

Thèse de Doctorat PhD

Présenté à :

L'Université du Québec en Outaouais

Comme exigence partielle
du Doctorat en Sciences et Technologies d'Information (3081)

Titre :

**Étude, conception et développement d'un algorithme de
la détection des pics thermiques dans le cas des sources
multiples de chaleur dans les circuits intégrés**

Réalisé par :

Fathi Ben Hamouda

Dirigé par :

Prof. Ahmed Lakhssassi, Ph.D., ing.

Dédicace

A titre de reconnaissance, d'amour

Je dédie cette contribution scientifique à :

Bouya Sidi Taher Omrane & Oummi Assia, leurs enfants et leurs petits enfants

Les Kadiris de Baghdâd, Nefta, Sidi Bouzid, Sfax, Msaken, Tunis, Bizerte & partout.

Mes parents, mes frères et mes sœurs, mes cousins et cousines (Ben Hamouda et Mahjoub) leurs enfants et leurs petits enfants

*Mon âme sœur et bien aimée Souad,
ses parents, ses frères et ses sœurs, leurs enfants et leurs petits enfants*

*Toute ma petite famille,
Mes enfants, mes filles,*

*Mariem & son mari Hassen Kaänich, Assia & Hicham, Emna & Mahdi,
Si Mohammed Amine ,*

*Marwa BH & Maroua J, Sarra, Nouha, Mehdi, Lina, Wala, Wafa, Wiem, Zineb, Si
Mohamed Taher, Si Abdelkader, Innes, Med Ali , Marwa S, Asma BH, Med Nour, Med
Salah Nesrine et Sidi Abdelkader BH ses enfants Latifa ,Mariem et Yassine BH*

Et mes Petits-Enfants,

Remerciements

Tout d'abord, je remercie mon premier professeur qui m'a recommandé pour continuer mes études graduées Sidi Taher Omrane puis la professeure Mme Souad Ben Hamouda qui m'a accompagnée, m'a beaucoup soutenu au début de la thèse et a sacrifié tout pour ma réussite .Puis, je remercie toute ma famille, mes filles Mariem & son mari Hassen Kaänich, Assia & Emna qui m'ont encouragé durant tout mon parcours de cette réalisation scientifique.

Ensuite, je remercie le professeur Monsieur Ahmed LAKHSSASSI, directeur du laboratoire LIMA (Laboratoire d'Ingénierie des Microsystèmes Avancés), pour m'avoir accueilli au sein de son laboratoire et pour l'honneur qu'il me fait en acceptant l'encadrement de ce doctorat. J'exprime toute ma gratitude pour son encadrement efficace, ses précieux conseils, son support moral, sa confiance qu'il m'a accordée et son assistance bienveillante m'ont permis de passer à travers tous les difficultés. Par la même occasion je rends hommage à sa rigueur scientifique et technique et son professionnalisme.

Mes remerciements s'adressent aussi à mes amis (ies) et à tous ceux et celles qui de près ou de loin m'ont permis de bien mener à terme mes activités de recherches doctorales.

Table des matières

Liste des figures.....	8
Liste de tableaux	10
Chapitre 1 : Introduction générale.....	11
1.1 Introduction	12
1.2 Motivations de la thèse.....	12
1.3 Méthodologie.....	12
1.4 Approche utilisée	13
1.4 Contributions.....	13
Chapitre 2 : État de l’art.....	15
2.1 Introduction à l’état de l’art.....	16
2.2 Contributions en dehors du laboratoire LIMA de l’UQO :	17
2.2.1 Contribution d’Andrzej Napieralski :	18
2.2.2 Contribution de Boemo :	18
2.3 Contributions au sein du laboratoire LIMA à l’UQO.....	23
2.3.1 Contribution de Boustany	24
2.3.2 Contribution de Mohammed Bougataya.....	25
2.3.3 Contribution de Saydé.....	26
2.3.4 Contribution de Oukaira	27
Chapitre 3 : Modélisation et Simulation.....	29
3.1 Introduction :	30
3.2 Objectif et Méthodologie	30
3.2.1 Objectif :	30
3.2.2 Méthodologie.....	31
3.3 Modélisation et simulation par NISA	32
3.3.1 Modèle de test.....	32
3.3.2 Hypothèses :	34
3.3.3 Résultats : Tableaux des données, images thermiques.....	35
3.4 Analyse des Résultats	39
3.4.1 Choix de la valeur instantanée appropriée pour étude et analyse	39
3.4.2 Analyse des pics thermiques en évolution	40

3.5	Solutions algorithmiques explorées	45
3.5.1	Contexte de la problématique.....	45
3.5.2	Objectif	46
3.5.3	Tâches de la procédure générale à suivre.....	46
3.5.4	Solutions Algorithmiques explorées.....	46
Chapitre 4 : Algorithme du k plus proche voisin (KPPV-SODA).....		47
4.1	Introduction de la méthode de recherche des K plus proches voisins :	48
4.2	KPPV-SODA pour la détection des pics thermiques dans les CI :.....	48
4.2.1	Prototype conçu pour vérification et validation de l'algorithme (KPPV-SODA)	49
4.2.2	Pseudocode de l'algorithme (KPPV-SODA)	51
4.2.3	Diagramme de flux de l'algorithme (KPPV-SODA) :.....	52
4.3	Analyse et discussion de l'application de l'algorithme (KPPV-SODA)	53
Chapitre 5 : Algorithme basé sur la matrice contour (MSTPDA)		54
5.1	Introduction :.....	55
5.2	Recueil, organisation et affichage des données.....	55
5.2.1	Modèle ou prototype sujet de l'étude.....	55
5.2.2	Recueil des données	55
5.2.3	Organisation et représentations des données	55
5.2.4	Comment détecter la position d'un nœud (i) sachant sa température.....	57
5.4	Diagramme de flux de l'Algorithme MSTPDA	62
5.5	Implémentation de l'Algorithme MSTPDA dans Matlab	63
5.6	La Matrice Contour "MCT"	63
Chapitre 6 : Algorithme de balayage thermique « MHSODA »		67
6.1	Modèle ou prototype conçu pour la validation de l'algorithme « MHSODA ».....	68
6.2	Principe de base de l'algorithme « MHSODA ».....	68
6.3	Développement de de l'algorithme « MHSODA »	68
6.4	Diagramme de flux de l'Algorithme de balayage MHSODA	70
	70
6.5	Exécution de « MHSODA » en C++ :	71
6.6	Analyse et Discussion de la 3e solution algorithmique de MHSODA	71
6.6.1	Analyse des résultats de l'exécution de MHSODA :	71
6.6.2	La performance de l'algorithme MHSODA	73
6.6.3	Discussion de la solution algorithmique MHSODA	74
Chapitre 7 : Algorithme de balayage thermique GDSCAN basé sur GDS		75
7.1	Introduction :.....	76

7.2	Modèle ou prototype conçu pour la validation du « GDSCAN »	76
7.4	Principe de l'algorithme GDS pour chaque cellule de capteurs	77
	77
7.5	Détails de l'application du GDS pour chaque cellule de capteurs	78
7.6	Progression de l'application de l'algorithme « GDS » en balayant les zones du CI	79
7.7	Détection de multiples sources de chaleur par GDSCAN	79
7.8	Validation numérique par MATLAB de l'Algorithme (GDSCAN).	79
7.9	Programmation en langage C++ de l'Algorithme (GDSCAN).	85
7.10	Étude de l'implémentation matérielle du (GDSCAN) sur carte FPGA	88
7.11	Analyse et discussion pour l'algorithme (GDSCAN)	88
	Conclusion générale :	90
	Travaux futurs de Recherches	93
	Références	94
	Annexes	97
	• Annexe A1 : Image thermique obtenue par NISA	98
	• Annexe A2 : Comparaison Image NISA Vs Image Matlab (MSTPDA)	99
	• Annexe A3 : Résultat de l'exécution de l'Algorithme MHSODA	100
	• Annexe A4 : Articles de l'auteur :	101
	• Annexe A5 : diagramme de flux Algorithme de balayage MHSODA	102
	• Annexe A6 : Étude technique de l'implémentation du GDSCAN sur une carte FPGA	103
	• Annexe A7 : ABSTRACT de l'article relative à la thèse	107
	• Annexe A8 : Affiche scientifique relative à la Thèse	108
	• Annexe A9 : Étude Bibliométrique sur la détection thermique dans les(CI)	109

Résumé

L'évolution de l'industrie des circuits intégrés (CI) durant la dernière décennie a été tellement rapide qu'il est possible d'intégrer actuellement des systèmes complexes sur une seule puce (SoC, System on Chip). Cette évolution vers des niveaux d'intégration de plus en plus élevés est motivée par les besoins de systèmes de plus en plus performants et par conséquent dissipant d'énormes densités de puissance. L'augmentation de la densité de puissance dissipée des processeurs provoque une augmentation de la température des CI et par conséquent de ce fait on assiste à une dégradation graduelle des performances des microsystèmes électroniques accentuée avec la diminution de la taille des CI. Ce développement a généré beaucoup de problèmes en ce qui concerne le contrôle de la température et la gestion des courants de fuite au niveau transistors sachant que ces courants augmentent exponentiellement avec la température; d'où la nécessité de la vérification et de la gestion thermique interne de ces microsystèmes complexes. Actuellement des chercheurs ont déjà conçu un algorithme de détection de pics thermiques dans le cas d'une seule source de chaleur basée sur la technique GDS (Gradient Direction Sensors). Cependant, ceci ne résout pas le problème du réchauffement dans les circuits LAIC (circuits de grandes tailles), là où il y a un besoin criant, et qui est causé pratiquement par plusieurs sources de chaleur, d'où la nécessité de chercher une solution algorithmique qui permet de détecter plusieurs sources de chaleur; c'est mon sujet de thèse. En effet, ma thèse consiste à étudier, concevoir et développer des solutions algorithmiques pour la détection de pics thermiques dans le cas des sources multiples de chaleur (un nombre de sources de chaleur supérieur à « un »). La méthodologie de l'algorithme GDSCAN qui a été utilisée est basée sur la technique GDS généralisée pour le cas des sources de chaleur multiples. Une configuration de quatre (4) cellules GDS a été proposée pour la détection des pics thermiques dans le cas de sources multiples. Nous avons visé en particulier un prototype de test de notre algorithme avec quatre sources de chaleur. Dans la même optique, on a fait une exploration de quelques algorithmes utilisés dans les domaines de l'ingénierie et du génie logiciel applicables au domaine de détection thermique de sources de chaleur dans les (CI). Nous avons exploré l'algorithme de recherche du « K Plus Proche Voisins » (KPPV) ou (KNN en Anglais), l'algorithme de traitement d'image basé sur la Matrice Contour (MC) et l'algorithme de balayage de surface (Scanning). Les quels algorithmes donnent des résultats satisfaisants mais pas autant que l'algorithme GDSCAN qui offre des estimations de températures très précises avec une marge d'erreur ne dépassant pas les 1.5% et une architecture de réseau de capteurs optimale qui inclut un minimum de capteurs thermiques.

Liste des figures

Figure 1 : Deux schémas de reconfiguration dynamique sur FPGA : (a) et (b)	20
Figure 2: Connexion optimale des capteurs RO pour détecter la température dans une FPGA [5]....	20
Figure 3 : Schéma du capteur de température RO reconfigurable[5].....	21
Figure 4 : Disposition superposée des capteurs RO en plaquette[6]	22
Figure 5: Effet de l'augmentation de la température au niveau de la source de chaleur [6].....	22
Figure 6: Méthode de Direction du gradient de température (GDS)	24
Figure 7 : Diagramme de flux de l'algorithme STPDA[7].....	25
Figure 8: Un modèle en éléments finis (FEM) de 2 triplets de capteurs et 2 sources de chaleur [8]	25
Figure 9 : Distribution thermique sous forme d'isothermes autour d'une source de chaleur [8].....	26
Figure 10: Montage et mesures avec une caméra infrarouge sur un « Wafer IC TM » [10].....	27
Figure 11: Implémentation matérielle du modèle à 2 pics thermiques dans un « Wafer IC TM » [10].	27
Figure 12: Implémentation du modèle dans une plateforme Altera [11].....	28
Figure 13: Effet de variation de la température ΔT dans un CI. [5].....	31
Figure 14: Modélisation d'un circuit intégré comportant quatre (4) sources de chaleurs[38].....	32
Figure 15: Prototype d'un CI avec 4 points rouges (pics thermiques) [38].....	33
Figure 16: Expérimentation numérique -puce avec 4 points chauds (devenus pics de chaleur) [38]..	34
Figure 17. Cas de quatre sources de chaleurs qui fusionnent en une seule.....	39
Figure 18: les valeurs instantanées appropriées (16) sujets de l'étude [38]	40
Figure 19: Modèle d'un circuit intégré en 2D [38].....	41
Figure 20 : Modèle d'un circuit intégré en 3D montrant ses couches de silicium superposées[38]....	41
Figure 21: Snapshot sélectionnée pour l'étude[38]	42
Figure 22 : Commencement de toucher des 2 sources de chaleur[38]	42
Figure 23 : Commencement de fusionnement des deux (2) sources de chaleur[38].....	43
Figure 24: Commencement de fusionnement des trois (3) sources en une seule (1) [38].....	43
Figure 25 : Commencement de fusionnement des quatre (4) sources en une seule (1) [38].....	44
Figure 26: Les quatre sources de chaleur en cours de fusionnement en une seule (1) [38].....	44
Figure 27 : La variation de la température dans l'espace[38].....	45
Figure 28: Schéma du prototype à modéliser. [38].....	45
Figure 29 : Schéma du prototype à modéliser [38].....	49
Figure 30 : Schéma du prototype de l'étude de l'algorithme (KPPV-SODA)	50
Figure 31: pseudocode de l'algorithme KPPV-SODA [38].....	51
Figure 32: Diagramme de flux de l'algorithme KPPV-SODA [38].....	52
Figure 33: Algorithme MSTPDA,[38].....	62
Figure 34: l'image thermique générée par la validation de MSTPDA dans Matlab,[38].....	65
Figure 35: comparaison de 2 images celle obtenue par NISA et celle par MSTPDA,[38]	66
Figure 36 : Diagramme de flux Algorithme MHSODA [38]	70
Figure 37 : Résultats d'exécution du code MHSODA,[38]	71
Figure 38: Listage du pic thermique (1) avec ses isothermes, de l'exécution de MHSODA,[38]	72
Figure 39: Pics thermiques (2 et 3), listage des résultats d'exécution du MHSODA,[38].....	73
Figure 40 : Prototype conçu pour la validation de l'algorithme « GDSCAN ».....	76
Figure 41 : Principe de l'algorithme du GDS pour la détection d'une seule source de chaleur,[7].....	77
Figure 42 : Cellule (i) de capteurs thermiques (6RO) orientée vers la droite,[38]	80
Figure 43: Résultats du GDSCAN et estimation des températures des sources de chaleur	84

Figure 44: Résultats du GDSCAN pour l'estimation des positions des sources de chaleur.....	85
Figure 45:Diagramme de flux détaillé de l'Algorithme (GDSCAN)pour C++.....	86
Figure 46:Pseudocode détaillé de l'Algorithme (GDSCAN)pour C++.....	87
Figure 47:Listage des résultats de l'exécution de l'Algorithme (GDSCAN) en C++.....	88

Liste de tableaux

Tableau 1 : Principales contributions au domaine de détection thermique dans les CI[38]	17
Tableau 2 : Contribution des membres du LIMA en utilisant le GDS[38].....	23
Tableau 3 : A-Liste des numéros des nœuds [38].....	36
Tableau 4 : B-Liste des coordonnées des nœuds[38].....	37
Tableau 5 : C-Liste des températures des nœuds[38]	38
Tableau 6: Matrice Température (MT),[38]	58
Tableau 7: Matrice des nœuds (MN),[38].....	59
Tableau 8 : Matrice des coordonnées des nœuds (MXY),[38]	60
Tableau 9: Feuille Excel des images thermiques obtenus par NISA (avec et sans quadrillage),[38]..	61

Chapitre 1 : Introduction générale

1.1 Introduction

L'évolution de l'industrie des circuits intégrés (CI) durant la dernière décennie a été tellement rapide qu'il est possible d'intégrer actuellement des systèmes complexes sur une seule puce (SoC, System on Chip).

Cette évolution vers des niveaux d'intégration de plus en plus élevés est motivée par les besoins de systèmes de plus en plus performants et par conséquent dissipant d'énormes densités de puissance.

Cela, provoque l'augmentation critique de la température au fur et à mesure que l'augmentation de densité de puissance dissipée des processeurs et par conséquent à une dégradation graduelle des performances des microsystèmes électroniques avec la réduction de la taille des CI.

1.2 Motivations de la thèse

Cette évolution a généré beaucoup de problèmes en ce qui concerne le contrôle de la température et la gestion des courants de fuite au niveau des transistors sachant que ces courants augmentent exponentiellement avec la température et doublent à chaque degré Celsius d'augmentation ; d'où la nécessité du monitoring, de la vérification et de la gestion thermique à l'intérieur de ces microsystèmes complexes.

Actuellement des chercheurs ont conçu déjà un algorithme de détection de pics thermiques dans le cas d'une seule source de chaleur. Cependant, cela ne résout pas le problème du réchauffement dans les circuits LAIC (Large Area Integrated Circuits) circuits de grandes tailles, là où il y a le besoin criant, et qui est causé pratiquement par plusieurs sources de chaleur, d'où la nécessité de chercher une solution algorithmique qui permet de détecter plusieurs sources de chaleur ; c'est notre proposition dans cette thèse. En effet, l'objectif de cette thèse consiste à étudier, concevoir et développer un algorithme de détection de pics thermiques dans le cas des sources multiples de chaleur (un nombre de sources de chaleur supérieur à « un »).

1.3 Méthodologie

La méthodologie proposée dans cette thèse est basée sur la technique GDS généralisée pour le cas des sources de chaleur multiples. Une configuration de cellules GDS sera proposée pour la détection des pics thermiques dans le cas de sources multiples. Nous visons en particulier un prototype de test de notre algorithme avec quatre sources de chaleur dans un CI (Circuit Intégré). Les résultats de test font partie de cette thèse. Dans la même optique, on a fait une exploration de quelques algorithmes utilisées

dans les domaines de l'ingénierie et du génie logiciel et qui s'apparentent applicable au domaine de détection thermique de source de chaleur dans les CI.

1.4 Approche utilisée

L'approche utilisée dans cette thèse a été guidée par les considérations suivantes :

- On a essayé d'améliorer le procédé de détection en vue de garantir des résultats pertinents.
- On a réglé d'abord la question du nombre et du positionnement approprié des capteurs de températures pour une configuration donnée de source de chaleur.
- On a mis en hypothèse « un nombre donné de sources de chaleur qui est quatre (4) » puis on a cherché à optimiser « le nombre approprié de capteurs » et configurer « l'architecture des cellules et de leurs emplacements sur le CI ».
- Une fois les endroits optimaux des capteurs ont été déterminés, le problème de « l'estimation de la température aux niveaux des pics thermiques » à partir des valeurs de températures détectées aux niveaux des emplacements des capteurs de chaleur peut être étudié en détails.

Des expériences simulées numériquement ont été réalisées pour obtenir les données d'entrée générées par simulation pour servir à concevoir un prototype (ou modèle) générique pour les calculs et le traitement dans les équations trigonométriques.

Les résultats des traitements des données ont servi pour la détection de pics thermiques dans le CI.

Toutes les investigations thermiques ont été réalisées avec l'outil NISA basé sur la méthode des éléments finis (FEM) en utilisant le modèle thermique 3D au laboratoire LIMA à l'UQO. Les résultats des simulations thermiques vont servir d'indicateurs pour la conception « d'un prototype réel d'une unité de détection de pics thermiques dédiée aux circuits intégrés dans lequel tout sera implémenté ».

1.4 Contributions

La contribution dans cette thèse consiste principalement en « la conception et le développement d'un algorithme de détection de pics thermiques dans le cas des sources multiples de chaleur ».

Autres contributions de plus, se manifestent implicitement par le faite que « la conception de cet algorithme sera basée sur une configuration en un réseau de capteurs regroupés en cellules pour capter les pics thermiques avec « le minimum de cellules de capteurs » sur la grandeur de la puce ».

En plus, notre approche est basée sur la technique GDS (Gradient Direction Sensor) appliquée au problème de la variation thermique ou différentielle (ΔT) dans les CI et sur l'utilisation de la technologie des capteurs RO (« Ring Oscillator » ou oscillateur en anneau) considérée comme une nouvelle contribution dans le domaine de la détection des pics thermiques. L'optimisation du nombre

de cellules, l'architecture de leurs emplacements et les tests ont été réalisées et validé numériquement par Matlab en vue de leurs implémentations futures sur FPGA [18]

Dans la même optique, on a fait une exploration de quelques algorithmes utilisés dans les domaines de l'ingénierie et du génie logiciel applicables au domaine de détection thermique de sources de chaleur dans les (CI) tel que : 1- l'algorithme de recherche du « K Plus Proche Voisins » (KPPV) ou (KNN en Anglais) ; 2-l'algorithme de traitement d'image basé sur la Matrice Contour (MC) et le Nombre d'Euler(r); 3- l'algorithme de balayage (Scanning). Les quelques algorithmes donnent des résultats satisfaisants mais sous contraintes techniques, opérationnelles et économiques.

Chapitre 2 : État de l'art

2.1 Introduction à l'état de l'art

Il y a plusieurs chercheurs provenant de plusieurs disciplines qui essaient de résoudre le problème de détection de pics thermiques. Les approches de ces chercheurs diffèrent les unes des autres par plusieurs aspects dont :

- Les fondements théoriques et mathématiques qui décrivent la variation thermique sur les «CI» (l'équation de Hough, la fonction de Green, la méthode de GDS avec toutes ses équations).
- Le type de capteurs utilisés dans la détection thermique sont du type standards ou RO (Ring-Oscillator ou Oscillateur en Anneau).
- Le positionnement des capteurs par rapport au CI (à l'intérieur ou l'extérieur du CI).
- Le mode expérimental utilisé pour étudier la détection soit par modélisation et simulation numérique (Méthode des Éléments Finis FME ou autre) ou autre
- Le mode d'implémentation dans un système réel de capteurs sur les CI tel que Michel Saydé a réalisé, ou sur une plateforme numérique Xilinx ou Altera.
- Le raisonnement algorithmique utilisé.

On constate que la problématique de la détection d'une seule source de chaleur a été résolue par succès au sein du laboratoire LIMA à l'université (UQO) alors que dans le cas des sources multiples de chaleur n'est pas résolu et il n'y a pas de solution valide et vérifiée jusqu'à date.

Les principales contributions le domaine de détection de pics thermiques sont illustrées dans le tableau suivant (*Tableau 1*) qui a été établi en se référant à une étude bibliométrique faite par SCOPUS (Voir Annexe A9, [36]) sur la citation des chercheurs du monde entier qui ont contribué d'une façon distinctive et importante dans l'étude de la détection des pics thermiques dans les CI et notamment ceux parmi eux qui ont utilisé la méthode GDS.

Parmi eux ceux qui ont traité la détection thermique d'une seule source de chaleur d'autres qui ont traité la détection thermique de plusieurs sources de chaleur. Chacun a traité un ou plusieurs aspects du domaine de la détection thermique dans les CI.

Tableau 1 : Principales contributions au domaine de détection thermique dans les CI[38]

Chercheurs	Contribution Distinctive
Boemo E.	<ul style="list-style-type: none"> ❖ Boemo a vérifié l’insertion, l’exploitation et l’élimination dynamiques des différentes configurations de systèmes munis de capteurs thermiques RO implémentés dans une FPGA. ❖ Il a réalisé un système de détection de sources de chaleur sur les CI et de monitoring thermique en ligne en installant une unité de traitement des données en dehors du puce (CI) [16].
Napieralski A.	<ul style="list-style-type: none"> ❖ Napieralski a traité la problématique de détection d’une source seule de chaleur et par après celle de multiple sources de chaleur. Pour ses simulations thermiques et pour la génération de résultats, Napieralski s’est basé sur son propre « solutionneur thermique tridimensionnel utilisant la solution de la fonction de Green de l’équation de la chaleur » [21], [22], [23] , [24] , [25].
Boustany CH.	<ul style="list-style-type: none"> ❖ Il a été le premier à concevoir un modèle et valider l’algorithme STPDA de détection d’une seule source de chaleur basée sur la méthode de GDS en utilisant des capteurs RO
Bougataya M.	<ul style="list-style-type: none"> ❖ il a implémenté le modèle dans un puce VLSI et a validé matériellement (dans le hardware) l’algorithme de détection d’une seule source de chaleur STPDA basé sur la méthode du GDS et utilisant des capteurs de type RO
Saydé M.	<ul style="list-style-type: none"> ❖ Contribution Michel Saydé s’est distinguée par les réalisations suivantes : <ul style="list-style-type: none"> • Il a utilisé les mesures d’une caméra infrarouge pour tester un algorithme de détection thermique de sources de chaleur dans un espace spatiotemporel. • Il a montré, caractérisé et étalonné un capteur de température à base d’un oscillateur en anneau (RO) dans un FPGA; • Il a étudié, adapté, et appliqué des algorithme d’amélioration et de linéarisation sur la réponse d’un capteur en anneau (RO) [34],[35].
Oukaira A.	<ul style="list-style-type: none"> ❖ Sur une plateforme ALTERA , Il a implémenté un modèle de détection d’une seule source de chaleur utilisant l’algorithme STPDA dans une FPGA munie de deux cellules incluant six capteurs RO .

2.2 Contributions en dehors du laboratoire LIMA de l’UQO :

Parmi les chercheurs qui sont parmi les plus connus pour leurs contributions dans le domaine de la détection des pics thermiques dans les CI on s’intéresse aux contributions du polonais Prof. Andrzej

Napieralski de l'espagnol Bomeo et du Canadien Dr Lakhssassi professeur de l'UQO et directeur du laboratoire LIMA) [17], [20], [21].

2.2.1 Contribution d'Andrzej Napieralski :

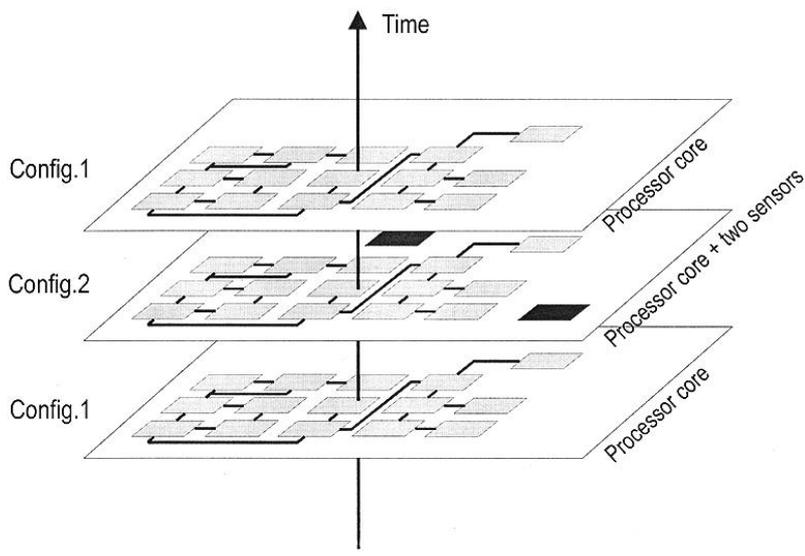
Le chef de file des études théoriques dans le domaine est le professeur polonais Dr Andrzej Napieralski [36] qui a essayé de le résoudre de plusieurs façons dont l'algorithme fondé sur les concepts du problème inverse qui pose mal le problème [1], [26], [27]. Dans ses études de gestion thermique dans les CI il s'est basé sur la transformé de Hough [2]. Il a été l'un des premiers à traiter la problématique de la détection d'une seule source de chaleur. Une de ses dernières tentatives est qu'il suppose un nombre de sources de chaleur supérieur à « un » (qui est de 4), puis il optimise le nombre et le positionnement des capteurs standards (non pas des RO) connectés aux bord du CI et non pas dans le CI. Le monitoring thermique dans le CI se fait à distance, en ligne et le traitement des données se fait dans une unité centrale à part afin de réaliser une estimation en temps réel de la température des sources de chaleur. Pour ses simulations thermiques présentées aussi bien que pour la génération des données d'entrée des expériences Napieralski s'est basé sur son propre « *solutionneur thermique tridimensionnel utilisant la solution de la fonction de Green de l'équation de la chaleur* » [3], [16], [22], [23], [24], [36]. Contrairement à ce qu'il s'attendait, les résultats obtenus n'ont pas été significatifs, comme il a reconnu lui-même, les incertitudes sont trop importantes de façon qu'on ne peut pas valider les valeurs de températures obtenues par estimation en temps réel aux niveaux des sources de chaleur.

2.2.2 Contribution de Boemo :

Boemo a réalisé des travaux concernant la détection de sources de chaleur sur les CI d'une façon générale, il n'a pas précisé dans ses objectifs s'il traite la détection thermique d'une ou de plusieurs sources de chaleur. Sur le plan technologique, il a su tirer profit de l'utilisation des capteurs RO grâce à leur efficacité [4], [16].

Dans un premier temps Boemo a réalisé un système de détection de sources de chaleur sur les CI en modélisant un système de monitoring en ligne en installant une unité de traitement des données et de gestion thermique en dehors du puce (CI) [5] (voir Figure 1(a) et (b)). Aussi Boemo a vérifié l'insertion, l'exploitation et l'élimination dynamiques des différentes configurations de systèmes (ou prototypes) comportant des capteurs thermiques RO implémentés dans une FPGA [5] (voir Figure 2 et Figure 3). Mais à cette étape, Boemo a obtenu des solutions qui se sont avérées inefficaces à cause des délais et des imprécisions des résultats considérables [16], [34], [35].

Par après, la contribution distinctive de Boemo a été le fait d'utiliser une plaquette contenant des capteurs RO uniformément distribués en forme de grille au niveau du CI pour enregistrer la tension en avant de la jonction (laquelle va nous renseigner sur la valeur de la température détectée), (voir Figure 4) [6]. Certes cette solution est réaliste tout en considérant que la puce est logée dans un boîtier muni d'un ventilateur et d'un dissipateur de chaleur. Mais cette solution n'a pas été satisfaisante car elle a généré beaucoup d'erreurs significatifs aux niveaux des valeurs des températures détectées ; lesquels sont dues à la surchauffe causée par la plaquette des capteurs RO. Toutes ces erreurs rendent le monitoring de la chaleur difficile dans le CI, il y a un risque d'endommagement de celui-ci (voir Figure 5) [6] [16].



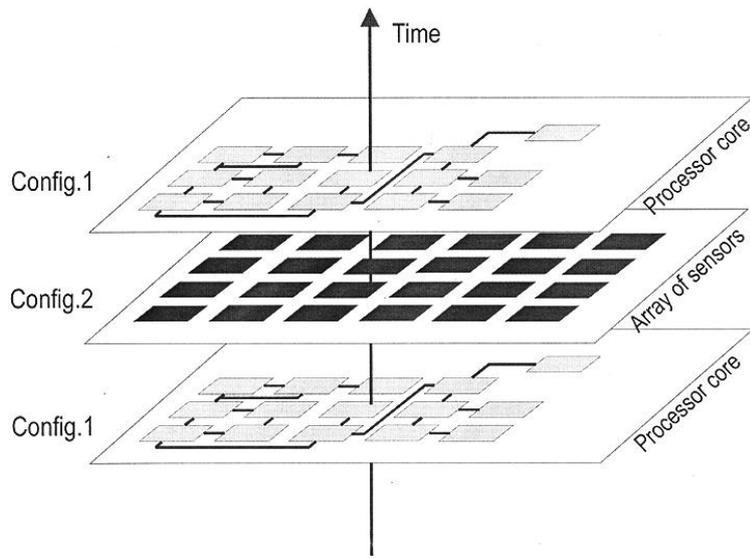


Figure 1 : Deux schémas de reconfiguration dynamique sur FPGA : (a) et (b)

(NB : Les deux reconfiguration s'intitulent : (a) monitoring thermique en ligne et (b) plaquelette de capteurs RO pour mesurer la température. [5])

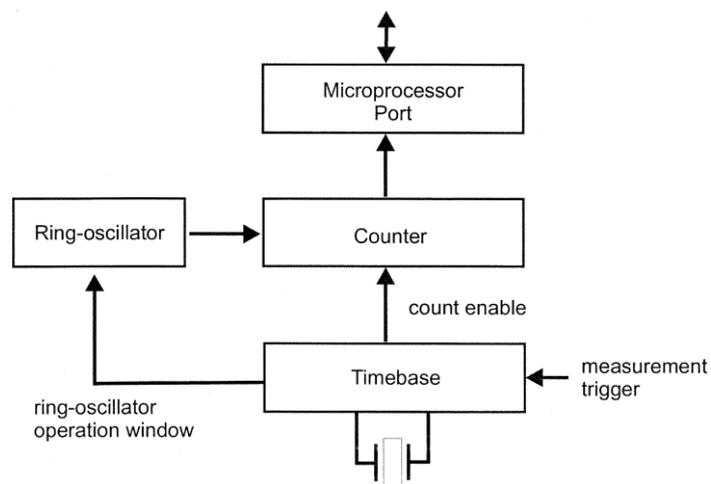


Figure 2: Connexion optimale des capteurs RO pour détecter la température dans une FPGA [5]

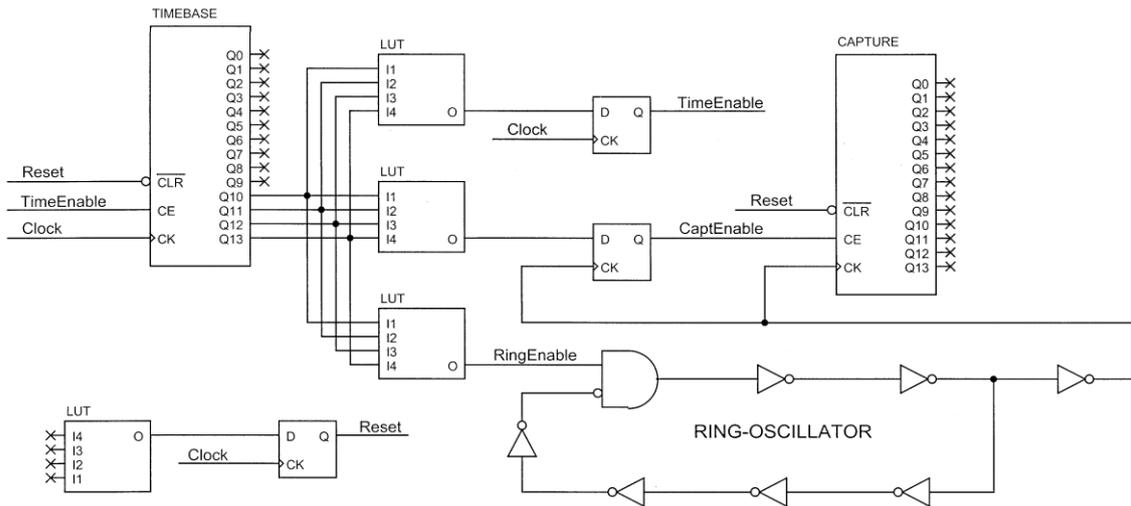


Figure 3 : Schéma du capteur de température RO reconfigurable[5]

2.2.3 Un capteur de température à base de RO et ses avantages (voir Figure 3)

Un capteur de température à base de RO enregistre une variation de fréquence lors de la détection d'une valeur de température.

