



U n i v e r s i t é d u Q u é b e c e n O u t a o u a i s

Mémoire présenté comme exigence partielle du programme de maîtrise en science et
technologie de l'information

Méthode de détection de la contrainte
thermomécanique dans les
microsystèmes intégrés

Par

Michel Saydé

Avril 2013

Jury d'évaluation

Président du Jury : Dr Marek Zaremba

Membre du Jury : Dr Emmanuel Kengne

Directeur de recherche : Dr Ahmed Lakhssassi

REMERCIEMENTS

Nous remercions Gestion TechnoCap Inc. pour nous avoir accordé cette opportunité précieuse de travailler sur le projet DreamWafer.

Nous remercions MITACS-Accélération, le ReSMiQ (Regroupement Stratégique en Microsystèmes du Québec) et M. Yvon Savaria de l'École Polytechnique de Montréal pour leur support financier.

Nous remercions également le Professeur Mohand S. Allili de l'UQO pour avoir mis à notre disposition la caméra infrarouge pour les mesures thermiques.

TABLE DES MATIÈRES

Jury d'évaluation	ii
Remerciements	iii
Table des matières.....	iv
LISTE DES FIGURES	vii
LISTE DES TABLEAUX	x
LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES.....	xi
1. Introduction.....	1
2. Objectifs	5
3. Identification du problème et motivation	6
3.1. Les sources de chaleur	6
3.2. Définition de la contrainte thermomécanique dans le WaferIC™	7
3.3. Identification de la contrainte thermomécanique dans le WaferIC™	8
4. État de l'art.....	11
4.1. Principe de transfert de la chaleur.....	11
4.2. Relations entre contrainte et déformation.....	16
4.3. Les contraintes dans les couches minces	19
4.3.1. Contrainte plane	19
4.3.2. Contrainte intrinsèque (résiduelle) entre deux couches	21
4.3.3. Contrainte thermomécanique entre deux couches.....	23
4.4. Le capteur de température.....	24
4.4.1. Principe de fonctionnement d'un RO.....	25
4.4.2. Études et caractérisation des capteurs à base de RO.....	26

4.4.2.1.	Effet d'auto réchauffement « Self heating » sur la réponse d'un capteur	26
4.4.2.2.	Effet de changement de la tension d'alimentation sur la réponse du capteur	28
4.4.2.3.	Linéarité de la réponse des capteurs de température à base de RO	33
4.4.2.4.	Algorithme de linéarisation de la réponse d'un RO	39
4.5.	Les algorithmes et le traitement de données provenant des capteurs de température à base de RO	40
4.5.1.	Méthode de direction du gradient de température	40
4.5.2.	Méthode avec des capteurs distribués	43
4.5.3.	Méthode de détection des cercles	45
4.5.4.	Méthode avec opérateur spatiotemporel	48
5.	Méthodologie et conception	49
5.1.	Limitations et contraintes	49
5.1.1.	Contraintes budgétaires et échancier	50
5.1.2.	Contraintes d'implémentation physique des capteurs, de communication, et de logiciel	50
5.2.	Conception matérielle des capteurs de température	52
5.2.1.	Architecture du capteur	52
5.2.2.	Distribution des RO au niveau de la cellule	53
5.2.3.	Définition des paramètres du circuit de contrôle	55
5.2.3.1.	Signaux d'entrée et de sortie du capteur	56
5.3.	Unité de contrôle	56
5.4.	Algorithme de traitement de données	57
5.4.1.	Méthodes possibles pour l'algorithme de traitement de données	58
5.4.2.	Méthodes adaptées pour l'algorithme de traitement de données	59
6.	Implémentation et mesures	61
6.1.	Implémentation du montage de l'expérience sur l'algorithme de détection de la contrainte thermomécanique avec une caméra infrarouge	62
6.2.	Mesure et traitement de données avec l'algorithme de détection de la contrainte thermomécanique	64
6.2.1.	Détection des pics thermiques ou sources de chaleurs dangereuses	65
6.2.2.	Détection de la vitesse de changement de la température locale	66
6.2.3.	Production des cartes isothermes	67

6.2.4.	Détection de la contrainte associée au choc thermique.....	71
6.3.	Implémentation et caractérisation des capteurs de température à base de RO sur un FPGA.....	73
6.3.1.	Définition des paramètres du test.....	75
6.3.2.	Test préliminaire du capteur dans un four.....	76
6.3.2.1.	Étude de la stabilité du circuit du contrôle par rapport à la variation de la température.....	78
6.3.2.2.	Mesure de la variation de fréquence du RO par rapport à une variation de température.....	82
6.4.	Amélioration de la réponse du capteur et l'étude de la méthode de linéarisation.....	85
6.4.1.	Amélioration de la réponse de F_{sub}	85
6.4.2.	Étude de la méthode de linéarisation sur la réponse de F_{out} corrigée..	87
7.	Conclusion.....	92
Annexe A	Conception du circuit du capteur et mesure sur une carte spartan-E3 de Xilinx	94
Annexe B	Code VHDL du circuit de test du chapitre 6.....	119
Annexe C	Code Matlab de l'amélioration des résultats de sortie du RO et de l'algorithme de linéarisation	126
Annexe D	Code Matlab de quelques fonctions de traitement d'image utilisés dans ce mémoire (Ch.6)	129
Annexe E	Publications.....	134
	Références.....	157

LISTE DES FIGURES

FIG. 1.1. LE WAFERBOARD™ COMME PRODUIT FINAL.	2
FIG. 1.2. LA COMPOSITION PRINCIPALE DU WAFERBOARD™.	3
FIG. 4.1. EXEMPLE D'UN PROCESSUS DE REFROIDISSEMENT OU DE TRANSFERT DE LA CHALEUR [3].	13
FIG. 4.2. A) LE WAFERBOARD™ AVEC DES COMPOSANTES. B) COUPE DE VUE DE FACE DU WAFERBOARD™ LORSQUE LA COUVERTURE EST FERMÉE ET LA PRESSION EST APPLIQUÉE.	15
FIG. 4.3. MODES DE TRANSFERT DE LA CHALEUR À PARTIR DU CENTRE D'UN CI.	15
FIG. 4.4. MODÈLE DE L'ENSEMBLE DES CONTRAINTES QUI AGISSENT SUR UN ÉLÉMENT D'UN CORPS [2].	16
FIG. 4.5. CONTRAINTE PLANE, SUR UNE COUCHE MINCE LOIN DES EXTRÉMITÉS [2].	19
FIG. 4.6. CONTRAINTES ET FORCES SUR LES EXTRÉMITÉS DE LA COUCHE MINCE [2].	20
FIG. 4.7. STRUCTURES À ÉCHELLE MICROSCOPIQUE PLIÉES SOUS L'EFFET DE LA CONTRAINTE INTRINSÈQUE [3].	21
FIG. 4.8. L'EFFET DE LA CONTRAINTE INTRINSÈQUE SUR DEUX COUCHES MINCES QUI ONT DES VALEURS DE LA CONTRAINTE INTRINSÈQUE DIFFÉRENTES [3].	22
FIG. 4.9. L'EFFET D'AUTO RÉCHAUFFEMENT SUR LA RÉPONSE D'UN RO [5].	27
FIG. 4.10. CIRCUIT AVEC UNE ALIMENTATION AJUSTABLE (VMID AJUSTABLE) EN FONCTION DE LA TEMPÉRATURE [6].	28
FIG. 4.11. LA RÉPONSE DE SORTIE DU CAPTEUR DE LA TEMPÉRATURE. A) EN HZ, B) EN NOMBRE DE COMPTES O [6].	30
FIG. 4.12 LA STABILITÉ PAR RAPPORT À LA VARIATION DE TENSION DE LA RÉPONSE DE SORTIE DU CAPTEUR RO EN FONCTION DE NOMBRE D'INVERSEUR [7].	32
FIG. 4.13. IMPLÉMENTATION DU CAPTEUR AVEC LA TECHNOLOGIE CMOS DE 0,13MM [6].	33
FIG. 4.14 LA RÉPONSE DE SORTIE D'UN CAPTEUR DE TEMPÉRATURE RO À TROIS INVERSEURS EN FONCTION DE LA TEMPÉRATURE [7].	34
FIG. 4.15. LA RÉPONSE DE SORTIE D'UN CAPTEUR DE TEMPÉRATURE RO -50 À 120 °C [9].	35
FIG. 4.16. A) LE MONTAGE DE L'EXPÉRIENCE. B) LA RÉPONSE DE SORTIE NORMALISÉE DU CAPTEUR RO EN FONCTION DE LA CONTRAINTE [11].	37
FIG. 4.17. STRUCTURE CASCADÉ À COUPLAGE CROISÉ INDÉPENDAMMENT DE LA TEMPÉRATURE [10].	38
FIG. 4.18. A) CAPTEUR CONVENTIONNEL; B) NOUVEAU CAPTEUR À DEUX RO [10].	39
FIG. 4.19. A) ET B) LE PRINCIPE DE DÉTECTION PAR GDS D'UNE SOURCE DE CHALEUR.	41
FIG. 4.20. DÉTAIL D'UN CAPTEUR À TROIS RO [14, 15].	42

