the temperature of the remaining area. A model adapted to our situation and inspired by [39] is described as follows in Equation (9):

$$T_{rem,sup} = \frac{T_{max,TGT}}{1 - e^{\left(-\frac{2R}{K}\right)}} \tag{9}$$

where $T_{rem,sup}$ is the estimated temperature of the remaining area, $T_{max,TGT}$ is the maximum temperature read on one of the sensors of the TGT algorithm, R is the length of the remaining area for thermal scanning, and finally K is a thermal constant, which depends on the thickness of the materials constituting the circuit to be monitored.

The flow diagram of the combined TGT-GDS algorithm is shown in Figure 12. The first and second cases correspond to the TGT algorithm. In the first case, the peak is inside the bounding area. In the second case, the peak is outside the bounding area. Thus, recursive scanning from the center is continued until we obtain an area and a temperature, estimated by Equation (7), suitable for applying the GDS technique. D_{min} is the parameter that determines the minimum distance that GDS can cover.

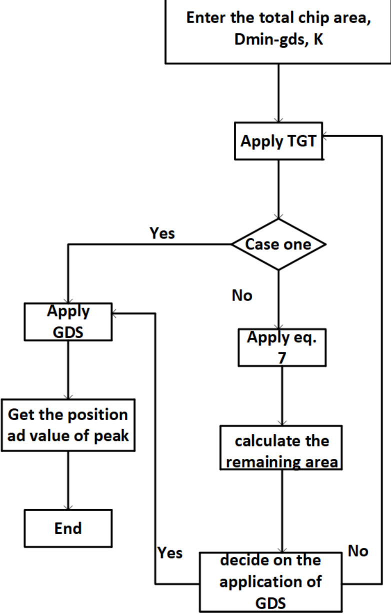


Figure 12. Flowchart of the proposed TGT-GDS algorithm.

4. Experimental Results and Validation

To test the efficiency of our algorithm in detecting the zones and temperatures of thermal peaks, we carried out a series of experiments, where the following parameters are variables. (1) The location of thermal peaks. (2) The temperature of thermal peaks. (3) The distance between the thermal peaks: We move away from the peaks and approach them for different peak temperatures. (4) The distance D_{pmin} between the points on which the sensors are placed. (5) The final area at which the division is made to stop (second stopping criterion).

To demonstrate the efficiency of our proposed method in this case study, we use a 2 ms time resolution power load to generate thermal data across a wide range of mission profiles in Ansys. All algorithms are implemented with MATLAB and run on an Intel(R) Core 3.9 GHz computer. Four sensor allocation methods are implemented for comparison reasons: grid [22], K-means [23], energy center [24], and the proposed method.

Figure 13 shows the average run time of these methods for various numbers of thermal sensors $TS > 25$. It is clear that the K-means, energy center, and grid-based methods require a longer run time, and this becomes even longer as the number of thermal sensors grows. Note that K-means is considerably more worse than the others. Note also that the proposed algorithm requires a short time to identify the most significant sensors and that the run time only increases slowly with the number of thermal sensors used. Compared with the grid-based method, the proposed method can significantly reduce the run time by up to 78 percent, which helps to further mitigate the threat of thermal peak evolution. The amount of time required to allocate sensors is crucial and depends on the requirements of dynamic thermal management systems, and for high performance [21,40,41], the amount of time is generally in order of hundreds of milliseconds. Therefore, the proposed algorithms ensure the efficiency of peak detection.