Un capteur de température à base de RO permet une détection de la température très précise grâce à ses caractéristiques distinctives suivantes :

- L'auto-réchauffement « Self heating » n'a pas d'effet de sur la réponse du capteur RO.
- La réponse d'un capteur de température à base de RO est parfaitement linéaire (la fréquence du RO décroît linéairement quand la température augmente)
- Le changement de la tension d'alimentation n'a pas d'effet sur la réponse du capteur RO, la température détecté reste stable.
- Utilisation d'un grand nombre d'inverseurs en cascade (pourvu qu'il soit impair) n'a pas d'effet de sur la réponse du capteur RO.

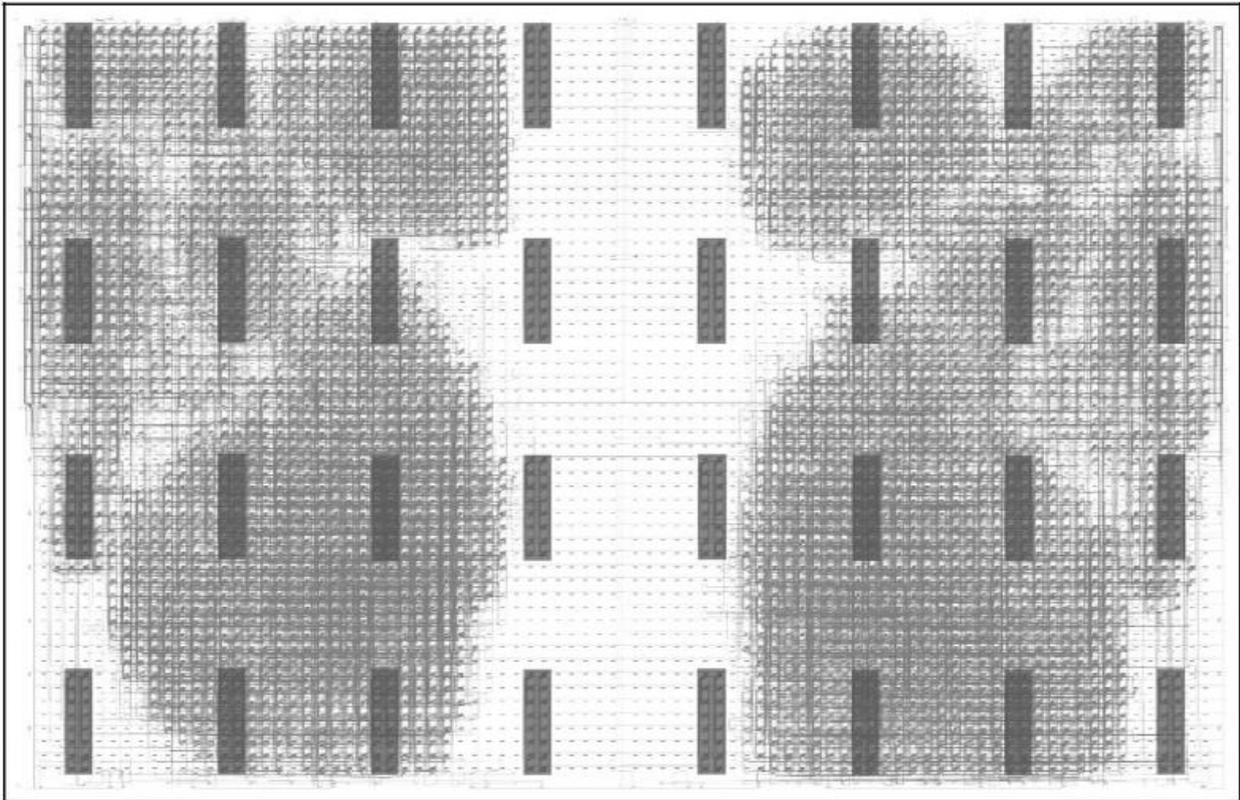


Figure 4 : Disposition superposée des capteurs RO en plaquette[6] .
Les capteurs RO disposés en plaquette sont uniformément distribués sous forme de grille de rectangles) et des deux processeurs jumeaux (zones sombres). Virtex XCV800HQ240-4C.

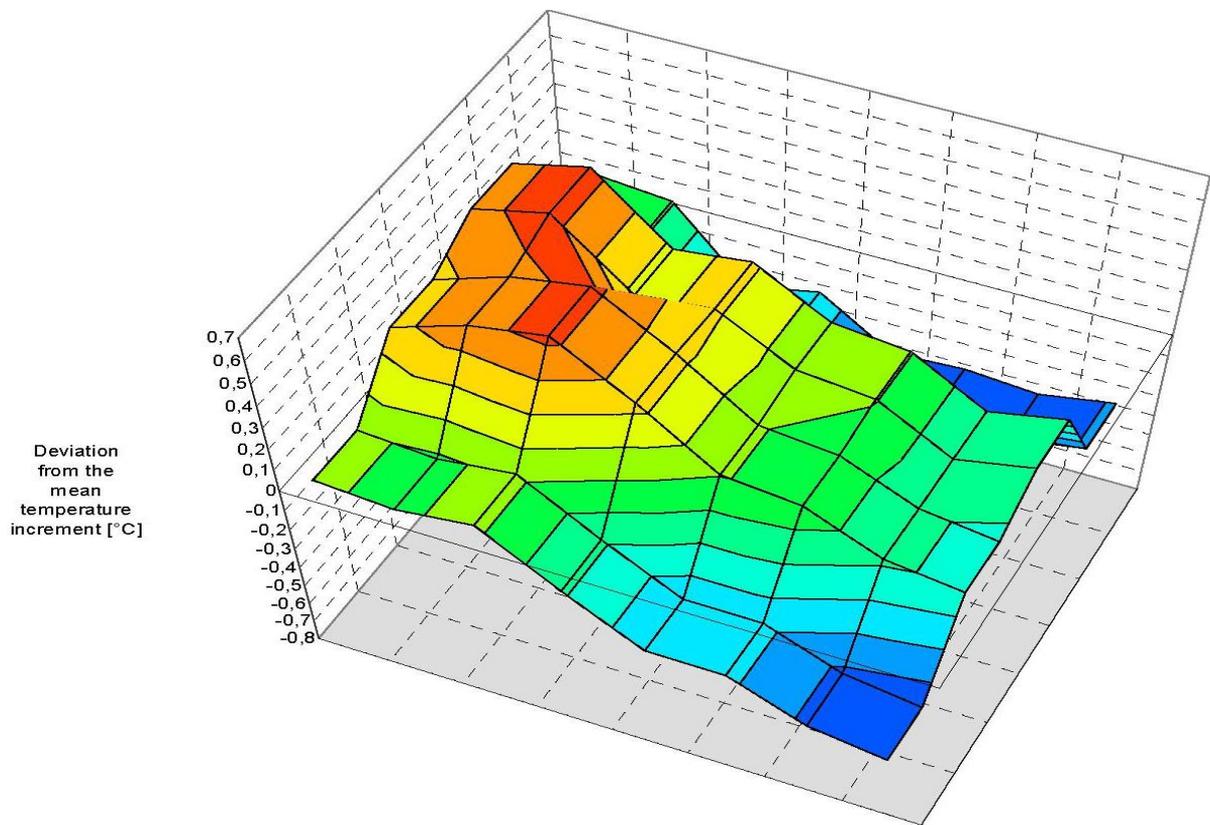


Figure 5: Effet de l'augmentation de la température au niveau de la source de chaleur [6]

La figure (5) montre la distribution de température au niveau du point chaud qui risque d'être endommageant pour le CI [6].

Beaucoup des membres du LIMA traitent le thème de la détection thermique de sources de chaleur dans les (CI). Le Tableau 1 résume les principales contributions utilisant la méthode du GDS et le Tableau 2 résume les principales contributions dans le monde [20], [21].

2.3 Contributions au sein du laboratoire LIMA à l'UQO

Tableau 2 : Contribution des membres du LIMA en utilisant le GDS[38]

Titres des Articles	Auteurs	Année	Source
Modelling and FPGA Implementation of a Thermal Peak Detection Unit for Complex System Design	Oukaira A. Ouafaa E. Lakhssassi A.	2017	International Journal of Advanced Computer Science and Applications (IJACSA) 8. 6 (2017)
SoC systems thermal monitoring using embedded sensor cells unit	Lakhssassi A., Saydé M.,	2012	In Circuits and Systems (NEWSCAS), 55th IEEE International Midwest Symposium
Thermal stress monitoring using gradient direction sensors	Lakhssassi, A., Bougataya M., Boustany, C., Massicotte, D.	2008	2008 Joint IEEE North-East Workshop on Circuits and Systems and TAISA Conference, NEWCAS-TAISA
VLSI thermal stress monitoring using GDS method	Lakhssassi, A., Bougataya, M., Boustany, C., Massicotte, D.	2008	Canadian Conference on Electrical and Computer Engineering
Steady state thermo-mechanical stress prediction for large VLSI circuits using GDS method	Bougataya, M., Lakhssassi, A., Daniel, M.	2007	Canadian Conference on Electrical and Computer Engineering
Practical approach to gradient direction sensor method in very large scale integration thermomechanical stress analysis	Lakhsasi, A., Bougataya, M., Massicotte, D.	2006	Journal of Vacuum Science and Technology A: Vacuum, Surfaces and Films
VLSI thermo-mechanical stress analysis by gradient direction sensor method	Bougataya M., Lakhsasi, A., Massicotte, D.	2005	Canadian Conference on Electrical and Computer Engineering

Ce tableau a été établi en se référant à une étude bibliométrique faites par SCOPUS sur l'utilisation et la citation des chercheurs qui ont utilisé la méthode GDS pour la détection des pics thermiques dans les CI. (Voir Annexe A9), [20].

2.3.1 Contribution de Boustany

Une des premières contributions réussies est celle réalisé au sein du laboratoire LIMA à l'UQO par Charbel Boustany, puis repris par Mohamed Bougataya qui ont solutionné le problème de détection de pics thermique pour une seule source de chaleur.

Théoriquement Charbel Boustany a découvert la possibilité de l'application de la technologie du GDS (Gradient Detection Sensor) dans le monitoring thermique des CI. Il a été le premier à concevoir un modèle de détection de pics thermique basé sur la méthode du GDS (dans laquelle on applique des règles mathématiques de trigonométrie dans un contexte thermo-spatiale d'un CI qui varie dans le temps), (voir Figure 6), [7].

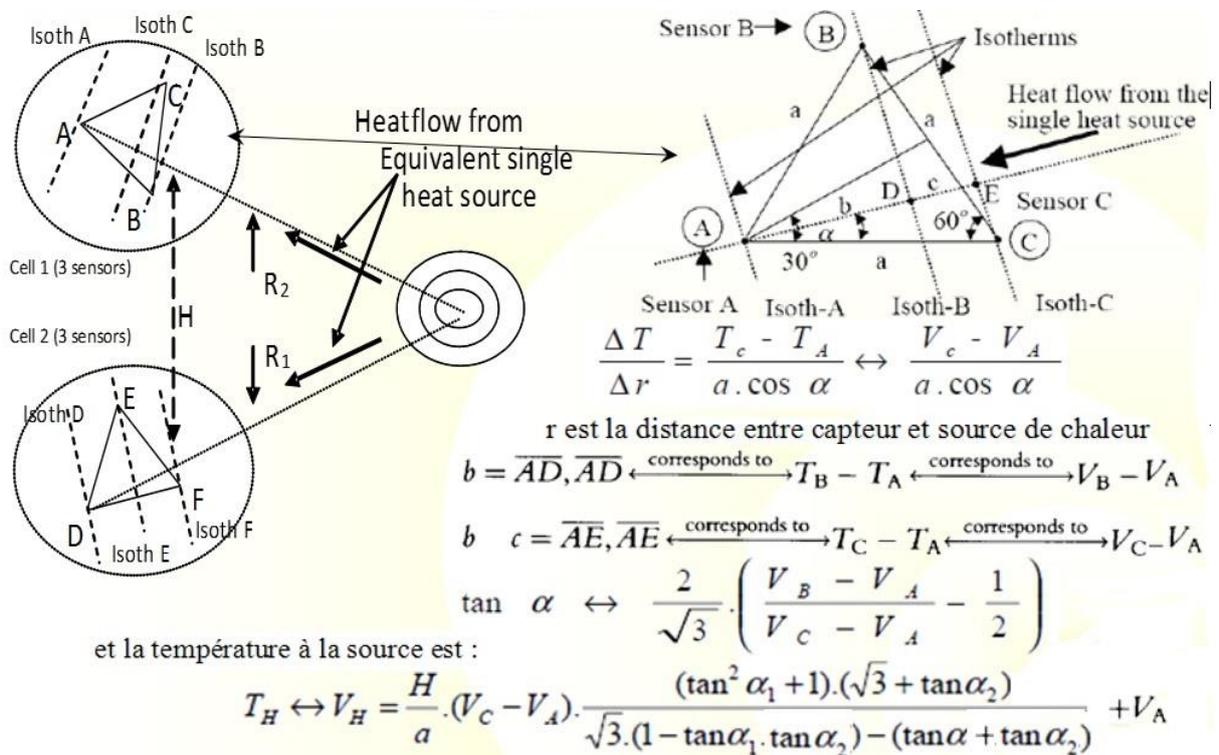


Figure 6: Méthode de Direction du gradient de température (GDS)

La méthode du GDS a été conçu pour la détection d'une seule source de chaleur. [1],[7],[12].

Les capteurs qu'ils ont utilisés pour la détection thermique sont du type RO. Il a conçu et validé numériquement l'algorithme de détection d'une seule source de chaleur STPDA (voir Figure 7).

Les détails de l'application du GDS pour détecter une seule source de chaleur tel que présenté dans (la Figure 6) pour chaque cellule de capteurs (i) seront explicités dans la section du sous-titre :

« 7. Détails de l'application du GDS pour chaque cellule de capteurs (i) » .

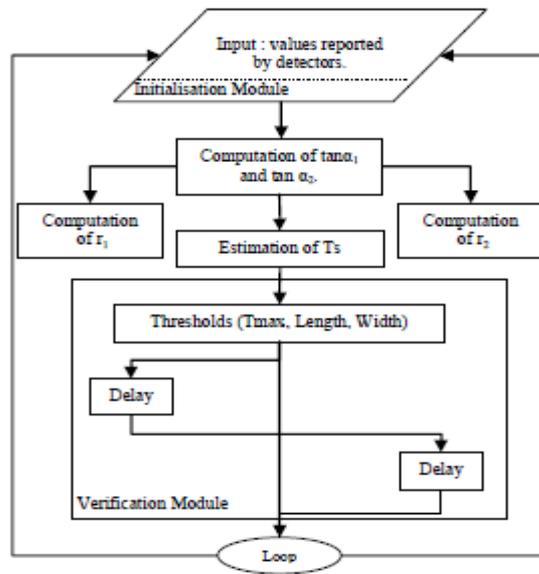


Figure 7 : Diagramme de flux de l’algorithme STPDA[7]

Boustany n’a pas abordé le sujet de détection thermique de plusieurs sources de chaleurs.

2.3.2 Contribution de Mohammed Bougataya

Mohamed Bougataya a repris le même modèle que Boustany puis il a validé l’algorithme STPDA numériquement dans un premier temps. Puis il a implémenté le modèle dans un puce VLSI et a validé matériellement (dans le hardware) l’algorithme de détection d’une seule source de chaleur STPDA en se basant sur la méthode du GDS et utilisant des capteurs de type RO (voir Figure 8) [8], [18].

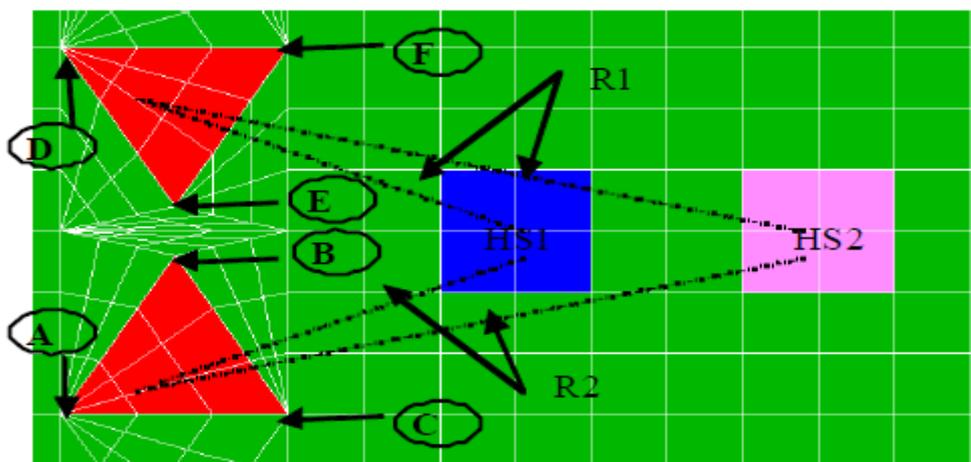


Figure 8: Un modèle en éléments finis (FEM) de 2 triplets de capteurs et 2 sources de chaleur [8]



Figure 9 : Distribution thermique sous forme d'isothermes autour d'une source de chaleur [8]

Mais Bougataya n'a pas résolu le problème de détection thermique pour le cas de plusieurs sources de chaleur. Les résultats qu'il a obtenus sont bonnes et distinctifs (voir Figure 9), [8].

2.3.3 Contribution de Saydé

La contribution de Michel Saydé se limite à l'étude de détection thermique d'une seule source de chaleur, il a repris le modèle que Bougataya [9], il a appliqué la méthode de GDS et il a utilisée des capteurs thermiques RO. Mais cette contribution se distingue par les réalisations suivantes :

- Il a implémenté matériellement (hardware) le modèle dans un puce VLSI, mais il a posé lui-même les capteurs RO directement dans le CI et il a validé la solution du problème de détection de pics thermique pour une seule source de chaleur [18].
- Saydé s'est exceptionnellement distingué par la faite qu'il a installé une caméra infra-rouge spécialement conçu pour visionner l'évolution thermique sur le circuit intégré dans lequel il a implémenté ses capteurs RO (voir Figure 10) [10].
- Malgré qu'il a conçu et implémenté matériellement un modèle comportant deux sources, il n'a pas traité la problématique de détection thermiques pour deux (2) sources comme il a implémenté. Les résultats obtenus sont exceptionnels avec des images réelles prises par caméra infrarouge du modèle matériel implémenté (voir Figure 11) [10].
- En plus, il a validé le modèle a une seule de chaleur, de l'algorithme STPDA et les performances des capteurs qu'il a implémentés.

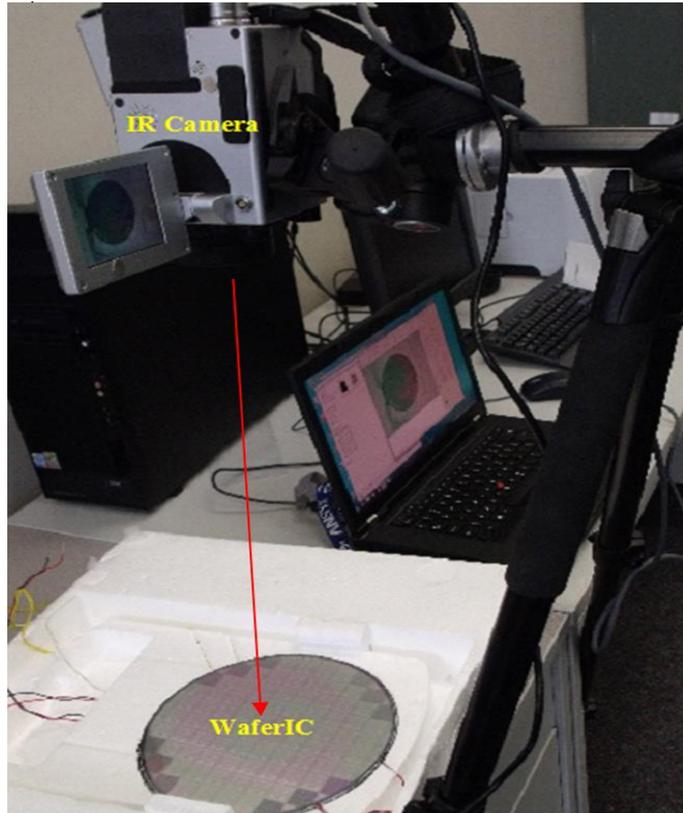


Figure 10: Montage et mesures avec une caméra infrarouge sur un « Wafer ICTM» [10]

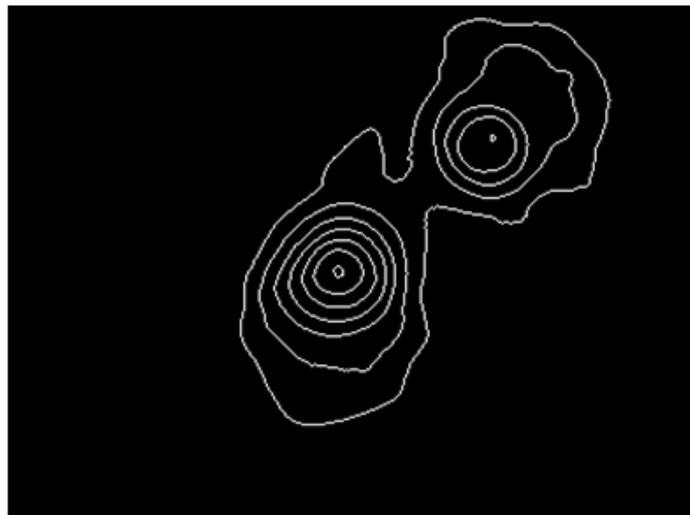
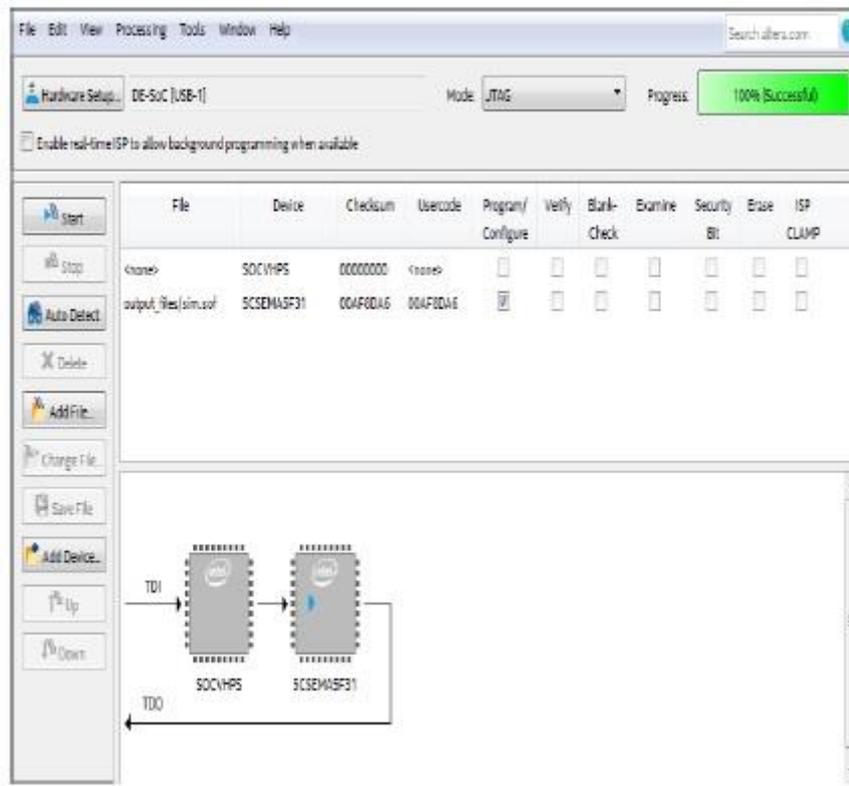


Figure 11: Implémentation matérielle du modèle à 2 pics thermiques dans un « Wafer ICTM» [10]

2.3.4 Contribution de Oukaira

Oukaira a repris le même modèle que Bougataya et Boustany et il s'est distingué par l'implémentation de ce modèle et de l'algorithme STPDA sur FPGA dans une plateforme ALTERA. (Voir Figure 12) [11]. Ainsi, Il a obtenu des résultats satisfaisants et similaires à ceux de ses collègues du LIMA Bougataya ([12], [13] et [18]) et Saydé [10] [17] [19].



Downloading the code of DE1 Altera cyclone V.

Figure 12: Implémentation du modèle dans une plateforme Altera [11]

Chapitre 3 : Modélisation et Simulation

3.1 Introduction :

Dans cette partie, on présente les travaux de recherches réalisées afin de résoudre la problématique de détection de multiples sources de chaleur énoncée précédemment. On va procéder selon les paragraphes suivants :

3.2 Objectif et Méthodologie

3.2.1 Objectif :

Trouver une solution algorithmique qui résout la problématique de la détection des sources multiples de chaleur. Dans la présente thèse, on est concerné seulement par la problématique de la détection et non pas par le problème de refroidissement.

La solution algorithmique qui sera proposée, va aider au monitoring de la chaleur dans le CI de telle façon à ce qu'on y atteigne un état d'équilibre thermique en y réduisant la température (refroidissement) ou en y chauffant des zones critiques afin d'y réduire la contrainte thermomécanique. Cette réduction de la contrainte thermomécanique implique la réduction du niveau de stress mécanique aux niveaux des couches de silicium dans le CI pour éviter de les endommager.

Le monitoring thermique dans les circuits intégrés

Le monitoring (ou contrôle) thermique dans le circuit intégré consiste à effectuer les tâches suivantes :

- Identifier le nombre de sources de chaleur (il ne devrait pas dépasser un certain nombre limite permmissible dans le CI, dépassant cette limite, on le met en Arrêt (OFF)).
- Estimer les positions exactes des pics thermiques.
- Estimer la valeur maximale de température de chaque pic.
- Calculer les variations de température entre ces pics ΔT .
- Si les variations ΔT sont très élevés, le risque de briser le CI est énorme on le met en Arrêt (OFF)).

En effet, si on risque de dépasser la limite élastique du CI, un étirement des couches de silicium peut survenir brusquement.

Pour un CI la limite de la contrainte thermomécanique est de 400 Méga Pascal, plus haut que cette limite les couches de silicium se déforme d'une façon permanente.

- Si les variations ΔT sont modestes on fait répartir la chaleur uniformément dans le CI de façon à avoir un équilibre thermique soit en refroidissant certaine zone du CI ou en chauffant d'autres zones. (Voir Figure 13)

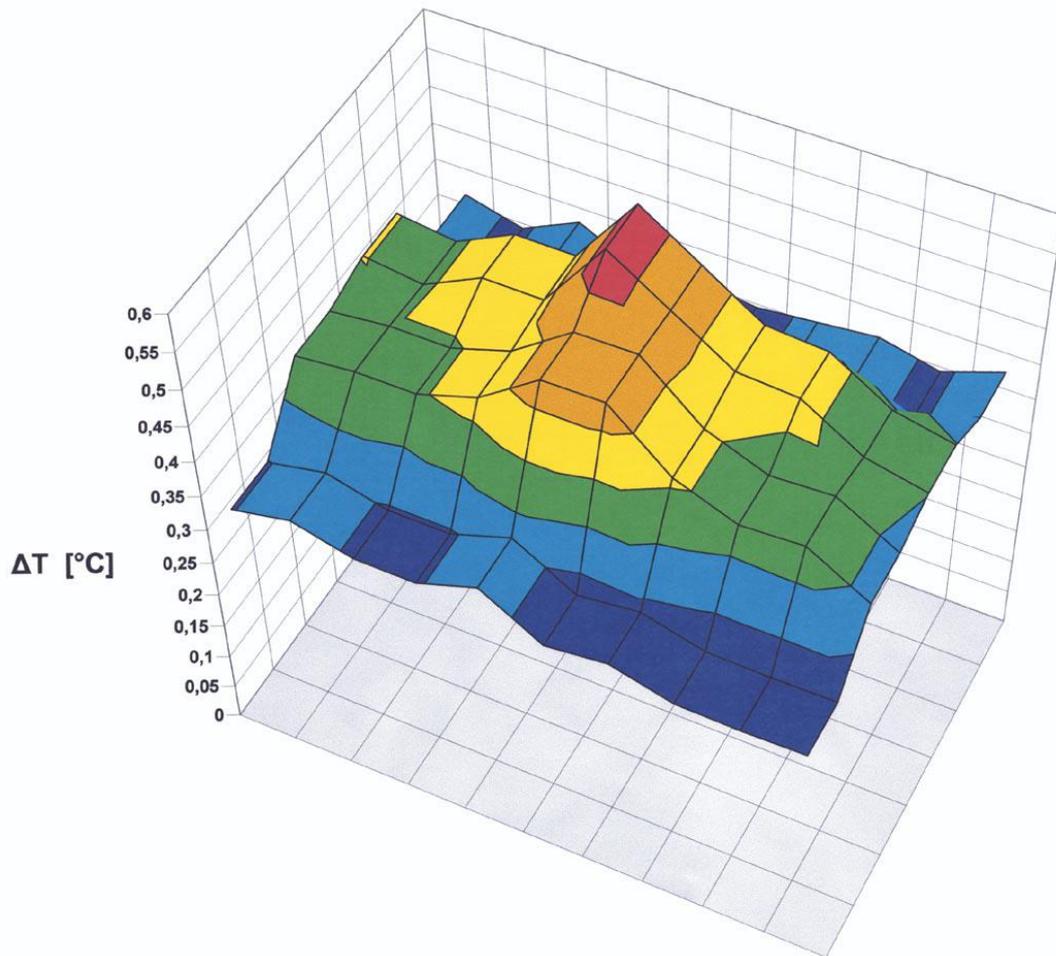


Figure 13: Effet de variation de la température ΔT dans un CI. [5]

La Figure 13 montre l'effet d'un pic thermique sur les couches de la matière composante du CI avec **la contrainte thermomécanique** qui risque d'être endommageant [5].

3.2.2 Méthodologie

Pour résoudre la problématique de détection de sources multiple de chaleur nous avons construit un modèle thermique de CI en 3D pour faire le développement de l'algorithme. Pour ce faire on va réaliser ce qui suit :

- Expérimentation numérique par la méthode FEM
- Modélisation d'un prototype et simulation transitoire par NISA
- Proposition de différentes solutions algorithmiques

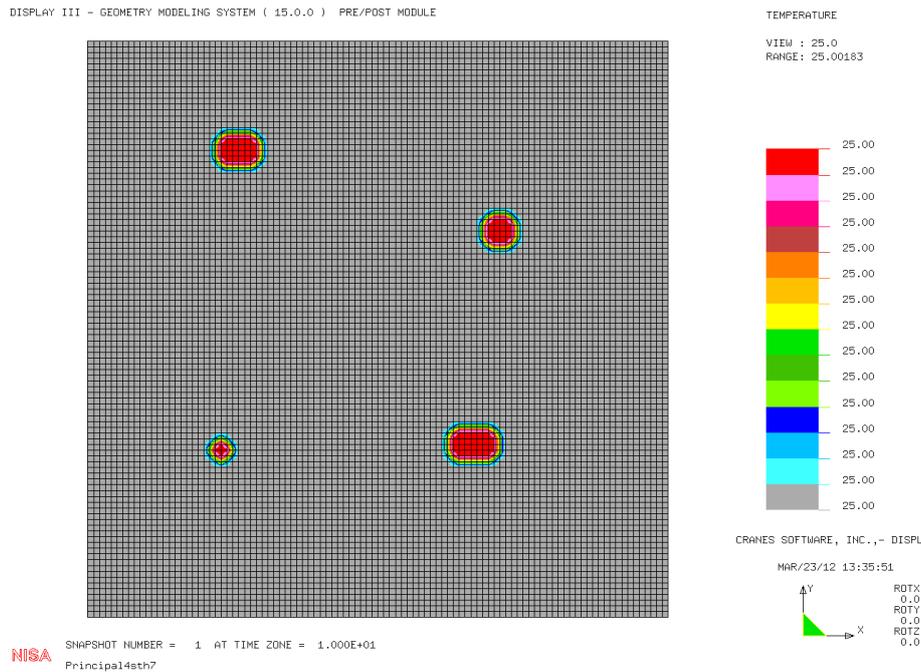
La solution va se baser sur l'utilisation de capteurs en anneau RO (Ring-Oscillator) configurés en cellule de détection. Nous favorisons l'option de l'usage de la technologie de GDS vérifiée pour la détection d'une source de chaleur.

3.3 Modélisation et simulation par NISA

Pour résoudre la problématique de détection de sources multiple de chaleur on va procéder numériquement pour modéliser et simuler la distribution thermique spatio-temporelle dans le cœur d'un circuit intégré avec quatre sources de chaleur.

La Figure 14 montre la diffusion de chaleur dans le CI avec quatre (4) sources de chaleurs.

Les résultats obtenus sont affichés durant le régime transitoire au temps $t = 10\mu s$.



3.3.1 Modèle de test

Les étapes de modélisation du système :

Construction d'un modèle 3D en éléments finis

Le circuit intégré (CI) est représenté géométriquement par un carré dont le côté est de dimension :

Figure 14: Modélisation d'un circuit intégré comportant quatre (4) sources de chaleurs[38] 0.5 cm

(soit 5000 micromètres) et divisé par cent (100) unités selon les axes des X et Y.