FIG. 4.21 CAPTEURS DISTRIBUÉS DE TEMPÉRATURE COMPOSÉS DES CELLULES RO SUR TOUTE LA SURFACE ($x \times y$) DE LA PUCE [16].....	44
FIG. 4.22. LA CARTE THERMOGRAPHIQUE D'UN FPGA PRODUITE AVEC UN TABLEAU DE (4×10) RO [16].	45
FIG. 4.23. A) IMAGE ORIGINALE, B) LA TRANSFORMÉ DE HOUGH SUR L'IMAGE ORIGINALE, C) UNE MODIFICATION À LA TRANSFORMÉE DE HOUGH POUR MONTRER LES CENTRES DES CERCLES [17].47	
FIG. 5.1. L'ARCHITECTURE DU WAFERIC™	51
FIG. 5.2. UN CIRCUIT D'UN SEUL RO POUR UNE CELLULE SUR LE WAFERIC™	54
FIG. 6.1. MONTAGE DE L'EXPÉRIENCE DE LA DÉTECTION THERMIQUE AVEC UNE CAMÉRA INFRAROUGE, UN WAFERIC ET DES SOURCES DE CHALEUR	63
FIG. 6.2. DE A. À C. 3 TRAMES DE VIDÉO QUI MONTRE LA DIFFUSION DE LA CHALEUR SUR LA SURFACE DU WAFERIC, ET QUI EST SÉPARÉS PAR UN DT DE 45 SECONDES.	64
FIG. 6.3. AVEC UN DT DE 45 SECONDES DE A. À C. UN SEUILLAGE À 44,04°C DÉTECTE UN PIC THERMIQUE DANS C.	66
FIG. 6.4. A. MONTRE UNE AUGMENTATION BRUSQUE DE LA TEMPÉRATURE QUI DÉPASSE UNE SEILLE T_{sl} , B ET C MONTRE QUE CETTE AUGMENTATION A CALMÉE. A. EST LE RÉSULTANT DE LA DÉRIVÉE DANS LE TEMPS DE LA FIGURE 6.2-A ET B. ET AINSI DE SUITE.....	66
FIG. 6.5. A. IMAGE ORIGINALE, B. APPLICATION DU LAPLACIEN SUR L'IMAGE EN A.	68
FIG. 6.6. A. SEUILLAGE À 16 NIVEAUX DE GRIS, B. APPLICATION DE LAPLACIEN SUR L'IMAGE DU A. ...	69
FIG. 6.7. A. SEUILLAGE À 16 NIVEAUX DE GRIS, B. APPLICATION D'UN FILTRE MÉDIAN DE 7X7, C. LAPLACIEN SUR L'IMAGE DU B.	70
FIG. 6.8. ISOTHERME À RÉOLUTION MAXIMALE, DONT UNE RÉOLUTION THERMIQUE DE 0,9375°C/ISOTHERME.....	71
FIG. 6.9. ÉVOLUTION DE LA TEMPÉRATURE SUR LA SURFACE DU WAFERIC. DE A. À C. LES IMAGES SONT SÉPARÉES D'UN DT DE 45 SECONDES ET UNE RÉOLUTION DE 0,9375°C/ISOTHERME.	
72	
FIG. 6.10. A. CAPTEUR DE TEMPÉRATURE À 9 RO, B. EMBLEMMENT PHYSIQUE DU CAPTEUR SUR LE FPGA.....	74
FIG. 6.11. LE CAPTEUR DE TEMPÉRATURE À 5 RO, AVEC LE CIRCUIT DE CONTRÔLE.	77
FIG. 6.12. LE CAPTEUR DE TEMPÉRATURE À 5 RO, AVEC LE CIRCUIT DE CONTRÔLE.	79
FIG. 6.13. MONTAGE EXPÉRIMENTAL.	80
FIG. 6.14. MESURE DE LA CONTRIBUTION DU CIRCUIT DE CONTRÔLE SUR LA RÉPONSE DU CAPTEUR; F_{OUT} À 25°C EST ÉGALE À 6000HZ.	81

FIG. 6.15. RÉPONSE DE LA FRÉQUENCE DU RO PAR RAPPORT À LA TEMPÉRATURE.	84
FIG. 6.16. COMPARAISON DE LA RÉPONSE DE LA FRÉQUENCE DU CAPTEUR (F_{SUB}) PAR RAPPORT À LA TEMPÉRATURE, AVANT (BLEU) ET APRÈS (ROUGE) L'AMÉLIORATION.	86
FIG. 6.17. COMPARAISON DE FOUT PAR RAPPORT À LA TEMPÉRATURE, AVANT (BLEU) ET APRÈS (ROUGE) L'AMÉLIORATION.	87
FIG. 6.18. LA COURBE ROUGE EST LA SORTIE DE L'ALGORITHME DE LINÉARISATION, EN BLEU EST LA SORTIE FOUT AMÉLIORÉE.	89
FIG. 6.19. LA COURBE ROUGE EST LA SORTIE DE L'ALGORITHME DE LINÉARISATION, EN BLEU EST LA SORTIE FOUT AMÉLIORÉE.	91

LISTE DES TABLEAUX

Tableau 6.I. Données thermiques de la fiche technique du FPGA xc3s500e-4fg320 FPGA 75

Tableau 6.II. Mesure de la fréquence de sortie F_{out} par rapport à la température 82

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ASIC	Application Specific Integrated Circuit
CBC	Clamped Boundary Condition
CI	Circuit Intégré
CMOS	Complementary Metal–Oxide–Semiconductor
CTE	Coefficient of Thermal Expansion
FPGA	Field Programmable Gate Array
GDS	Gradient Direction Sensor
GUI	Graphic User Interface
IR	Infra rouge (Infra Red)
ISE	Integrated Simulation Environment
JTAG	Joint Test Action Group
LAIC	Large Area Integrated Circuit
LFM	Linear Feet per Minute
LIMA	Laboratoire d’Ingénierie des Microsystèmes Avancés
LUT	Look Up Table
MEMS	Micro Electro Mechanical System
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor
PCB	Printed Circuit Board
R&D	Research and Development
RO	Ring Oscillator
RTA	Rapid Thermal Annealing
SoC	System on a Chip
SPTDA	Surface Peaks Thermal Detector Algorithm
ULSI	Ultra Large Scale integration
VCO	Voltage Controlled Oscillator
VHDL	Very Large Scale Integration Circuit Hardware Description
VLSI	Very Large Scale Integration

WSI	Wafer Scale Integration
XST	Xilinx Synthesis Tool

Résumé

La fabrication des circuits intégrés (CI) n'est pas nouvelle, mais elle est toujours en progression. Aujourd'hui les CI de type ULSI (Ultra Large Scale integration) peut intégrer plus qu'un million de transistors par un centimètre carré. Des nouvelles inventions comme le projet de DreamWafer (<http://www.dreamwafer.com>) nécessitent l'utilisation de la méthode WSI (Wafer Scale Integration), là où toute la surface de la gaufre (Wafer) de silicium sera utilisée comme un seul grand CI de 200 [mm] de diamètre.

Avec la grande intégration dans les ULSI, et avec l'augmentation dramatique de la surface d'un CI dans les WSI, la contrainte thermomécanique devient très problématique. Plus d'intégration cause plus de consommation d'énergie par conséquent plus de dissipation de chaleur (effet joule) par unité de surface, cela cause des pics thermiques plus intenses. Encore, plus la surface est grande, plus ces pics thermiques (notamment en mode transitoire) deviennent dangereux, à cause du fort niveau de gradient de température. Dans ces deux cas, la recette est là pour que la contrainte thermomécanique endommage le CI.

Dans ce mémoire, nous proposons une méthode pour gérer efficacement la problématique de la contrainte thermomécanique dans les LAIC (Large area integrated circuit). Une matrice de capteurs de température RO (Ring Oscillator) seront intégrés dans le CI et seront configurés pour effectuer les mesures en espace spatiotemporel. Un étalonnage, caractérisations et étude de la réponse de ces capteurs a été effectué. Également, une méthode de linéarisation et d'amélioration de la réponse a été étudiée en détail. Par la suite, nous avons utilisé les mesures thermiques que nous avons obtenues à l'aide d'une camera IR qui imite l'espace spatiotemporel

de notre capteur, pour tester un algorithme de détection de la contrainte thermomécanique sur la grandeur de la surface d'un LAIC.

Mots clés : Ring Oscillator (RO), Spatial, Spatiotemporal, Thermomechanical contrainte, Stress, Temperature sensor, Thermal Analysis, WaferIC, Wafer-scale system, LAIC.

1. Introduction

Le prototypage dans l'industrie de l'électronique est un défi majeur devant le lancement d'un nouveau produit sur le marché, c'est un processus lent qui utilise généralement les circuits imprimés (PCB : Printed Circuit Board) comme plateformes d'interconnexions des composantes électroniques. Donc à chaque modification ou amélioration, il faudra un nouveau PCB. Face à une grande compétition, les grands joueurs dans l'industrie cherchent toujours à améliorer leur processus de fabrication pour lancer leurs nouveaux produits sur le marché. Chaque minute compte, pas de place pour les erreurs.