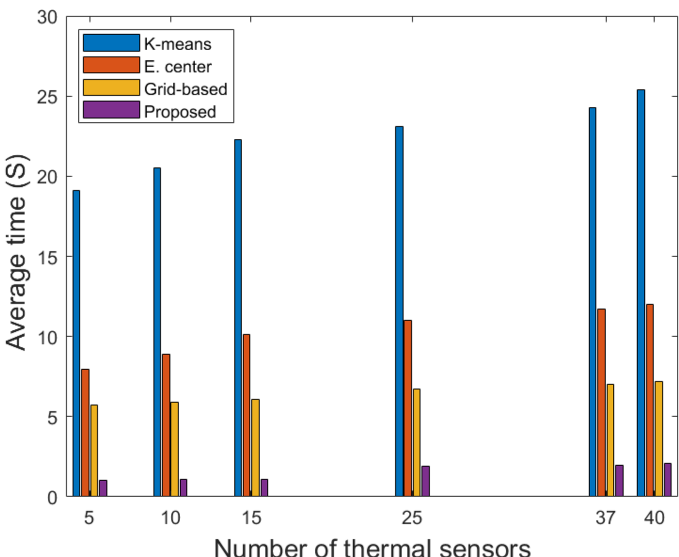


Figure 13. Comparison of average run time for different numbers of sensors.

In the first experiment reported below, we placed four heat sources separated by the same distance from each other, but each has its own temperature value. In this experiment, we varied the distance D_{pmin} between the points to be measured, and the area of the stopping zone was fixed at a value considered to be relatively small, namely 0.00006 mm^2. Table 1 shows the obtained results. For each zone, we give the simulated peak obtained by Ansys, the peak temperature estimated by our algorithm, and the number of sensors (NBS). The error here is defined as the difference between the simulated and the estimated peak, and its range in the case of zone 1, 2, 3, and 4 is approximately constant regardless of the peak temperature, and it depends only on D_{pmin} and the number of sensors. On the other hand, the thermal peak detection errors are relatively high.

Table 1. Relationship between D_{pmin} and detection accuracy.

		D_{pmin} (mm)			
		0.0015	**0.0010**	**0.0008**	**0.0007**
Zone 1	Simulated peak	50.5	68.3	61.3	49.8
	Estimated peak	47.4	66.6	58.7	48.4
	NBS	15	20	20	30
Zone 2	Simulated peak	67.22	72.1	71.9	66
	Estimated peak	65.89	69.5	68.4	63.9
	NBS	10	15	20	25
Zone 3	Simulated peak	65.4	54.5	68.8	70.5
	Estimated peak	62.9	52.7	67.2	69.1
	NBS	10	20	30	30
Zone 4	Simulated peak	70	55.9	70.3	59.4
	Estimated peak	68.6	54.5	68	57.1
	NBS	15	25	25	30

In this experiment, D_{pmin} has a negative impact on the estimation accuracy because the proposed algorithm assumes that the peak is in the middle of the hottest zone. In fact, this explains the large errors in detecting thermal peaks in some cases. Clearly, a larger distance between the measured points and the actual hot spot position increases the errors in the estimated peak temperature values.

In the second experiment, we change the area of the stopping criterion while keeping the distance D_{pmin} quite small. In this experiment, we want to determine the influence of the second stopping criterion on the performance of the TGT algorithm. To assess the impact of the second stopping criterion on the TGT algorithm, a set of simulations was conducted in which the area is gradually reduced. Table 2 shows the results obtained for different stopping criterion values. For a value of the stopping criterion equal to 0.005 mm^2, the error is noticeably large across all simulations, with a maximum error of 5 °C reported in zone 3. In addition, a reduction in estimation errors is observed when we reduce the area until we reach an error of 0.3 °C when the target area is pretty small, namely 0.00006 mm^2, when the number of thermal sensors is relatively high, 25 in the present case.

Table 2. Impact of stopping criteria on detection accuracy.

		Stopping Area (mm)^2			
		0.005	**0.002**	**0.00010**	**0.00006**
Zone 1	Simulated peak,°C	54.5	65.1	59.7	45.8
	Estimated peak,°C	50.2	62.3	57	44.3
	NBS	10	10	15	20
	Error, °C	4.3	2.8	2.7	1.5
Zone 2	Simulated peak,°C	65.6	77.4	78.3	57.6
	Estimated peak,°C	60.89	74	75.1	55.4
	NBS	10	15	3 15	20
	Error, °C	4.71	3.4	3.2	2.2
Zone 3	Simulated peak,°C	70.5	58.6	77	77.5
	Estimated peak,°C	65.5	55.8	77.2	76.2
	NBS	10	15	20	25
	Error, °C	5	2.8	0.8	0.3
Zone 4	Simulated peak,°C	72.8	60.7	73.9	69.3
	Estimated peak,°C	69.1	58.6	72.1	67.9
	NBS	15	15	20	20
	Error, °C	3.7	2.1	1.8	1.4

As the results reported in Table 2 confirm, the narrower the stopping zone, the less difference there is between the simulated peak and the estimated peak; therefore, the accuracy improves as the stopping zone decreases in size, but the error also decreases with the number of instantiated sensors. Finally, we conclude that the TGT algorithm is always closer to the thermal peak, and the accuracy of the aforementioned algorithm is area-dependent and requires a relatively large number of thermal sensors.