Chaque unité représente un (1) nœud ou points sur la surface. Donc théoriquement on a un carré dont la surface est de (101 x 101) nœuds.

- Injection de quatre sources de chaleur (rouges)** dispersés dans le carré représentant le CI, sur chacun desquels on applique une énergie chauffante (la source chaleur) variante d'un point à un autre (voir Figure 15), ci-dessous.

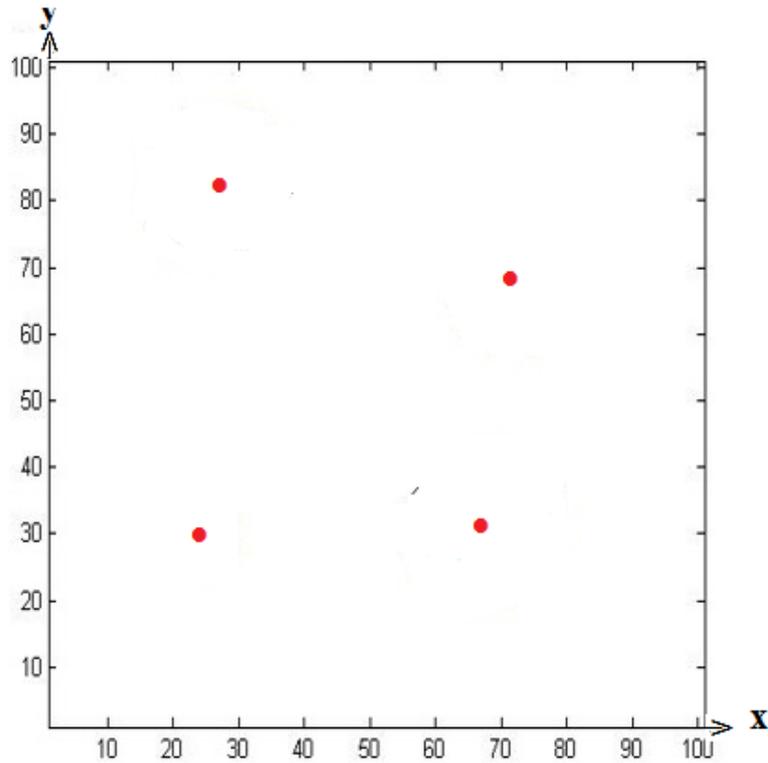


Figure 15: Prototype d'un CI avec 4 points rouges (pics thermiques) [38]

- b) **Application des conditions aux limites initiales (BC)** aux frontières du CI.
- c) **Simulation en transitoire** avec les quatre sources.
- d) **Enregistrement des résultats** de diffusion des quatre sources à différents instants.

Ces quatre points rouges vont évoluer jusqu'à devenir des sources de chaleur importantes (ou pics thermiques), (voir Figure 16) ci-dessous.

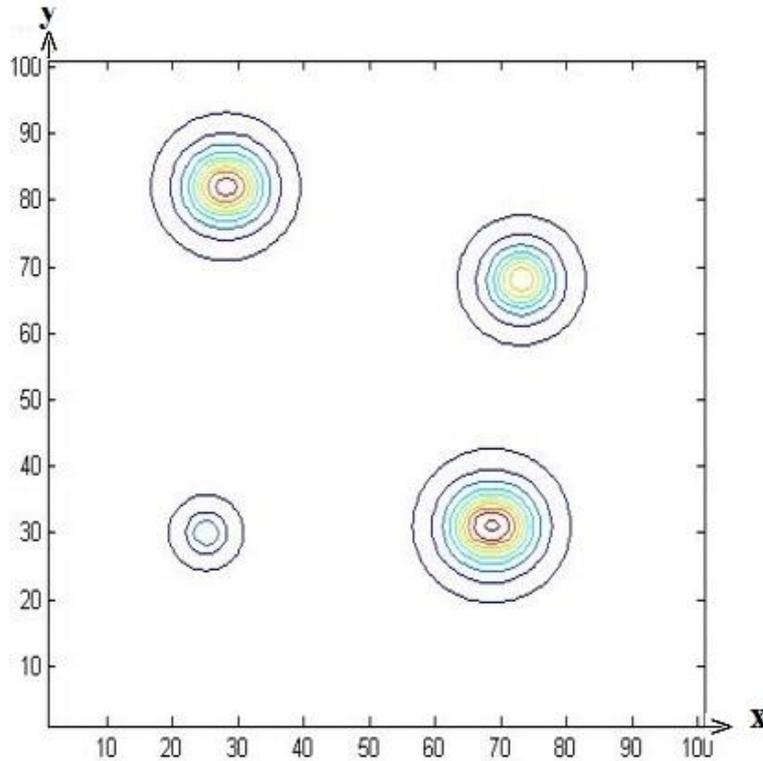


Figure 16: Expérimentation numérique -puce avec 4 points chauds (devenus pics de chaleur) [38]

3.3.2 Hypothèses :

Lors de la modélisation, on a pris en compte un ensemble d'hypothèses pour assurer une simulation qui nous donne des résultats pertinents, qui sont les suivantes :

- La température initiale sur le CI est de 25 °C, elle restera la même aux frontières des bords du CI et c'est aussi la valeur de la température ambiante dans la salle de laboratoire.
- La température maximale permise sur le CI après application de la puissance de chaleur T_{max} : 125 °C.
- À un instant $t_1=100 \mu s$, les températures aux niveaux des sources varient selon l'échelle de 25 °C à 77 °C lequel correspond à une échelle de couleur du bleu au rouge à droite de la Figure 18.
Si la température T atteint 98 °C, la surface du CI deviendra complètement rouge.
Si la température T est supérieure à T_{max} (125 °C), on signale à l'unité de contrôle thermomécanique au niveau de la jonction de faire un repositionnement des CI.
- Dans notre prototype de (CI), le nombre de sources de chaleur est fixé à quatre (4) sources distinctes dispersées dans quatre (4) zones pour les raisons suivantes :
 - Dans l'industrie de la fabrication électronique, on parle de zones ou cœurs dans le CI où il y a dissipation de chaleur. En général, les manufacturiers comme Intel sont intéressés à

savoir et contrôler la température pour refroidissement et suivre le monitoring de la température dans ces zones.

- En général, ils choisissent un nombre de zones divisible par quatre (4,8,16) pour couvrir les zones critiques situées dans les quatre directions conventionnelles
- Pour notre prototype on a choisi quatre car c'est le plus petit nombre optimal de sources qui va nous permettre d'implémenter un plus petit nombre optimal de capteurs possible.

3.3.3 Résultats : Tableaux des données, images thermiques

Comme résultats, NISA offre la possibilité de réaliser une série de snapshots décrivant l'état thermique du prototype (ou modèle) en spécifiant l'évolution dans le temps des conditions thermiques dans le CI tout en précisant le nombre de pics thermiques, leurs niveaux de températures maximums, leurs localisations et leurs tendances de fusionnement ou pas.

Chaque snapshot nous expose :

- ❖ **La liste des nœuds** avec la description des conditions de l'expérience du système (conditions des bornes, hypothèses, informations générales du système) (voir Tableau 3).
- ❖ **La liste des coordonnées** des nœuds (voir Tableau 4).

Cette liste présente les nœuds en spécifiant pour chacun :

- Le numéro
- Coordonnée selon l'axe des X
- Coordonnée selon l'axe des Y
- Coordonnée selon l'axe des Z

- ❖ **La liste des températures** des nœuds (Voir Tableau 5).

La liste des températures de tous les nœuds. Il présente pour chaque nœud :

- Le numéro
- La température correspondante (détectée à ce point)

- ❖ **Une image thermique de la surface du CI** avec un nombre de sources de chaleur en évolution thermique entourées de cercles isothermes avec la possibilité de fusionnement de quelques-uns. (Voir figures de 17 à 26).

Tableau 3 : A-Liste des numéros des nœuds [38]

numeros des noeuds - Notepad													
File Edit Format View Help													
ELEMENT DEFINITIONS (*ELEMENT DATA GROUP)													
ELEMENT NUMBER	MATERIAL INDEX	NSRL	REAL INDEX	CONST ANISOTROPIC INDEX	- - - E L E M E N T N O D E N U M B E R S - - -								
1	1	1	1	0	1	2	103	102	10202	10203	10304	10303	
2	1	1	1	0	2	3	104	103	10203	10204	10305	10304	
3	1	1	1	0	3	4	105	104	10204	10205	10306	10305	
4	1	1	1	0	4	5	106	105	10205	10206	10307	10306	
5	1	1	1	0	5	6	107	106	10206	10207	10308	10307	
6	1	1	1	0	6	7	108	107	10207	10208	10309	10308	
7	1	1	1	0	7	8	109	108	10208	10209	10310	10309	
8	1	1	1	0	8	9	110	109	10209	10210	10311	10310	
9	1	1	1	0	9	10	111	110	10210	10211	10312	10311	
10	1	1	1	0	10	11	112	111	10211	10212	10313	10312	
11	1	1	1	0	11	12	113	112	10212	10213	10314	10313	
12	1	1	1	0	12	13	114	113	10213	10214	10315	10314	
13	1	1	1	0	13	14	115	114	10214	10215	10316	10315	
14	1	1	1	0	14	15	116	115	10215	10216	10317	10316	
15	1	1	1	0	15	16	117	116	10216	10217	10318	10317	
16	1	1	1	0	16	17	118	117	10217	10218	10319	10318	
17	1	1	1	0	17	18	119	118	10218	10219	10320	10319	
18	1	1	1	0	18	19	120	119	10219	10220	10321	10320	
19	1	1	1	0	19	20	121	120	10220	10221	10322	10321	
20	1	1	1	0	20	21	122	121	10221	10222	10323	10322	
21	1	1	1	0	21	22	123	122	10222	10223	10324	10323	
22	1	1	1	0	22	23	124	123	10223	10224	10325	10324	
23	1	1	1	0	23	24	125	124	10224	10225	10326	10325	
24	1	1	1	0	24	25	126	125	10225	10226	10327	10326	
25	1	1	1	0	25	26	127	126	10226	10227	10328	10327	
26	1	1	1	0	26	27	128	127	10227	10228	10329	10328	
27	1	1	1	0	27	28	129	128	10228	10229	10330	10329	
28	1	1	1	0	28	29	130	129	10229	10230	10331	10330	
29	1	1	1	0	29	30	131	130	10230	10231	10332	10331	
30	1	1	1	0	30	31	132	131	10231	10232	10333	10332	
31	1	1	1	0	31	32	133	132	10232	10233	10334	10333	
32	1	1	1	0	32	33	134	133	10233	10234	10335	10334	
33	1	1	1	0	33	34	135	134	10234	10235	10336	10335	
34	1	1	1	0	34	35	136	135	10235	10236	10337	10336	

Tableau 4 : B-Liste des coordonnées des nœuds[38]

Coordonnees des 102010noeuds - Notepad				
File Edit Format View Help				
- NODAL COORDINATES (*NODES DATA GROUP) -				
NODE NO.	COORD SYS. ID.	COORDINATES		
1	0	0.00000E+00	0.00000E+00	0.00000E+00
2	0	5.00000E+01	0.00000E+00	0.00000E+00
3	0	1.00000E+02	0.00000E+00	0.00000E+00
4	0	1.50000E+02	0.00000E+00	0.00000E+00
5	0	2.00000E+02	0.00000E+00	0.00000E+00
6	0	2.50000E+02	0.00000E+00	0.00000E+00
7	0	3.00000E+02	0.00000E+00	0.00000E+00
8	0	3.50000E+02	0.00000E+00	0.00000E+00
9	0	4.00000E+02	0.00000E+00	0.00000E+00
10	0	4.50000E+02	0.00000E+00	0.00000E+00
11	0	5.00000E+02	0.00000E+00	0.00000E+00
12	0	5.50000E+02	0.00000E+00	0.00000E+00
13	0	6.00000E+02	0.00000E+00	0.00000E+00
14	0	6.50000E+02	0.00000E+00	0.00000E+00
15	0	7.00000E+02	0.00000E+00	0.00000E+00
16	0	7.50000E+02	0.00000E+00	0.00000E+00
17	0	8.00000E+02	0.00000E+00	0.00000E+00
18	0	8.50000E+02	0.00000E+00	0.00000E+00
19	0	9.00000E+02	0.00000E+00	0.00000E+00
20	0	9.50000E+02	0.00000E+00	0.00000E+00
21	0	1.00000E+03	0.00000E+00	0.00000E+00
22	0	1.05000E+03	0.00000E+00	0.00000E+00
23	0	1.10000E+03	0.00000E+00	0.00000E+00
24	0	1.15000E+03	0.00000E+00	0.00000E+00
25	0	1.20000E+03	0.00000E+00	0.00000E+00
26	0	1.25000E+03	0.00000E+00	0.00000E+00
27	0	1.30000E+03	0.00000E+00	0.00000E+00
28	0	1.35000E+03	0.00000E+00	0.00000E+00
29	0	1.40000E+03	0.00000E+00	0.00000E+00
30	0	1.45000E+03	0.00000E+00	0.00000E+00
31	0	1.50000E+03	0.00000E+00	0.00000E+00
32	0	1.55000E+03	0.00000E+00	0.00000E+00
33	0	1.60000E+03	0.00000E+00	0.00000E+00
34	0	1.65000E+03	0.00000E+00	0.00000E+00
35	0	1.70000E+03	0.00000E+00	0.00000E+00

Tableau 5 : C-Liste des températures des nœuds[38]

snapshot 17 - Notepad

File Edit Format View Help

***** TEMPERATURE SOLUTION *****

TIME = 3.0000E+05 TIME STEP NUMBER = 17

NODE	TEMPERATURE	NODE	TEMPERATURE	NODE	TEMPERATURE	NODE	TEMPERATURE
98533	2.5188577E+01	98534	2.5272073E+01	98535	2.5393631E+01	98536	2.5571225E+01
98537	2.5831789E+01	98538	2.6216037E+01	98539	2.6786162E+01	98540	2.7638400E+01
98541	2.8923969E+01	98542	3.0885253E+01	98543	3.3916381E+01	98544	3.8759314E+01
98545	4.3548239E+01	98546	4.6228530E+01	98547	4.7118706E+01	98548	4.6228520E+01
98549	4.3548218E+01	98550	3.8759282E+01	98551	3.3916339E+01	98552	3.0885201E+01
98553	2.8923906E+01	98554	2.7638326E+01	98555	2.6786076E+01	98556	2.6215937E+01
98557	2.5831672E+01	98558	2.5571086E+01	98559	2.5393462E+01	98560	2.5271865E+01
98561	2.5188317E+01	98562	2.5130730E+01	98563	2.5090930E+01	98564	2.5063359E+01
98565	2.5044220E+01	98566	2.5030912E+01	98567	2.5021646E+01	98568	2.5015189E+01
98569	2.5010689E+01	98570	2.5007558E+01	98571	2.5005388E+01	98572	2.5003901E+01
98573	2.5002904E+01	98574	2.5002270E+01	98575	2.5001920E+01	98576	2.5001808E+01
98577	2.5002335E+01	98578	2.5002443E+01	98579	2.5002775E+01	98580	2.5003357E+01
98581	2.5004234E+01	98582	2.5005470E+01	98583	2.5007155E+01	98584	2.5009402E+01
98585	2.5012354E+01	98586	2.5016187E+01	98587	2.5021103E+01	98588	2.5027337E+01
98589	2.5035139E+01	98590	2.5044767E+01	98591	2.5056455E+01	98592	2.5070386E+01
98593	2.5086643E+01	98594	2.5105156E+01	98595	2.5125658E+01	98596	2.5147633E+01
98597	2.5170304E+01	98598	2.5192647E+01	98599	2.5213449E+01	98600	2.5231416E+01
98601	2.5245312E+01	98602	2.5254106E+01	98603	2.5257118E+01	98604	2.5254108E+01
98605	2.5245316E+01	98606	2.5231423E+01	98607	2.5213459E+01	98608	2.5192662E+01
98609	2.5170327E+01	98610	2.5147667E+01	98611	2.5125707E+01	98612	2.5105228E+01
98613	2.5086745E+01	98614	2.5070533E+01	98615	2.5056665E+01	98616	2.5045066E+01
98617	2.5035565E+01	98618	2.5027943E+01	98619	2.5021965E+01	98620	2.5017412E+01
98621	2.5014097E+01	98622	2.5011882E+01	98623	2.5010686E+01	98624	2.5010503E+01
98625	2.5011412E+01	98626	2.5013605E+01	98627	2.5017421E+01	98628	2.5023399E+01
98629	2.5032357E+01	98630	2.5045512E+01	98631	2.5064654E+01	98632	2.5092409E+01
98633	2.5132629E+01	98634	2.5190976E+01	98635	2.5275804E+01	98636	2.5399507E+01
98637	2.5580592E+01	98638	2.5846916E+01	98639	2.6240837E+01	98640	2.6827575E+01
98641	2.7709133E+01	98642	2.9048297E+01	98643	3.1111278E+01	98644	3.4351240E+01
98645	3.9435317E+01	98646	4.4390139E+01	98647	4.7118677E+01	98648	4.7990406E+01
98649	4.7118669E+01	98650	4.4390123E+01	98651	3.9435293E+01	98652	3.4351207E+01
98653	3.1111237E+01	98654	2.9048246E+01	98655	2.7709072E+01	98656	2.6827502E+01
98657	2.6240749E+01	98658	2.5846808E+01	98659	2.5580459E+01	98660	2.5399339E+01
98661	2.5275589E+01	98662	2.5190697E+01	98663	2.5132264E+01	98664	2.5091925E+01
98665	2.5064008E+01	98666	2.5044646E+01	98667	2.5031192E+01	98668	2.5021831E+01

3.4 Analyse des Résultats

Pour analyser en détails l'évolution thermique dans le CI, il faut se référer aux snapshots réalisés auparavant et en choisir une snapshot optimale qui soit appropriée au contexte de la résolution de la problématique de la thèse. Ce snapshot doit nous présenter un puce avec à la surface quatre sources distinctes et indépendantes.

3.4.1 Choix de la valeur instantanée appropriée pour étude et analyse

-A- On évite les pics thermiques en fusionnement

On s'intéresse aux snapshots qui ne présentent pas les pics thermiques entrain de fusionner. Ceci parce qu'une fois les pics commencent à fusionner, on ne peut plus distinguer le nombre de sources de chaleur sur la surface du CI. On ne peut plus étudier, analyser et contrôler la dissipation thermique aux niveaux des pics individuellement d'une façon distincte (Voir Figure 17)

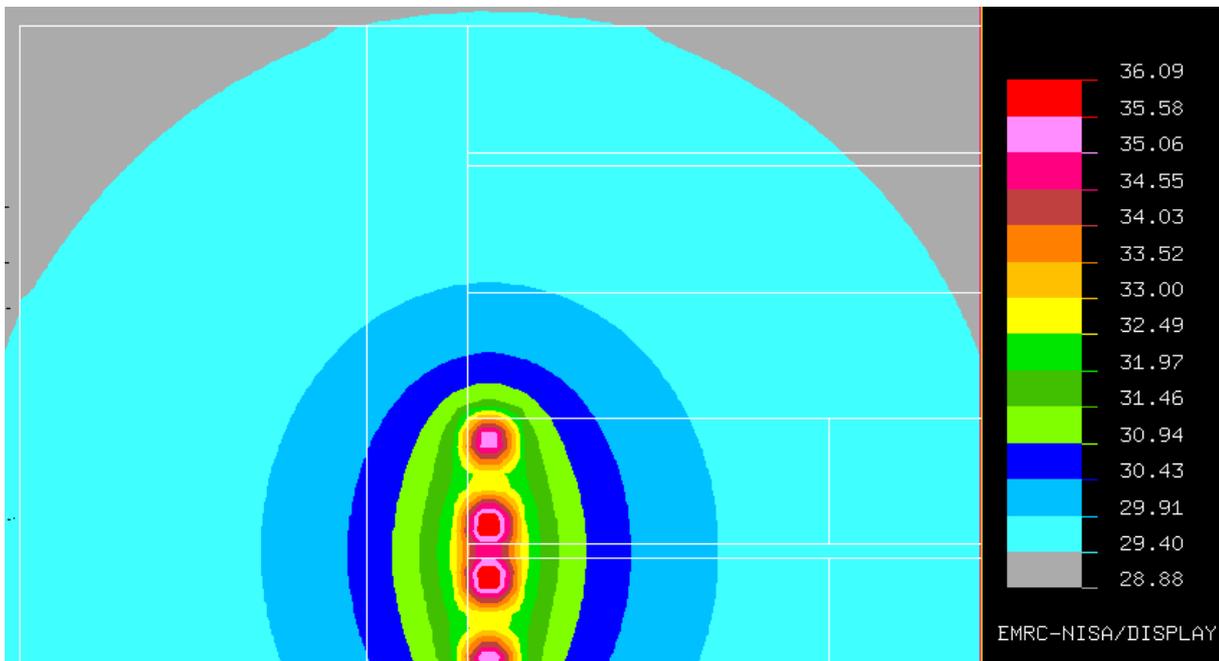


Figure 17. Cas de quatre sources de chaleurs qui fusionnent en un seule source

Dans cette figure, on peut voir la problématique de détection et la nécessité du « Choix approprié des valeurs instantanées » (de la snapshot) pour l'étude et l'analyse de la détection des pics thermiques dans le cas de plusieurs sources de chaleurs.

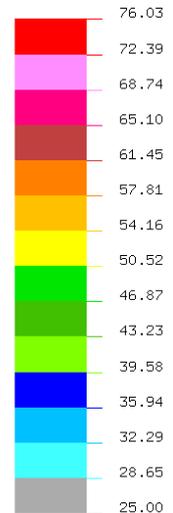
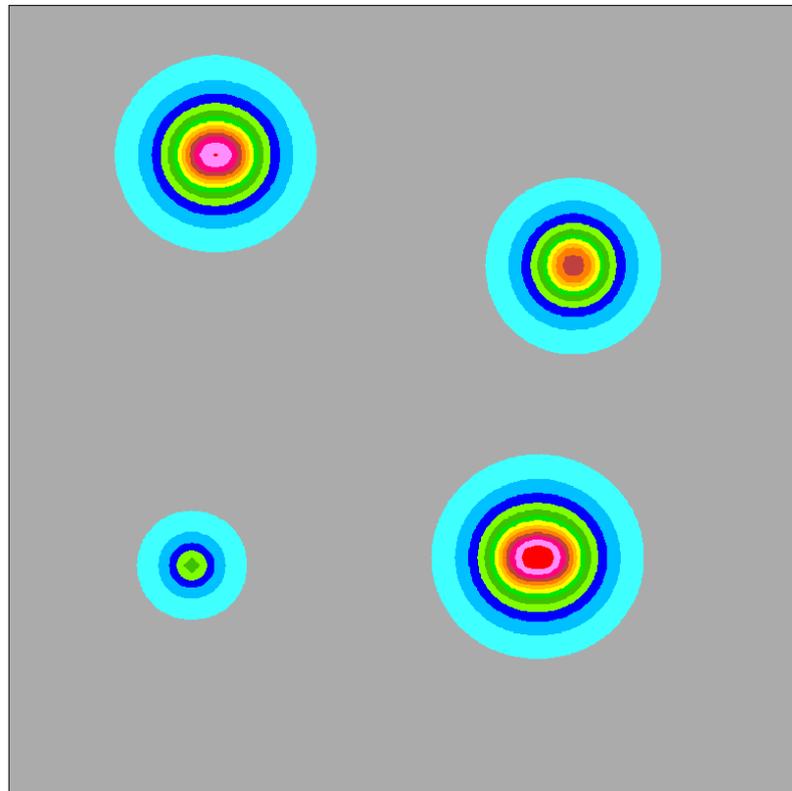
-B- On choisit la valeur instantanée appropriée pour l'étude : On sélectionne **l'instant approprié qui présente les valeurs maximales** instantanées des températures aux pics thermiques **et qui précède l'instant où les pics commencent à fusionner.**

Dans notre cas on a choisi **les valeurs instantanées appropriées (16)** (snapshot 16) (Voir Figure 18).

Les données obtenues à partir de la snapshot sont pris en considération comme étant les résultats de la simulation à analyser pour étudier la détection de sources chaleur multiples dans un (CI).

DISPLAY III - GEOMETRY MODELING SYSTEM (15.0.0) PRE/POST MODULE

TEMPERATURE
VIEW : 25.0
RANGE : 76.03304



CRANES SOFTWARE, INC., - DISPL

MAR/23/12 13:43:59



NISA

SNAPSHOT NUMBER = 16 AT TIME ZONE = 1.000E+06
Principal4eth7

Figure 18: les valeurs instantanées appropriées (16) sujets de l'étude [38]

3.4.2 Analyse des pics thermiques en évolution

Une fois on a appliqué une puissance de chaleur à chacun des quatre (4) points sur le CI quatre sources de chaleur surgissent à la surface qui vont évoluer en transmettant des vagues de chaleur tout autour. Ces vagues de chaleur se manifestent sous forme de cercles qui se succèdent graduellement tout en augmentant de diamètre et de taille. Ces cercles sont constitués de points isothermes. À chaque point de chaque cercle, la valeur de la température est la même. Ainsi, on les appelle : des cercles isothermes. Plus qu'un cercle isotherme s'éloigne de la source plus sa température est relativement moindre. Les couleurs des pics thermiques et des cercles isothermes qui l'entourent varient graduellement et proportionnellement avec les niveaux de température détectés et elles sont assignées respectivement selon une échelle standard d'intervalles de température prédéfinis par le simulateur NISA. A chaque couleur correspond un intervalle de niveaux de température (voir l'échelle de couleur située au côté droit de la Figure 18).

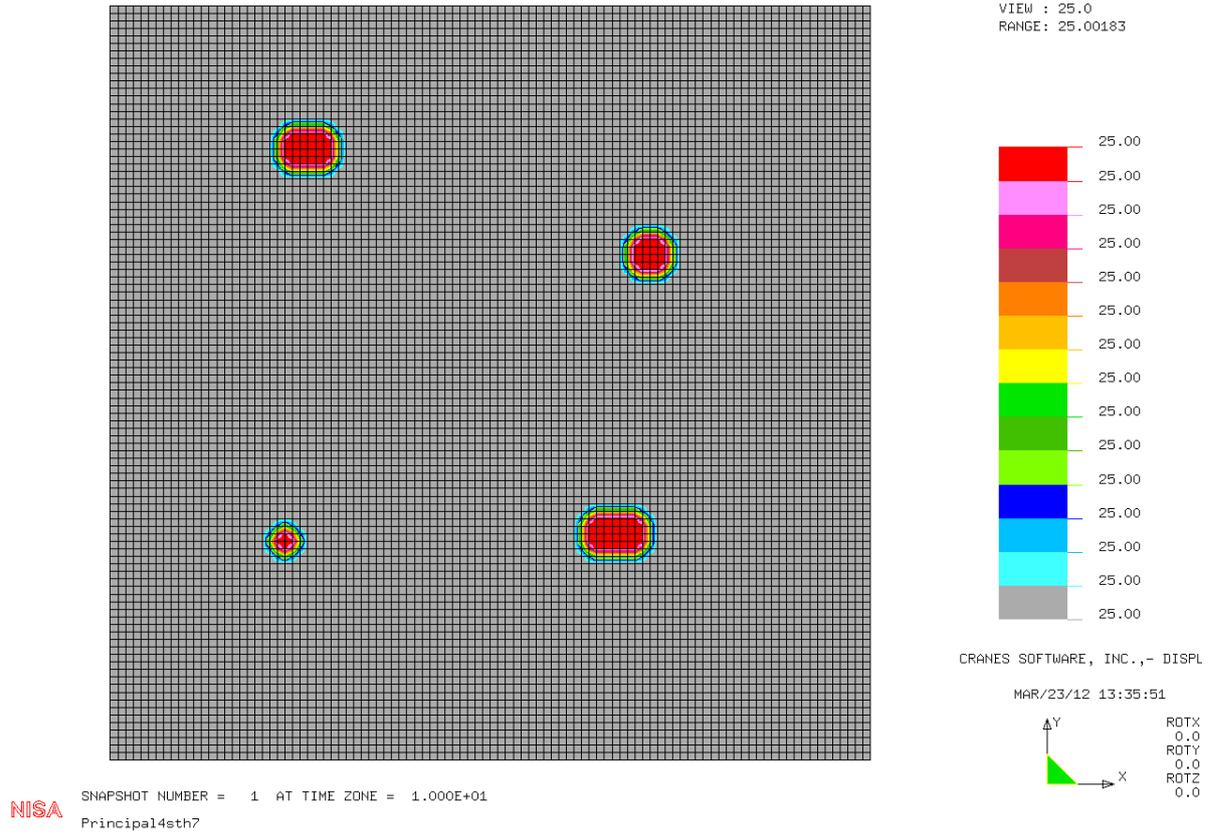


Figure 19: Modèle d'un circuit intégré en 2D [38]

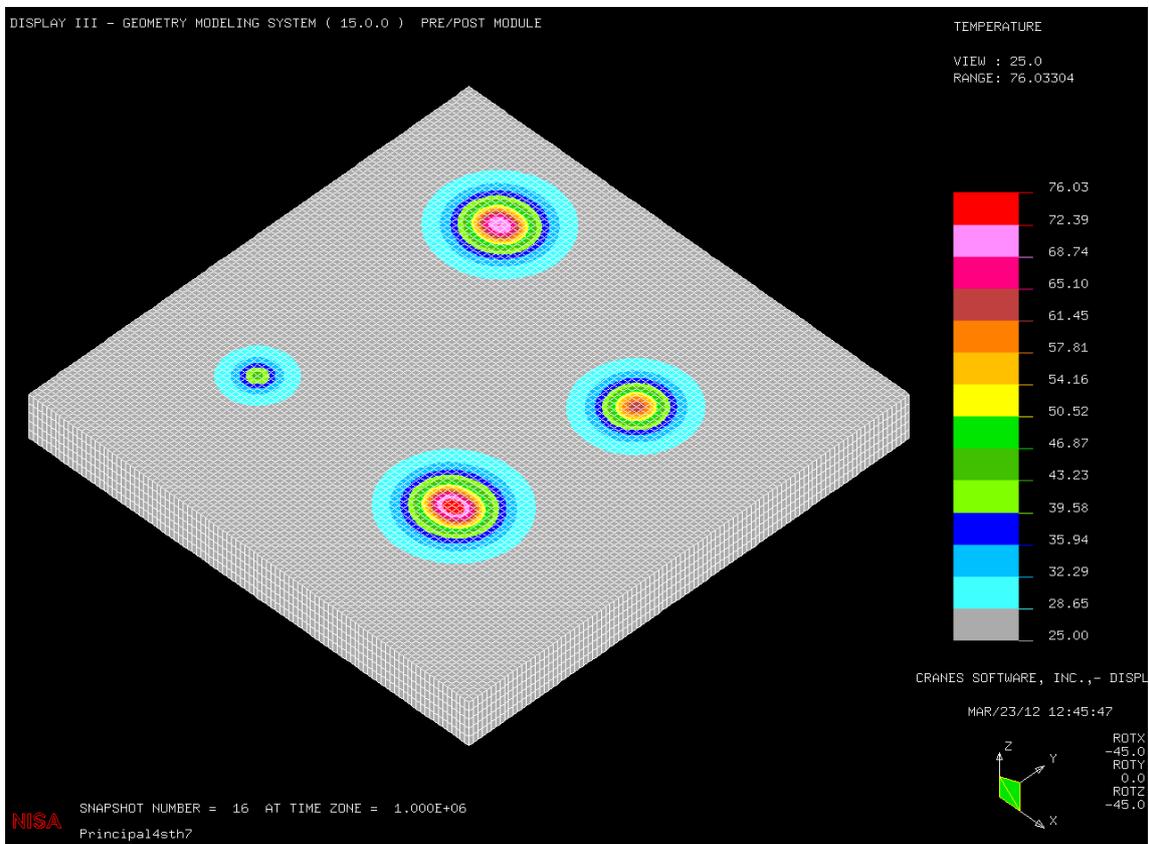
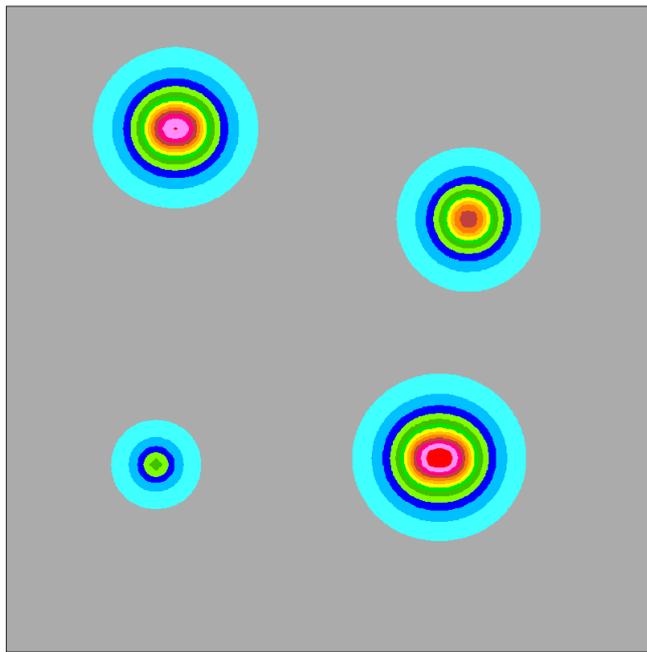


Figure 20 : Modèle d'un circuit intégré en 3D montrant ses couches de silicium superposées[38]