D'ici vient l'idée du projet DreamWafer™ qui est une plateforme de prototypage des circuits électroniques modernes, dans le but de faciliter et d'accélérer le processus de prototypage de ces derniers. Ce système de prototypage rapide consiste principalement d'un WaferBoard™ (figure 1.1) et d'une interface usager qui sera installée sur un ordinateur. L'élément central du WaferBoard™ est le WaferIC™ (figure 1.2). Le concepteur place simplement les composantes de son système sur le WaferIC™, ensuite le WaferBoard™ détecte, reconnaît, et interconnecte les circuits intégrés représentant le design. Ainsi, à partir de l'interface usager (sur l'ordinateur), le concepteur suit tout le processus et son prototype sera prêt à être testé en quelques minutes, donc le WaferBoard™ agit comme un circuit imprimé reconfigurable et dynamique. Par analogie, la différence entre un prototypage avec PCB et un prototypage avec le WaferBoard™, est comme la différence entre gravé une chanson sur un microsillon, ou de la faire enregistrer sur une cassette à bande magnétique tout simplement en appuyant sur le bouton « Record ».



Fig. 1.1. Le WaferBoard™ comme produit final.

Sans entrer dans les détails, le WaferIC™ est comme un circuit intégré qui est fabriqué tout en utilisant le processus de fabrication CMOS, la différence majeure entre un circuit intégré (CI) et le WaferIC™ est au niveau de l'échelle. Un seul Wafer de silicium sera découpé dans la fonderie pour fabriquer une dizaine des circuits intégrés. Dans le cas du WaferIC™ un seul WaferIC™ occupe toute la surface d'un wafer au complet pour former un LAIC (Large Area Integrated Circuit). La figure 1.2 montre la composition principale du WaferBoard™ et on voit que le WaferIC™ qui se compose d'une gaufre au complet et il occupe toute la surface du WaferBoard™ sur laquelle on place les composantes du circuit électronique à tester.

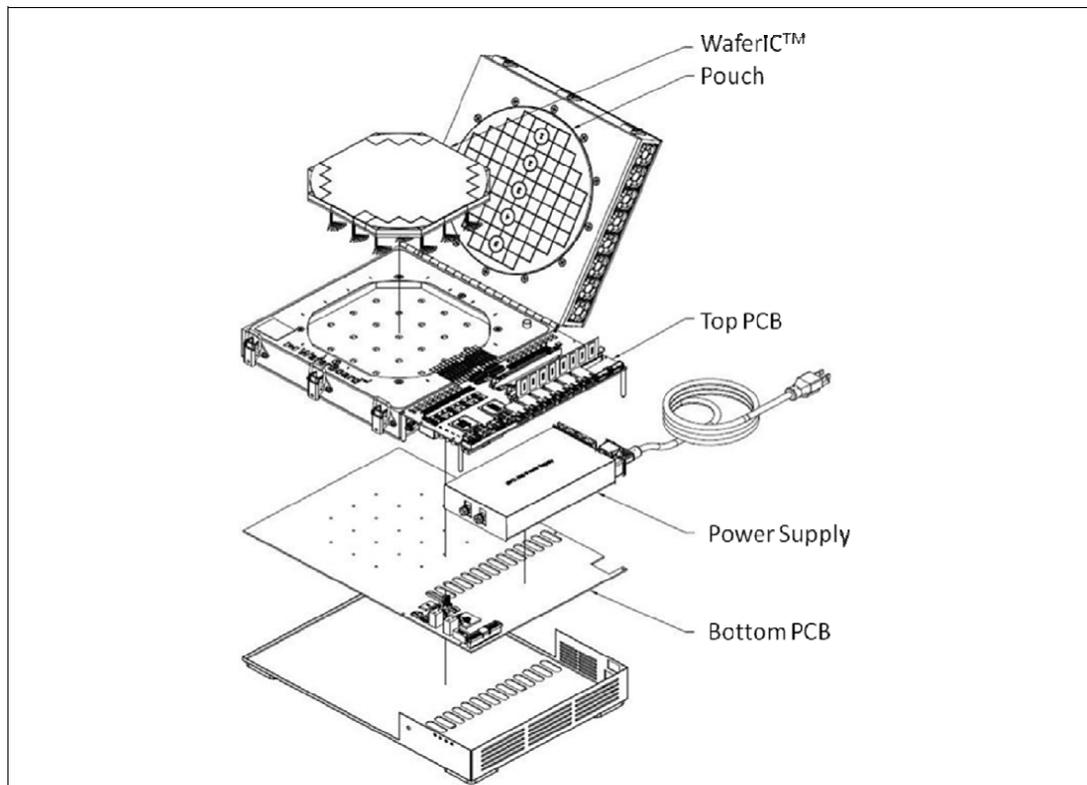


Fig. 1.2. La composition principale du WaferBoard™.

L'utilisation d'un Wafer au complet comme un seul circuit est nouvelle dans le domaine de la microélectronique. Il est toujours dans la phase de la recherche et il n'a pas encore rentré dans la phase de la production à une échelle commerciale. Il est connu qu'avec l'augmentation de la surface d'un CI, la contrainte thermomécanique causée par le gradient thermique dans les couches minces sur la surface du substrat devienne plus importante. Il est encore très connue que cette contrainte thermomécanique est à la l'origine de la fatigue dans les couches minces, donc à l'origine de la défaillance et de la réduction de la vie des CI. D'abord, il faut garder la contrainte thermomécanique sous contrôle en tout temps, car avec l'absence d'une

gestion thermique adéquate, toute la fiabilité et la crédibilité des LAIC seront mis en jeu.

2. Objectifs

L'objectif général de ce projet est de trouver une méthode qui détecte en temps réel la distribution de la chaleur sur la surface d'un WaferICTM, et de faire identifier les régions qui ont une contrainte thermomécanique d'une valeur critique et qui ont le potentiel d'endommager ce dernier.

Pour mesurer la température dans le WaferICTM, nous suggérons tout d'abord d'installer physiquement des capteurs de température embarqués (Embedded) dans le WaferICTM même et qui auront le potentiel de mesurer sélectivement ou discrètement la température dans chaque région de la surface de ce dernier. Par la suite, ces capteurs embarqués vont envoyer périodiquement les valeurs de températures observées vers un algorithme (que nous allons développer) qui va évaluer en espace spatiotemporel ces mesures, et afficher en temps réel la distribution de la température sur ce dernier tout en indiquant les régions où la contrainte thermomécanique est critique ou est sur le point de le devenir.

En résumé, notre contribution va être au niveau de l'implémentation des capteurs thermiques à la surface du wafer en silicium (SI) (WaferICTM), étudier améliorer, et trouver une méthode pour interpréter efficacement la réponse de ces capteurs pour le but de produire des images thermiques sur la grandeur du WaferICTM. À partir de ces mesures ou images, nous allons effectuer l'analyse de la contrainte thermomécanique à la grandeur d'un circuit LAIC, ce qui est en fait nouveau, innovateur et qui n'est pas encore abordé en profondeur par d'autres chercheurs. De plus, il est à noter que c'est la première fois que ces techniques de détection vont sortir des labos de recherche pour être appliqués à un projet industriel qui a le potentiel de commercialisation.

3. Identification du problème et motivation

Dans ce chapitre nous allons identifier les sources de chaleur qui sont à l'origine de la contrainte thermomécanique, pour ensuite définir et identifier cette dernière dans le WaferICTM.

3.1. Les sources de chaleur

Comme nous avons vu dans l'introduction précédemment, le WaferBoardTM, servira comme une plateforme de prototypage sur laquelle le concepteur place des composantes de son système sur la surface du WaferICTM, et le WaferBoardTM s'occupe de les faire interconnecter. Mais le problème majeur ici est la gestion thermique. Comme nous le savons, les composantes électroniques consomment de l'énergie électrique, une partie considérable de cette énergie sera transformée en chaleur (effet joule). De plus, les éléments du système (système du concepteur) d'interconnexion seront configurés directement dans le WaferICTM, par conséquent le tout va dégager de la chaleur. Toutes cette chaleur générée à la surface ou dans le WaferICTM, va se propager sur la surface de ce dernier en forme d'isothermes autour de chaque source de chaleur et par conséquent seront à l'origine de la contrainte thermomécanique dans le WaferICTM.

3.2. Définition de la contrainte thermomécanique dans le WaferIC™

Il est très connu que la contrainte thermomécanique est un grand défi pour la fabrication et l'utilisation des circuits intégrés, et surtout pour les LAIC (Large Area Integrated Circuit). Comme le WaferIC™ est fabriqué avec le procédé CMOS, qui est généralement composé de plusieurs couches de silicium, d'oxyde de silicium, de poly-silicium, et d'autres métaux, le coefficient de dilatation thermique (expansion ou compression CTE : Coefficient of Thermal Expansion) de chacune de ces couches n'est pas le même. Durant le procédé de fabrication, due au CTE différent pour chaque couche, une contrainte thermomécanique résiduelle est inévitable, cette contrainte est appelée la contrainte intrinsèque « Intrinsic Stress » [1]. Mais cette contrainte intrinsèque est minimisée par un traitement de recuits thermique rapide après le dépôt (RTA : Rapid Thermal Annealing) pour faire relaxer la contrainte. Ainsi, en sortant du four, le substrat et la couche vont refroidir et se contracter à des vitesses différentes. Si la couche du matériau déposée se contracte plus vite que le substrat, cela va générer une contrainte de tension dans cette couche, mais si la vitesse de contraction de la couche déposée est plus lente que celle du substrat, cela va générer une contrainte de contraction dans la même couche [2, Ch. 3, p. 88] et [3].