According to the results presented in the above tables, the detection error is high in some cases and low in other cases when the source is in the middle of the zone or close to the center with the smallest possible stopping surface. In fact, to solve this problem, the GDS technique is applied to interpolate the temperature of the heat source. This is the motivation for the third experiment, in which the influence of the GDS parameters H and R on estimation accuracy is studied.

To analyze the effects of the GDS parameters, a series of finite element thermal simulations were carried out using Ansys software. In the first simulation, we set the temperature source between 40 and 115 °C, and we changed the location of the center of two cells ABC and DEF relative to the location of the heat source; i.e., we changed the value of R. In this simulation, the distance H between the two cell centers is fixed to a value that we have previously determined. For each R, we apply the GDS technique to estimate the heat source temperature and position. The chosen R values and the temperature source estimation results of this first simulation are shown in Table 3.

Table 3. Influence of R parameter on GDS accuracy.

Heat Source °C	H µm	R µm	Estimation Error °C
45.5	200	210	0.0044
45.5	200	215	0.0039
73.3	200	240	0.00375
73.3	200	250	0.003358
85.4	200	271	0.00296
85.4	200	277	0.00314
90.5	200	283	0.00322
90.5	200	286	0.0033
106.3	200	292	0.00425
90.5	200	297	0.0043

This table reports estimates of each thermal peak for two different values of R and the corresponding error for each estimate. We observe a reduction in the error of the estimates when we adjust the value of R. For example, the estimation error of the thermal peak thermal is 0.0044 when the temperature value is 45.5 °C with R = 210 µm, and this error keeps decreasing as R reaches 215 µm. It is of interest that we achieve a smaller error while increasing R in the case where the temperature value is 85.4 °C It can be seen that in Table 3, the estimation precision changes according to R. It can also be noted that with equal precision, the higher the temperature of the source, the greater the R. It is therefore important when developing a temperature monitoring system based on GDS to consider the R parameter.

Now we take the same data from the previous simulation and change the parameter H. The parameter R is fixed at a value taken from Table 3, and the value chosen for R is that which resulted in the best estimate of temperature in the previous simulation. The obtained results are shown in Table 4. We remark that when H < 260 µm, the error gradually decreases from 0.0064 to 0.00259 as the length H increases, and it increases thereafter. Compared to the earlier results of Table 3, the precision is higher, and the error was reduced to 0.00259 in the best case. This is expected, since we have chosen the value of R that results in a better estimate, and we have adjusted the parameter H.

The parameter H also plays a determining role in the precision of estimation. Indeed, we achieved a better estimation than that noted in the previous simulation in Table 3 by changing the parameter H. According to the results presented in Tables 3 and 4, the couple (H, R) are critical points in developing the GDS technique, and they play a determining role in the accuracy of estimating the temperature and position of the heat source.

Table 4. Influence of H parameter on the GDS accuracy.

Heat Source °C	R μm	H μm	Error of Estimation °C
45.5	240	180	0.0064
45.5	240	184	0.0067
73.3	240	233	0.0036
73.3	240	241	0.0038
85.4	240	260	0.00259
85.4	240	269	0.00298
90.5	240	277	0.003
90.5	240	282	0.0031
106.3	240	300	0.00401
106.3	240	306	0.00411

In the fourth experiment, we applied our automatic search algorithm using different source temperatures. The objective is to quantify the relation between the parameters (H, R) and the heat source as well as to help the hybrid algorithm to adjust its stopping criteria according to the situation. The obtained results are shown in Table 5. We note that the estimation error is not only a function of the number of sensors but also depends on the stopping criteria defined by (H, R). Indeed, the error changes with the GDS parameters for the same number of thermal sensors used. For example, to estimate the same heat source in the first two lines of Table 5, the error changes from 0.00022 for (H, R) = $(140, 420)$ to 0.000207 for (H, R) = $(145, 470)$.