DISPLAY III - GEOMETRY MODELING SYSTEM (15.0.0) PRE/POST MODULE



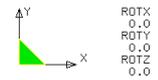
TEMPERATURE

VIEW : 25.0
RANGE : 76.03304



CRANES SOFTWARE, INC., - DISPL

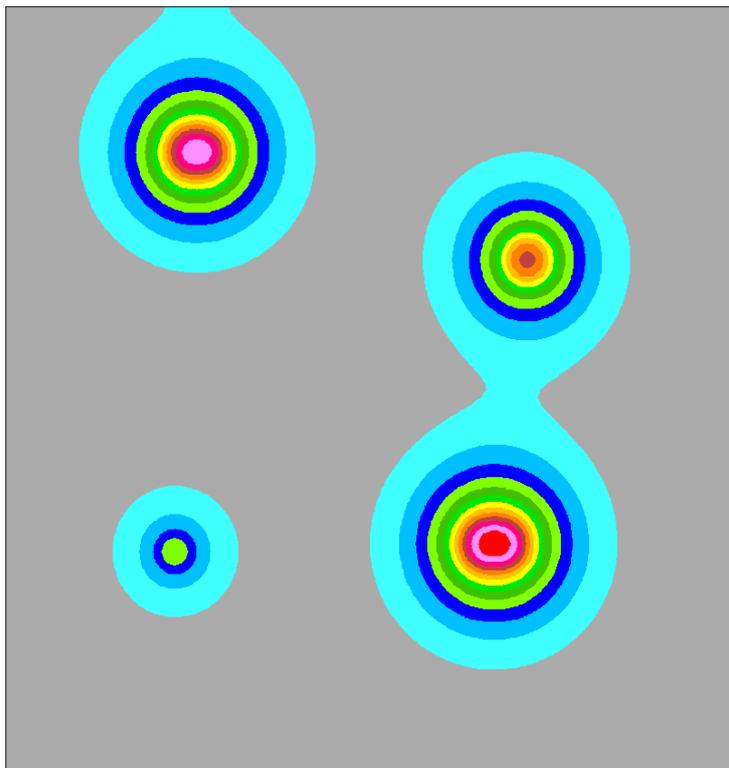
MAR/23/12 13:43:59



NISA SNAPSHOT NUMBER = 16 AT TIME ZONE = 1.000E+06
Principal4sth7

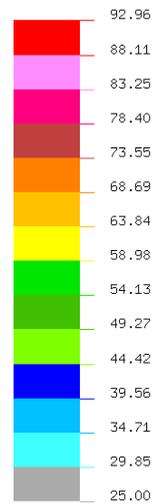
Figure 21: Snapshot sélectionnée pour l'étude[38]

DISPLAY III - GEOMETRY MODELING SYSTEM (15.0.0) PRE/POST MODULE



TEMPERATURE

VIEW : 25.0
RANGE : 92.96326



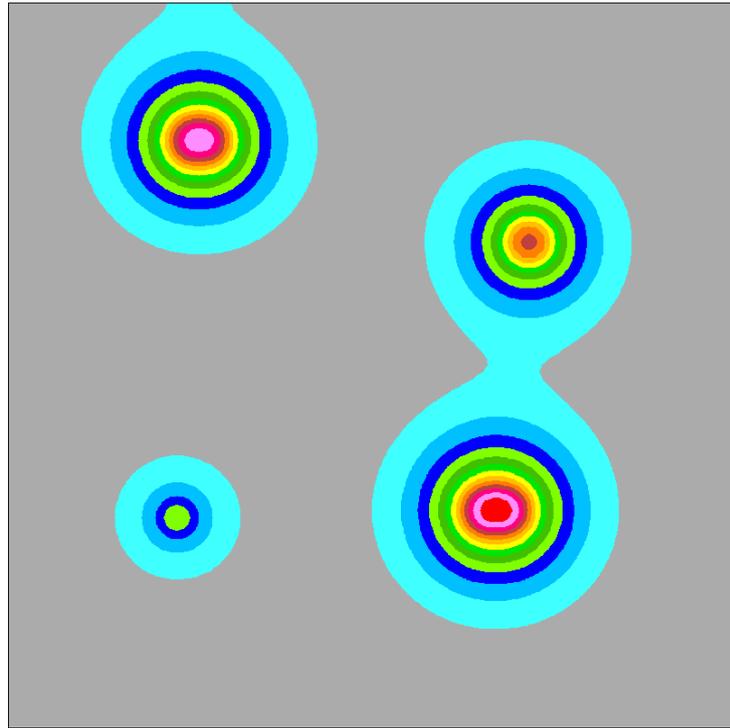
CRANES SOFTWARE, INC., - DISPL

MAR/23/12 21:43:18



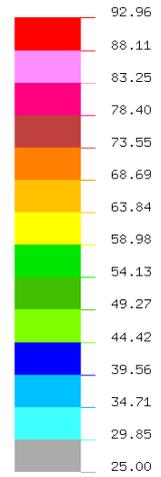
NISA SNAPSHOT NUMBER = 17 AT TIME ZONE = 2.000E+06
Principal4sth7

Figure 22 : Commencement de toucher des 2 sources de chaleur[38]



TEMPERATURE

VIEW : 25.0
RANGE: 92.96326



CRANES SOFTWARE, INC., - DISPL

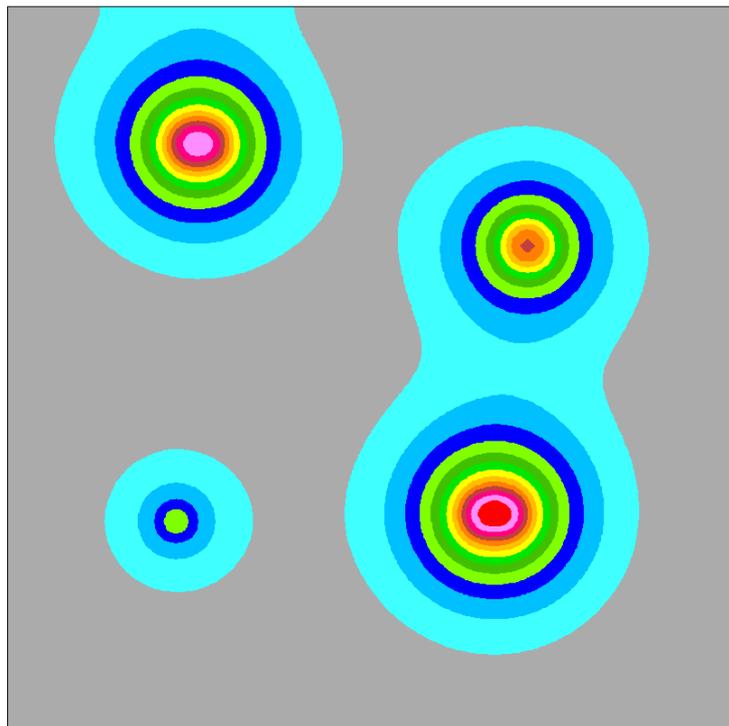
MAR/23/12 21:43:18



NISA

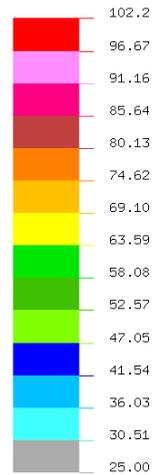
SNAPSHOT NUMBER = 17 AT TIME ZONE = 2.000E+06
Principal4sth7

Figure 23 : Commencement de fusionnement des deux (2) sources de chaleur[38]
On assiste à un déplacement de la source gauche vers l'extrémité du haut de la puce ce qui explique l'échappement de la chaleur du côté le plus chaud vers l'extrémité qui est relativement plus froide.



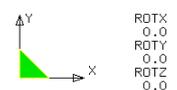
TEMPERATURE

VIEW : 25.0
RANGE: 102.182



CRANES SOFTWARE, INC., - DISPL

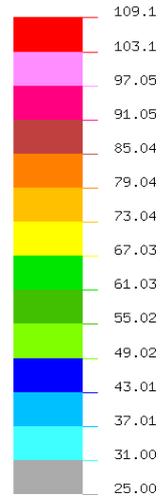
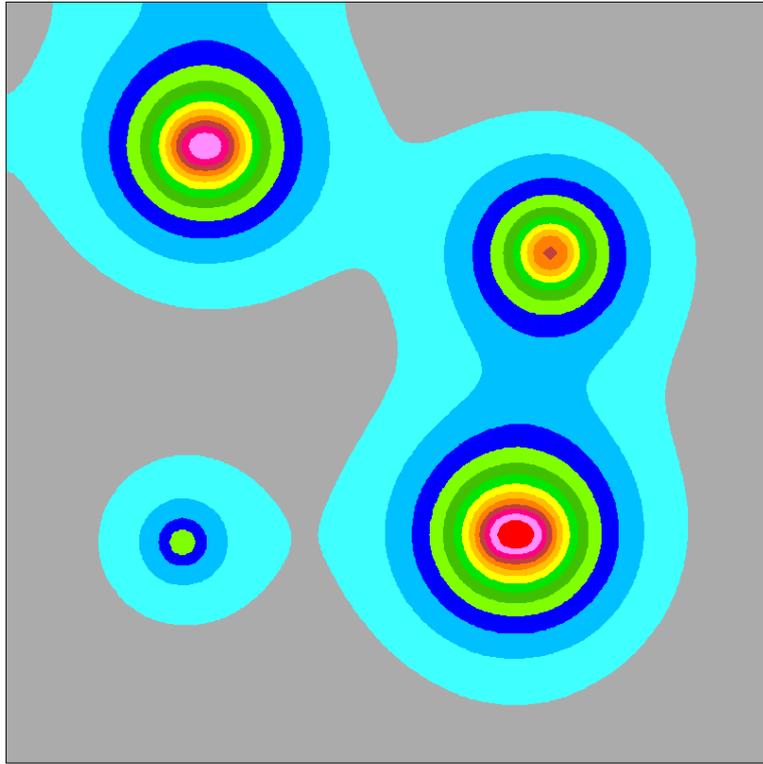
MAR/23/12 21:46:20



NISA

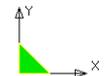
SNAPSHOT NUMBER = 18 AT TIME ZONE = 3.000E+06
Principal4sth7

Figure 24: Commencement de fusionnement des trois (3) sources en une seule (1) [38]



CRANES SOFTWARE, INC., - DISPL

MAR/23/12 21:48:25

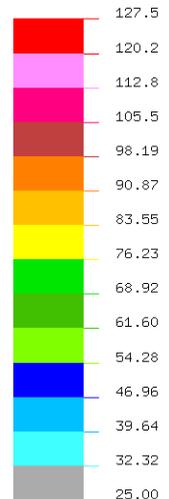
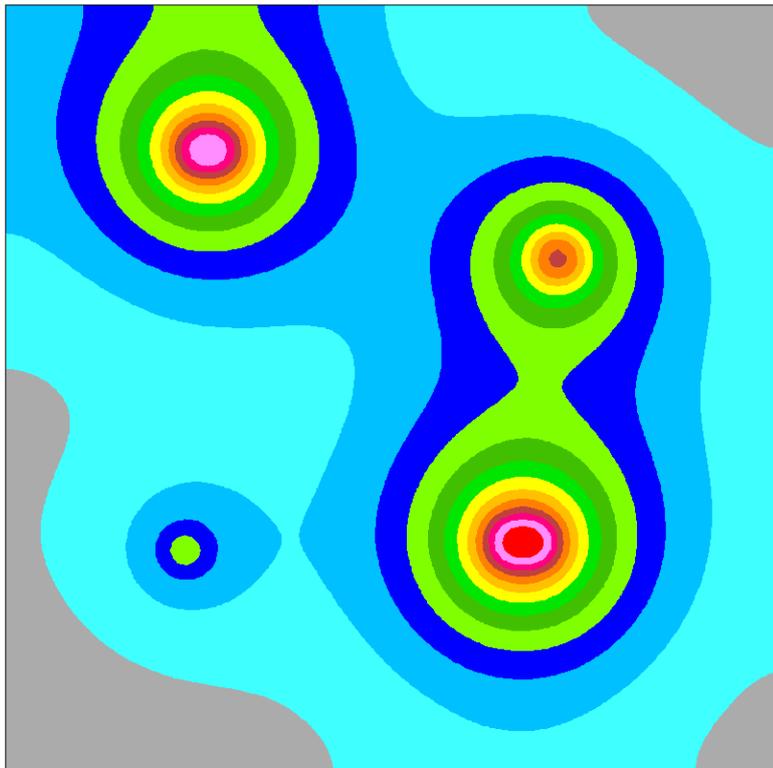


ROTX
0.0
ROTY
0.0
ROTZ
0.0

NISA

SNAPSHOT NUMBER = 19 AT TIME ZONE = 4.000E+06
Principal4sth7

Figure 25 : Commencement de fusionnement d quatre (4) sources en une seule (1) [38]



CRANES SOFTWARE, INC., - DISPL

MAR/23/12 21:51:26



ROTX
0.0
ROTY
0.0
ROTZ
0.0

NISA

SNAPSHOT NUMBER = 23 AT TIME ZONE = 8.000E+06
Principal4sth7

Figure 266: Les quatre sources de chaleur en cours de fusionnement en une seule (1) [38]

On peut voir la problématique de détection et la nécessité de configuration de cellules appropriées pour la détection des pics thermiques dans le cas de plusieurs sources de chaleurs (Voir Figure 26).

SS 16, (0/1500/500), (5000,1500,500)

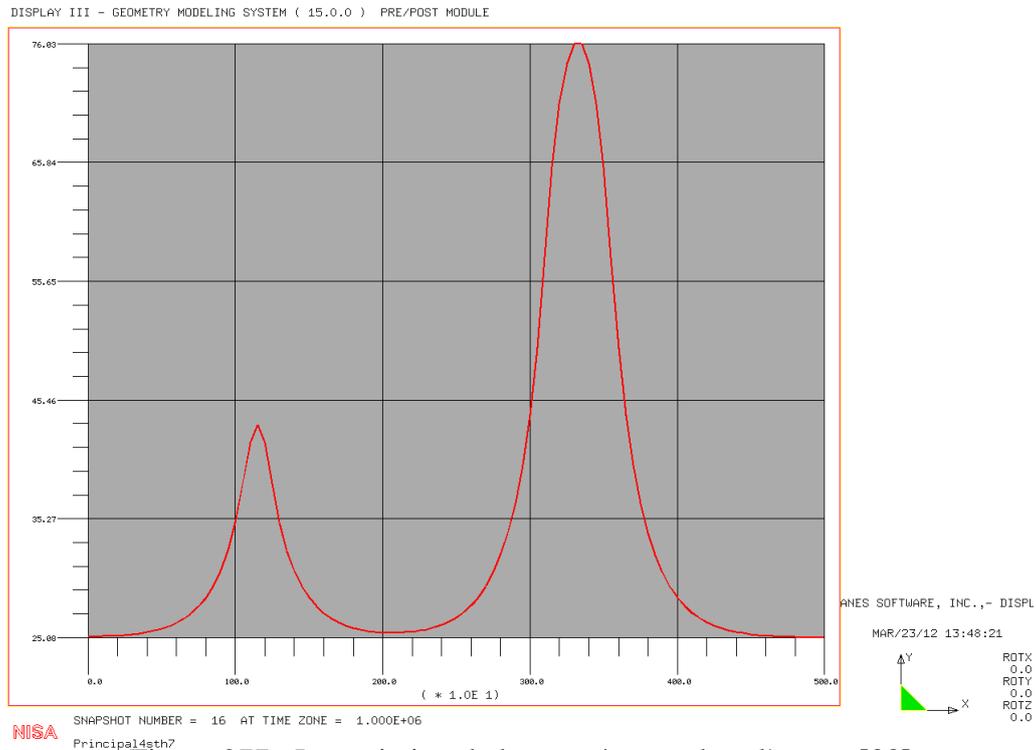


Figure 277 : La variation de la température dans l'espace[38]

L'augmentation de la température brusque dans l'espace augmente la contrainte thermomécanique qui endommage le CI.

3.5 Solutions algorithmiques explorées

3.5.1 Contexte de la problématique

D'après le modèle illustré à la figure 28, on a 10201(101x101) points dont on connaît leur positions (Xi, Yi) et leur température Ti appartenant à une surface rectangulaire (101 points x101 points).

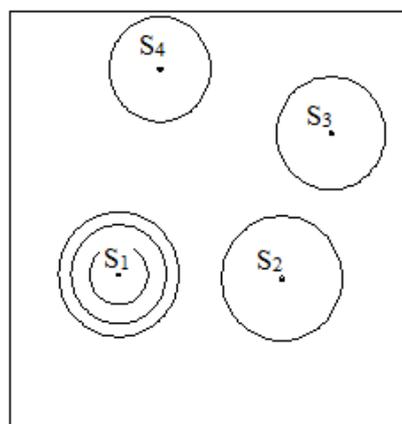


Figure 288: Schéma du prototype à modéliser. [38]

On veut déterminer dans un premier temps quels sont les points qui sont sources de chaleur. On peut avoir une source ou 2 ou 3 ...etc. Si une source (S_j) existe alors elle sera le centre des cercles isothermes concentriques » [33].

3.5.2 Objectif

Déterminer les (k) points sources de chaleurs les plus chauds (pics thermiques) qui sont les plus proches voisins du point le plus chaud dans tout le CI, y détecter les valeurs maximales de températures respectivement et les localiser sur la surface du CI. Ainsi on détecte toutes les sources de chaleur sur ce CI.

3.5.3 Tâches de la procédure générale à suivre

- 1-Lire le fichier de données parmi les (10201 points, $(X_i, Y_i), T_i$) ;
- 2-En se basant sur un raisonnement algorithmique appropriée détecter les (k) points sources de chaleur les plus chauds (S_j) parmi les (10201 points, $(X_i, Y_i), T_i$) et les localiser.

X_i : abscisse du point (i) sur la surface du CI, avec ($n = 0, 10000$)

Y_i : ordonnée du point (i) sur la surface du CI, avec ($n = 0, 10000$).

T_i : valeurs de la température au point (i) sur la surface du CI,

S_j : source de chaleur (j) ou Pic thermique, avec ($j = 1, k$)

Dans l'exemple la figure ci-dessus montre quatre sources : S_1, S_2, S_3 et S_4 ; avec ($j = 1, 4$).

3.5.4 Solutions Algorithmiques explorées

On a exploré quatre (4) solutions algorithmiques :

- 1-** L'algorithme du k plus proche voisin (KPPV-SODA).
- 2-** L'algorithme MSTPDA basé sur recherche de la Matrice Contour.
- 3-** L'algorithme MHSODA, basé sur le balayage thermique de la surface du CI.
- 4-** L'algorithme (GDSCAN) de balayage thermique basé sur le GDS.

Chapitre 4 : Algorithme du k plus proche voisin (KPPV- SODA)

4.1 Introduction de la méthode de recherche des K plus proches voisins :

La recherche de voisinage est utilisée dans de nombreux domaines, telle la reconnaissance de formes, le *clustering*, l'approximation de fonctions, la prédiction de séries temporelles et même les algorithmes de compression (recherche d'un groupe de données le plus proche possible du groupe de données à compresser pour minimiser l'apport d'information).

La méthode de recherche des K plus proches voisins est utilisée par les chercheurs dans plusieurs domaines pour trouver des solutions à leurs problématiques [29].

Chaque chercheur bâtit un algorithme de KPPV approprié au domaine de sa recherche en respectant les principes de la méthode du KPPV. Le paramètre le plus important sur le lequel est basée cette méthode est la distance (d) qui sépare un point requête et un point d'intérêt. D'où il s'avère important de bien la définir, la calculer et la représenter [30].

4.2 KPPV-SODA pour la détection des pics thermiques dans les CI :

Dans notre cas on va appeler KPPV-SODA (KPPV Sources of heat Detection Algorithm) l'algorithme qui permet la détection des pics thermiques en se basant sur la méthode de recherche des K plus proches voisins (KPPV / KNN).

Les points requêtes sont les « pics thermiques » et les points d'intérêts sont tous les points sur la surface du CI qui sont les plus proches voisins susceptibles d'être aussi des Pics thermiques. On définit (d) comme la distance euclidienne (en mm) qui sépare un Pic thermique déjà détecté est vérifié en tant que source de chaleur) et un point voisin susceptible d'être aussi un pic thermique. (d) étant différent de zéro et supérieur ou égale à d_{min} . On suppose (d_{min}) comme étant définie, fixée préalablement et désigne «la distance minimum » la plus petite entre deux pics thermiques. Cette hypothèse garantit une expérimentation numérique valide du système en mode transitoire [31].

Ayant déjà détecté un Pic thermique on peut découvrir plusieurs autres Pics thermiques en appliquant la méthode de KPPV sur chaque point existant dans l'espace euclidien de la surface du CI. On peut vérifier que chaque pic thermique est entouré de cercles isothermes et qu'un maximum de température y est détecté. Les cercles isothermes sont détectés et vérifiés par un long calcul de trigonométrie appliqué sur les points détectés. Ces longs calculs d'équations et de fonction vérifiées et validées dans le domaine de détection thermique (tel que la transformée de Hough ou la fonction de Green ou même le GDS), nous permettent de localiser avec précision le pic thermique local dans une zone donnée du CI.

On présente dans ce qui suit les détails de l'application de la méthode du KPPV dans la détection thermique et de l'implémentation de l'algorithme (KPPV-SODA) pour résoudre la problématique de détection de plusieurs sources de chaleur dans les circuits intégrés (CI) :

- Prototype conçu pour vérification et validation de l'algorithme (KPPV-SODA) voir figure 30.
- Pseudocode de l'algorithme (KPPV-SODA) (Voir Figure 31).
- Diagramme de flux de l'algorithme (KPPV-SODA) (Voir Figure 32).

4.2.1 Prototype conçu pour vérification et validation de l'algorithme (KPPV-SODA)

Dans cette section on introduit le prototype du modèle sur lequel la méthode algorithmique du KPPV (K Plus Proche Voisins) va être appliquée pour résoudre la détection de multi-sources de chaleurs sur un CI.

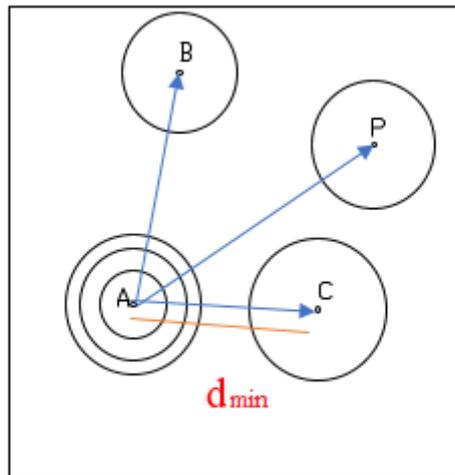


Figure 29 : Schéma du prototype à modéliser [38]

La surface de CI inclut quatre pics thermiques : A, B, C et P.

Déterminer les (k) points les plus chaud (pics thermiques voisins du point le plus chaud (A) ($T_A = T_{Max}$) dans la surface du (CI) situés à une distance $d > d_{min}$.

A-Hypothèses

- On se propose de fixer une distance minimum (d_{min}) qui sépare deux (2) sources de chaleurs. On peut se poser la question suivante : Pourquoi la distance d_{min} ?
La réponse est simple : c'est juste pour être sûr de ne pas prendre des points appartenant à la même source. Ainsi les cercles isothermes des deux sources de chaleur ne se touchent pas (pas de fusionnement de sources).
- On suppose qu'on a 10201 capteurs, chaque nœud y installé un capteur RO.

B-Prototype de l'étude

Notre prototype qui va servir pour l'étude est celui décrit dans la figure 30 .Il se refere aux donnees résultants de la simulation effectuée précédemment . Ainsi,Il possède les même caractéristiques et contexte problématique du modèle décrit précédemment dans la figure 20 , la figure 28 et inspiré de la figure 29 .

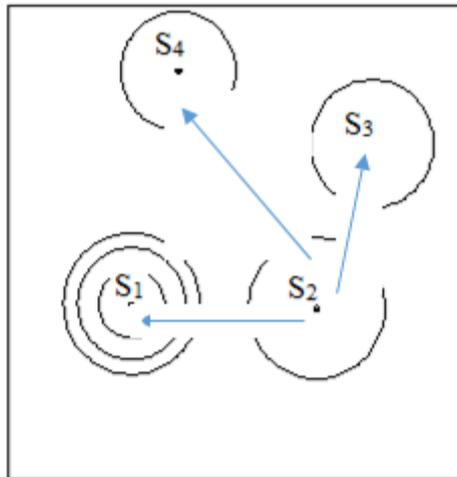


Figure 30 : Schéma du prototype de l'étude de l'algorithme (KPPV-SODA)

C-Tâches de la procédure générale à suivre de l'algorithme (KPPV-SODA)

La procédure générale à suivre de l'application de l'algorithme (KPPV-SODA) diffère de celle énoncée à la section (5.3.2) par la particularité du paradigme de la méthode algorithmique du KPPV.

Généralement, on a procédé par les étapes suivantes comme suit :

- 1) **Lire le fichier de données** (10201 points, $(X_i, Y_i), T_i$) ;
- 2) **Extraire le point le plus chaud (A)** sur la surface du CI : $(A, (X_A, Y_A), T_A = \text{Max}(T_i, i=1,10201))$;
- 3) **Déterminer les K Plus Proches Voisins de (A)**, au niveau température (T), qui sont Situés à une distance ($d > d_{\min}$) ;
- 4) Lister les K plus proches voisins de A ($P_j, (X_{pj}, Y_{pj}), T_{pj}$) qui sont des pics thermiques. Dans l'exemple de la figure 26, ce sont : B, C et P.

4.2.2 Pseudocode de l'algorithme (KPPV-SODA)

Initiation :

$N_s = 0$ (N_s : nombre de sources de chaleur)

Designier « N_{Max} le nombre maximum de sources de chaleur tolérable sur le CI »

Designier « d_{min} qui sépare deux (2) sources de chaleurs »

Déterminer les (k+1) sources de chaleurs

1-lire le fichier de données (NISA) des nœuds (points) $N_i (x_i, y_i)$, T_i , avec ($i=1$ à 10201)

2-Extraire le point le plus chaud à la surface du CI : A (X_A, Y_A), $T_A = T_{Max}$

parmi tous les T_i de tous les nœuds N_i avec ($i=1$ à 10201)

- $N_s = N_s + 1$

3-Extraire les K plus proches voisins de A au niveau de T^0 situés à une distance $< d_{min} >$

a-déterminer les KPPVI Isothermes de $P_j (j=1, k)$

b-vérifier si les KPPVI $\in C (P_j, R \leq d_{min}/2)$

c-Si <oui> $P_j \equiv S_{j+1}$, P_j est une source de chaleur, elle devient S_{j+1}

c₁- $k = k + 1$

c₂- $N_s = K + 1$ (Déduire le nombre total de sources de chaleur)

c₃-verifier si $N_s > N_{Max}$ (*Décider si on doit mettre le CI sur OFF pour qu'il ne s'endommage pas tout en se basant surtout sur le nombre maximum de sources tolérable*).

si oui

Lister

-les K plus proches voisins de A ($P_j (X_{pj}, Y_{pj}), T_{pj}$).

-tous les (k+1) sources de chaleur, leurs coordonnées

et les valeurs de température aux niveaux de ces sources.

Finir (**FIN**).

mettre le CI sur (**OFF**)

si non

laisser le CI sur (**ON**)

Répéter étape (3.a).

d- Si <non> P_j n'est pas une source de chaleur

Répéter étape (3.a).

Figure 31:pseudocode de l'algorithme KPPV-SODA [38]

4.2.3 Diagramme de flux de l'algorithme (KPPV-SODA) :

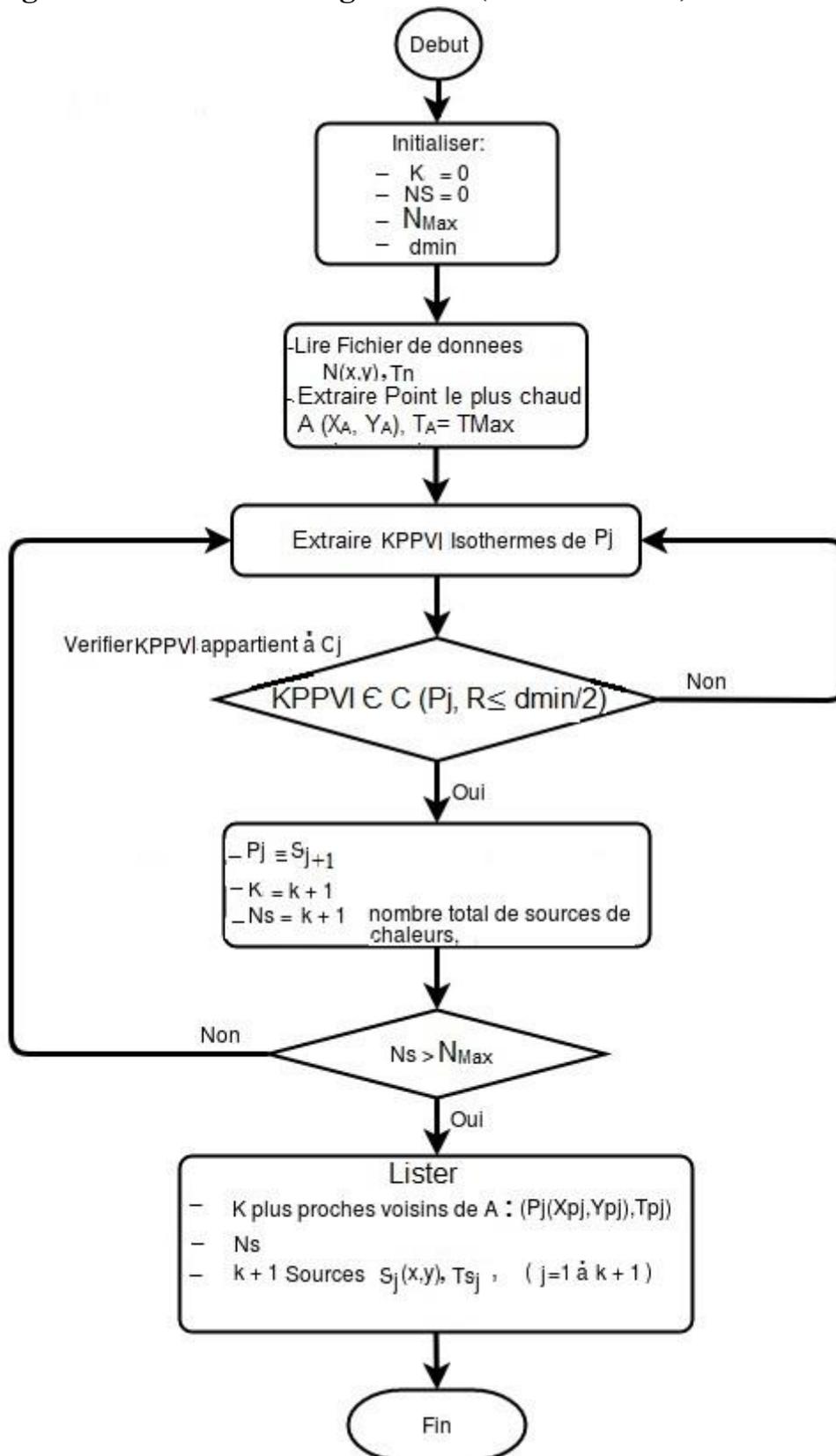


Figure 32: Diagramme de flux de l'algorithme KPPV-SODA [38]

4.3 Analyse et discussion de l'application de l'algorithme (KPPV-SODA)

L'algorithme (KPPV-SODA) s'avère applicable et efficace sur le plan algorithmique et logiciel en tant qu'algorithme générique qui peut tester n'importe quel point du CI s'il est source de chaleur ou pas. Mais la faiblesse de cet algorithme est le fait suivant : pour connaître les plus proches voisins pics d'un « pic thermique » détecté déjà sur la surface du CI on est obligé de faire un nombre très excessif de calcul et de comparaisons de distances séparant ce pic au reste de tous les points constituant (10200) la surface du CI. On doit aussi répéter ceci pour chaque point du CI vis-à-vis de tout le reste des points du CI. Cette multitude excessive d'opérations et de calculs va consommer trop de temps d'exécution et de calculs ce qui rend l'application de l'algorithme (KPPV-SODA) trop lente à appliquer avec des délais d'exécution et de réponses trop longs, non efficace et être irréalisable face à une vitesse relativement croissante du réchauffement du CI.

Dans le futur, des recherches pour introduire des hypothèses et des conditions prédéfinies pourraient rendre plus efficace et réaliste l'application de l'algorithme (KPPV-SODA) afin de résoudre la problématique de détection de plusieurs sources de chaleur dans les circuits intégrés (CI).

Chapitre 5 : Algorithme basé sur la matrice contour (MSTPDA)

5.1 Introduction :

MSTPDA (Multi-Source Temperature Peak Detection Algorithm) est une méthode Algorithmique qui aboutit à la détection "de plusieurs sources de chaleur" à la surface d'un CI en se basant essentiellement sur l'algorithme de recherche de la Matrice Contour.

Elle est plus complexe que celle qui détermine "une seule source" de chaleur, nommée STPDA" et qui repose seulement sur le « GDS ». On suppose l'utilisation des capteurs de type RO.

En effet cette méthode inclut deux méthodes algorithmiques qui sont respectivement les suivantes:

1- Algorithme MACA de "Recherche de Contour de Matrice " qui aboutit à " l'obtention de la Matrice Contour MCT"

2- Algorithme EUNA de "Recherche du nombre d'inclusions (ou nombre d'Euler) " qui aboutit à l'obtention du « nombre d'inclusions(r) », dont on en déduit le nombre sources de chaleur.

5.2 Recueil, organisation et affichage des données

5.2.1 Modèle ou prototype sujet de l'étude

Dans cette étude on a considéré le même modèle conçu précédemment (Voir Figure 28), avec l'hypothèse importante suivante : le nombre de capteurs de capteurs implémentés sur la surface du CI est 1021 capteurs lors du scan (balayage) et on les interroge tous (on les met en marche (ON)) rapidement, progressivement, une à la fois et en un temps relativement court.

5.2.2 Recueil des données

Après l'étape d'expérimentation, cette étape consiste à recueillir les informations obtenues par "scan" complet de la surface du CI. En effet, On a configuré la surface du CI de façon qu'elle soit de forme carré, divisée en 10201 nœuds. Pour "chaque nœud", on a relevé les données suivantes:

- numéro de nœud (i)
- coordonnées du nœud selon les 3 axes (X_i , Y_i , Z_i)
- la valeur de température détectée au niveau de ce nœud.