Une variation locale de la température sur différents points de la surface d'un LAIC, va introduire des contraintes thermomécaniques locales avec des valeurs différentes sur toute la surface de ce dernier. Toutes ces nouvelles valeurs de la contrainte, vont s'additionner ou se soustraire avec les valeurs des contraintes résiduelles qui existent déjà sur chaque région de la surface du LAIC. Tout d'abord, il faut surveiller chaque valeur de ces contraintes locales, car il suffit qu'une de ces dernières dépasse un certain seuil, pour endommager le WaferIC™.

Une différence de température entre deux régions adjacentes (choc thermique) quelconques, va à son tour introduire une nouvelle contrainte que nous allons appeler une contrainte thermomécanique interrégionale (ou contrainte due au gradient de température). La valeur de cette contrainte interrégionale est surtout importante en régime transitoire dû à la valeur du gradient thermique qui est significatif dans ce mode transitoire (mise en tension du système). Il faut surveiller en tout temps chaque valeur de toutes ces contraintes thermomécaniques interrégionales, car s'ils dépassent un certain seuil prédéfini, ils peuvent aussi endommager le WaferICTM.

3.3. Identification de la contrainte thermomécanique dans le WaferICTM

Normalement sur un circuit intégré, à petite surface, la variation temporelle de la contrainte thermomécanique pourrait être estimée comme une variation linéaire sur toute la surface, car c'est une variable dépendante de la variation de température qui est à son tour estimée être linéaire. La vitesse de variation de la température sur un seul point est :

$$v = dT / dt \quad (3.1)$$

T est la température en °C et t est le temps en seconde

Mais pour les LAIC à grande surface (comme le cas du WaferICTM), dans un temps transitoire t_{tr} , la température se propage sur la surface par conduction. Or comme nous l'avons indiqué dans la section précédente, une variation temporelle locale de la température sur un point P_1 de la surface de ce dernier va introduire un gradient de température par rapport à un autre point P_2 avec

$$D = P_2 - P_1 \quad (3.2)$$

D est la distance entre les deux points et ;

$$T_{(tr)} = T_{p1} - T_{p2} \quad (3.3)$$

T_{p1} est la température sur le point P_1 en °C et T_{p2} est la température sur le point P_2 en °C et $T_{(tr)}$ est la différence de température entre T_{p1} et T_{p2} en °C.

La magnitude de ce gradient est :

$$m = T_{(tr)} / D \quad (3.4)$$

Comme nous avons précisé dans la partie 3.2, la valeur m devient problématique lorsqu'elle atteint une valeur ou un seuil critique prédéfini que nous allons appeler m_c .

Dans un temps infini t , la différence de température entre T_{p1} et T_{p2} ($T_{(tr)}$) va prendre une valeur finale théoriquement nulle, mais en pratique ce n'est pas le cas, car il faut considérer la résistivité thermique de la surface du WaferICTM, et la perte par convection et par radiation [3, Ch. 5, pp. 153-157] sur les deux faces du WaferICTM.

Finalement pour finir cette section, deux paramètres à surveiller : la température locale à chaque point du WaferICTM, pour qu'elle ne dépasse pas un seuil S_1 , et m_c pour une différence de température entre deux régions adjacentes pour qu'elle ne dépasse pas un seuil S_2 .

4. État de l'art

Dans ce chapitre nous allons exposer la théorie, des principes, et une revue critique de la littérature sur des sujets en relation avec notre projet de recherche, tout en analysant la pertinence de ces sujets avec notre problématique.

4.1. Principe de transfert de la chaleur

Il existe quatre principes connus pour le transfert de la chaleur [3] [3, Ch. 5, pp. 153-157] :

1. Par conduction dans les solides avec la présence du gradient de température,

$$\text{Error! Bookmark not defined. } q = -\kappa \frac{dT}{dx}$$

(4.1)

Avec q qui est le flux de la chaleur [W/m^2] à la longueur de l'axe des x , et κ la conductivité thermique [W/m];

2. Par convection naturelle (ou passive) représentée par un transfert de chaleur d'une surface vers un gaz ou un fluide stationnaire (voir formule 4.2);
3. Par convection forcée représentée par un transfert de chaleur vers un gaz ou un fluide en mouvement.

Pour les deux types de convection on a :

$$q = h(T_s - T_\infty) \quad (4.2)$$

où h est le coefficient de la chaleur convective, T_s est la température de la surface en [°C], et T_∞ est la température du gaz ou fluide en [°C].

4. Par radiation, qui est la perte ou le gain de la chaleur par radiation électromagnétique dans l'espace ou dans l'air avec :

$$E = \varepsilon \sigma T_R^4 \quad (4.3)$$

E est la puissance émissive [W/m²], T_R est la température absolue [°K], σ est la constante de Boltzmann (5,67x10⁻⁸ [W/m²°K⁴]), et ε est l'émissivité radiative (0 ≤ ε ≤ 1).

Pour illustrer ces principes, la figure 4.1 montre un exemple simplifié de transfert de chaleur entre une source de chaleur refroidie avec un radiateur rempli avec de l'huile refroidissante. Le courant électrique passe dans la résistance électrique qui à son tour va générer de la chaleur qui va être dissipée dans les objets à l'entourage. Dans le modèle ou le circuit équivalent de la figure 4.1, R est la résistance thermique et P_D est la puissance dissipée

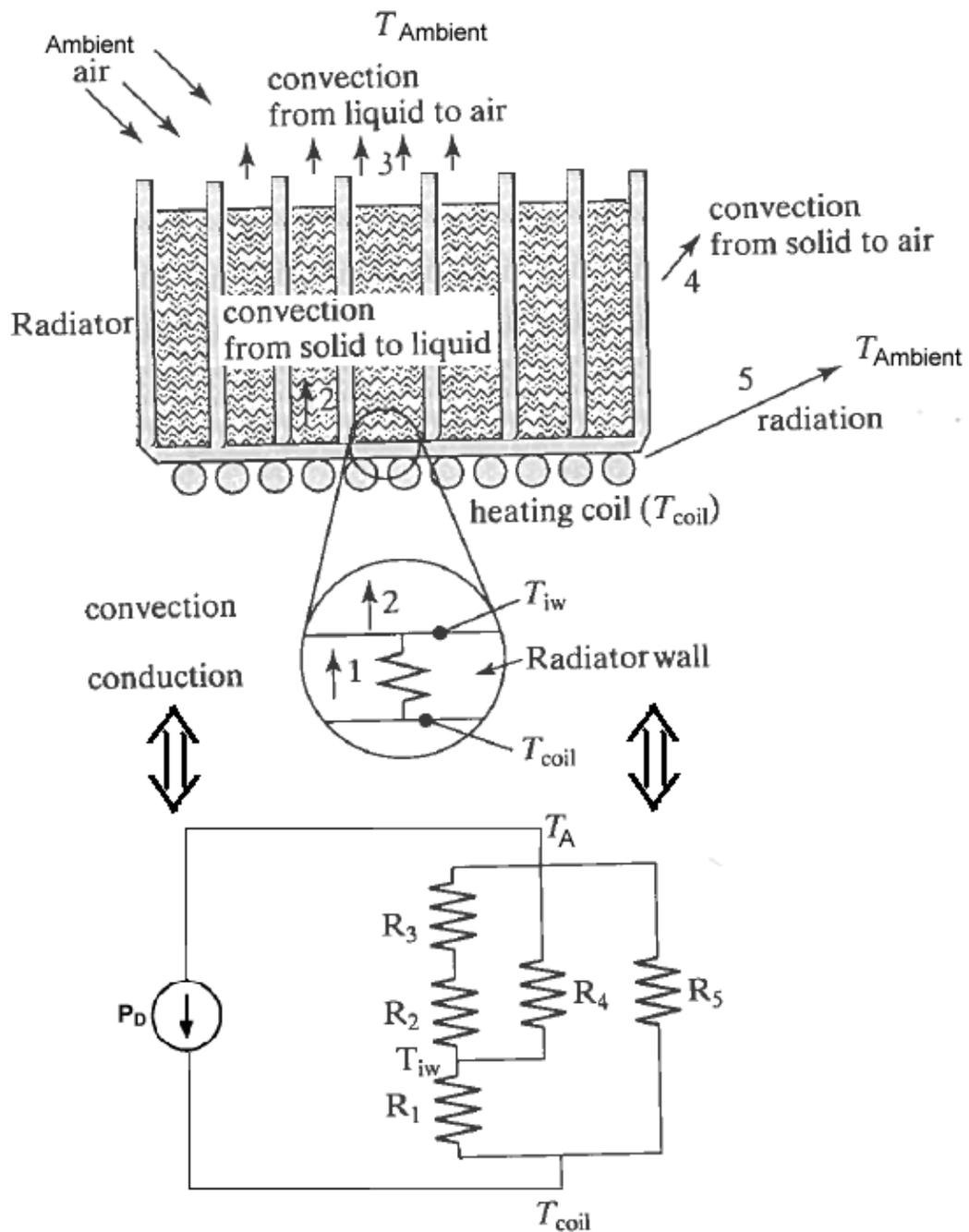


Fig. 4.1. Exemple d'un processus de refroidissement ou de transfert de la chaleur [3].