Table 5. Results obtained using the automatic search algorithm.

Heat Source °C	Stopping Area (mm)^2	(H, R) μm	Error of Estimation °C	NBS
116.9	256.25	(145, 470)	0.000207	8
116.9	240.6	(140, 420)	0.00022	8
80.3	169	(161, 313)	0.000223	12
85.8	198	(163, 394)	0.000227	8
103.7	289	(192, 580)	0.00033	5
110.6	292.49	(197, 610)	0.00061	5
112.5	259.16	(135, 450)	0.0042	8
115.4	263.45	(137, 459)	0.0044	8

All results shown in Table 5 are of great importance. On the one hand, these results show that R increases with the increase in the source temperature, as in the case of estimating the peak temperature values of 80.3 and 85.8 with couplet (H, R) = (161, 313) and (163, 394), respectively; and they show us that R can be adjusted according to the estimation error. It is noted that for a fairly small error difference, a relatively large thermal coverage area can be obtained. On the other hand, these results show that for the temperatures of two close sources, the parameter (H, R) can be the same as shown in Table 5. Across all the results shown in this table, the reduction in the number of thermal sensors that must be used is remarkable, as R and H increase as well as the stopping area, since all these parameters are related, and the greater the (H, R) the greater the coverage area and fewer sensors that are needed.

Instead of using (H, R) = (140, 420), the hybrid technique may use (H, R) = (145, 470) in order to improve the accuracy of thermal peak detection while reducing the number of sensors and the time needed to identify thermal peaks, since it has the ability to select a (H, R) tuple that results in better accuracy with a large thermal coverage area. As the second stop criterion must be sufficiently reduced for the TGT algorithm to give acceptable results, the accuracy of the hybrid algorithm depends on the second stopping criterion, which depends on R and H. Larger (H, R) values increase the thermal coverage area, which reduces the number of sensors needed for thermal monitoring. Table 6 shows the detection results for the hybrid algorithm, which shows improvement over the TGT algorithm.

Table 6. The advantage of the hybrid algorithm compared to the *TGT* algorithm.

Sim Peak °C	est. TGT °C	NBS #	TGT.ac %	est. TGT-GDS °C	NBS #	TGT-GDS.ac %
80.23	75.18	25	93.69	79.35	8	98.89
90.25	87.9	30	97.39	90.05	12	99.77
75.83	73.49	25	96.91	75.54	12	99.61
62	57.1	20	92.09	61.2	8	98.71

Using the hybrid algorithm together with the results obtained by the automatic parameter search algorithm (H, R), better precision can be achieved with a reduced number of sensors compared to the TGT algorithm. For the efficiency of our algorithm, a full comparison in terms of the number of sensors used for peak detection and the average error in hot spot estimation with the results provided in [24,42] is given in Figure 14.

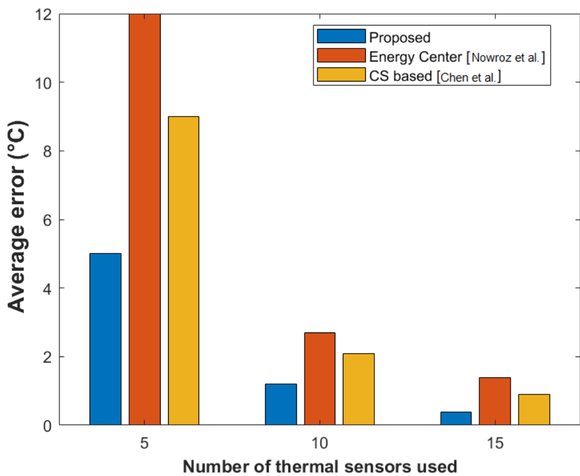


Figure 14. Comparison of the average error in hot spot detection [24,42].