5.2.3 Organisation et représentations des données

Toutes les données citées précédemment sont organisées et représentées dans un grand tableau GT dont les cellules sont configurées de la même façon que la configuration des cellules de la surface du CI dans le modèle. Les mêmes données correspondantes aux cellules (ou nœuds) de la surface CI (numéro, coordonnées, températures) sont rapportées respectivement dans les cellules du tableau générale GT symboliquement représenté ainsi :

Tableau GT de (100x100) contenant tous les données relatives à chaque nœud (les coordonnées (x , y), un numéro de nœud ,la température a ce nœud)

$$\begin{pmatrix} (x_0, y_{100}), 91810, T_{0,100}^0 & \dots & \dots & \dots & \dots & (x_{100}, y_{100}, 91909, T_{100,100}^0) \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & & \\ \vdots & & & & & \\ \vdots & & & & & \\ (x_0, y_0), \dots, T_{0,0}^0 & \dots & \dots & \dots & \dots & (x_{100}, y_0), \dots, T_{100,0}^0 \end{pmatrix}$$

On a trouvé que MS Excel est le meilleur outil logiciel pour représenter ,organiser et traiter les données de GT extraites de notre système de prototype.

Le tableau général de la matrice GT est décrit dans un fichier (ou classeur) Excel comportant principalement trois (3) feuilles. Chaque « feuille Excel » du classeur représente un tableau de données qu'on a formulé mathématiquement en une « matrice ». On a essentiellement 3 tableaux correspondants à 3 matrices :

Tableau 6 : On y a représenté les données relatives aux valeurs de la température des nœuds

rapportées dans une Matrice MT : *Matrice MT*

$$\begin{pmatrix} t_{0,100} & \dots & \dots & \dots & \dots & t_{100,100} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ t_{0,0} & \dots & \dots & \dots & \dots & t_{100,0} \end{pmatrix}$$

Tableau 7 : au quelle on fait correspondre la matrice des numéros des nœuds (MN).

Matrice MN

$$\begin{pmatrix} i_{0,100} & \dots & \dots & \dots & \dots & i_{100,100} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ i_{0,0} & \dots & \dots & \dots & \dots & i_{100,0} \end{pmatrix}$$

Tableau 8 : est attribué à la matrice des coordonnées des nœuds (MXY) :

Matrice MXY

$$\begin{pmatrix} (x_0, y_{100}) & \dots & \dots & \dots & \dots & (x_{100}, y_{100}) \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ (x_0, y_0) & \dots & \dots & \dots & \dots & (x_{100}, y_0) \end{pmatrix}$$

5.2.4 Comment détecter la position d'un nœud (i) sachant sa température

Une fois on connaît la température (T_i) d'un nœud (i) détecté dans la matrice MT , on peut aller chercher ses coordonnées (X_i, Y_i) dans la matrice des coordonnées MXY , en récupérant le numéro du nœud (i) dans la matrice MN

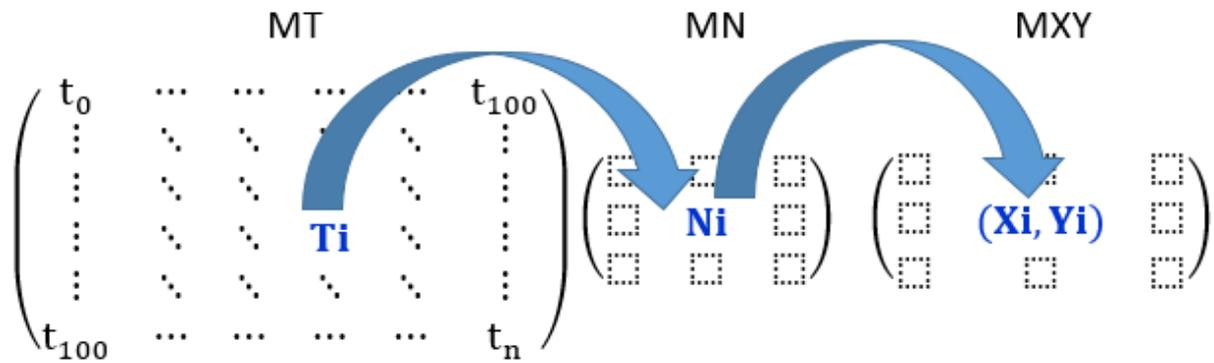


Tableau 6: Matrice Température (MT),[38]

	A	B	C	D	E	F	G	H
1	TIME	=	1.00E+06	TIME	STEP	NUMBER	=	18
2								
3	NODE	TEMPERATURE	NODE	TEMPERATURE	NODE	TEMPERATURE	NODE	TEMPERATURE
4	91801	2.50E+01	91802	2.50E+01	91803	2.50E+01	91804	2.50E+01
5	91805	2.50E+01	91806	2.50E+01	91807	2.50E+01	91808	2.50E+01
6	91809	2.50E+01	91810	2.50E+01	91811	2.50E+01	91812	2.50E+01
7	91813	2.50E+01	91814	2.50E+01	91815	2.50E+01	91816	2.50E+01
8	91817	2.50E+01	91818	2.50E+01	91819	2.50E+01	91820	2.50E+01
9	91821	2.50E+01	91822	2.50E+01	91823	2.50E+01	91824	2.50E+01
10	91825	2.50E+01	91826	2.50E+01	91827	2.50E+01	91828	2.50E+01
11	91829	2.50E+01	91830	2.50E+01	91831	2.50E+01	91832	2.50E+01
12	91833	2.50E+01	91834	2.50E+01	91835	2.50E+01	91836	2.50E+01
13	91837	2.50E+01	91838	2.50E+01	91839	2.50E+01	91840	2.50E+01
14	91841	2.50E+01	91842	2.50E+01	91843	2.50E+01	91844	2.50E+01
15	91845	2.50E+01	91846	2.50E+01	91847	2.50E+01	91848	2.50E+01
16	91849	2.50E+01	91850	2.50E+01	91851	2.50E+01	91852	2.50E+01
17	91853	2.50E+01	91854	2.50E+01	91855	2.50E+01	91856	2.50E+01
18	91857	2.51E+01	91858	2.51E+01	91859	2.51E+01	91860	2.51E+01
19	91861	2.51E+01	91862	2.51E+01	91863	2.51E+01	91864	2.51E+01
20	91865	2.51E+01	91866	2.51E+01	91867	2.51E+01	91868	2.51E+01
21	91869	2.51E+01	91870	2.51E+01	91871	2.51E+01	91872	2.51E+01
22	91873	2.51E+01	91874	2.51E+01	91875	2.52E+01	91876	2.52E+01
23	91877	2.52E+01	91878	2.51E+01	91879	2.51E+01	91880	2.51E+01

Tableau 7: Matrice des nœuds (MN),[38]

Row	Node	Temperature
1		
2		
3	NODE	TEMPERATURE
4	91801	2.50E+01
5	91802	2.50E+01
6	91803	2.50E+01
7	91804	2.50E+01
8	91805	2.50E+01
9	91806	2.50E+01
10	91807	2.50E+01
11	91808	2.50E+01
12	91809	2.50E+01
13	91810	2.50E+01
14	91811	2.50E+01
15	91812	2.50E+01
16	91813	2.50E+01
17	91814	2.50E+01
18	91815	2.50E+01
19	91816	2.50E+01
20	91817	2.50E+01
21	91818	2.50E+01
22	91819	2.50E+01
23	91820	2.50E+01

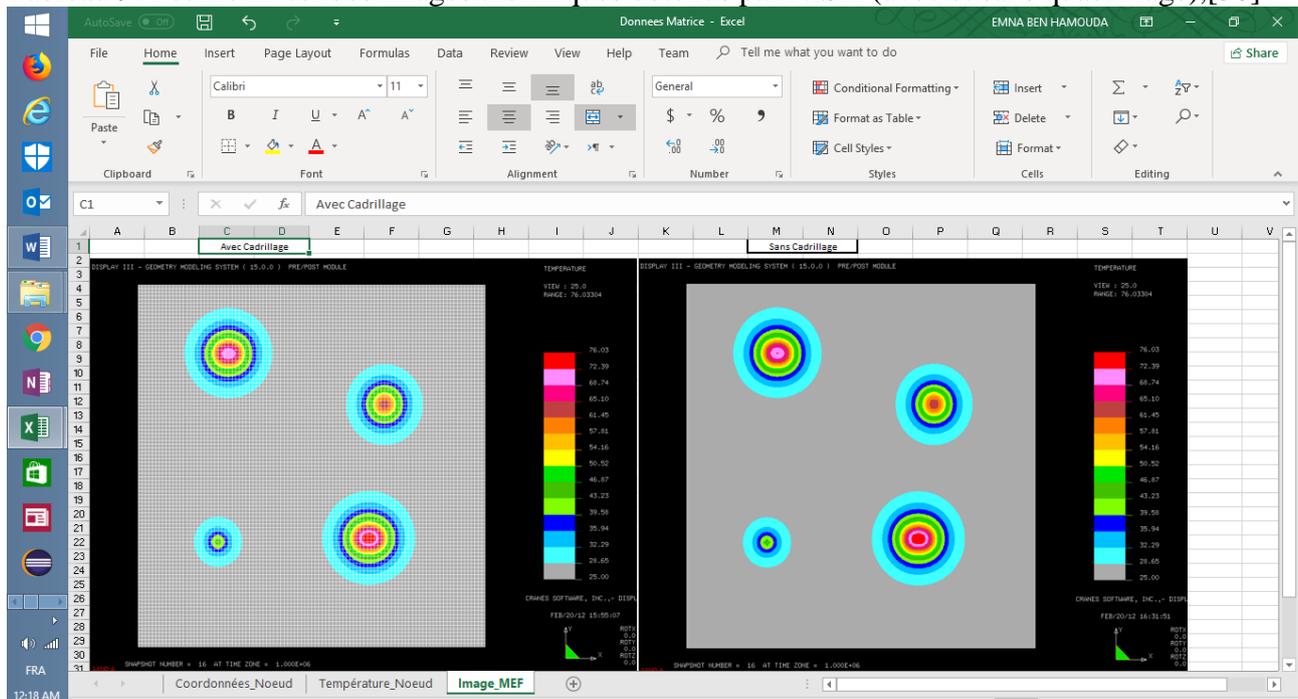
Row	Node	Temperature
98	91895	2.50E+01
99	91896	2.50E+01
100	91897	2.50E+01
101	91898	2.50E+01
102	91899	2.50E+01
103	91900	2.50E+01
104	91901	2.50E+01
105	91902	2.50E+01
106	91903	2.50E+01
107	91904	2.50E+01
108	91905	2.50E+01
109	91906	2.50E+01
110	91907	2.50E+01
111	91908	2.50E+01
112	91909	2.50E+01
113	91910	2.50E+01
114	91911	2.50E+01
115	91912	2.50E+01
116	91913	2.50E+01
117	91914	2.50E+01
118	91915	2.50E+01
119	91916	2.50E+01
120	91917	2.50E+01

Tableau 8 : Matrice des coordonnées des nœuds (MXY),[38]

	A	B	C	D	E	F	G	H	I
1	Nœud		X	Y	Z				
2	91810	0	0.00000E+00	0.00000E+00	5.00000E+02				
3	91811	0	5.00000E+01	0.00000E+00	5.00000E+02				
4	91812	0	1.00000E+02	0.00000E+00	5.00000E+02				
5	91813	0	1.50000E+02	0.00000E+00	5.00000E+02				
6	91814	0	2.00000E+02	0.00000E+00	5.00000E+02				
7	91815	0	2.50000E+02	0.00000E+00	5.00000E+02				
8	91816	0	3.00000E+02	0.00000E+00	5.00000E+02				
9	91817	0	3.50000E+02	0.00000E+00	5.00000E+02				
10	91818	0	4.00000E+02	0.00000E+00	5.00000E+02				
11	91819	0	4.50000E+02	0.00000E+00	5.00000E+02				
12	91820	0	5.00000E+02	0.00000E+00	5.00000E+02				
13	91821	0	5.50000E+02	0.00000E+00	5.00000E+02				
14	91822	0	6.00000E+02	0.00000E+00	5.00000E+02				
15	91823	0	6.50000E+02	0.00000E+00	5.00000E+02				
16	91824	0	7.00000E+02	0.00000E+00	5.00000E+02				
17	91825	0	7.50000E+02	0.00000E+00	5.00000E+02				
18	91826	0	8.00000E+02	0.00000E+00	5.00000E+02				
19	91827	0	8.50000E+02	0.00000E+00	5.00000E+02				
20	91828	0	9.00000E+02	0.00000E+00	5.00000E+02				
21	91829	0	9.50000E+02	0.00000E+00	5.00000E+02				
22	91830	0	1.00000E+03	0.00000E+00	5.00000E+02				
23	91831	0	1.05000E+03	0.00000E+00	5.00000E+02				

Une quatrième feuille est rajoutée au tableau Excel pour y enregistrer les résultats des simulations et les images thermiques obtenues par NISA (voir Tableau 9).

Tableau 9: Feuille Excel des images thermiques obtenus par NISA (avec et sans quadrillage),[38]



(Voir ce tableau dans un format plus grand mis en Annexe A2)

5.3 L’algorithme de recherche de « la matrice contour »

L’algorithme de recherche de la matrice contour est une solution algorithmique basée sur un calcul matriciel. L’aspect matriciel de la représentation des données sous forme de matrices m’a inspiré à chercher une solution algorithmique basée sur le calcul matriciel. En plus les résultats de la simulation offrent une image thermique qui décrit la distribution thermique dans la surface du CI. Sachant que « la matrice contour » est un outil matriciel efficace utilisé dans le domaine du traitement d’image qui permet le scan complet de la surface du CI et permet de produire une image thermique très net et précise sur l’état thermique du modèle, on a suggéré « l’Algorithme de recherche de la matrice contour » qu’on a nommé « MACA » (Matrix Contour Algorithm) pour résoudre la problématique de la détection thermique dans le cas de sources multiples de chaleur. En effet, MACA permet de détecter tous les contours à la surface du CI et en générer une image thermique avec toutes les inclusions qui sont sous formes de cercles de différentes couleurs qui ne sont autres que les cercles

isothermes renfermant à leur centre les pics thermiques recherchés. Ainsi, on a ajouté l'algorithme de recherche du nombre d'Euler (r). (r) permet de renseigner sur le nombre sources de chaleur détectées.

5.4 Diagramme de flux de l'Algorithme MSTPDA

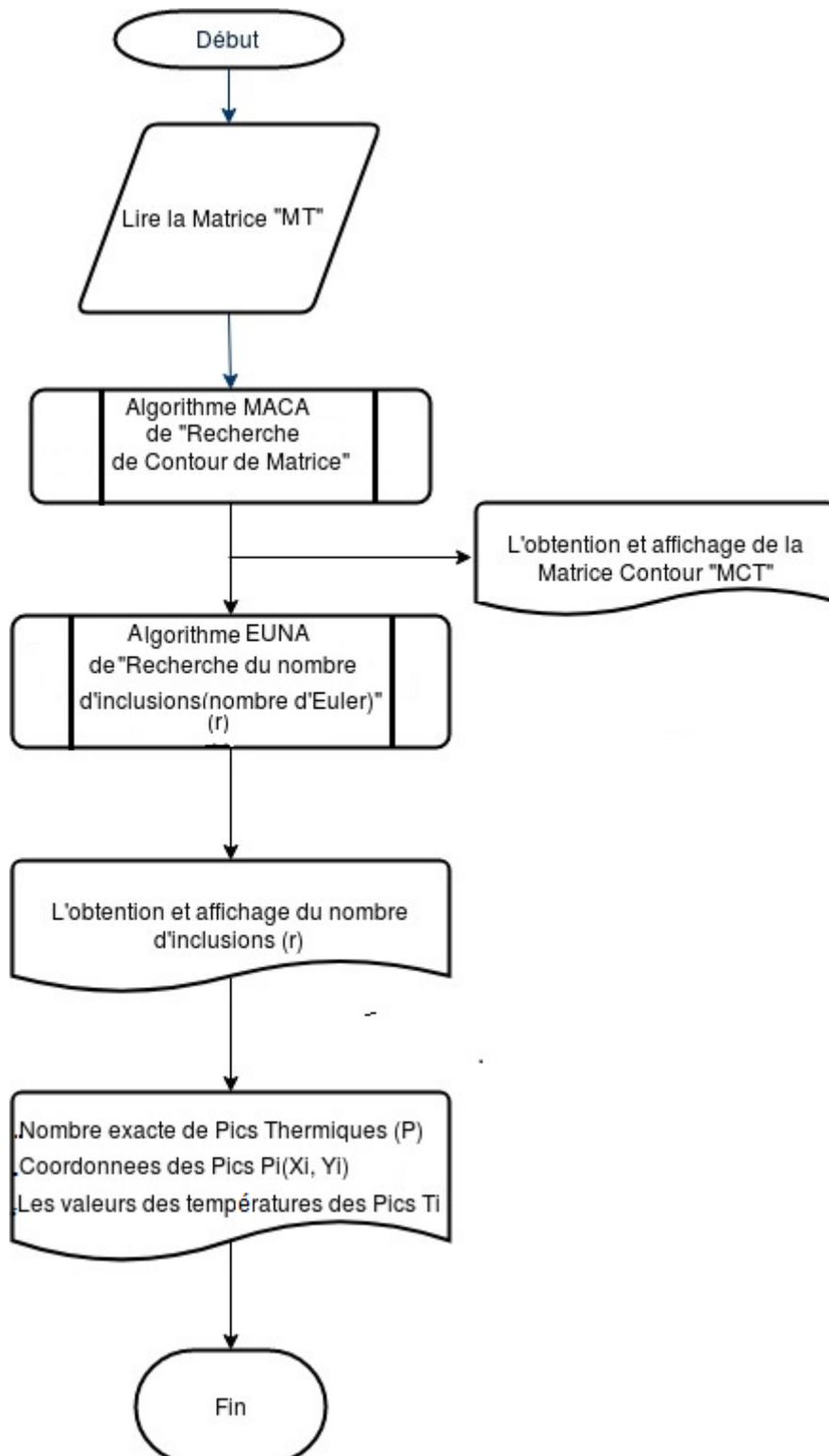


Figure 33: Algorithme MSTPDA,[38]

5.5 Implémentation de l'Algorithme MSTPDA dans Matlab

L'Algorithme de la Matrice Contour est implémenté dans l'outil logiciel de la modélisation et simulation MATLAB sous la commande « **Contours MT** ». En appliquant la commande « **Contours MT** » qui exécute l'Algorithme MACA de "Recherche de Contour de Matrice " et l'Algorithme EUNA de "Recherche du nombre d'inclusions (ou nombre d'Euler) " on obtient :

5.6 La Matrice Contour "MCT"

La matrice contour MCT est obtenue après un long calcul matriciel complexe et contient seulement des valeurs binaires (0 ou 1).

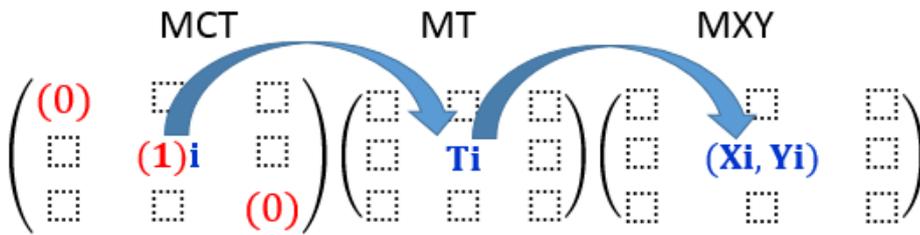
$$MCT \begin{pmatrix} 0 & 0 & 0 & 1 & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{pmatrix}$$

5.7 Comment identifier les points de MCT qui sont sur les isotherme) ?

Si au niveau de la matrice contour on détecte l'entier '1' à une position donnée (i) cela correspondant à un contour, on devrait à ce moment faire correspondre à cette position sa température et ses coordonnées en passant par les matrices suivantes respectivement : MCT (contour), MT

(température), et MXY(coordonnées). Cette procédure est illustrée dans ce qui suit :

$$\text{MCT} \begin{pmatrix} 0 & 0 & 0 & 1 & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \end{pmatrix}$$



5.8 Nombre d'inclusions « ou (nombre d'Euler (r)) »

La valeur de (r) renseigne sur le nombre de sources de chaleur dans le CI. Dans notre cas (r) est égal à (4). Ces quatre sources sont illustrées dans « l'image thermique du modèle » générée par Matlab.

5.9 L' image thermique résultante

Générée par l'exécution de la commande Contour de Matlab montre la détection des contours qui ne sont autres que les cercles des isothermes avec les pics thermiques aux niveaux de leurs centres.

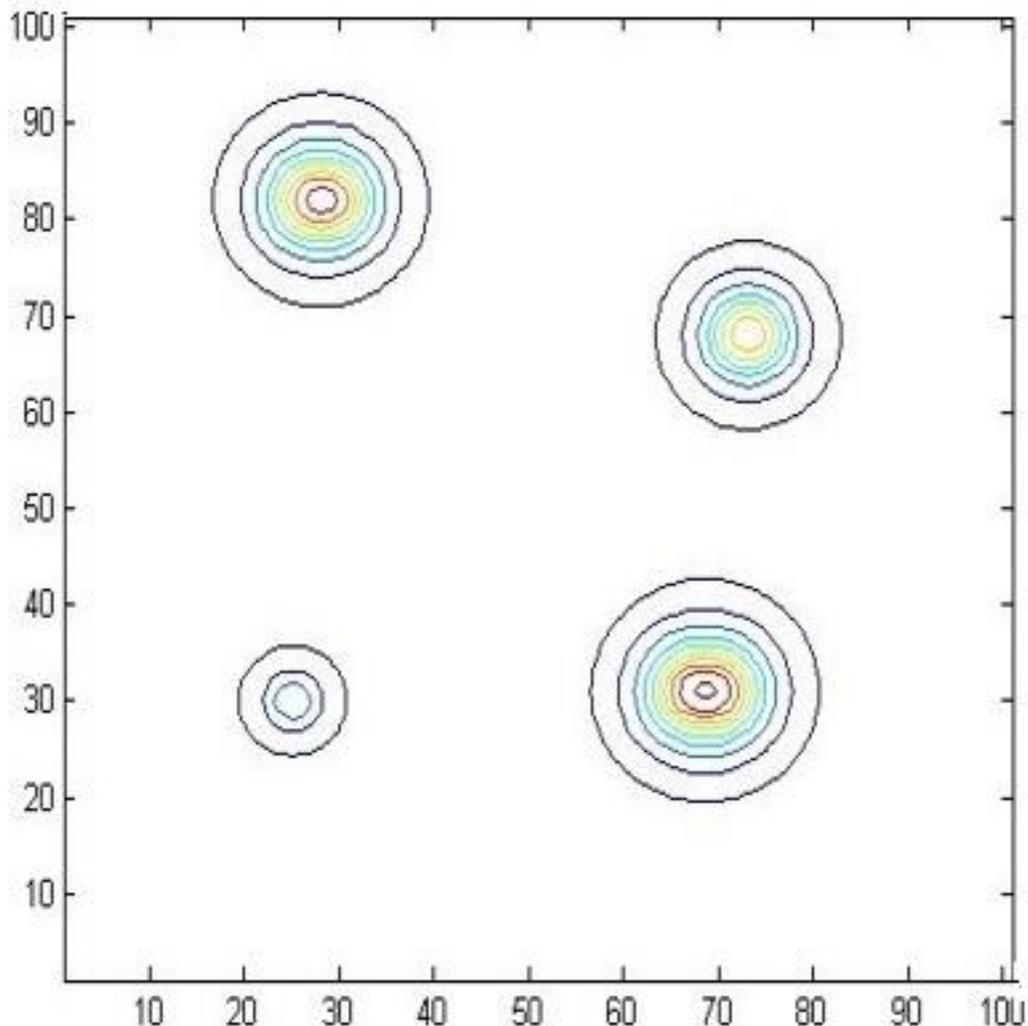


Figure 34: l'image thermique générée par la validation de MSTPDA dans Matlab,[38]

5.10 Validation de l'algorithme MSTPDA

On a fait validation de l'algorithme MSTPDA en comparant les résultats de l'exécution de MSTPDA dans Matlab à ceux du modèle généré par NISA.

Dans la feuille Excel qui suit, on peut faire la comparaison entre les 2 images thermiques, celle obtenue par NISA et celle obtenue par la validation de MSTPDA dans Matlab, (voir Figure 34).

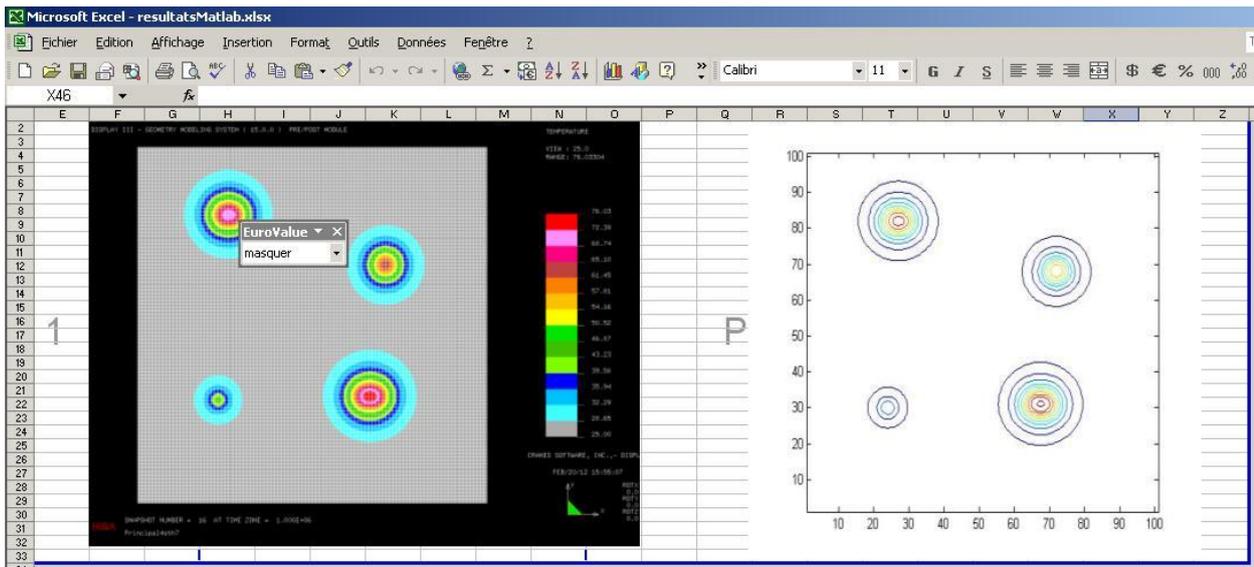


Figure 35: comparaison de 2 images celle obtenue par NISA et celle par MSTPDA,[38]

(NB : Veuillez voir cette figure35 dans un format plus grand dans l'Annexe A).

En comparant ces deux images, celle obtenue par NISA (à gauche) et celle générée par MSTPDA (à droite), on a constaté que ces images thermiques sont presque similaires, ce qui prouve que l'algorithme MSTPDA est efficace pour la détection des quatre sources de chaleurs et il a généré l'information recherché d'une façon précise :la température, la position de chaque source de chaleur et leur nombre. Ainsi on a validé l'algorithme MSTPDA dans Matlab.

5.11 Analyse et discussion de la validation de l'algorithme

MSTPDA :

En tant qu'algorithme de traitement d'image par excellence MSTPDA est efficace pour traiter l'image thermique d'un CI, détecter plusieurs sources de chaleur et générer avec précision la température, la position de chaque source de chaleur et leur nombre total.

MSTPDA est un algorithme de détection thermique par excellence, il est efficace et recommandé pour la gestion thermique dans les grands circuits LAIC y compris le Wafer ICTM.

Par contre dans l'industrie électronique on s'intéresse aussi à ce qui se passe réellement dans un CI de petite taille (non LAIC), ce qui n'est pas le cas du domaine d'application de l'algorithme MSTPDA. Donc cette solution est irréaliste dans le cas des CI de petite taille (non LAIC), on ne peut jamais risquer à implémenter dans un petit CI (non LAIC) et interroger 10201 capteurs en même temps car il endommagera certainement le CI. D'où la nécessité de suggérer une autre troisième solution qui est le sujet du chapitre suivant de ce rapport.

Chapitre 6 : Algorithme de balayage thermique « MHSODA »

6.1 Modèle ou prototype conçu pour la validation de l'algorithme « MHSODA »

Dans notre étude on a considéré le même modèle conçu précédemment (voir Figure 28) sauf qu'on a ajouté l'hypothèse importante suivante : le nombre de capteurs de capteurs implémentés sur la surface du CI est 1021 capteurs mais lors du scan (balayage), on interroge (on met en marche (ON)) seulement trois (3) capteurs en même temps pour éviter le risque d'endommagement du circuit intégré CI causée par l'augmentation de la dissipation de chaleur provenant de la mise en marche d'un nombre excessif de capteurs.

6.2 Principe de base de l'algorithme « MHSODA »

Une façon simple de résoudre la problématique de la thèse et de procéder par raisonnement algorithmique séquentielle et qui permet de balayer (scanner) progressivement la surface du CI selon les deux axes (x et y) tout en mettant en marche (ON) seulement trois (3) capteurs en même temps. Ceci évite le risque de surchauffe du CI causé par l'allumage des capteurs.

Puis on détecte les trois valeurs de température significatives avec des erreurs négligeables aux niveaux des trois nœuds des capteurs. On détecte le pic thermique en comparant ces trois valeurs de température détectées. Si l'un des nœuds (ou points) enregistre une température maximum plus élevée que celle détectée dans le point qui le précède et si en même temps elle est plus élevée que celle détectée dans le point suivant, alors on peut conclure que ce point est un pic thermique.

6.3 Développement de de l'algorithme « MHSODA »

MHSODA détecte les pics thermiques en effectuant un balayage selon l'axe des (X) et un autre selon l'axe des (Y). Ceci est démontré dans la première et la deuxième boucle du diagramme de flux de MHSODA (Voir la figure 36).

6.3.1 Le balayage thermique selon l'axe des (X)

Il progresse le long de chaque rangée (ou la ligne) tout en maintenant la valeur de l'ordonnée (Y) fixe (ou constante) et en scannant trois points à la fois, puis on passe aux trois points suivants. On continue le balayage de la même façon jusqu'à finir le scan de toute la rangée. Puis on passe à la rangée d'après. A chaque fois on détecte un pic thermique sur la rangée, on mentionne qu'il y a un pic thermique en précisant sa température et ses coordonnées. Ce balayage selon l'axe des (X) est démontré dans la première boucle du diagramme de flux du MHSODA de la figure 36 et dans la section (**Recherche des nœuds des pics thermiques par rangée**) de l'Annexe A5 (exposé dans l'Annexe du présent rapport).

6.3.2 Le balayage thermique selon l'axe des (Y)

Il progresse le long de chaque Colonne (ou la verticale) tout en maintenant la valeur de l'abscisse (X) fixe (ou constante) et en scannant trois points à la fois, puis on passe aux trois points suivants. On continue le balayage de la même façon jusqu'à finir le scan de toute la rangée. Puis on passe à la colonne d'après. A chaque fois on détecte un pic thermique sur la rangée, on mentionne qu'il y a un pic thermique en précisant sa température et ses coordonnées. Ce balayage selon l'axe des (Y) est démontré dans la première boucle du Diagramme de flux du MHSODA, de la figure 36 et dans la section (**Recherche des nœuds des pics thermiques par Colonne**) de l'Annexe A5 (exposé dans l'Annexe du présent rapport).

6.3.3 La Confirmation de Détection des Sources Multiples après comparaison

La Confirmation de Détection des Sources Multiples après comparaison de chaque pic thermique détecté à la suite du balayage selon l'axe des (X) à un autre pic thermique détecté à la suite du balayage des (Y). Si ces deux pics thermiques possèdent les mêmes caractéristiques, on confirme qu'on a détecté le même pic thermique et qu'il est « une seule source de chaleur ». De la même manière, on compare tous les pics thermiques et on confirme la détection de ceux qui sont « sources de chaleur ».

6.3.4 Impressions de tous les résultats

Ainsi finit par détecter et imprimer toutes les sources de chaleurs dans le CI en précisant pour chacune d'elles son numéro de nœud, sa température et ses coordonnées. En plus, on finit par déduire et imprimer : le nombre total de sources de chaleur et la « sources de chaleur » la plus chaude avec son numéro de nœud, sa température et ses coordonnées.

Cette procédure de Confirmation de Détection des Sources Multiples est démontrée dans la troisième boucle du Diagramme de flux du MHSODA de à la figure 36, et dans la section (**Partie C : imprimer les résultats des pics thermiques à l'écran**) de l'Annexe A5(exposé dans l'Annexe du présent rapport).

Toutes les étapes de la procédure d'Implémentation de de l'algorithme « MHSODA » de détection du Pics thermiques sont détaillées dans le Diagramme de flux de l'algorithme MHSODA qui est imprimé en grand format dans l'Annexe A5 (exposé dans l'Annexe du présent rapport). Et elle est bien explicitée dans les commentaires du code de MHSODA qu'on a programmé en langage C++.