La température de la source de la figure 4.1 peut être calculée :

$$T_{coil} = ((R_1 + ((R_2 + R_3) // R_4)) // R_5) \times P_d + T_A \quad (4.4)$$

T est la température [°C], R est la résistance thermique [°C/W] et P est la puissance dissipée [W]

Par analogie du module thermique de la figure 4.1, nous pouvons l'appliquer pour développer le module thermique du WaferBoard™ de la figure 4.2. Mais celui-ci va être beaucoup plus complexe à cause des sources de chaleur multiples à plusieurs niveaux ou couches du système. La figure 4.2 b), montre la coupe de vue de face du WaferBoard™ lorsque le couvercle est fermé et la pression est appliquée pour donner un bon contact électrique entre les composantes sur la surface du WaferIC™ et le WaferIC™ même. Un fluide thermique est utilisé pour aider à dissiper la chaleur produite dans le système.

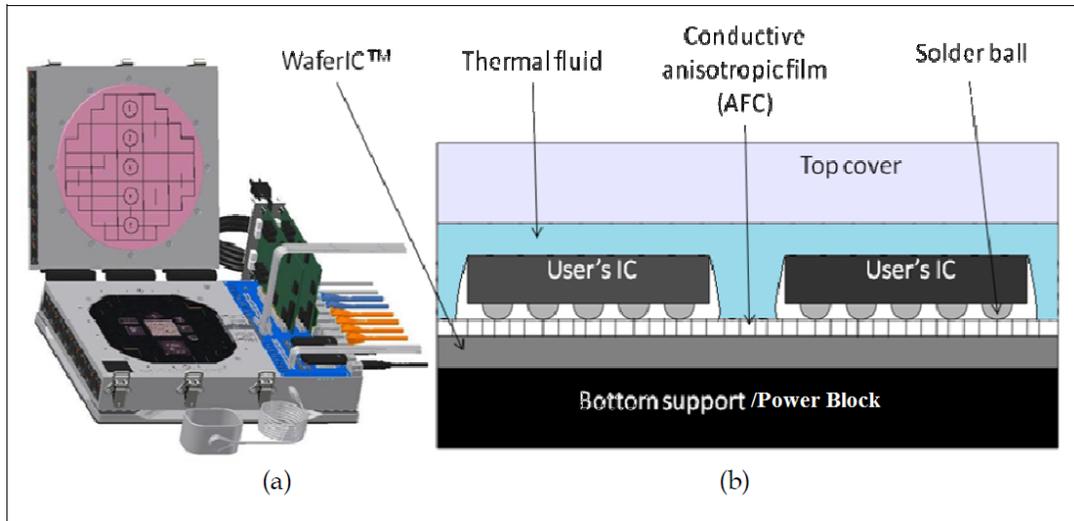


Fig. 4.2. a) le WaferBoard™ avec des composants. b) Coupe de vue de face du WaferBoard™ lorsque la couverture est fermée et la pression est appliquée.

Pour les CI sur la surface du WaferIC™, la figure 4.3 montre les modes de transfert de la chaleur impliquée à partir du noyau d'un CI.

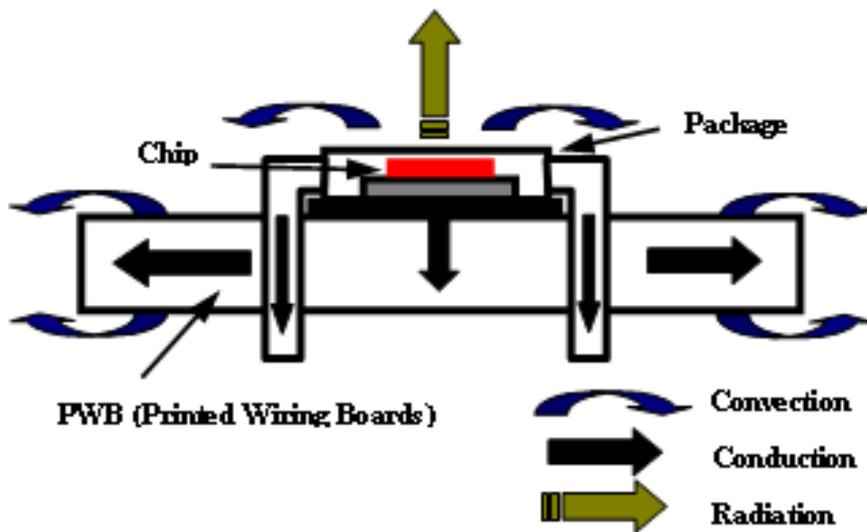


Fig. 4.3. Modes de transfert de la chaleur à partir du centre d'un CI.

4.2. Relations entre contrainte et déformation

La figure 4.4 montre l'ensemble des contraintes qui agissent sur un élément d'un corps. Il y a encore des forces semblables à ce que nous voyons ici sur les trois surfaces que nous ne voyons pas du cube, d'une façon à ce que la somme des forces est nulle.

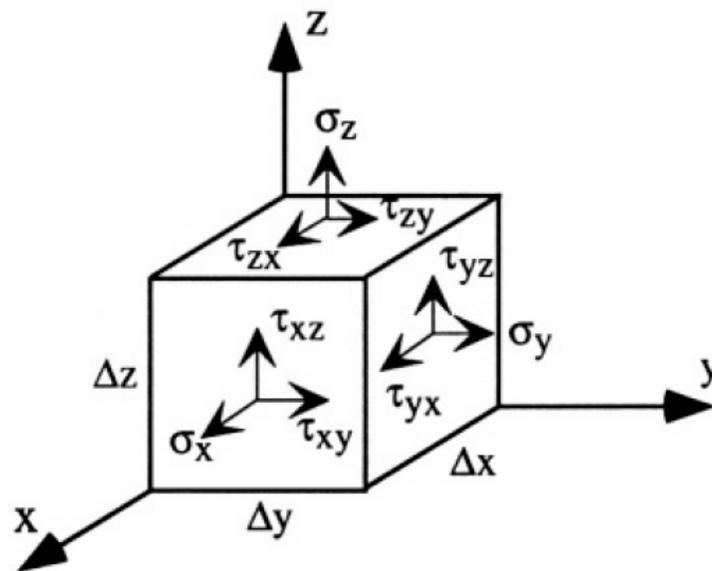


Fig. 4.4. Modèle de l'ensemble des contraintes qui agissent sur un élément d'un corps [2].

La figure 4.4 présente le modèle de l'ensemble des contraintes qui agissent sur un élément d'un corps. Ainsi, la matrice ou le tenseur des contraintes σ_{ij} est le suivant :

$$[\sigma] = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} \quad (4.5)$$

La diagonale de la matrice 4.5 montre les contraintes normales parallèles aux axes spécifiques (x, y et z), et les autres contraintes sont des contraintes de cisaillement là où chaque paire sur deux faces voisines de ces dernières sont égales.

$$\begin{aligned} \tau_{xy} &= \tau_{yx} \\ \tau_{xz} &= \tau_{zx} \\ \tau_{yz} &= \tau_{zy} \end{aligned} \quad (4.6)$$

Supposons que l'élément de la figure 4.4 soumis à chacune de ces forces se déforme. Alors, un point P quelconque dans cet élément se déplace en P', et les composantes de ce vecteur par rapport à l'origine sont u, v, et w, alors, les déformations seront :

$$\begin{aligned} [\varepsilon_x] &= \frac{\partial u}{\partial x}, & [\gamma_x] &= \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ [\varepsilon_y] &= \frac{\partial v}{\partial y}, & [\gamma_y] &= \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ [\varepsilon_z] &= \frac{\partial w}{\partial z}; & [\gamma_z] &= \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \end{aligned} \quad (4.7)$$

Et la relation compète entre les contraintes et les déformations pour les solides isotopiques et élastiques sont donnés par la loi généralisée de Hooke comme [2] :

$$\varepsilon_x = \frac{1}{E}[\sigma_x - \nu(\sigma_y + \sigma_z)] \quad (4.8)$$

$$\varepsilon_y = \frac{1}{E}[\sigma_y - \nu(\sigma_z + \sigma_x)] \quad (4.9)$$

$$\varepsilon_z = \frac{1}{E}[\sigma_z - \nu(\sigma_x + \sigma_y)] \quad (4.10)$$

$$\gamma_{xy} = \frac{1}{G}\tau_{xy} \quad (4.11)$$

$$\gamma_{yz} = \frac{1}{G}\tau_{yz} \quad (4.12)$$

$$\gamma_{zx} = \frac{1}{G}\tau_{zx} \quad (4.13)$$

E [GPa] est le module d'Young, G [GPa] est le module de Coulomb et ν est le coefficient de poisson et qui sont reliées par la relation suivante [3] :

$$G = \frac{E}{2(1+\nu)} \quad (4.14)$$

4.3. Les contraintes dans les couches minces

4.3.1. Contrainte plane

C'est un cas particulier aux couches minces sur un substrat [2]. La figure 4.5 montre la contrainte plane sur une couche mince qui a été déposée sur un substrat.

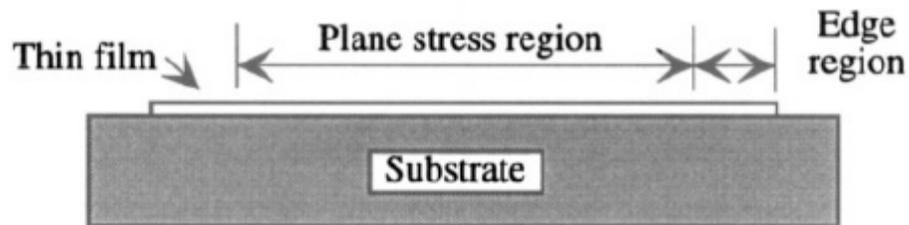


Fig. 4.5. Contrainte plane, sur une couche mince loin des extrémités [2].