The CS-based method [42] and energy clustering method [24] both use the frequency characterization of the heat distribution as an index to allocate the thermal sensors. The spectral representation is a non-limited band, and filtering can lead to a lack of necessary information, which in turn results in an overestimation of temperature. Unlike these two methods, our proposed hybrid method, TGT-based GDS, intelligently scans for potential thermal spike locations and automatically adjusts the (H, R) parameters to find the best position of the thermal sensors for better estimation of spike value and placement, which can reduce the number of sensors needed for better estimation, as noted in Figure 14.

To verify that the proposed algorithms meet practical requirements and to confirm their validity, we simulate different thermal profiles each containing a different number of thermal peaks, placed at different distances from each other. The results of this simulation are shown in Table 7. Table 7 shows the best and worst peak detection accuracy and the number of thermal peaks successfully detected. As reported in this table, the algorithm can accurately detect up to 18 act spikes in the chip. For more than 18 peaks, we note that the algorithm fails to detect some peaks, which is due to the phenomenon of heat fusion, as some peaks join together and form a new peak.

Table 7. The validity of the proposed algorithm in detecting different numbers of peaks.

# peak	1	3	6	10	18	20	24
# detected peak	1	3	6	10	18	17	19
Best accuracy %	99.99	99.97	99.87	99.75	98.64	97.97	95.14
Worst accuracy %	NA	99.96	99.84	99.31	98.01	96.22	93.53

5. Conclusions

In this paper, two techniques were presented to obtain accurate runtime temperature sensing with a reduced number of thermal sensors and a reduced time to detect thermal peaks. The first technique, called TGT, can determine the areas of thermal peaks and estimate their positions and values in a short time by scanning from the middle of the potential zones and eliminating all those without thermal peaks. A hybrid method that combines GDS with TGT was proposed. It was shown that it can improve the precision of thermal peak detection and reduce the number of thermal sensors required to do so. Thanks to the proposed automatic search algorithm, the configuration of the GDS parameters can easily be set up, and it was shown that they play an important role in determining the accuracy of thermal peak estimated values. The simulation results show that the proposed TGT-based GDS thermal sensor allocation algorithm leads to more accurate peak detection.