6.4 Diagramme de flux de l'Algorithme de balayage MHSODA

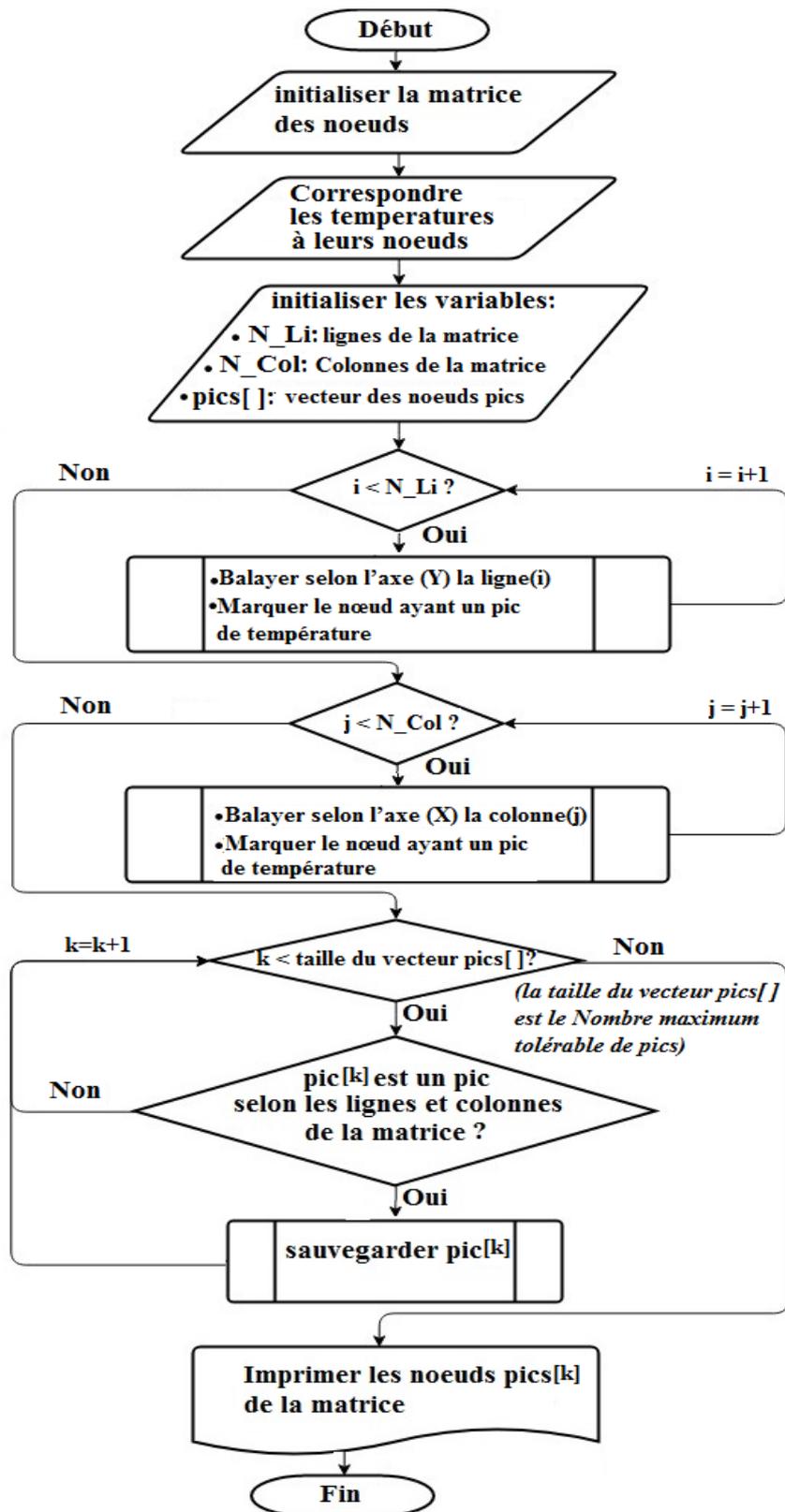


Figure 36 : Diagramme de flux Algorithme MHSODA [38]

(Voir cette figure 36 dans un format plus grand et commentée dans l'Annexe A5).

6.5 Exécution de « MHSODA » en C++ :

```
FILE* pFile = fopen(FILENAME_NODE_COORD, "r"); // Open file
if (pFile==NULL) {
    perror("Error opening Coord_noeuds.txt"); // Verify no errors
}

// Read line by line until end of file
for(row_idx = 0; row_idx < ROW_SIZE; row_idx++) {
    for(col_idx = 0; (col_idx < COL_SIZE) && (fgets(f_line_buf, 200, pFile)
        //Read line in given format: "Node_Num [space] Dummy_Value [space] X
        sscanf(f_line_buf, "%10s %10s %10s %10s %10s %10s", node, dum1, x, y,
        if(atol(node) <= 0) { // Verify that Node_Num value is a number
            continue; // if not, we skip and read the next line
        }
    }
}
```

Node Peaks:
Node num=94762 coord=(1150, 1450) temp=44.543152
Node num=94907 coord=(3350, 1500) temp=76.033043
Node num=98648 coord=(3550, 3350) temp=64.114830
Node num=100017 coord=(1300, 4050) temp=72.695702

SUMMARY:
Number of Peaks: 4
Highest Peak: Node num=94907 coord=(3350, 1500) temp=76.033043

Figure 37 : Résultats d'exécution du code MHSODA,[38]

(Voir cette figure dans un format plus grand mis en Annexe A).

6.6 Analyse et Discussion de la 3e solution algorithmique de MHSODA

6.6.1 Analyse des résultats de l'exécution de MHSODA :

Les résultats de l'exécution du code MHSODA sont très satisfaisants comme montré en bas de l'image de l'interface de la plateforme Eclipse (Figure 36). On a obtenu avec précision, pour chaque source de chaleur son numéro de nœud, sa température et ses coordonnées.

En plus, il est décrit dans le sommaire : le nombre de sources de chaleur et le pic thermique le plus chaud avec son numéro de nœud, sa température et ses coordonnées.

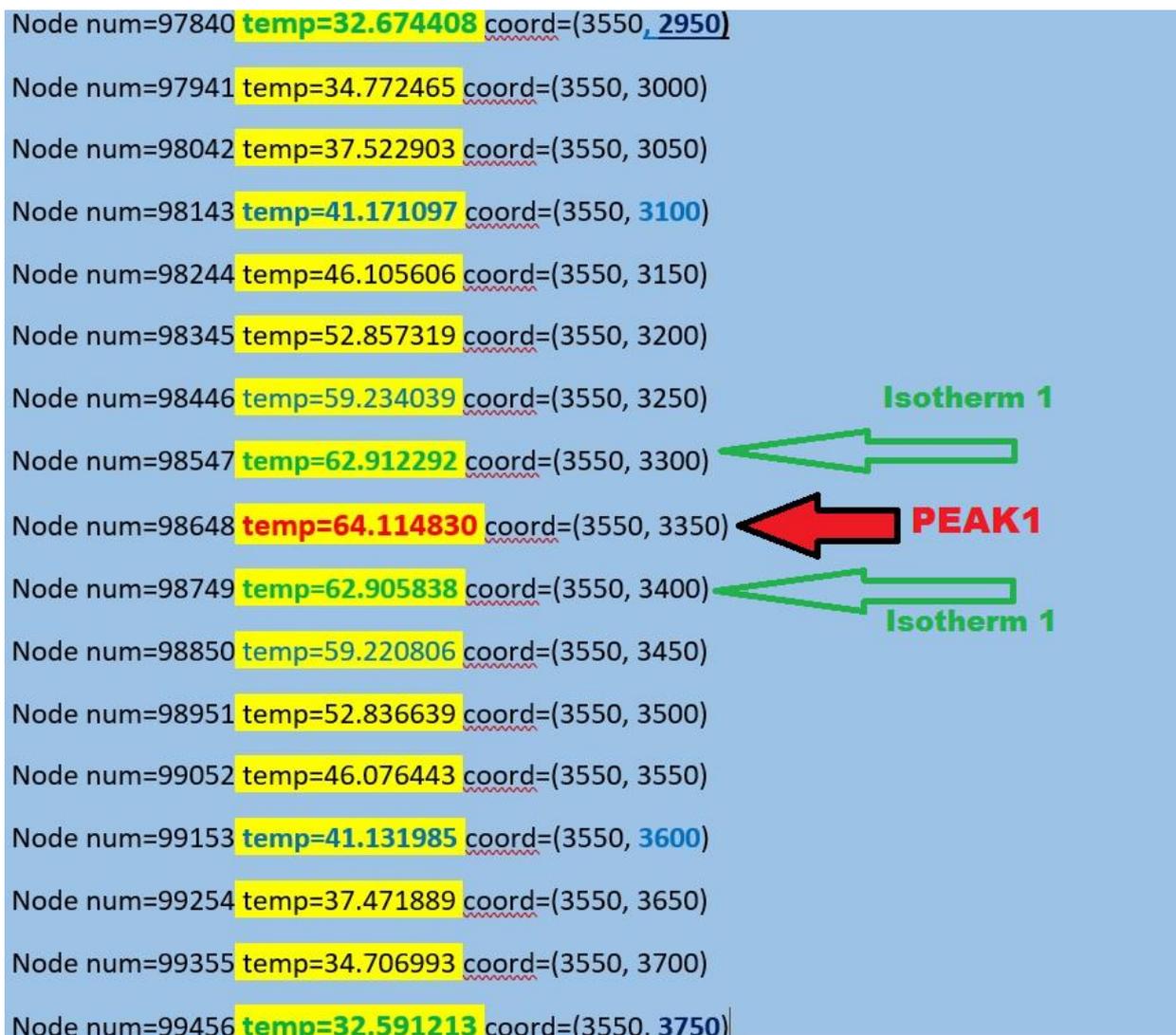


Figure 38: Listage du pic thermique (1) avec ses isothermes, de l'exécution de MHSODA,[38]

Les résultats listés dans les Figures 37 et 38 sont extraites des listages d'exécution du MHSODA confirment les résultats décrits dans la figure 36, mais il se distinguent par l'exposition des isothermes en montrant les valeurs de températures similaires inscrites de façon symétrique en bas et en haut des pics thermiques.

```

Node num=93997 temp=33.912956 coord=(3300, 1050)-----
Node num=94098 temp=36.287338 coord=(3300, 1100)
Node num=94199 temp=39.317997 coord=(3300, 1150)
Node num=94300 temp=43.196796 coord=(3300, 1200)
Node num=94401 temp=48.173355 coord=(3300, 1250)
Node num=94502 temp=54.559250 coord=(3300, 1300)
Node num=94603 temp=62.710392 coord=(3300, 1350)
Node num=94704 temp=70.213394 coord=(3300, 1400) ← Isotherm2
Node num=94762 temp=44.543152 coord=(1150, 1450) ← PEAK3
Node num=94907 temp=76.033043 coord=(3350, 1500) ← PEAK2
Node num=95008 temp=74.596901 coord=(3350, 1550)
Node num=95109 temp=70.222191 coord=(3350, 1600) ← Isotherm2
Node num=95210 temp=62.724113 coord=(3350, 1650)
Node num=95311 temp=54.578575 coord=(3350, 1700)
Node num=95412 temp=48.199245 coord=(3350, 1750)
Node num=95513 temp=43.230530 coord=(3350, 1800)
Node num=95614 temp=39.361237 coord=(3350, 1850)
Node num=95715 temp=36.342220 coord=(3350, 1900)
Node num=95816 temp=33.982197 coord=(3350, 1950)-----

```

Figure 39: Pics thermiques (2 et 3), listage des résultats d'exécution du MHSODA,[38] Pour mieux apprécier l'efficacité et la performance du programme veuillez lire l'analyse de la performance de l'algorithme dans la section qui suit.

6.6.2 La performance de l'algorithme MHSODA

La comparaison asymptotique est la méthode utilisée pour étudier la complexité et le comportement d'une fonction ou d'un algorithme. Elle peut être appliquée à l'analyse des scénarios défavorables par rapport au temps d'exécution ou à l'espace de mémoire utilisé par un algorithme. La notation Grand-O d'Edmund Landau [14] dénote le caractère dominé d'une fonction par rapport à une autre.

Complexité temporelle linéaire :

Un algorithme dont la performance accroit linéairement et proportionnellement à la taille des données entrées est dit d'avoir un Temps Linéaire et est décrit par la notation $O(N)$. Par exemple, une procédure qui additionne tous les éléments d'une liste nécessite un temps proportionnel à la taille de la liste si le temps d'addition est constant. L'exemple ci-dessous montre également comment $O(N)$ favorise le scénario de performance le plus défavorable : une chaîne correspondante pourrait être trouvée à chaque

itération de la boucle-for et la fonction reviendrait plus tôt, mais $O(N)$ supposera toujours la limite supérieure où l'algorithme effectuera le nombre maximal d'itérations [15].

Complexité temporelle quadratique:

Un algorithme dont la performance est proportionnelle au carré de la taille de l'ensemble des données entrées est dit d'avoir un Temps Quadratique et décrit par $O(N^2)$. Ceci est courant avec les algorithmes utilisant des itérations imbriquées sur l'ensemble des données. D'autres itérations imbriquées entraîneront des degrés de complexité $O(N^3)$, $O(N^4)$, etc. [15]

Notre programme analyse une disposition matricielle des nœuds dans la grille. Il emploie plusieurs algorithmes de recherche qui parcourent les rangées et les colonnes des matrices, les rendant ainsi d'une complexité temporelle quadratique $O(N^2)$.

6.6.3 Discussion de la solution algorithmique MHSODA

L'algorithme MHSODA proposé est excellent, car il est simple, rapide, performant, facilement implémentable, ne présente pas de risque de surchauffe rapide causé par les capteurs, efficace et permet d'obtenir des résultats précis du nombre de sources de chaleur ainsi que leurs positions et les températures qui y sont détectées. Mais son désavantage est qu'il utilise 10201 capteurs ce qui est irréaliste. Car dans la réalité de l'industrie électronique, on ne va pas installer 10201 capteurs dans un CI de petite taille à cause des contraintes économiques et opérationnelles de fabrication. Pour ceci on a cherché à optimiser le nombre de capteurs et le réduire au maximum, c'est la raison pour laquelle on a proposé une quatrième solution algorithme dans la partie suivante du rapport. MHSODA demeure un excellent algorithme à implémenter dans les CI de grandes tailles (LAIC) tel que Wafer IC TM.

Chapitre 7 : Algorithme de balayage thermique GDSCAN basé sur GDS

7.1 Introduction :

« GDSCAN » est un algorithme de balayage thermique basée sur le GDS pour la détection multiple de sources de chaleur dans les circuits intégrés.

7.2 Modèle ou prototype conçu pour la validation du « GDSCAN »

Dans notre étude on a développé le même modèle conçu précédemment sauf le nombre de capteurs qui va être restreint à 24 capteurs distribués en quatre (4) cellules(i), avec (i=1 à 4) RO.

Chaque cellule (i)est constituée de six (6) RO (3RO : Ai,Bi,Ci +3RO : Di,Ei,Fi),avec(i=1 à 4).

Pour pouvoir utiliser ce prototype, on a positionné chaque cellule de six (6) RO dans la zone environnante de la source de chaleur (Si), avec (i=1 à 4).On a réalisé le prototype suivant :

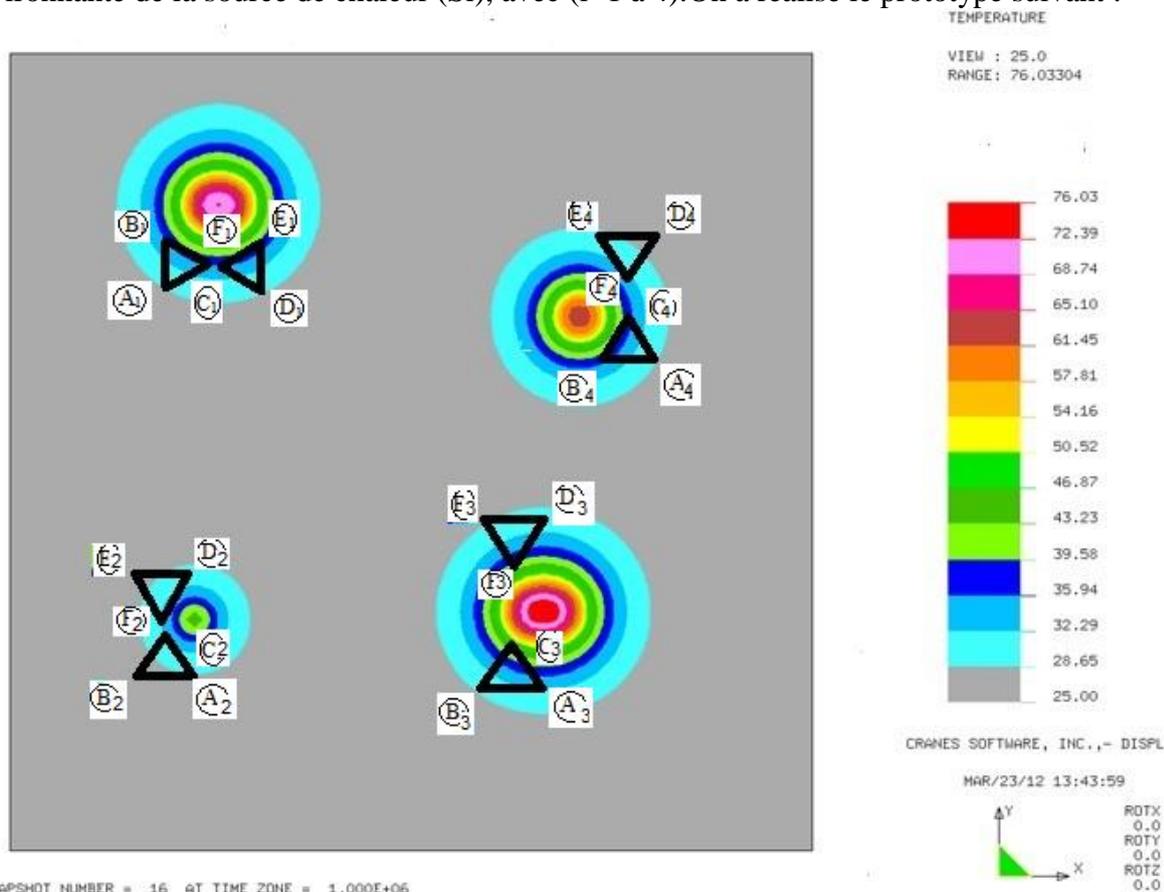


Figure 40 : Prototype conçu pour la validation de l'algorithme « GDSCAN »

7.3 Principe de l'algorithme GDSCAN basé sur GDS

Du point de vue algorithmique, le balayage est une solution pertinente, mais vu qu'en réalité sur un CI on ne doit pas utiliser 10201 capteurs. On va optimiser le nombre de capteurs nécessaires et le réduire au maximum pour couvrir toute la surface du CI et scanner seulement un nombre restreint de cellules de capteurs lors du monitoring thermique. C'est le cas de notre quatrième proposition de solution algorithmique : **Algorithme de balayage « GDSCAN » basé sur le GDS-étendu.**

Pour ce faire, on a utilisé la méthode du GDS et on a procédé par l'implémentation d'un ensemble de capteurs RO sous forme de cellules de six (6) capteurs RO qui sont positionnée selon une configuration de deux triangles équilatéraux dont les extrémités sont munies de 3 capteurs chacun. Cette configuration des cellules de capteurs RO va permettre l'application de la technologie du GDS pour détecter les pics thermiques aux environs de ces cellules RO, Ainsi on étend l'application du GDS pour détecter plusieurs sources de chaleur dans les CI.

7.4 Principe de l'algorithme GDS pour chaque cellule de capteurs

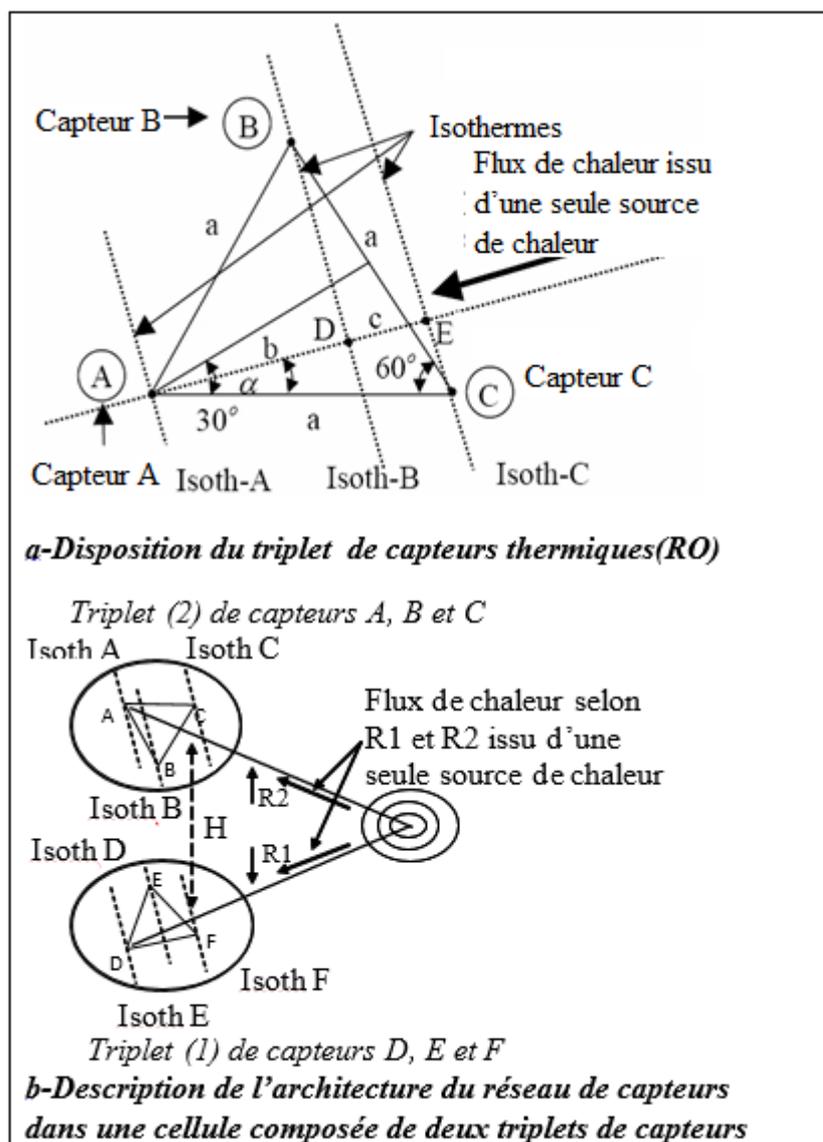


Figure 41 : Principe de l'algorithme du GDS pour la détection d'une seule source de chaleur,[7]

La Figure (41) décrit les détails du principe GDS pour détecter une seule source de chaleur. Pour chacun des capteurs RO implémenté sur le CI on a son numéro de nœud et ses coordonnées de nœud.

On applique l'algorithme GDS pour détecter supposément un seul pic thermique à la fois dans la cellule de capteurs de la zone (i) puis on passe la zone (i+1) pour détecter supposément un seul pic thermique dans la cellule de capteurs de la zone (i+1). De la même façon le principe du GDS est appliqué jusqu'à atteindre la cellule de capteurs la zone (4).

7.5 Détails de l'application du GDS pour chaque cellule de capteurs

Puis, on peut en déduire d'autres données telles que :

- **La distance (r_i).** Dans ce qui suit la définition générale de (r) et son expression de calcul :
On définit (r) comme étant la distance entre le triplet de capteurs et la source de chaleur.

(r) est déduite de l'expression suivante :

$$\frac{\Delta T}{\Delta r} = \frac{T_C - T_A}{a \cdot \cos \alpha} \leftrightarrow \frac{V_C - V_A}{a \cdot \cos \alpha} \quad [7]$$

La valeur métrique du côté (**a**) du triangle équilatéral est supposé connu.

Pour chaque cellule de six (6) capteurs RO, on donne les valeurs des côtés (a) suivantes :

- (a_{1i}) pour chaque triplet de capteurs (D_i, E_i, F_i), avec (i=1 à 4).
- (a_{2i}) pour chaque triplet de capteurs (A_i, B_i, C_i), avec (i=1 à 4).

D'où pour chaque cellule de six (6) capteurs RO, on obtient les valeurs des (**r_i**) suivantes:

- (**R_{1i}**) pour chaque triplet de capteurs (D_i, E_i, F_i), avec (i=1 à 4).
- (**R_{2i}**) pour chaque triplet de capteurs (A_i, B_i, C_i), avec (i=1 à 4).
- **Les distances (b) et (c).** Dans ce qui suit leurs expressions générales :

$$b = \overline{AD}, \overline{AD} \leftrightarrow T_B - T_A \leftrightarrow V_B - V_A$$

$$b + c = \overline{AE}, \overline{AE} \leftrightarrow T_C - T_A \leftrightarrow V_C - V_A \quad [7]$$

Ainsi pour chaque cellule de six (6) capteurs RO, on obtient les valeurs des (b et c) suivantes:

- (b_{1i}) pour chaque triplet de capteurs (D_i, E_i, F_i), avec (i=1 à 4).
- (c_{2i}) pour chaque triplet de capteurs (A_i, B_i, C_i), avec (i=1 à 4).

Puis lors de l'exécution de l'algorithme de balayage en ces capteurs on va y détecter :

- **Les valeurs températures des capteurs** ((T_{Ai}, T_{Bi}, T_{Ci}) et (T_{Di}, T_{Ei}, T_{Fi}), avec (i=1 à 4)).

Pour chaque zone de source de chaleur (S_i), on applique le principe du GDS (voir Figure 41).

On obtient par « estimation » toutes les valeurs des paramètres suivants :

- (X_{Si}, Y_{Si}) : les coordonnées de la source de chaleur (S_i), avec (i=1 à 4)
- (T_{Si}) : la température de la source de chaleur (S_i), avec (i=1 à 4)

(T_{si}) est estimée par l'expression générale de (T_H) suivante :

$$T_S \leftrightarrow V_S = \frac{H}{a} \cdot (V_{C1} - V_{A1}) \cdot \frac{(\tan^2 \alpha_1 + 1) \cdot (\sqrt{3} + \tan \alpha_2)}{\sqrt{3} \cdot (1 - \tan \alpha_1 \cdot \tan \alpha_2) - (\tan \alpha_1 + \tan \alpha_2)} + V_{A1} \quad [7]$$

- **La tangente de l'angle (α)** de chaque triangle de capteurs, qui est estimée par l'expressions générale de ($\tan \alpha$) suivante :

$$\tan \alpha \leftrightarrow \frac{2}{\sqrt{3}} \cdot \left(\frac{V_B - V_A}{V_C - V_A} - \frac{1}{2} \right) \quad [7]$$

Ainsi pour chaque Source de chaleur on obtient les valeurs des tangentes :

- ($\tan \alpha_{1i}$) pour chaque triplet de capteurs (D_i, E_i, F_i), avec ($i=1$ à 4).
- ($\tan \alpha_{2i}$) pour chaque triplet de capteurs (A_i, B_i, C_i), avec ($i=1$ à 4).

7.6 Progression de l'application de l'algorithme « GDS » en balayant les zones du CI

Ayant fini de scanner la zone de la cellule(i), on passe à la zone de la cellule (i+1) pour la scanner et y appliquer le «GDS ». Ainsi cet algorithme serra appliquer progressivement d'une zone de cellule(i) à une autre zone de cellule (i+1) d'une façon alternative. L'application du « GDS » se fait dans une zone de cellule à la fois, puis on passe à la zone de cellule d'après jusqu'à atteindre la dernière cellule.

7.7 Détection de multiples sources de chaleur par GDSCAN

Par l'application de l'algorithme « GDS » tout en balayant toutes les zones des cellules de capteurs, on finit par détecter toutes les sources de chaleurs dans le CI, les localiser avec précision et y estimer de façon précise les valeurs de température maximum (Pic thermique) dans ces zones.

Ainsi, on a réalisé :

- Une exécution de l'Algorithme (GDSCAN) a été faites en exécutant le GDS d'une façon séquentielle étape par étape, d'une zone a une autre de la surface du CI.
- Une validation numérique par MATLAB de l'Algorithme (GDSCAN) a été réalisée et illustrée dans la section qui suit (7.8)
- Cet algorithme (GDSCAN) a été aussi sujet de programmation en langage C++ (7.9).

7.8 Validation numérique par MATLAB de l'Algorithme (GDSCAN).

Une implémentation numérique par MATLAB a été exécutée avec succès. Cette implémentation par Matlab est suffisante pour valider GDSCAN. Elle nous offre une configuration de réseau de capteurs fiable lequel inclut quatre (4) cellules avec un nombre minimum de capteurs (24 RO) qui nous ont permis de détecter quatre (4) sources de chaleur. Les températures estimées sont relativement les

même que celles des valeurs de températures prédites au début avec des erreurs qui ne dépassent pas les 1.2% (voir Figure 42).

Les positions estimées de ces sources de chaleur sont dans les zones de hautes températures entourant celles-ci qui sont relativement très proches de leurs positions prédites au début (voir Figure 43).

Étapes de développement de la validation GDSCAN en Matlab pour une cellule (i) générique de Capteurs (Orientée vers la droite)

1-Prédiction de la zone cible de contrôle thermique

Dans cette section, on prévoit « la position de la source de chaleur suspecte » et on va installer dans son environnement la cellule de 6 capteurs (RO).

Choix des Coordonnées de la source de chaleur suspecte

On prévoit que la chaleur augmentera dans la zone ou (S_i) est prévue être une source de chaleur :

$$\boxed{S_i (X_{si}, Y_{si})} \quad (i= 1 \text{ à } 4)$$

NB: Dans notre cas on va chercher les valeurs des positions de (S_i) obtenues par simulation

S1(1150, 1450) ; S2(3350, 1500) ; S3(3550, 3350) ; S4(1300, 4050)

2-Choix des caractéristiques de la configuration de la cellule de (6) capteurs RO

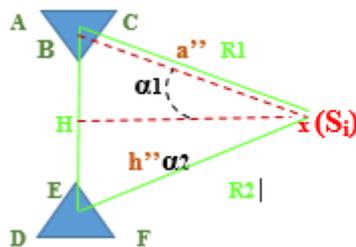


Figure 42 : Cellule (i)de capteurs thermiques (6RO) orientée vers la droite,[38]

Figure d'une cellule (i) générique de Capteurs (Orientée vers la droite)

2-1 Choix de (Hi) est la distance entre les 2 centres O1 ET O2 des triangles du 1^{er} et du 2^{eme} triplet de capteurs

On met : $\boxed{(Hi) = (a'')}$

On suppose : O1 (Si) O2 forme un triangle équilatéral de côté (**a''**) et de hauteur (**h''**) avec, O1 & O2 sont respectivement les centres du 1^{er} et du 2^{eme} triplet de capteurs ABC et DEF

Choix de (a'') ; Calcul de : $\boxed{(h'') = (a'') \cos \alpha}$; avec $\boxed{\cos \alpha = 0.87}$

exemple : (a'')=300 , (h'') = 300 x 0.87=261= ~250

2-2 Choix de (a) : est le côté du Triangle équilatéral de chaque triplet de capteurs ABC et DEF

Calcul de : $\boxed{h=a \cos(\alpha)}$ exemple: a=175 → h=151.55=150

2-3 Calcul des Coordonnées des centres O1 ET O2 du 1^{er} et du 2^{ème} triplet de capteurs :

$O_1(X_{O1}, Y_{O1}) ; O_2(X_{O2}, Y_{O2})$

Etant donné que : $O_1(S_i)O_2$ forme un triangle équilatéral

$$\boxed{X_{O1} = X_{Si} - h''}$$

$$\boxed{Y_{O1} = Y_{Si} + (a''/2)}$$

$$\boxed{X_{O2} = X_{O1}}$$

$$\boxed{Y_{O2} = Y_{O1} + (a''/2)}$$

3-Calcul des coordonnées des capteurs de la cellule (1)

3.1-Calcul des Coordonnées du 1er Capteur A : (XA ,YA)

$$\boxed{X_A = X_{O1} - (a/2)}$$

$$\boxed{X_A = X_D}$$

$$\boxed{Y_A = Y_{O1} + (h/3)} \text{ ainsi on obtient la position de } \boxed{A(X_A, Y_A)}$$

3.2-Coordonnées des Capteurs du 1^{er} Triplet

Soit :A(XA, YA)

$$\boxed{X_B = X_A + (a/2)}$$

$$\boxed{Y_B = Y_A - h}$$

$$\boxed{X_C = X_A + a}$$

$$\boxed{Y_C = Y_A}$$

3.3-Coordonnées des Capteurs du 2^{ème} Triplet

$$\boxed{X_D = X_{O2} - (a/2)}$$

$$\boxed{Y_D = Y_{O2} - (h/3)}$$

$$\boxed{X_E = X_D + (a/2)}$$

$$\boxed{Y_E = Y_D + h}$$

$$\boxed{X_F = X_D + a}$$

$$\boxed{Y_F = Y_D}$$

On obtient : A (XA, YA), B(XB, YB), C(XC, YC),D(XD, YD),E(XE, YE),F(XF,YF) (et imprimer)

4-Détection des températures aux niveaux des capteurs ABCDEF (imprimer tableau de leurs valeurs)

5-Calcul de l'estimation de T(Si) (voir section du principe (GDS)) (et imprimer)

6-Calcul de l'estimation des coordonnées de la source chaleur : (Si)_e

$$\boxed{X_{Si_e} = X_{O1} + ((R_1) \times \cos(\alpha))}$$

$$\boxed{Y_{Si_e} = Y_{O1} - (H/2)}$$

7- Impression de tous les Résultats

8-Comparaison entre la prédiction de la position de la source(Si) à celle estimée (Si)_e

NB : Rappelons que (Si)_e (estimée) est obtenue par le calcul fondée sur l'implémentation de la méthode GDS).

9-Conditions strictes à respecter

a) rapporter les coordonnées des capteurs sans approximations:

- Les coordonnées de (A...O1,, O2,, F) seront calculées et rapportées sous leur forme décimale avec aucun arrondissement ni approximation dans les valeurs des coordonnées pour que le Plot soit parfaitement dessiné avec des triangles parfaitement équilatéraux.
- Mais lors d'extraction des températures (TA, ,,,,,, a TF),on a besoin de prendre la valeur de la température des coordonnées approximées par excès en se référant au plus proche nœud.
- Pas d'approximation pour les paramètres décrites dans l'expression de calcul de (Ts).

b) Contrainte à Respecter : les 2 triangles ne doivent pas se toucher

----> $H > 2x(2h/3)$ ----> $H > 4 (.87xa) /3$ ----> $H > 1.16 x (a)$

$H > 1.16 a$

NB : Dans notre cas on va chercher les valeurs des positions de (Si) obtenues par simulation

(S1) (1150, 1450) temp=44.543152 ; (S2)coord=(3350, 1500) temp=76.033043

(S3)coord=(3550, 3350) temp=64.114830 ; (S4)coord=(1300, 4050) temp=72.695702

Rq : Si nécessaire, on peut faire varier H et (a).