Une couche mince déposée sur un substrat a une certaine quantité de contraintes planes, qui provient soit du processus de fabrication, soit de coefficient de dilatation thermique qui n'est pas le même pour la couche et le substrat. Loin des bordures de la couche, la contrainte principale est celle qui agit dans le plan (x et y), car la surface de la couche est relaxée et les déformations sur x et y sont données par :

$$\begin{aligned}\varepsilon_x &= \frac{1}{E}(\sigma_x - \nu\sigma_y) \\ \varepsilon_y &= \frac{1}{E}(\sigma_y - \nu\sigma_x)\end{aligned}\quad (4.15)$$

Sur les extrémités ou le périmètre de la couche, la situation est différente (figure 4.6)

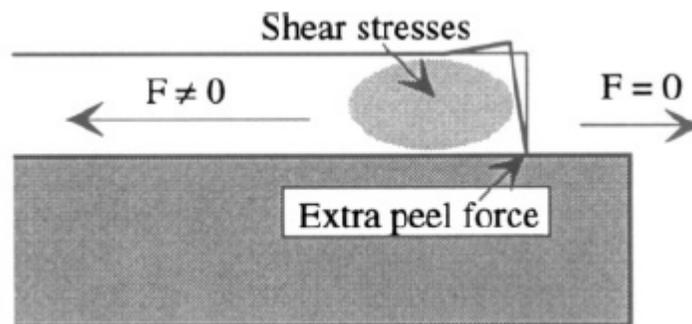


Fig. 4.6. Contraintes et forces sur les extrémités de la couche mince [2].

La force plane est zéro sur les extrémités, mais on a la libération de la contraction verticale de Poisson [2] ce qui fait que le côté vertical est incliné vers le centre de la couche et plus épais que le reste de la couche. Par conséquent, la région sombre sur la figure 4.6 a une distribution complexe de contraintes de cisaillements (créant un moment de fléchissement) qui font la transition entre les deux régions de la couche (haut-bas). Donc «une concentration de contraintes» ou «forces de pèle» (peel forces) qui essaient d'arracher la couche du substrat.

4.3.2. Contrainte intrinsèque (résiduelle) entre deux couches

Comme spécifié dans la section 3.2, la contrainte intrinsèque est une contrainte interne à la température ambiante dans les couches minces déposées sur la surface d'un substrat [3, Ch. 5, pp. 153-157]. La figure 4.7 montre deux structures à échelle microscopique pliées sous l'effet de la contrainte intrinsèque.

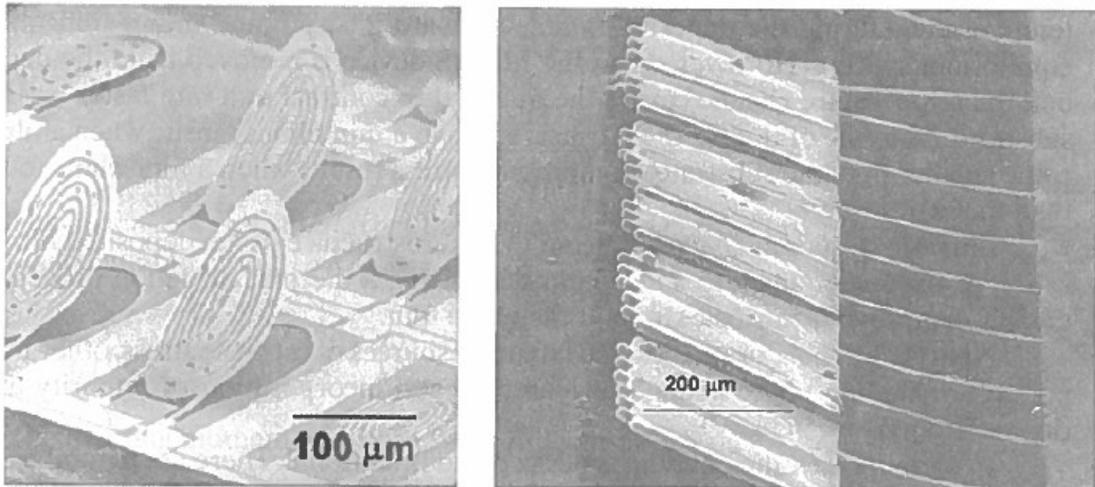


Fig. 4.7. Structures à échelle microscopique pliées sous l'effet de la contrainte intrinsèque [3].

Pour bien formuler la contrainte intrinsèque mécanique entre deux couches ou deux matériaux différents, la figure 4.8 montre l'effet de courbure entre deux couches qui ont des contraintes résiduelles intrinsèques avec des valeurs différentes.

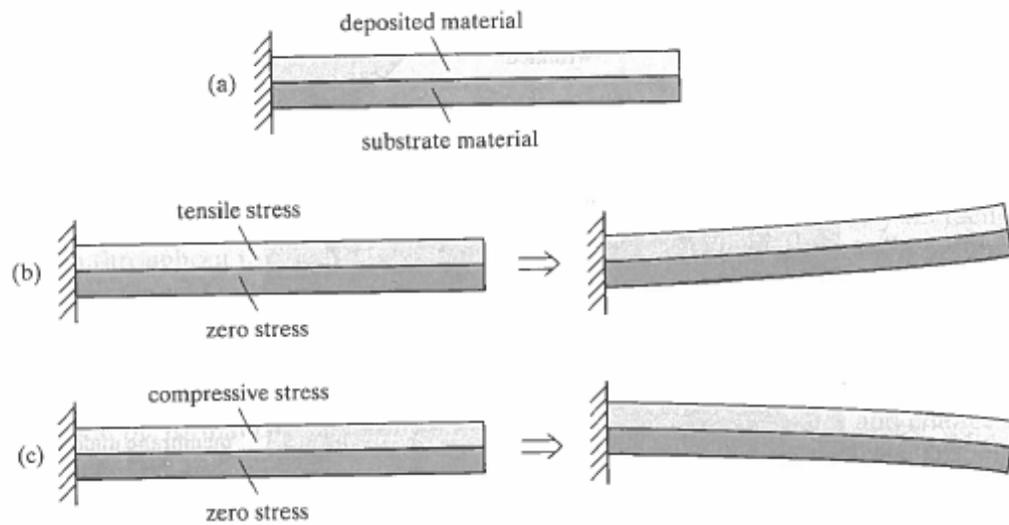


Fig. 4.8. L'effet de la contrainte intrinsèque sur deux couches minces qui ont des valeurs de la contrainte intrinsèque différentes [3].

Pour un calcul simplifié de différents paramètres de contrainte intrinsèques sur la structure de la figure 4.8, nous allons assumer que les deux couches ont la même longueur (l) et la même largeur (w).

La position de l'axe du neutre à partir du bas de la couche inférieure est :

$$\bar{y} = \frac{\frac{1}{2}(E_1 t_1^2 + E_2 t_2^2) + E_2 t_1 t_2}{E_1 t_1 + E_2 t_2} \quad (4.16)$$

La rigidité de flexion effective est

$$I_{eff}E_0 = w \left(E_1 t_1 \left(\frac{t_1^2}{12} + \left(\frac{t_1}{2} - \bar{y} \right)^2 \right) + E_2 t_2 \left(\frac{t_2^2}{12} + \left(\frac{t_2}{2} + t_1 - \bar{y} \right)^2 \right) \right) \quad (4.17)$$

Le moment de flexion agissant sur la structure est :

$$M = w \left[\left(\frac{t_1^2}{2} \right) \left(\sigma_1 (1 - \nu_1) - E_1 \frac{t_1 \sigma_1 (1 - \nu_1) + t_2 \sigma_2 (1 - \nu_2)}{E_1 t_1 + E_2 t_2} \right) \right] + \quad (4.18)$$

$$w \left[\left(\frac{t_2^2 + t_1 t_2}{2} \right) \left(\sigma_2 (1 - \nu_2) - E_2 \frac{t_1 \sigma_1 (1 - \nu_1) + t_2 \sigma_2 (1 - \nu_2)}{E_1 t_1 + E_2 t_2} \right) \right]$$

Et le rayon de la courbature R est :

$$R = \frac{I_{eff} E_0}{M} \quad (4.19)$$

t_i est l'épaisseur, σ_i est la contrainte intrinsèque statique ou résiduelle, E_i est le Module de Young, et $i = 1$ pour la couche #1 (haut), et 2 pour la couche #2 (bas).

4.3.3. Contrainte thermomécanique entre deux couches

Si nous prenons maintenant deux couches minces avec des coefficients de dilatation thermique différents et nous augmentons la température, et si en plus ces

deux couches ne peuvent pas dilater aux deux extrémités (CBC : Clamped Boundary Condition), la couche qui a le coefficient de dilatation thermique le plus bas va empêcher l'autre couche qui a coefficient de dilatation thermique le plus élevé de se déformer, par conséquent cette dernière va subir des forces de compression. Chacune des couches donc va endurer une nouvelle contrainte en fonction de la différence des coefficients de dilatation thermique entre ces deux couches et qui est :

$$\sigma_{mn} = \frac{E_{mn}}{1 - \nu_{mn}} (\alpha_2 - \alpha_1) \cdot (T_1 - T_0) \quad (4.20)$$

n est pour la couche 1 ou 2, E_m module de Young,

ν_m coefficient de poisson, α coefficient de dilatation, et T la température [4].