Author Contributions: Conceptualization, D.E.T.; Writing—original draft, D.E.T.; Writing—review & editing, D.E.T.; Supervision, A.O., A.H., M.A., A.L. and Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Huang, X.; Sukharev, V.; Kim, T.; Tan, S.X.D. Dynamic electromigration modeling for transient stress evolution and recovery under time-dependent current and temperature stressing. *Integration* **2017**, *58*, 518–527. [CrossRef]
2. Oukaira, A.; Hassan, A.; Savaria, Y.; Lakhssassi, A. Foster-based transient thermal analysis of SiP for thermomechanical studies. In Proceedings of the IEEE International New Circuits and Systems Conference (NEWCAS), Toulon, France, 13–16 June 2021; pp. 1–4.
3. Oukaira, A.; Touati, D.E.; Hassan, A.; Ali, M.; Savaria, Y.; Lakhssassi, A. Thermo-mechanical Analysis and Fatigue Life Prediction for Integrated Circuits (ICs). In Proceedings of the IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), East Lansing, MI, USA, 8–11 August 2021; pp. 630–634.
4. Meyer, B.H.; Hartman, A.S.; Thomas, D.E. Cost-effective lifetime and yield optimization for NoC-based MPSoCs. *ACM Trans. Des. Autom. Electron. Syst. (TODAES)* **2014**, *19*, 1–33. [CrossRef]
5. Mutapcic, A.; Boyd, S.; Murali, S.; Atienza, D.; De Micheli, G.; Gupta, R. Processor speed control with thermal constraints. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2008**, *56*, 1994–2008. [CrossRef]
6. Brooks, D.; Dick, R.P.; Joseph, R.; Shang, L. Power, thermal, and reliability modelling in nanometer-scale microprocessors. *IEEE Micro* **2007**, *27*, 49–62. [CrossRef]
7. Pedram, M.; Nazarian, S. Thermal modeling, analysis, and management in VLSI circuits: Principles and methods. *Proc. IEEE* **2006**, *94*, 1487–1501. [CrossRef]
8. Lu, S.; Tessier, R.; Burleson, W. Dynamic on-chip thermal sensor calibration using performance counters. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2014**, *33*, 1487–1501. [CrossRef]
9. Shi, B.; Zhang, Y.; Srivastava, A. Dynamic Thermal Management under Soft Thermal Constraints. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2012**, *21*, 2045–2054. [CrossRef]
10. Oukaira, A.; Touati, D.E.; Hassan, A.; Ali, M.; Savaria, Y.; Lakhssassi, A. FEM-based Thermal Profile Prediction for Thermal Management of System-on-Chips. In Proceedings of the IEEE International Conference on Optimization and Applications (ICOA), Genoa, Italy, 6–7 October 2022; pp. 1–4.
11. Oukaira, A.; Mellal, I.; Ettahri, O.; Kengne, E.; Lakhssassi, A. Thermal management and monitoring based on embedded ring oscillator network sensors for complex system design. *Int. J. Comput. Eng. Inf. Technol.* **2017**, *9*, 127–134.
12. Ahmed, M.; Fisher, N.; Wang, S.; Hettiarachchi, P. Minimizing peak temperature in embedded real-time systems via thermal-aware periodic resources. *Sustain. Comput. Inform. Syst.* **2011**, *1*, 226–240. [CrossRef]
13. Zhou, M.; Cheng, L.; Dell'Antonio, M.; Wang, X.; Bing, Z.; Nasseri, M.A.; Huang, K.; Knoll, A. Peak Temperature Minimization for Hard Real-Time Systems Using DVS and DPM. *J. Circuits Syst. Comput.* **2019**, *28*, 1950102. [CrossRef]
14. Lopez-Buedo, S.; Garrido, J.; Boemo, E.I. Dynamically inserting, operating, and eliminating thermal sensors of FPGA-based systems. *IEEE Trans. Components Packag. Technol.* **2002**, *25*, 561–566. [CrossRef]
15. Memik, S.O. *Heat Management in Integrated Circuits: On-Chip and System-Level Monitoring and Cooling*, 1st ed.; Institution of Engineering and Technology: New York, NY, USA, 2016; pp. 154–196.
16. Choobineh, L.; Jain, A. Analytical solution for steady-state and transient temperature fields in vertically stacked 3-D integrated circuits. *IEEE Trans. Components Packag. Manuf. Technol.* **2012**, *2*, 2031–2039. [CrossRef]