10-Étapes de l'exécution en Matlab de l'algorithme GDSCAN

1. Entrée des données : $S_i (X_{si} , Y_{si}) ,H, (a),$
2. on calcule respectivement la position de O1,A,B,C,D,E,F
3. on extrait les valeurs de TA,TB,TC,TD,TE,TF
4. on calcule (Ts)
5. on calcule la position de (Si)_e
6. on compare la position de (Si) à celle estimée (Si)_e
7. on compare la valeur de (Ts) a celle obtenu par simulation (T).
8. on compare la position de chaque source(Si) à celle estimée (Si)_e

Validation MATLAB de GDSCAN pour la détection de 4 sources de chaleurs sur un CI

Après le développement dans Matlab de l'algorithme GDSCAN pour la détection de 4 sources de chaleurs sur un CI, nous l'avons exécuté sur Matlab et on a obtenu les résultats illustrés dans le tableau qui suit :

Table10: Tableau des résultats de l'exécution sur Matlab de l'algorithme GDSCAN,[38]

Source n°1- estimée		Source n°2- estimée	
Postions S (x , y) en (µm) (micromètres)	[1150, 1450]	Postions S (x , y) en (µm) (micromètres)	[3351, 1500]
Valeurs de température de capteurs en (°C)	VA1=41.80724716 VB1=43.34230042 VC1=43.34314728 VD1=31.86423874 VE1=34.92364502 VF1=35.91907120	Valeurs de température de capteurs en (°C)	VA2=60.25504684 VB2=69.53496552 VC2=68.70415497 VD2=60.24752808 VE2=69.53110504 VF2=68.69589996
Température TS en (°C)	44.5397	Température TS en (°C)	75.13438
Tan alpha 1	-2.0936e+03	Tan alpha 1	12.3203
Tan alpha 2	30.675	Tan alpha 2	71.4758
Alpha 1	-1.5703	Alpha 1	1.4898
Alpha 2	1.5382	Alpha 2	1.5568
RTD(Delta T/delta R)	0.0964	RTD(Delta T/delta R)	0.1281
Source n°3- estimée		Source n° 4 estimée	
Postions S (x , y) en (µm) (micromètres)	[3551 , 3350]	Postions S (x , y) en (µm) (micromètres)	[1301, 4050]
Valeurs de température de capteurs en (°C)	VA3= 43.70559311 VB3= 51.90933990 VC3= 54.38533401 VD3= 43.71912003 VE3= 51.91598511 VF3= 54.39896011	Valeurs de température de capteurs en (°C)	VA4= 52.34053421 VB4= 62.65264130 VC4= 63.61520004 VD4= 52.33092880 VE4= 62.64786530 VF4= 63.60529327
Température TS en (°C)	63.59521	Température TS en (°C)	72.14274
Tan alpha 1	-4.4032	Tan alpha 1	-12.9479
Tan alpha 2	54.6432	Tan alpha 2	66.4025
Alpha 1	-1.3475	Alpha 1	-1.4937
Alpha 2	1.5525	Alpha 2	1.5557
RTD(Delta T/delta)	-0.5162	RTD(Delta T/delta)	0.1787

Estimation des valeurs des Températures des sources de chaleur T_(si) par GDSCAN

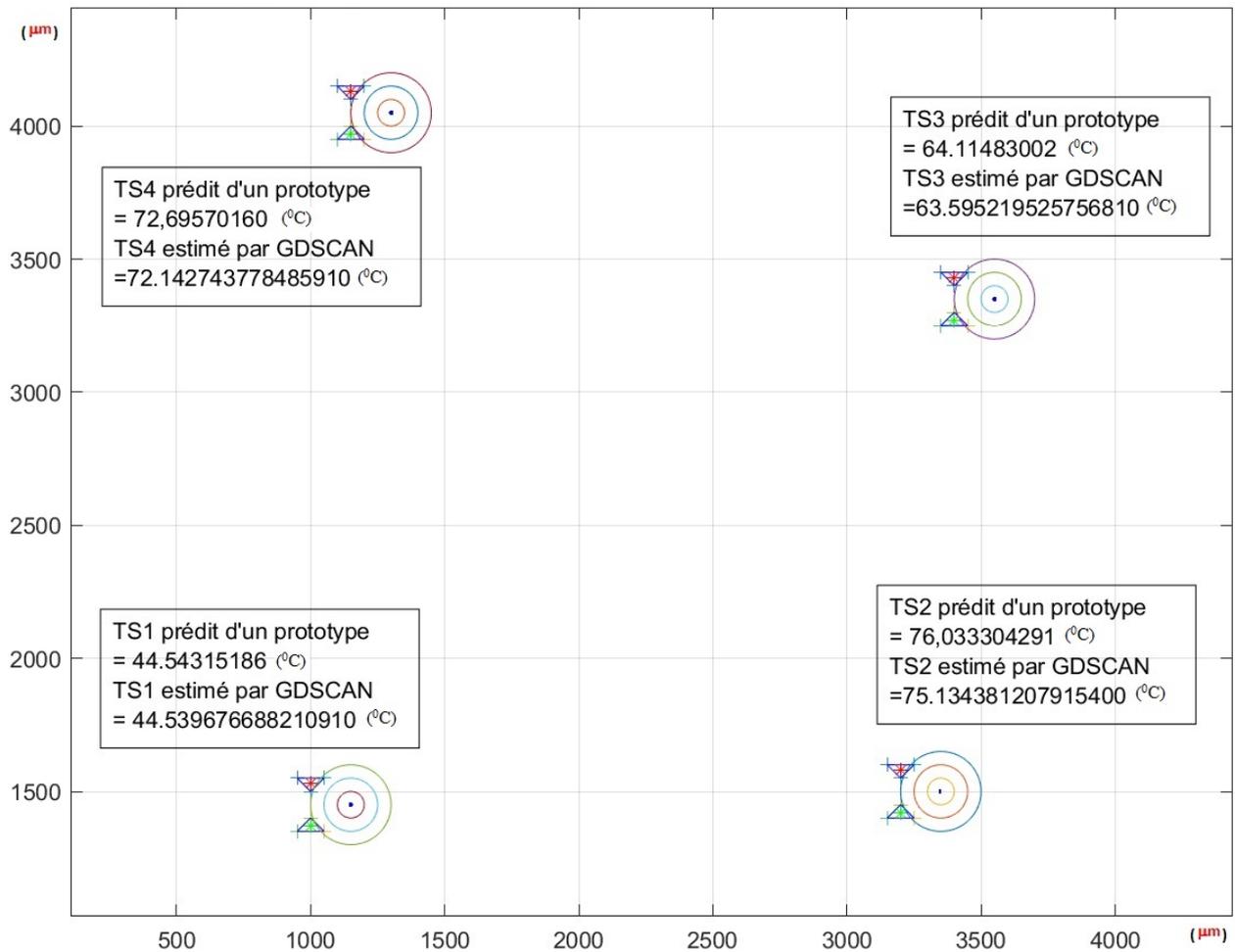


Figure 43: Résultats du GDSCAN et estimation des températures des sources de chaleur

L'estimation des valeurs des températures(T_{si})des sources de chaleur est faites en **degrés Celsius**, [39]

Calcul des erreurs des estimations des valeurs de températures des sources.

$$e_1 = 44.54315 - 44.53967 = 0.00348/44.54315 = 0.0078 = 0.78 \% .$$

$$e_2 = 76.03330 - 75.13438 = 0.89892/76.03330 = 0.0118 = 1.18 \% .$$

$$e_3 = 64.11483 - 63.59521 = 0.51961/64.11483 = 0.0081 = 0.81 \% .$$

$$e_4 = 72.69570 - 72.14274 = 0.55299/72.69570 = 0.0076 = 0.76 \% .$$

Estimation des positions des sources de chaleur $T(s_i)$ par GDSCAN

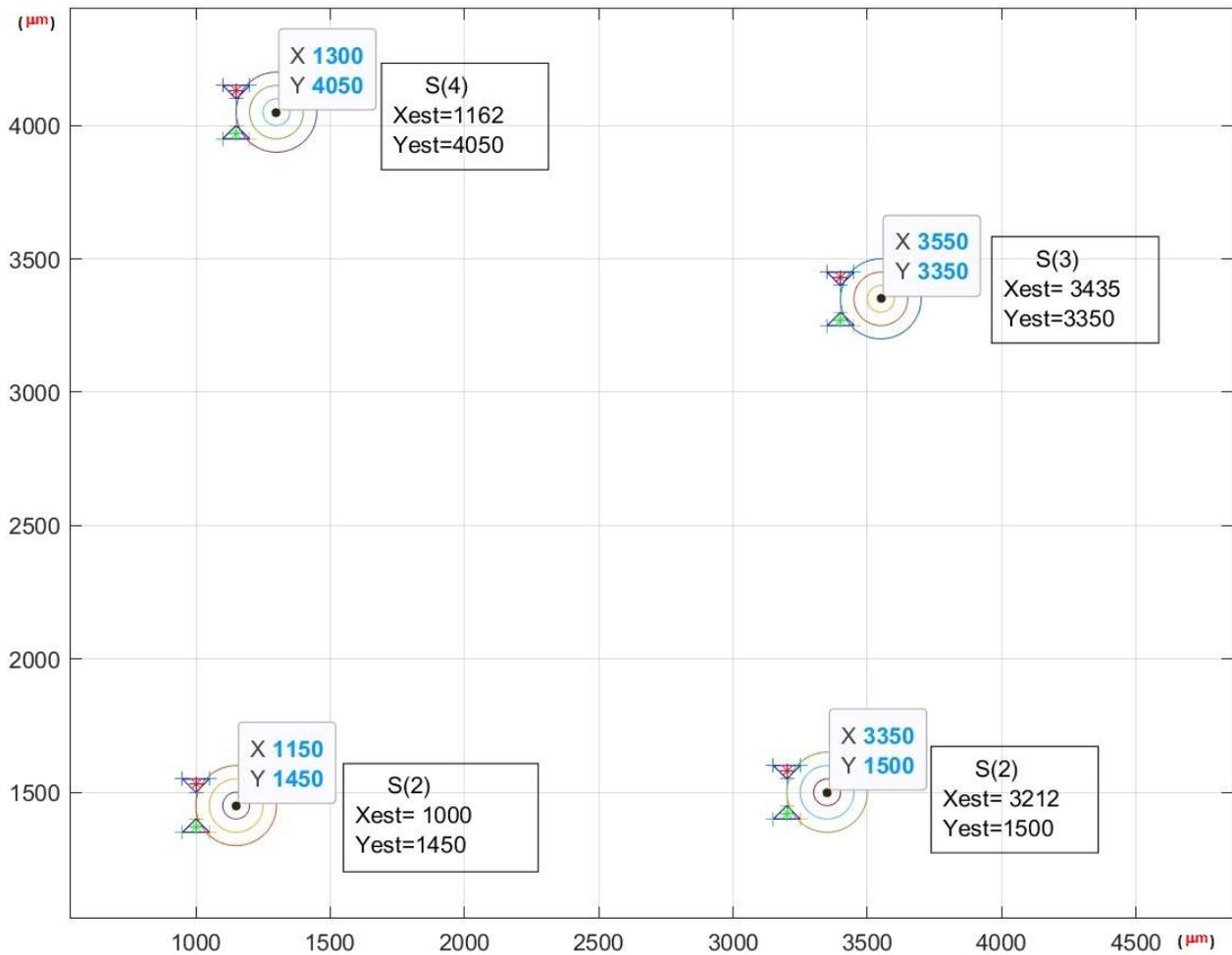


Figure 44: Résultats du GDSCAN pour l'estimation des positions des sources de chaleur

L'estimation des positions des sources de chaleur $S_i (x_i, y_i)$ est faite en (micromètre) [39]

7.9 Programmation en langage C++ de l'Algorithme (GDSCAN).

La programmation de l'Algorithme (GDSCAN) en C++ se résume en une suite de calculs simples, mais elle est plus complexe que la validation Matlab au niveau de la définition des structures de données. Cet aspect influence la finalité des résultats qui sont plus précises en Matlab.

Pour ceci, un diagramme de flux et pseudocode de l'algorithme (GDSCAN) ont été réalisés.

7.9.1 Le diagramme de flux de l'algorithme (GDSCAN)

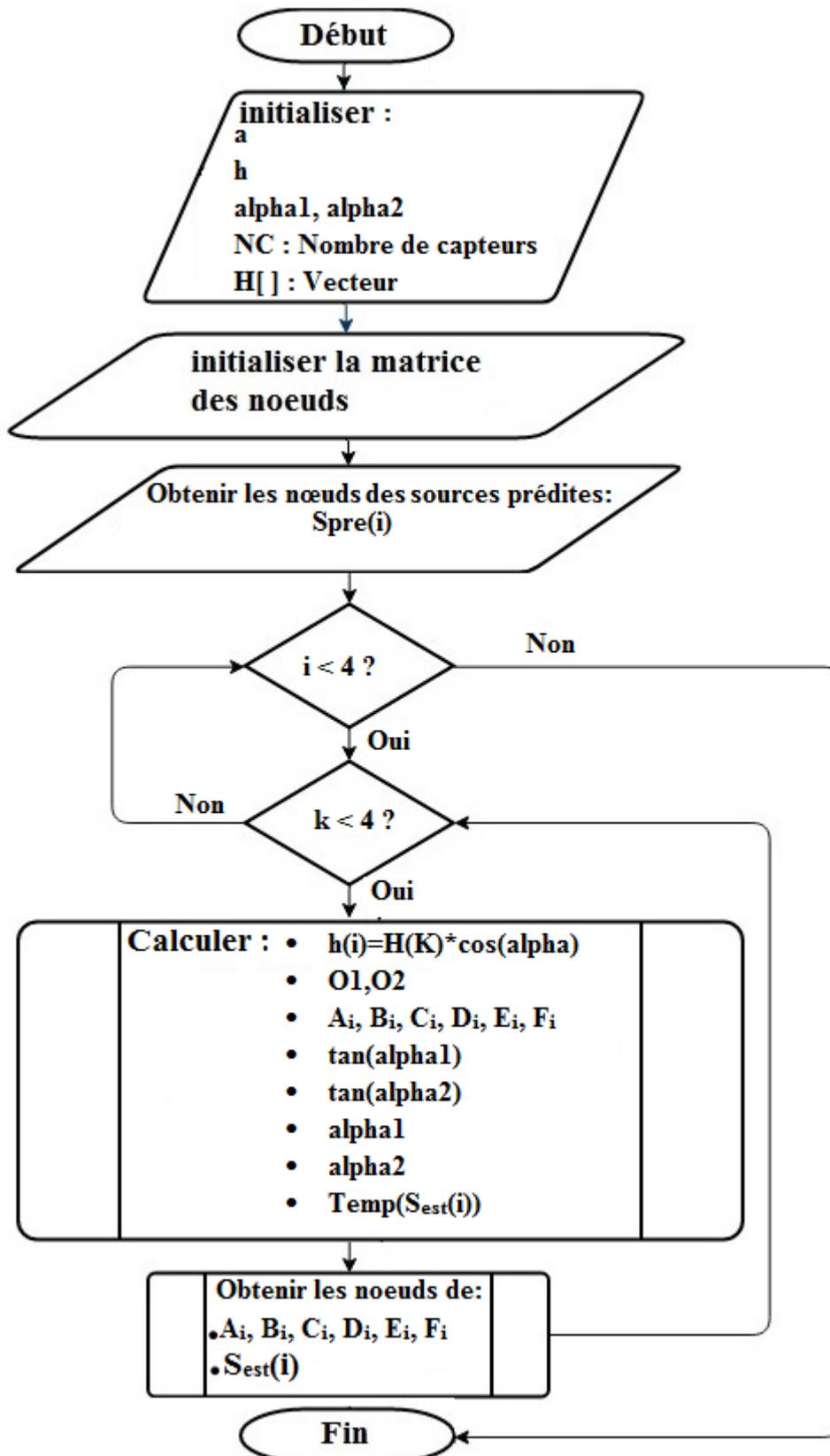


Figure 45:Diagramme de flux détaillé de l'Algorithme (GDSCAN) pour C++

(GDSCAN) est un algorithme de balayage de la surface d'un CI qui est basé sur le principe du GDS pour la détection des pics thermiques dans le cas de sources de chaleur multiples.

7.9.2 Pseudocode de l'Algorithme (GDSCAN)

Dans ce qui suit on détaille le pseudocode de l'algorithme GDSCAN afin de le codifier en C++

Pseudocode de GDSCAN

Début

```

- Choisir H //cote du triangle équilatéral [O1, S, O2]
- hpp = H * cos(alpha) //hauteur de [O1, S, O2]:
    //cos(alpha) = 0.87
- Calculer coordonnées de O1 et O2:
  O1x = Sx - hpp
  O1y = Sy + H/2
  O2x = O1x
  O2y = Sy - H/2
- Calculer coordonnées des capteurs:
  Ax = O1x - a/2
  Ay = O1y + h/3
  Bx = Ax + a/2
  By = Ay - h
  Cx = Ax + a
  Cy = Ay

  Dx = O2x - a/2
  Dy = O2y - h/3
  Ex = Dx + a/2
  Ey = Dy + h
  Fx = Dx + a
  Fy = Dy

- Obtenir températures des capteurs ABCDEF:
  TA = getTemp(Ax, Ay)
  TB = getTemp(Bx, By)
  TC = getTemp(Cx, Cy)
  TD = getTemp(Dx, Dy)
  TE = getTemp(Ex, Ey)
  TF = getTemp(Fx, Fy)

- Estimer TS:
  TS = (((H/a) * (TC - TA) * (tan_alpha1^2 + 1) * (tan_alpha2 + sqrt(3.0))) /
  ((sqrt(3.0) * (1 - tan_alpha1*tan_alpha2)) - (tan_alpha1 + tan_alpha2))) + TA;

```

Fin

Figure 46:Pseudocode détaillé de l'Algorithme (GDSCAN)pour C++

7.9.3 Résultats de l'exécution de l'Algorithme (GDSCAN) en C++

Dans ce qui suit récapitulatif des résultats de l'exécution du programme C++ de GDSCAN

```
*****
S_pre(1) = Node 94762 x 1150 y 1450 T 44.54315186
X(S_pre) & Y(S_pre) from Node matrix
*****
H = 175
S_est(1)->temp = 42.80595
*****
S_pre(2) = Node 94907 x 3350 y 1500 T 76.03304291
X(S_pre) & Y(S_pre) from Node matrix
*****
H = 175
S_est(2)->temp = 71.36054
*****
S_pred(3) = Node 98648 x 3550 y 3350 T 64.11483002
X(S_pre) & Y(S_pre) from Node matrix
*****
H = 175
S_est(3)->temp = 60.10724
*****
S_pre (4) = Node 100017 x 1300 y 4050 T 72.69570160
X(S_pre) & Y(S_pre) from Node matrix
*****
H = 175
S_est(4)->temp = 71.12550
*****
```

Figure 47:Listage des résultats de l'exécution de l'Algorithme (GDSCAN) en C++

7.9.4 Analyse des résultats de l'exécution du GDSCAN) en C++

GDSCAN a été relativement validé par les résultats de son code C++ , mais il faut l'améliorer plus.

En effet , les résultats obtenus par le logiciel en C++ n'étaient pas très proches de ce que nous attendions comme les ont montrés les résultats du logiciel en Matlab.

Nous avons trouvé des résultats assez valides mais qui n'incluaient pas des valeurs hors de la norme.

L'amélioration pourra inclure une révision de l'algorithme ou de la méthode de programmation en C++, ce qui pourraient améliorer les résultats finaux.

7.10 Étude de l'implémentation matérielle du (GDSCAN) sur carte

FPGA

Bien que l'implémentation matérielle du GDSCAN n'est pas de notre charge, mais une étude de l'implémentation matérielle sur une puce sur carte FPGA a été détaillée dans les Annexes (A6) [37].

7.11 Analyse et discussion pour l'algorithme (GDSCAN)

On a réalisé une solution algorithmique prometteuse qui est facile à implémenter et réaliste pour résoudre la problématique de la thèse en concevant, développant et validant l'algorithme GDSCAN

pour la détection de 4 sources de chaleurs. On peut généraliser ceci pour multiple sources de chaleur pourvu que le nombre soit supérieur à « un » (1).

Ainsi, on a réalisé pour l'algorithme GDSCAN :

- Le diagramme de flux de l'algorithme de balayage des cellules de capteurs RO basé sur le GDS pour la détection des pics thermiques dans le cas de sources de chaleur multiples.
- Une programmation en langage C++.
- Une validation numérique par MATLAB.
- Une étude de l'implémentation matérielle sur une puce sur carte FPGA.

On a tiré profit des avantages de tous les techniques algorithmiques de balayage de MHSODA , des techniques expérimentales de l'usage des capteurs en anneau (RO) et de la technique du (GDS) qui ont été vérifiés auparavant pour **garantir la validation de cette 4ème solution algorithmique prometteuse du GDS-étendu : (GDSCAN).**

Néanmoins on peut mentionner une limite de (GDSCAN) est que **par une seule (1) cellule de capteurs thermiques (6RO) l'on ne peut détecter qu'une seule source de chaleur à la fois.**

Conclusion générale :

Cette thèse constitue une contribution importante dans le génie des microsystèmes électroniques avancés et plus particulièrement dans le domaine de la détection des pics thermiques dans le cas des sources multiples de chaleur dans les CI (Circuits Intégrés).

Cette thèse illustre la réalisation d'activités de recherches dans le domaine génie logiciel appliqué aux systèmes avancés électroniques qui vise à résoudre la problématique de la détection de pics thermiques dans le cas de sources multiples de chaleur dans les circuits intégrés. Pour ce faire on a retracé les travaux précédents qui ont constitué l'état de l'art dans la détection des sources de chaleur dans les CI. Par la suite, on s'en est inspiré pour procéder méthodologiquement à bâtir une résolution numérique en FEM pour modéliser et un prototype sur lequel on a simulé quatre sources de chaleurs surgissant à la surface d'un CI. Les résultats de ces simulations ont constitué des données pour explorer des solutions algorithmiques efficace permettant la détection de multiples sources de chaleurs dans les CI tout en précisant, leur nombre, la température maximale au niveau de chaque source et sa position tout en optimisant le nombre de capteurs et permettre un monitoring thermique permanent d'assurer un équilibre de différentielles thermiques dans tout le CI. Ces dernières sont très importantes à détecter puisqu'elles causent des pics de contraintes thermomécaniques transistorisées. Technologiquement nous avons choisi d'utiliser des capteurs en anneau à base de RO (Ring Oscillator).

On a exploré les quatre solutions algorithmiques suivantes : l'algorithme (KPPV-SODA) basé sur les K Plus Proches Voisins, l'algorithme (MSTPDA) basé sur recherche de la Matrice Contour, l'algorithme MHSODA, basé sur le balayage de la surface du CI et enfin l'algorithme GDSCAN basé sur la méthode du GDS et sur le balayage de la surface du CI.

L'algorithme (KPPV-SODA) s'est avéré efficace sur le plan algorithmique et logiciel en tant qu'algorithme générique qui peut tester n'importe quel point du CI avec une présence d'une source de chaleur ou pas.

Mais la faiblesse de cet algorithme réside dans le fait suivant : pour connaître les plus proches pics voisins d'un « pic thermique » détecté déjà sur la surface du CI on est obligé de faire un nombre très excessif de calcul et de comparaisons de distances séparant ce pic au reste de tous les points constituants (10201) sur la surface du CI. Cependant, on doit répéter ceci pour chaque point du CI vis-à-vis de tout le reste des points du CI. Cette multitude excessive d'opérations et de calculs va consommer trop de temps d'exécutions et de calculs ce qui rend l'application de l'algorithme

(KPPV-SODA) trop lente à appliquer avec des délais d'exécution et de réponses trop longs et non efficace sans tenir compte de la vitesse relativement croissante du réchauffement du CI.

MSTPDA est un algorithme de détection thermique par excellence, il est efficace pour la gestion thermique dans les grands circuits LAIC y compris le Wafer ICTM, la détection relativement rapide de plusieurs sources de chaleurs et la génération avec précision des estimations de la température et de la position de chaque source de chaleur et leur nombre total.

L'algorithme MHSODA proposé est excellent dans notre cas, car il est une solution valide qui résout la problématique de la thèse. Elle est simple, rapide, performant, facilement implémentable, ne présente pas de risque de « surchauffe rapide » causé la chaleur générée par les capteurs.

Il est efficace pour la détection thermique et permet d'obtenir des résultats précis du nombre de sources de chaleur ainsi que leurs positions et les températures qui y sont détectées. MHSODA est efficace pour la gestion thermique dans les grands circuits LAIC dans le Wafer ICTM, Mais son désavantage est qu'il utilise 10201 capteurs ce qui n'est pas une solution économique à implémenter dans les petits CI.

Car dans la réalité de l'industrie des microsystemes électronique, on ne va pas planifier l'espace pour intégrer sur la puce 10201 capteurs dans un petit CI à cause des contraintes économiques et opérationnelles de fabrication. Pour ceci on a cherché à optimiser le nombre de capteurs et le réduire au minimum en explorant une quatrième solution algorithmique soit l'algorithme GDSCAN.

L'algorithme GDSCAN comme étant une solution algorithmique prometteuse qu'on a validé numériquement par l'outil mathématique Matlab qui a permis la détection de quatre (4) sources de chaleurs sur un CI. L'algorithme GDSCAN est basé sur la méthode du GDS et sur le balayage de la surface du CI. En effet pour le bâtir, on a tiré profit des avantages :

- Du procédé de balayage (inspiré de l'algorithme MHSODA),
- De l'utilisation de capteurs en anneau (RO)
- De la technique du (GDS) (validé pour la détection d'une seule source de chaleur)

Ces avantages ont été vérifiés expérimentalement auparavant, ce qui a facilité la validation de cette 4ème solution algorithmique prometteuse (GDSCAN) avec des estimations relativement précises sur les positions et les températures des sources de chaleur dont les erreurs ne dépassent pas les 1.5%.

Cette dernière solution nous offre aussi, une configuration de réseau de capteurs fiable lequel inclut quatre (4) cellules avec « un nombre optimal minimum » de 24 capteurs de type Ring Oscillator (24 RO).

Chaque cellule est formée de deux triplet de capteurs distant l'un de l'autre par une distance « H ». Chaque triplet de capteurs forme une architecture trigonométrique spécifique soit : un triangle équilatéral.

Travaux futurs de Recherches

L'implémentation matérielle du (GDSCAN)

Des travaux futurs ont été suggérés pour compléter l'implémentation matérielle sur une puce dans une carte FPGA de l'algorithme du (GDSCAN).

Bien que ça ne fait pas partie de notre sujet de thèse, on a réalisé une étude détaillée de l'implémentation matérielle sur une puce dans une carte FPGA afin de faciliter aux futurs chercheurs et étudiants l'implémentation matérielle de l'algorithme (GDSCAN).

Programmation de l'algorithme GDSCAN en JAVA

Parmi les travaux futurs, on peut envisager de programmer l'algorithme GDSCAN en JAVA pour faciliter sa portabilité pour le monitoring de la gestion thermique dans les CI à distance et en ligne (par internet).

Références

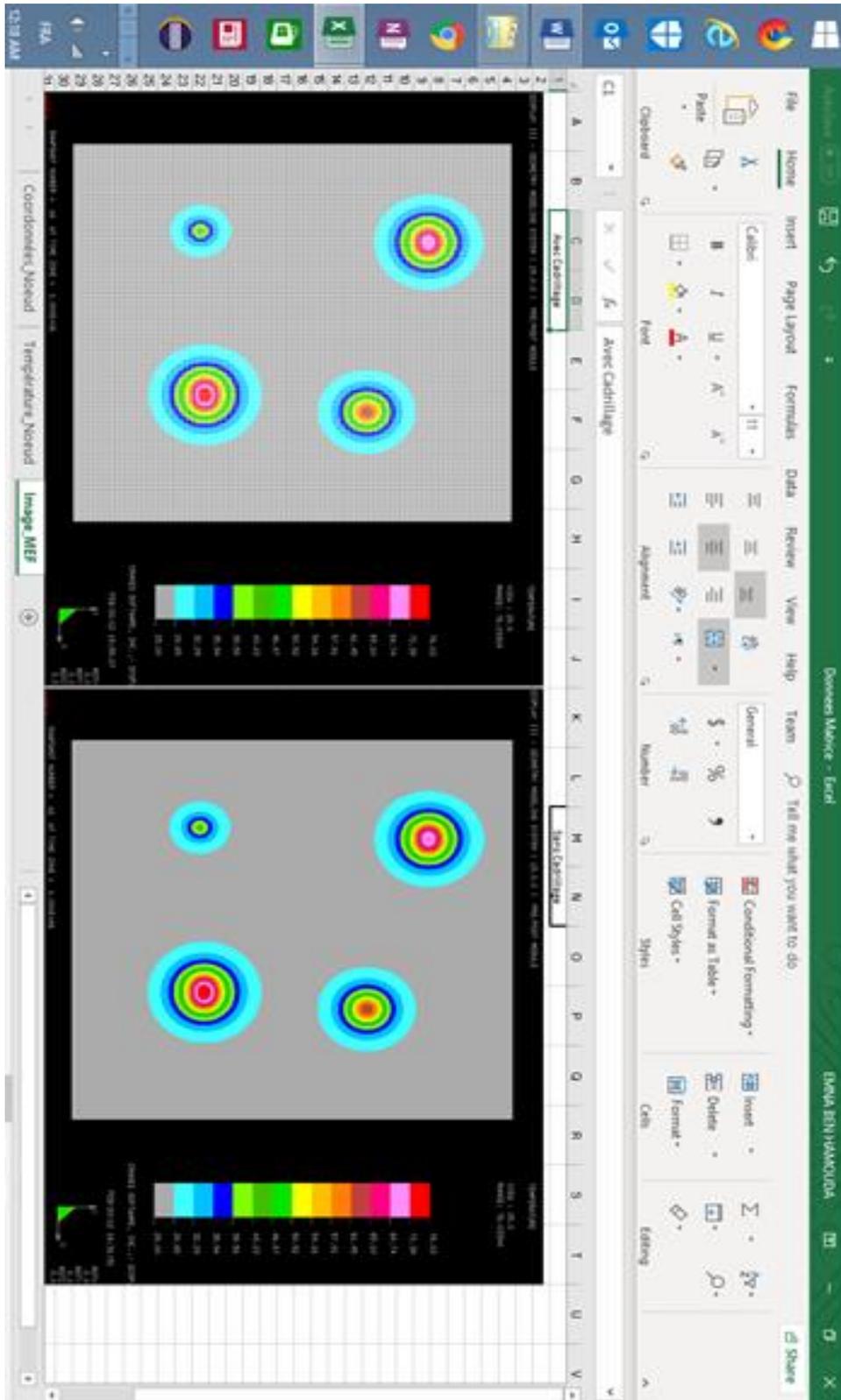
- [1] W. Wójciak and A. Napieralski, "Thermal monitoring of a single heat source in semiconductor devices — the first approach," *Microelectronics Journal*, vol. 28, pp. 313-316, 1997.
- [2] Janicki, M, Zubert, M, & Napieralski, A, "Application of inverse problem algorithms for integrated circuit temperature estimation", *Microelectronics journal*, pp. 1099-1107, (1999).
- [3] Janicki M and Napieralski A, Real time temperature estimation of heat sources in integrated circuits with remote temperature sensors, *Journal of Physics: Conference Series*, Vol. 124 (2008) 012027, pp. 1-11.
- [4] E. Boemo and S. Lopez-Buedo, "Thermal Monitoring on FPGAs using Ring-Oscillators", *Lecture Notes in Computer Science*, No.1304, pp.69-78, Berlin: Springer-Verlag, 1997.
- [5] S. Lopez-Buedo, J. Garrido, and E. I. Boemo, "Dynamically inserting, operating, and eliminating thermal sensors of FPGA-based systems," *Components and Packaging Technologies, IEEE Transactions on*, vol. 25, pp. 561-566, 2002.
- [6] E. Boemo and S. Lopez-Buedo, "Thermal verification on FPGAs," in *NORCHIP Conference, 2005. 23rd*, 2005, pp. 48-53.
- [7] Lakhssassi, A, Bougataya, M, Boustany, C, & Massicotte, D, "Thermal stress monitoring using gradient direction sensors", In *Circuits and Systems (NEWCAS-TAISA), 6th IEEE International Northeast Workshop*, pp. 177-180, (June. 2008).
- [8] A. Lakhsasi, M. Bougataya, D. Massicotte: Practical approach to gradient direction sensor method in very large scale integration thermomechanical stress analysis, *J. Vac. Sci. Technol. A* 24(3), pp. 758-763, May 2006.
- [9] Bougataya, M, Lakhsasi, A, Savaria, Y & Massicotte, D, "Thermomechanical Stress Analysis of VLSI devices by partially coupled finite element methods", *IEEE Proceedings; ISBN*, pp. 509- 513, (May. 2004).
- [10] Sayde, M, Lakhssassi, A, Bougataya, M, Terkawi, O, & Blaquiere, Y, "SoC systems thermal monitoring using embedded sensor cells unit", In *Circuits and Systems (NWSCAS), 55th IEEE International Midwest Symposium*, pp. 1052-1055, (Aug. 2012).