4.4. Le capteur de température

Il existe plusieurs méthodes de mesure de la température. La méthode que nous allons utiliser dans notre projet est celle qui utilise les RO (Ring Oscillator), car elle pourra être intégrée facilement dans la puce durant le processus de fabrication CMOS et de plus il convient à l'architecture du WaferICTM (présence des RO dans ses cellules).

4.4.1. Principe de fonctionnement d'un RO

Comme tous les oscillateurs, le principe de fonctionnement d'un capteur RO est basé sur la contre-réaction du signal de sortie vers l'entrée d'un amplificateur avec un gain ≥ 1 et avec un déphasage de 180° . Supposons que cet amplificateur a une constante de temps τ , alors qu'une période de 2π a besoin 2τ , ce qui donne une fréquence de sortie

$$f = \frac{1}{2\tau} \text{ [Hz]} \quad (4.21)$$

L'unité de base de notre RO est un inverseur logique avec une constante de temps égale à τ , donc pour obtenir le délai souhaitable un nombre n d'inverseurs sera utilisé.

$$n_i = (2n-1) \quad (4.22)$$

Ce qui donne une fréquence de sortie :

$$f = \frac{n_i}{2\tau} \text{ [Hz]} \quad (4.23)$$

Un phénomène très connu dans le monde de semi-conducteur est que la constante de temps initiale (à la température ambiante) d'une jonction change légèrement avec la température. Donc, la fréquence f est donnée par :

$$f = (f_0 - \Delta f) \text{ avec } f_0 = \frac{ni}{2\tau_0} \text{ et } f = \frac{ni}{2\tau}, \quad (4.24)$$

Alors :

$$\Delta f = (f_0 - f) = \left(\frac{ni}{2\tau_0} - \frac{ni}{2\tau} \right), \quad (4.25)$$

Avec une sensibilité du capteur S égale à :

$$S = \frac{\Delta f}{f} \quad (4.26)$$

4.4.2. Études et caractérisation des capteurs à base de RO

4.4.2.1. Effet d'auto réchauffement « Self heating » sur la réponse d'un capteur

Dans la littérature les RO sont utilisés pour mesurer la température dans un FPGA [5], le RO est activé et la mesure est effectuée après 0,2 ms pour s'assurer que la sortie de ce dernier sera stable. Cependant, il faut activer le RO pour une petite période de mesure pour éviter le phénomène d'auto réchauffement « self heating » dû à la dissipation de l'énergie du RO lui-même. La figure 4.9 montre que dans ce cas

spécifique, la mesure effectuée dans 4,2ms, avec une répétition à tout les 250ms réduit ce problème.

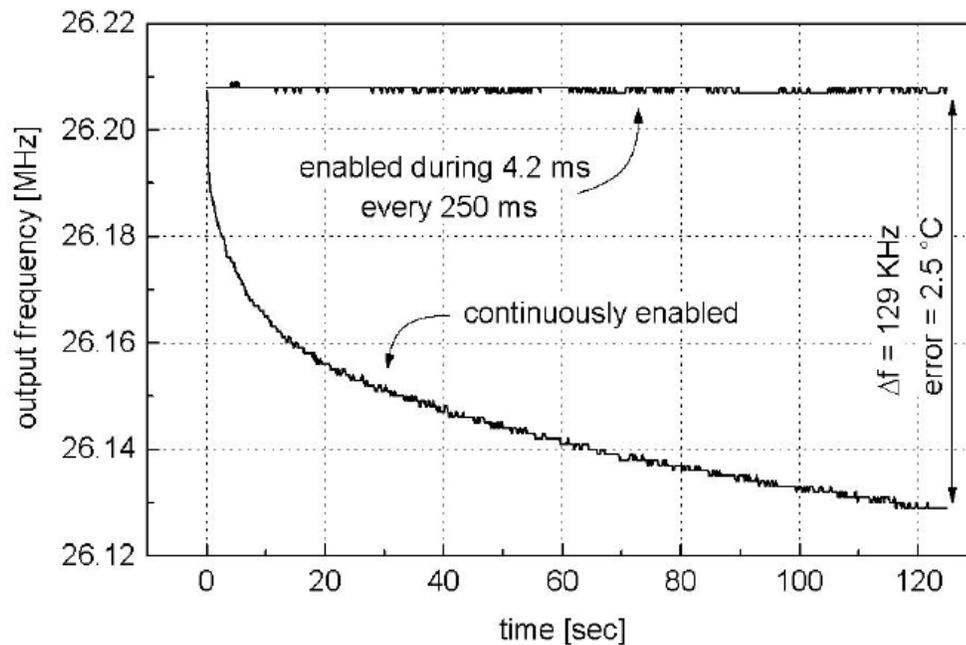


Fig. 4.9. L'effet d'auto réchauffement sur la réponse d'un RO [5].

Pour effectuer un calibrage précis du capteur [5], le RO doit être placé dans un four avec une température contrôlée. L'article montre que le capteur est capable de détecter une variation de température, mais il ne montre pas la réponse de sortie du capteur.

4.4.2.2. Effet de changement de la tension d'alimentation sur la réponse du capteur

La figure 4.10 montre le circuit d'un capteur de température à base d'un RO [6] avec une tension d'alimentation ajustable en fonction de la température. Avec l'augmentation de la température, la tension sur le RO va augmenter d'une certaine proportion et en même temps réduire la même proportion du circuit du haut. Cela sera achevé par une tension de référence variable nommée V_{mid} . Avec l'augmentation de la température, le circuit du haut augmente sa consommation en énergie beaucoup plus vite que le circuit en bas, en augmentant V_{mid} , on réduit la tension sur le circuit du haut en en même temps on l'augmente sur le circuit du RO.

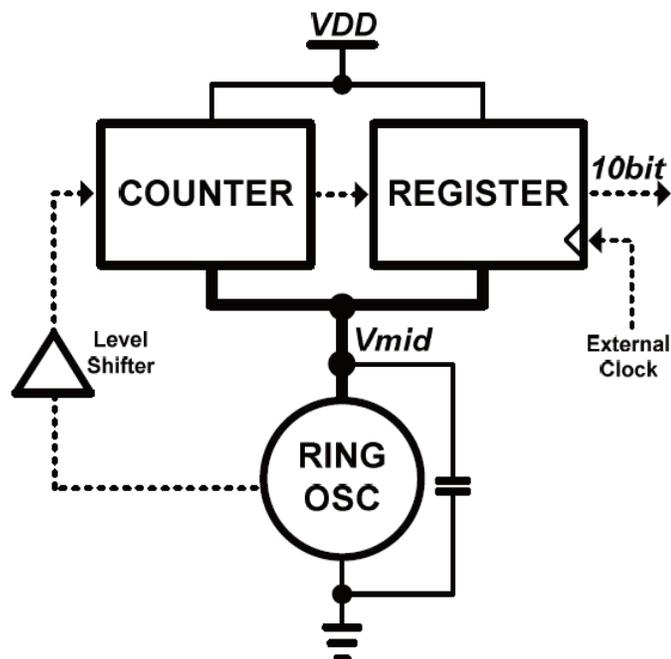


Fig. 4.10. Circuit avec une alimentation ajustable (V_{mid} ajustable) en fonction de la température [6].

La figure 4.11 montre l'effet de cette méthode sur la réponse. Nous remarquons que la réponse du circuit entre la température ambiante et 100 °C varie en même temps en fonction de la tension et de la température. Comme nous le constatons, cette réponse n'est pas linéaire, mais plutôt exponentielle, et elle dépend de deux variables (la température et la tension), mais elle pourra être considérée comme linéaire sur un petit segment de 10 °C (30 °C à 40 °C), juste pour la mesure de la température du corps de l'être humain [6].