17. Li, P.; Dong, Y.; Tang, M.; Mao, J.; Jiang, L.J.; Bagcı, H. Transient Thermal Analysis of 3-D Integrated Circuits Packages by the DGTD Method. *IEEE Trans. Components Packag. Manuf. Technol.* **2017**, *7*, 862–871. [CrossRef]
18. Savidis, I.; Vaisb, B.; Friedman, E.G. Experimental analysis of thermal coupling in 3-D integrated circuits. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2014**, *23*, 2077–2089. [CrossRef]
19. Huang, W.; Ghosh, S.; Velusamy, S.; Sankaranarayanan, K.; Skadron, K.; Stan, M.R. A compact thermal modeling methodology for early-stage VLSI design. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2006**, *14*, 501–513. [CrossRef]
20. Sharifi, S.; Rosing, T.S. Accurate direct and indirect on-chip temperature sensing for efficient dynamic thermal management. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2010**, *29*, 1586–1599. [CrossRef]
21. Zhang, Y.; Srivastava, A. Accurate temperature estimation using noisy thermal sensors for Gaussian and non-Gaussian cases. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2010**, *19*, 1617–1626. [CrossRef]
22. Long, J.; Memik, S.O.; Memik, G.; Mukherjee, R. Thermal monitoring mechanisms for chip multiprocessors. *ACM Trans. Archit. Code Optim. (TACO)* **2008**, *5*, 1–33. [CrossRef]
23. Memik, S.O.; Mukherjee, R.; Ni, M.; Long, J. Optimizing thermal sensor allocation for microprocessors. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2008**, *27*, 516–527. [CrossRef]
24. Nowroz, A.N.; Cochran, R.; Reda, S. Thermal monitoring of real processors: Techniques for sensor allocation and full characterization. In Proceedings of the 47th Design Automation Conference, Anaheim, CA, USA, 13–18 June 2010; Volume 5, pp. 56–61.
25. Reda, S.; Dev, K.; Belouchrani, A. Blind identification of thermal models and power sources from thermal measurements. *IEEE Sens. J.* **2017**, *18*, 680–691. [CrossRef]
26. Cochran, R.; Nowroz, A.N.; Reda, S. Post-silicon power characterization using thermal infrared emissions. In Proceedings of the 16th ACM/IEEE International Symposium on Low Power Electronics and Design, Austin, TX, USA, 18–20 August 2010; Volume 5, pp. 331–336.
27. Nowroz, A.N.; Woods, G.; Reda, S. Power mapping of integrated circuits using AC-based thermography. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2012**, *21*, 1398–1409.
28. Paek, S.; Shin, W.; Sim, J.; Kim, L.S. A probabilistic approach for temperature-to-power conversion based on Markov random field theory. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2013**, *32*, 1509–1519. [CrossRef]
29. Reda, S.; Cochran, R.; Nowroz, A.N. Improved thermal tracking for processors using hard and soft sensor allocation techniques. *IEEE Trans. Comput.* **2011**, *60*, 841–851. [CrossRef]
30. Li, J.; Feng, S.; Zhang, Y.; Wang, C.; He, X. Optimized thermal sensor allocation for field-programmable gate array temperature measurements based on self-heating test. *Microelectron. J.* **2017**, *60*, 841–851 [CrossRef]
31. Wang, S.; Feng, S.; Xiao, Y.; Hu, C.; Pan, S. Build-in compact and efficient temperature sensor array on field programmable gate array. *Microelectron. J.* **2021**, *111*, 105–118 [CrossRef]
32. Yu, W.; Feng, S.; Zhang, Y.; Shi, B. Temperature distribution measurement based on field-programmable gate array embedded ring oscillators. *Solid-State Electron.* **2019**, *158*, 16–21
33. Rabah, H.; Amira, A.; Mohanty, B.K.; Almaadeed, S.; Meher, P.K. FPGA implementation of orthogonal matching pursuit for compressive sensing reconstruction. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2014**, *23*, 2209–2220 [CrossRef]
34. Oukaira, A.; Ettahri, O.; Tabaa, M.; Taheri, S.; Lakhssassi, A. Simulation Validation and FPGA Implementation of a Ring Oscillator Sensor for Thermal Management and Monitoring. *Microelectron. J.* **2019**, *155*, 83–85 [CrossRef]
35. 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.
36. 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]
37. Wójciak, W.; Napieralski, A. Monitoring of a single heat source in semiconductor devices—The first approach. *Microelectron. J.* **1997**, *28*, 313–316 [CrossRef]
38. ANSYS Fluent Theory Guide. Available online: https://ansyshelp.ansys.com (accessed on 22 May 2023).
39. Lee, K.J.; Skadron, K.; Huang, W. Analytical model for sensor placement on microprocessors. *Int. Conf. Comput. Des.* **2005**, *124*, 24–27
40. Mesa-Martinez, F.J.; Ardestani, E.K.; Renau, J. Characterizing processor thermal behavior. *ACM SIGARCH Comput. Archit. News* **2010**, *38*, 193–204 [CrossRef]
41. Ettahri, O.; Oukaira, A.; Ali, M.; Hassan, A.; Nabavi, M.; Savaria, Y.; Lakhssassi, A. A real-time thermal monitoring system intended for embedded sensors interfaces. *Sensors* **2020**, *20*, 5657. [CrossRef] [PubMed]
42. Chen, K.C.; Chen, Y.H.; Lin, Y.P. Thermal sensor allocation and full-system temperature characterization for thermal-aware mesh-based NoC system by using compressive sensing technique. In Proceedings of the 2017 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, Taiwan, 24–27 April 2017; Volume 1109, pp. 1–4

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.