- [11] Oukaira, Aziz, Ouafaa Ettahri, Ahmed Lakhssassi. Modelling and FPGA Implementation of a Thermal Peak Detection Unit for Complex System Design. *International Journal of Advanced Computer Science and Applications (IJACSA)* 8. 6 (2017) :
- [12] M. Bougataya, A. Lakhssassi, and D. Massicotte, "Steady State Thermomechanical Stress Prediction for Large VLSI circuits using GDS Method," in *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, 2006, pp. 1205-1209.
- [13] Bougataya, M, Lakhsasi, A, & Massicotte, D, "VLSI thermo-mechanical stress analysis by gradient direction sensor method", In *Electrical and Computer Engineering, IEEE Canadian Conference*, pp. 710-713, (May. 2015).
- [14] "Handbuch der Lehre von der Verteilung der Primzahlen", Edmund Landau
- [15] [Internet]
<https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/>
- [16] Lopez-Buedo S., Garrido J. and Boemo E. Thermal Testing on Reconfigurable Computers. *IEEE Design & Test of Computers*, 17, I(Jan.-Mar.) 2000, 84-90.
- [17] Oukaira, A, Lakhssassi, A, Fontaine, R, & Lecomte, R, "Thermal Model Development for LabPET II Scanner Adapter Board Detector Module", *Proceedings of the COMSOL Conference*, pp. 1-5, (Oct. 2015).
- [18] Lakhssassi, A, M, Bougataya, "VLSI Thermal Analysis and Monitoring", In-Tech, Kirchengasse, A-1070 Vienna, Austria, pp. 441-456, (2009).
- [19] Oukaira, Aziz; Mellal, Idir; Ettahri, Ouafaa; Kengne, Emmanuel; Lakhssassi, Ahmed "Thermal Management and Monitoring Based on Embedded Ring Oscillator Network Sensors for Complex System Design", *International Journal of Computer Engineering and Information Technology*; Dubai, UAE, Vol. 9, Iss. 7, (Jul 2017): 127-134.
- [20] Internet:
<https://www.scopus.com/authid/detail.uri?authorId=24923171000>
- [21] - Janicki M, De Mey G and Napieralski A 2007 Thermal analysis of layered electronic circuits with Green's functions *Microelectron. J.* 38 177-84
- [22] - Janicki M, De Mey G and Napieralski A 2002 Application of Green's functions for analysis of transient thermal states in electronic circuits *Microelectron. J.* 33 733-8
- [23] - Janicki M, De Mey G and Napieralski A 2002 Transient thermal analysis of multilayered structures using Green's functions *Microelectron. Reliab.* 42 1059-64
- [24] - Janicki M, De Mey G and Napieralski A 2007 Thermal analysis of layered electronic circuits with Green's functions *Microelectron. J.* 38 177-84

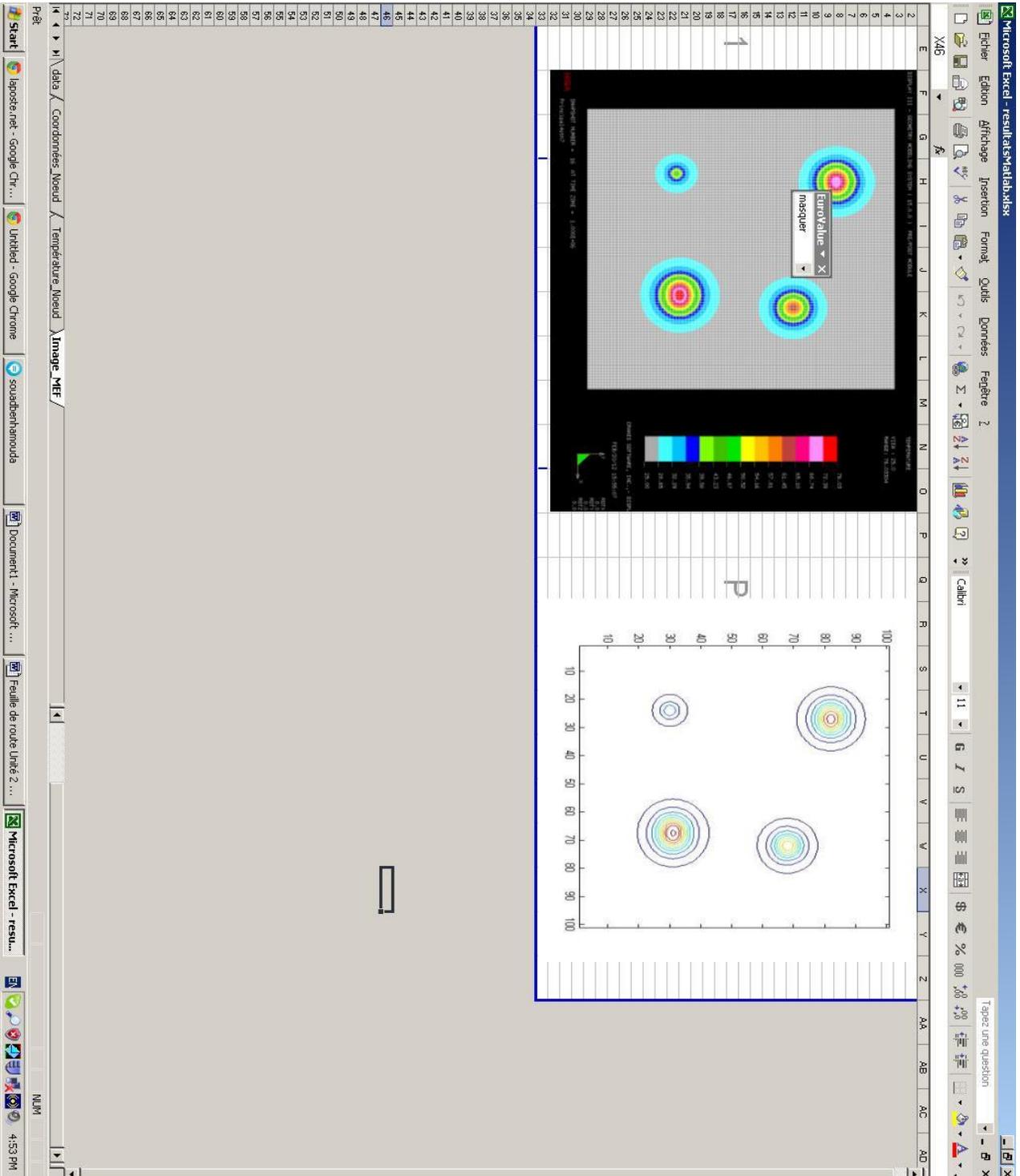
- [25] - Janicki M and Napieralski A 2002 Inverse heat conduction problems in electronic circuits Proc. 9th Int. Conf. Mixed Design of Integrated Circuits and Systems (Wroclaw, Poland, 20-22 June 2002) pp 385-8
- [26] - Janicki M, Zubert M and Napieralski A 1998 Application of inverse heat conduction methods in temperature monitoring of integrated circuits Sensor Actuat A 71 51-7
- [27] - Janicki M, Zubert M and Napieralski A 1999 Application of inverse problem algorithms for integrated circuit temperature estimation Microelectron. J. 30 1099-107
- [28] - Emmanuel, Kengne; Fathi, Ben Hamouda and Ahmed Lakhssassi, "Extended generalized Riccati equation mapping for thermal traveling-wave distribution in biological tissues through a bio-heat transfer model with linear/quadratic temperature-dependent blood perfusion", Journal of Applied Mathematics, 4(10), 1471-1484, 2013
- [29] - Ferchichi, H., & Akaichi, J. (2013, May). A novel approach for the continuous search of the K nearest neighbors on road networks. *In Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on(pp. 878-881). IEEE.*
- [30] - Berchtold, S., Böhm, C., Keim, D. A., & Kriegel, H. P. (1997, May). A cost model for nearest neighbor search in high-dimensional data space. *In Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems (pp. 78-86).ACM.*
- [31]- R. Zhong, G. Li, K.-L. Tan, and L. Zhou. (2013) G-tree: an efficient index for knn search on road networks. *In CIKM*, pages 39–48.
- [32]- Song. Z and Roussopoulos. N, (2001). K-Nearest Neighbor Search for moving query Point, *In SSTD*, Heidelberg, Berlin.
- [33] Beckmann, N, Kriegel, H.P, Schneider, R. and Seeger, B. (1990) An efficient and robust access method for points and rectangles, *In Proc.ACMSIGMOD*, Atlantic City, NJ, USA,pp.47–57, 1990.
- [34]- Lopez-Buedo, S, & Boemo, E, "Making visible the thermal behaviour of embedded microprocessors on FPGAs: a progress report", In Proceedings of the ACM/SIGDA 12th international symposium on Field programmable gate arrays, pp. 79-86, (2005).
- [35]- E. Boemo and S. Lopez-Buedo, "Thermal Monitoring on FPGAs using Ring-Oscillators", Lecture Notes in Computer Science, No.1304, pp.69-78, Berlin: Springer-Verlag, 1997.
- [36] [Internet] <http://www.napieralski.dmcs.pl/>
- [37] Guide d'implémentation d'un RO, Publication interne du laboratoire LIMA de l'UQO.
- [38] Rapport de projet de thèse de Fathi Ben Hamouda déposé le 21.12.2018 soumis au Département d'ingénierie et Informatique de l'UQO (Université du Québec en Outaouais).

Annexes

- Annexe A1 : Image thermique obtenue par NISA



- Annexe A2 : Comparaison Image NISA Vs Image Matlab (MSTPDA)



- Annexe A3 : Résultat de l'exécution de l'Algorithme MHSODA

The screenshot shows the Eclipse IDE interface. The main editor displays the source code of `MHSODA.cpp`. The console window shows the output of the program, which is summarized in the following table:

Node num	coord	temp
94762	(1150, 1450)	4.543152
94907	(3350, 1500)	76.033043
98648	(3550, 3350)	64.114830
100017	(1300, 4050)	72.695702

The console output also includes a summary:

```

SUMMARY:
Number of Peaks: 4
Highest Peak: Node num=94907 coord=(3350, 1500) temp=76.033043
  
```

A blue oval highlights the console output, and a separate window titled "Peaks informations" displays the same data in a structured format:

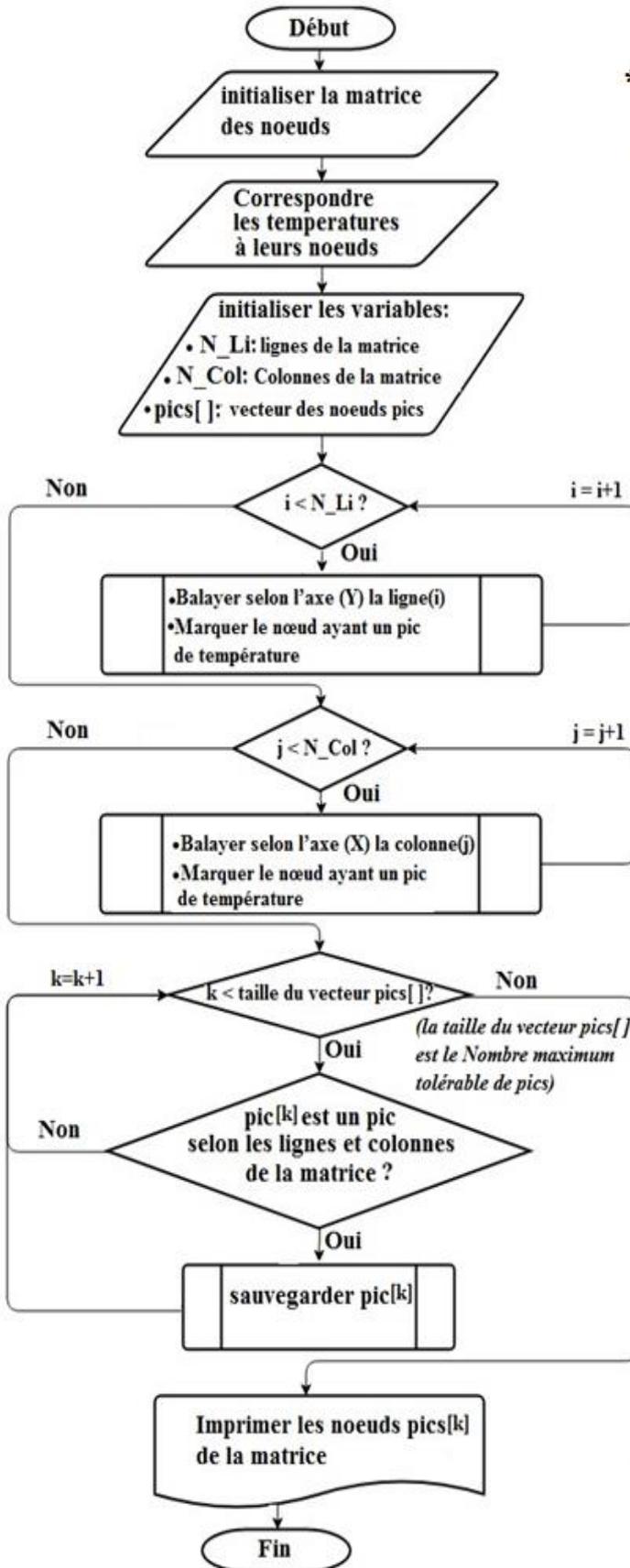
```

Peaks informations
-----
Node num: 94762 coord: (1150, 1450) temp: 4.543152
Node num: 94907 coord: (3350, 1500) temp: 76.033043
Node num: 98648 coord: (3550, 3350) temp: 64.114830
Node num: 100017 coord: (1300, 4050) temp: 72.695702
  
```

● **Annexe A4 : Articles de l'auteur :**

- 1) Emmanuel, Kengne; Fathi, Ben Hamouda and Ahmed Lakhssassi, "Extended generalized Riccati equation mapping for thermal traveling-wave distribution in biological tissues through a bio-heat transfer model with linear/quadratic temperature-dependent blood perfusion", Journal of Applied Mathematics, 4(10), 1471-1484, 2013.
- 2) Fathi Ben Hamouda, Aziz Oukaira, Ahmed Lakhssassi, Multi-Heat Sources Silicon-Die Thermal Monitoring Using Embedded Sensor Cells Unit, Athens Institute for Education and Research, An Annual International Conference on Computer & Software Engineering, 22-25 July 2019, Athens, Greece.
 - Article en soumission
- 3) Fathi Ben Hamouda, Aziz Oukaira, Ahmed Lakhssassi, Multi-Heat Sources Silicon-Die Thermal Monitoring Using Embedded Sensor Cells Unit, (en consideration pour possible publication dans Athens Journal of Technology & Engineering e-ISSN: 2241-8237 (listed in the National Library of Greece) DOI: 10.30958/ajte.

• **Annexe A5 : diagramme de flux Algorithme de balayage MHSODA**



***Partie A : lecture des fichiers des données et construction des Matrices Données**

***Partie B : Recherche des valeurs Pics thermiques !**

+ Recherche des nœuds des pics thermiques par rangée

+ Recherche des nœuds des pics thermiques par Colonne

***Partie C : imprimer les résultats des pics thermiques à l'écran**

• **Annexe A6 : Étude technique de l'implémentation du GDSCAN sur une carte FPGA**

Objectif Spécifique de l'implémentation du l'algorithme GDSCAN sur une carte FPGA

L'objectif est de réaliser l'implémentation sur FPGA d'une configuration de 24 ROs configurés en 8 cellules de 3 ROs chacune à proximité des sources de chaleur.

Voici les étapes pour l'implémentation :

1)Partie source de chaleur : Selon la photo (du prototype présenté pour la validation de l'algorithme GDSCAN) de la distribution de chaleur au niveau du dé disponible placer sur un FPGA manuellement 4 sources de chaleur éloignées chacune par rapport à l'autre. Les sources de chaleur peuvent être sous forme de petits modules ou compteurs 8 bits qu'on peut démarrer et qui finissent par dissiper de la chaleur aux quatre coins du CI.

2)Partie cellules RO : Selon la photo placer sur le FPGA manuellement comme indiqué sur la photo à proximité des 4 sources de chaleur 3 ROs placés chacune au sommet d'un triangle équilatéral comme cela est prévu pour la détection par la méthode GDS de la position d'une source de chaleur et sa valeur dans le cas d'une seule source.

3)Partie compteur de fréquence : sur la carte FPGA, programmer un compteur de fréquence pour afficher les résultats des 4 sources de chaleur dans l'afficheur 8segments une par une avec un bouton poussoir.

4)Partie LUT fréquence -température: Selon les résultats des oscillateurs RO construire une LUT (Look-UP Table) fréquence –température pour la correspondance.

Méthodologie : On va procéder comme suit :

1. Il est mieux de commencer par l'Implémentation sur une carte FPGA d'une cellule six oscillateurs en anneaux (RO) pour détecter une seule source de chaleur
2. Puis il faut effectuer l'Implémentation manuelle sur une carte FPGA de quatre (4) cellules de capteurs thermiques, chacune contenant six oscillateurs en anneaux (RO) pour détecter quatre (4) sources de chaleur

L'implémentation avec la technologie Xilinx serait plus simple qu'avec la technologie Altera plus précisément sur la carte FPGA et avec le logiciel de conception.

Le professeur encadreur, Dr Ahmed Lakhssassi nous a fourni un guide pratique pour l'implémentation d'une cellule de six oscillateurs en anneaux.

L'implémentation des cellules de détection thermique sur FPGA

Choix entre deux options de cette implémentation

Il est difficile de favoriser le choix d'une option par rapport à l'autre entre les deux options possibles pour faire cette implémentation, car elles se défendent selon les moyens disponibles. Dans le cas présent, l'on a le choix entre réaliser l'implantation du projet sur une carte FPGA

D'Altera et une carte SPARTAN 3E de Xilinx. Bien qu'on soit déjà familiarisé à la carte FPGA et qu'elle soit performante, le meilleur des choix est d'utiliser la carte SPARTAN 3E pour l'implantation, car on dispose grâce au laboratoire LIMA d'un guide d'implantation d'oscillateur en anneaux pour la SPARTAN.

Codes d'implémentations

L'implémentation se présente sous la forme de différents codes. L'ensemble de ces codes implémentés sur la carte DE1 est composé de deux (2) codes fondamentaux que sont :

- Le code des oscillateurs en anneaux
- Le code des sources de chaleur

Concernant le code des oscillateurs en anneaux, il s'agissait de réaliser un schéma bloc sur Quartus Prime composé essentiellement de deux parties importantes que sont : des ensembles de trois inverseurs groupés et une porte NAND inversible. Ces deux parties se présentent conformément aux figures (a) et (b) suivantes.

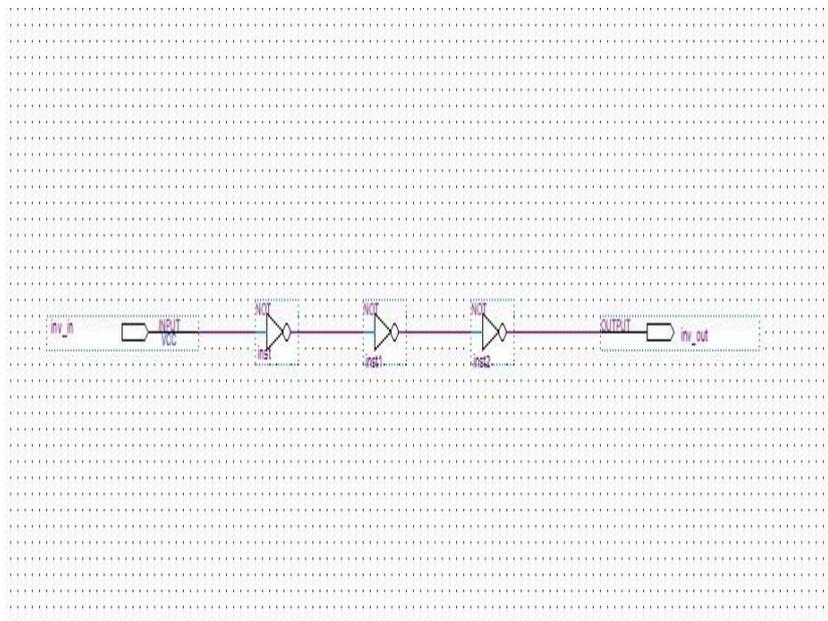


Figure (a) Schéma du triple inverseur

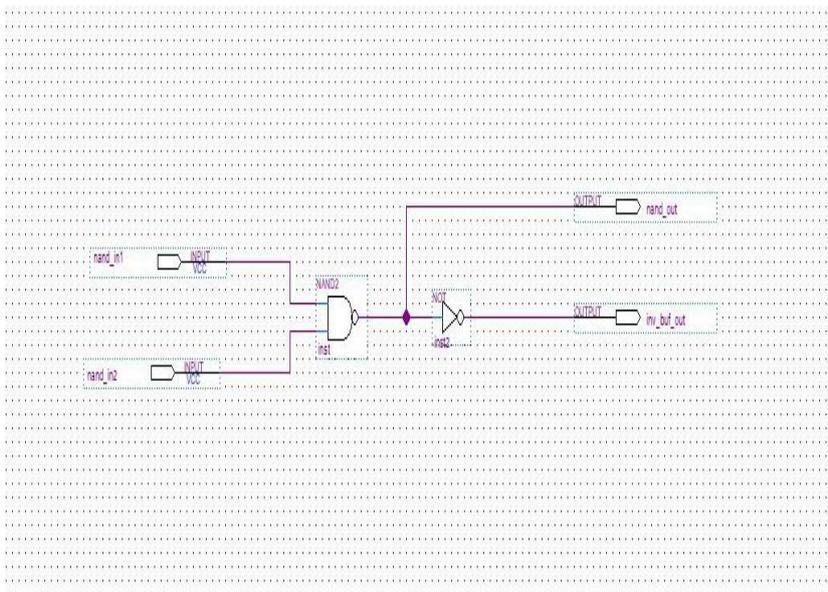


Figure (b) Schéma de la porte NAND inversible et non inversible

Ces figures montrant respectivement la triple inversion d'un signal d'entrée et une porte NAND dont la sortie peut être obtenue inversée et non inversée forment ensemble conformément à la figure (c) l'oscillateur en anneaux type de ce projet.

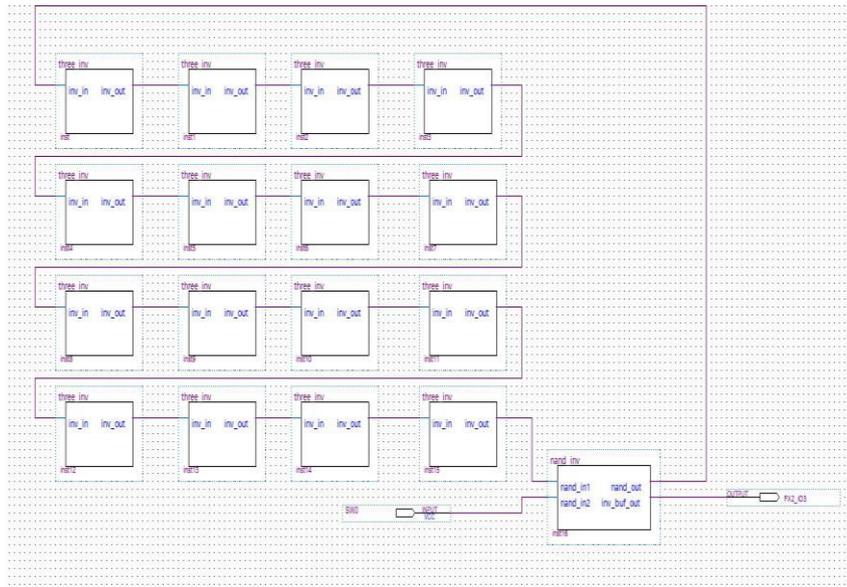


Figure (c) Schéma d'un oscillateur en anneau.

Cet oscillateur servira à capter les températures à certains endroits du dé de silicium grâce à la fréquence du signal obtenue en sortie de l'oscillateur en anneau. Aussi l'on peut voir à la figure suivante le placement manuel sur la carte d'un RO grâce à la fonction Chip planner du logiciel Quartus Prime.

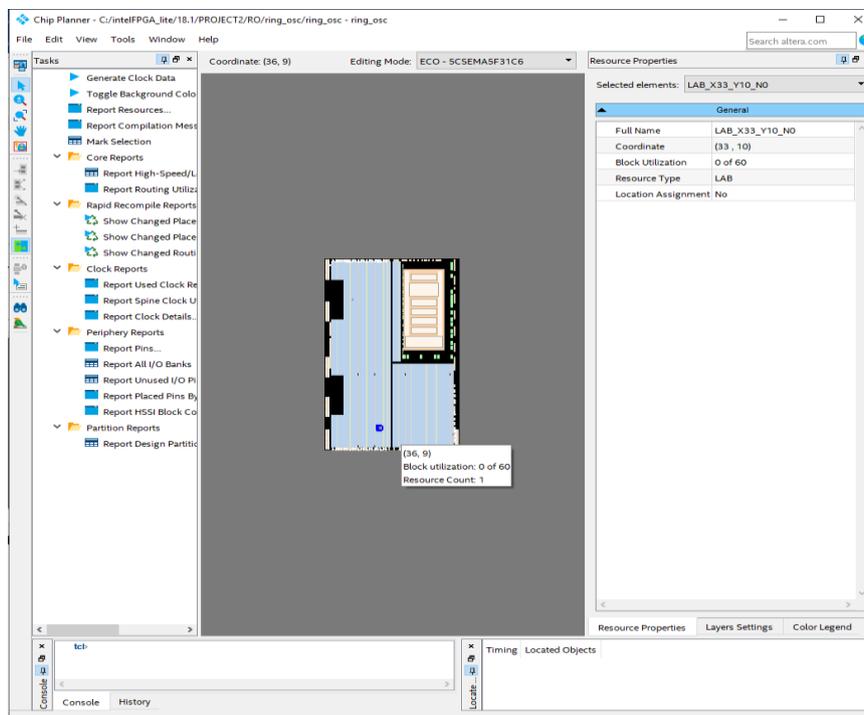


Figure (d) Placement manuel d'un RO sur la carte DE1 à la position (36, 9)

En dernier, le code des sources de chaleur consiste en l'implémentation d'un compteur infini ayant 9 à 12 bits.

Pour une implémentation matérielle, on dispose composantes suivantes :

- Carte SPARTAN 3E de Xilinx
- Câble USB et ordinateur

Vue les contraintes de rupture de stock de matériel électronique au laboratoire, pour l'implémentation du GDSCAN sur FPGA, au lieu d'utiliser (la carte SPARTAN 3E et le logiciel ISE Project), on peut utiliser le matériel disponible au laboratoire LIMA :

- Carte Altera DE1
- Logiciel Quartus Prime
- Câble USB et ordinateur

<p><u>Conclusion</u> : Cette étude technique [37] facilitera au futur chercheurs du laboratoire LIMA d'implémenter l'algorithme GDSCAN.</p>

● **Annexe A7 : ABSTRACT de l'article relative à la thèse**

<u>Athens Institute for Education and Research</u>	
Abstract Submitting Form	
Conference	<u>An Annual International Conference on Computer & Software Engineering 22-25 July 2019, Athens, Greece</u>
Title of Paper	Multi-Heat Sources Silicon-Die Thermal Monitoring Using Embedded Sensor Cells Unit
AUTHORS: Fathi Ben Hamouda, Aziz Oukaira and Ahmed Lakhssassi	
University/ Organization	UNIVERSITY OF QUEBEC AT OUTAOUAIS
Country	CANADA
<u>Abstract</u>	
<p>Nowadays, real-time thermal monitoring is essential in integrated circuit (IC) and VLSI chip which are a multilayer structure and a stack of different materials. The evolution of the integrated circuits industry (IC) during the decade has been so rapid that it is possible to integrate complex systems on a single chip (SOC, system on Chip). This trend towards increasingly high levels of integration is driven by the need for increasingly efficient systems and thus dissipating enormous power densities. Currently researchers have already designed an algorithm for detecting thermal peaks in the case of a single heat source based on the GDS technique(Gradient Direction Sensors). However, this does not solve the problem of thermal peaks monitoring in LAIC circuits. In this paper we presents an algorithmic and the experimental result of multi-sources silicon-die thermal monitoring method using embedded sensor cells unit. The methodology that has been used is based on the generalized GDS technique for the case of multiple heat sources. The test results of a configuration of embedded four (4) GDS sensor cells unit has been proposed for the detection of thermal peaks in the case of multiple sources. Our results shows that our algorithmic solution give a satisfactory thermal peak prediction with less than 1.2 % error.</p>	
Keywords (at least three)	Keywords: GDS method , GDS generalized method , KNN algorithm, contour matrix (MC), Euler number (r), algorithm of scanning, thermal peak, heat source, Isotherm, RO sensors, ring oscillator

• Annexe A8 : Affiche scientifique relative à la Thèse



UNIVERSITÉ
DU QUÉBEC
EN OUTAOUAIS

Etude de détection de pics thermiques dans le cas des sources multiples de chaleur dans les circuits intégrés



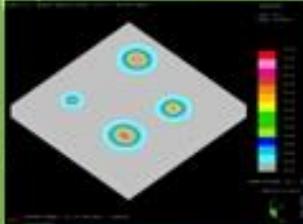
RESMIQ
Regroupement Stratégique
en Microélectronique du Québec

Fathi Ben Hamouda¹, A. Lakhssassi¹,
¹ Université du Québec en Outaouais, Gatineau, QC, Canada

MODÈLE & PROTOTYPE

PROBLÉMATIQUE

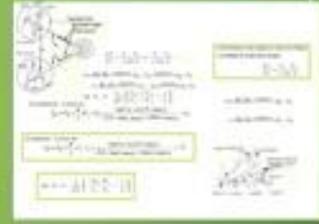
PRINCIPE DU « GDS »



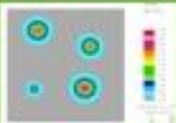
L'évolution de l'industrie des circuits intégrés (CI) provoque l'augmentation critique de la température au fur et à mesure que l'augmentation de densité de puissance diminue des processeurs et par conséquent à une dégradation graduelle des performances des microsystèmes électroniques avec la diminution de la taille des CI.

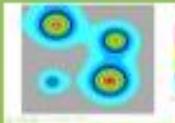
OBJECTIF

- Conseils à étudier et concevoir un algorithme de détection de pics thermiques dans le cas des sources multiples de chaleur.
- Conception de l'algorithme basée sur une configuration en un réseau de capteurs regroupés en cellules pour capter les pics thermiques avec le minimum de cellules sur la grandeur de la puce.



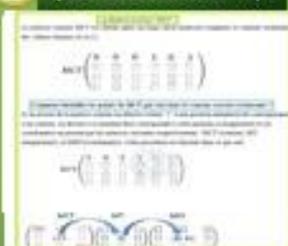
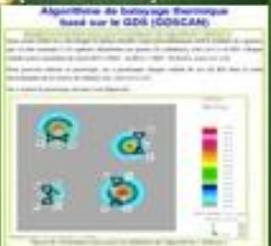
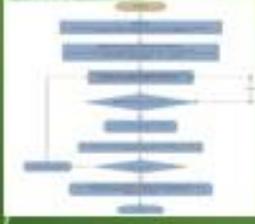
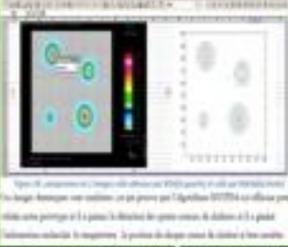
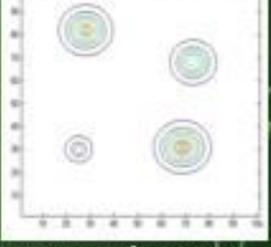
EVOLUTION THERMIQUE DANS LE (CI)







4 SOLUTIONS ALGORITHMIQUES DE DÉTECTION THERMIQUE DANS LES CI

	1 Solution basé sur (KPPV)	2 Solution basé sur (MATRICE CONTOUR)	3 Solution basé sur (SCAN/BALAYAGE)	4 Solution basé sur (GDS étendu)
PRINCIPES				
DIAG. FLUX				
RÉSULTATS				
CONCLUSION	<p>En conclusion sur le plan algorithmique et expérimentale l'algorithme MHSODA constitue une solution implémentable valide et suffisante pour résoudre la problématique de la détection des pics thermiques dans le cas de sources de chaleur multiple dans les CI. Mais pour offrir une solution implémentable et économique pour l'industrie électronique, on doit continuer à valider l'algorithme (GDSscan). Les algorithmes (KPPV-SODA et MSTPDA) sont insoutenables.</p>			

- **Annexe A9 : Étude Bibliométrique sur la détection thermique dans les(CI)**

Scopus Preview

10 documents have cited:

Thermal monitoring of a single heat source in semiconductor devices - The first approach

Wojciak W., Napieralski A.

(1997) *Microelectronics Journal*, 28 (3), pp. 313-316.

[Set feed](#)

You are in Preview mode, only the first 20 documents are visible.

[Search within results...](#)



[Analyze search results](#)

[Refine results](#)

All [v](#)

[Export](#)

[Download](#)

[View citation overview](#)

	Document title	Authors	Year	Source	Cited by
<input type="checkbox"/> 1	PTAT sensor for chip overheat protection	Szermer, M., Janicki, M., Pietrzak, P., Kulesza, Z., Napieralski, A.	2010	Annual IEEE Semiconductor Thermal Measurement and Management Symposium	0
	View abstract  Related documents				
<input type="checkbox"/> 2	Practical realization of PTAT sensor for ASIC overheat protection	Szermer, M., Janicki, M., Kulesza, Z., Napieralski, A.	2009	15th International Workshop on Thermal Investigations of ICs and Systems, THERMINIC 2009	1
	View abstract  Related documents				
<input type="checkbox"/> 3	Thermal stress monitoring using gradient direction sensors	Lakhssassi, A., Bougataya, M., Boustany, C., Massicotte, D.	2008	2008 Joint IEEE North-East Workshop on Circuits and Systems and TAISA Conference, NEWCAS-TAISA	4
	View abstract  Related documents				
<input type="checkbox"/> 4	VLSI thermal stress monitoring using GDS method	Lakhsasi, A., Bougataya, M., Boustany, C., Massicotte, D.	2008	Canadian Conference on Electrical and Computer Engineering	2
	View abstract  Related documents				
<input type="checkbox"/> 5	Steady state thermo-mechanical stress prediction for large VLSI circuits using GDS method	Bougataya, M., Lakhsasi, A., Daniel, M.	2007	Canadian Conference on Electrical and Computer Engineering	3
	View abstract  Related documents				
<input type="checkbox"/> 6	Practical approach to gradient direction sensor method in very large scale integration thermomechanical stress analysis	Lakhsasi, A., Bougataya, M., Massicotte, D.	2006	Journal of Vacuum Science and Technology A: Vacuum, Surfaces and Films	6
	View abstract  Related documents				

	Document title	Authors	Year	Source	Cited by
<input type="checkbox"/> 7	VLSI thermo-mechanical stress analysis by gradient direction sensor method	Bougataya, M., Lakhsasi, A., Massicotte, D.	2005	Canadian Conference on Electrical and Computer Engineering	0
	View abstract  Related documents				
<input type="checkbox"/> 8	Laser probes for the thermal and thermomechanical characterisation of microelectronic devices	Claeys, W., Dilhaire, S., Jorez, S., Patiño-Lopez, L.-D.	2001	Microelectronics Journal	18
	View abstract  Related documents				
<input type="checkbox"/> 9	Localisation of heat sources in electronic circuits by microthermal laser probing	Dilhaire, S., Schaub, E., Claeys, W., Altet, J., Rubio, A.	2000	International Journal of Thermal Sciences	13
	View abstract  Related documents				
<input type="checkbox"/> 10	Fault localisation in ICs by goniometric laser probing of thermal induced surface waves	Dilhaire, S., Altet, J., Jorez, S., (...), Rubio, A., Claeys, W.	1999	Microelectronics Reliability	7
	View abstract  Related documents				

Display: 20  results per page **1**

 Top of page