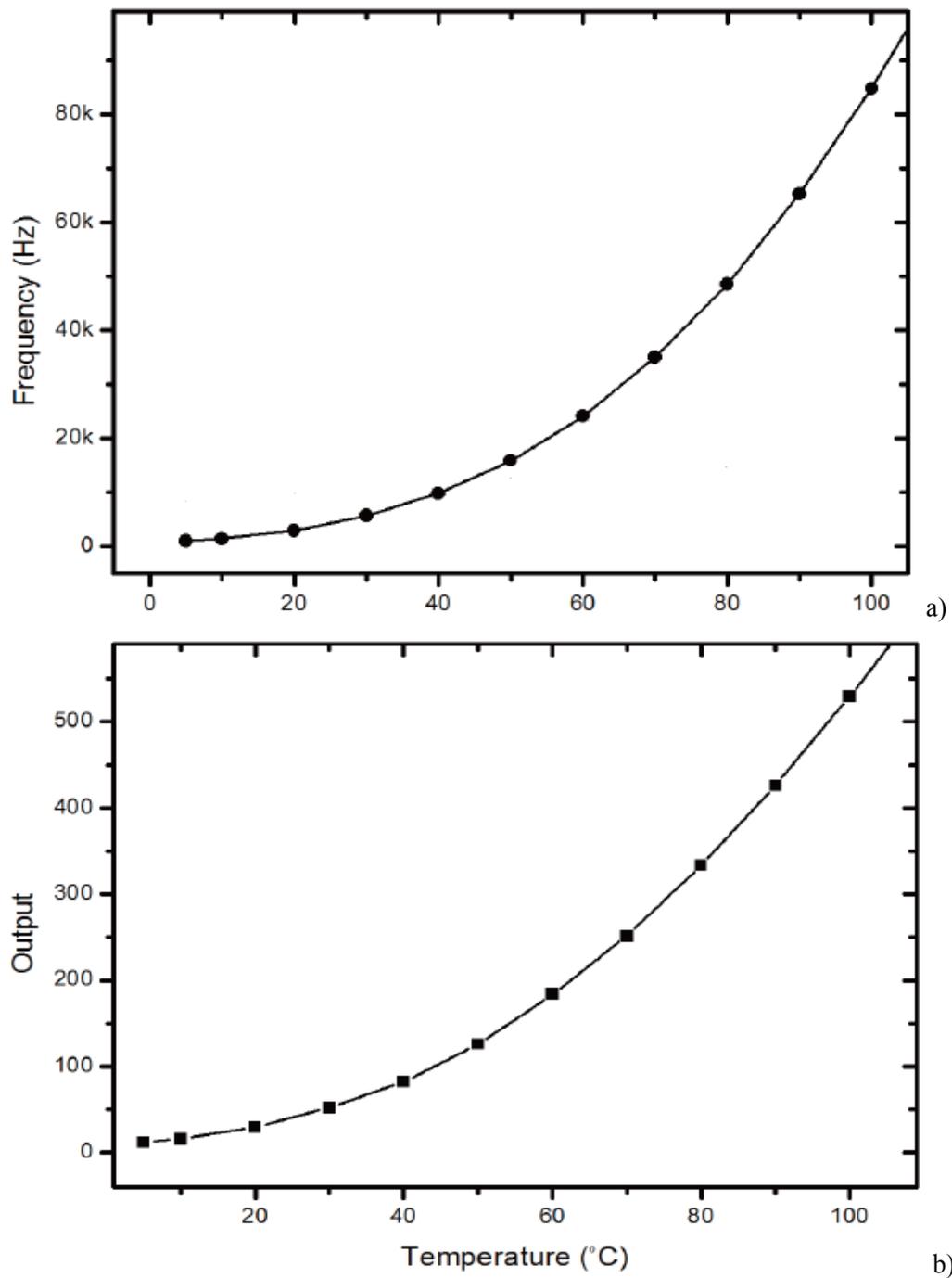


Fig. 4.11. La réponse de sortie du capteur de la température. a) en Hz, b) en nombre de comptes O [6].

Cette méthode stabilise la consommation et améliore la résolution du capteur [6]. La résolution obtenue est calculée avec la formule suivante :

$$Résolution \cong \frac{T_{\max} - T_{\min}}{O_{\max} - O_{\min}} \quad (4.27)$$

T = Température [°C];

O = Nombre de comptes.

À partir des données de la figure 2.8-b et avec N = 1, la résolution du capteur est :

$$Résolution \cong \frac{100 - 5}{530 - 12} = 0,18^{\circ}C / LBS \quad (4.28)$$

Une autre étude a été effectuée sur l'effet de la variation de la tension sur la réponse d'un capteur de température basée sur les RO, en rapport avec le nombre d'inverseurs dans la boucle [7]. La température a été gardée stable à 40 °C et la tension d'alimentation du capteur a été changée de 0,95V à 1,05V avec des pas de 10 mV. La mesure a été répétée 254 fois pour produire la figure 4.12 avec la moyenne de la déviation standard de ces mesures.

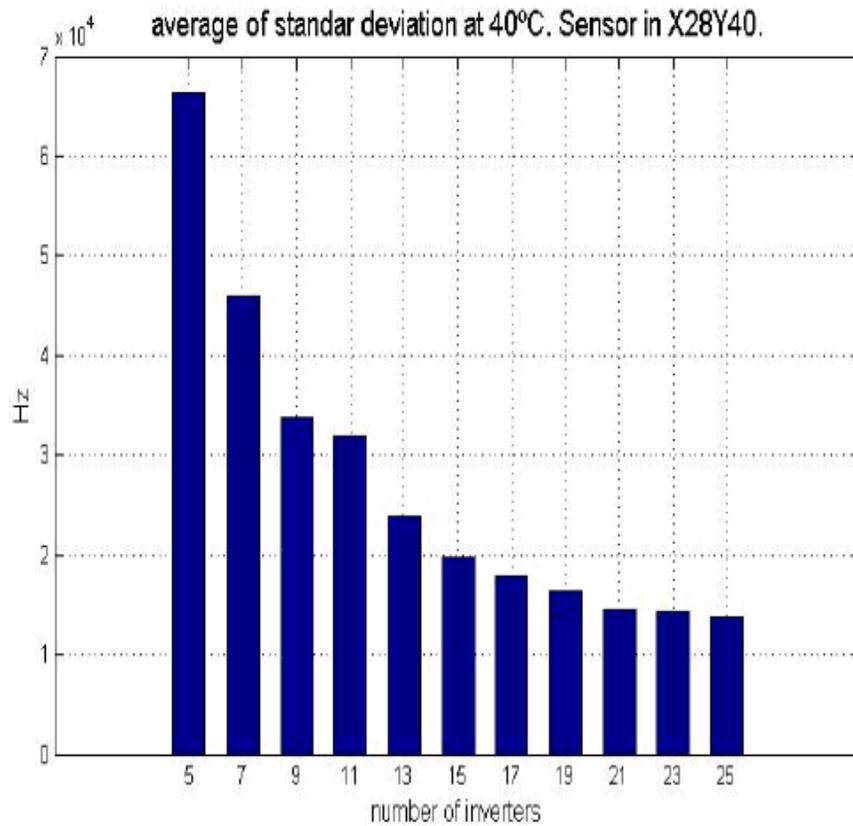


Fig. 4.12 La stabilité par rapport à la variation de tension de la réponse de sortie du capteur RO en fonction de nombre d'inverseur [7].

Ces résultats montrent qu'avec l'augmentation de nombre d'inverseurs, le capteur devient moins sensible à la fluctuation de la tension d'alimentation, ce qui est désirable pour un capteur de température plus stable.

4.4.2.3. Linéarité de la réponse des capteurs de température à base de RO

Comme nous l'avons vu dans le paragraphe précédent (figure 4.11), [6] a obtenu une réponse exponentielle. Mais la réponse est obtenue avec la contribution de deux variables dépendantes. Il suffit que l'une de ces deux variables soit non-linéaire pour que la réponse soit non-linéaire. Le circuit dans [6] a été implémenté et simulé avec la technologie CMOS de $0,13\mu\text{m}$ (figure 4.13).

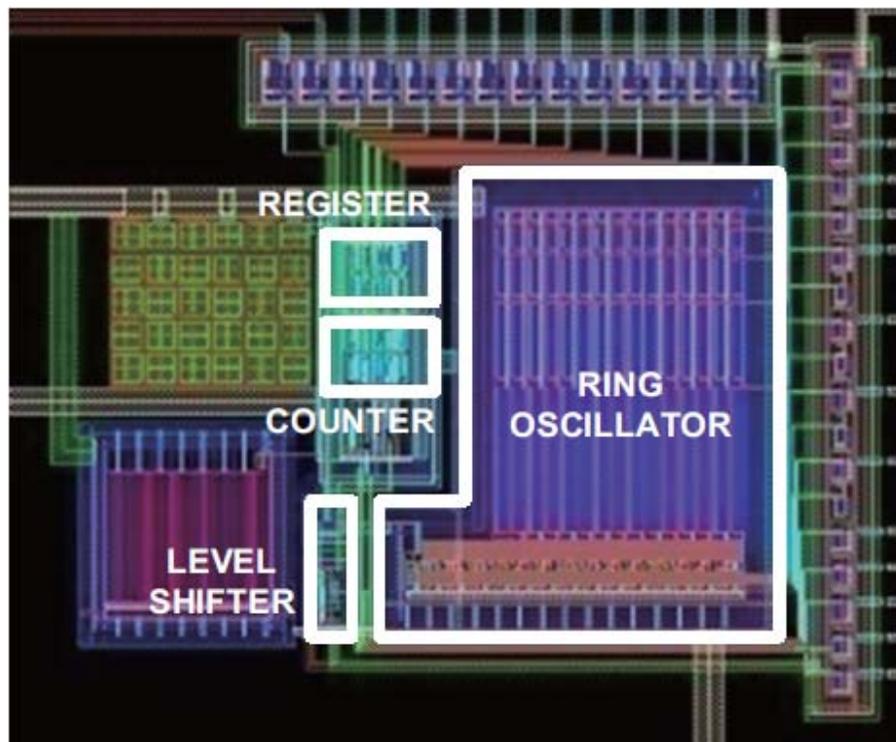


Fig. 4.13. Implémentation du capteur avec la technologie CMOS de $0,13\mu\text{m}$ [6].

Une autre étude [7] montre la réponse d'un capteur RO à trois inverseurs en fonction de la température tout en gardant la tension constante (figure 4.14). Le capteur a été implémenté sur un FPGA Vertex-5 LX30FF676-1 de Xilinx, comme nous le constatons la réponse est exponentielle.

Frequency response of one sensor (X26Y40) with constant voltage core (1.0 V)

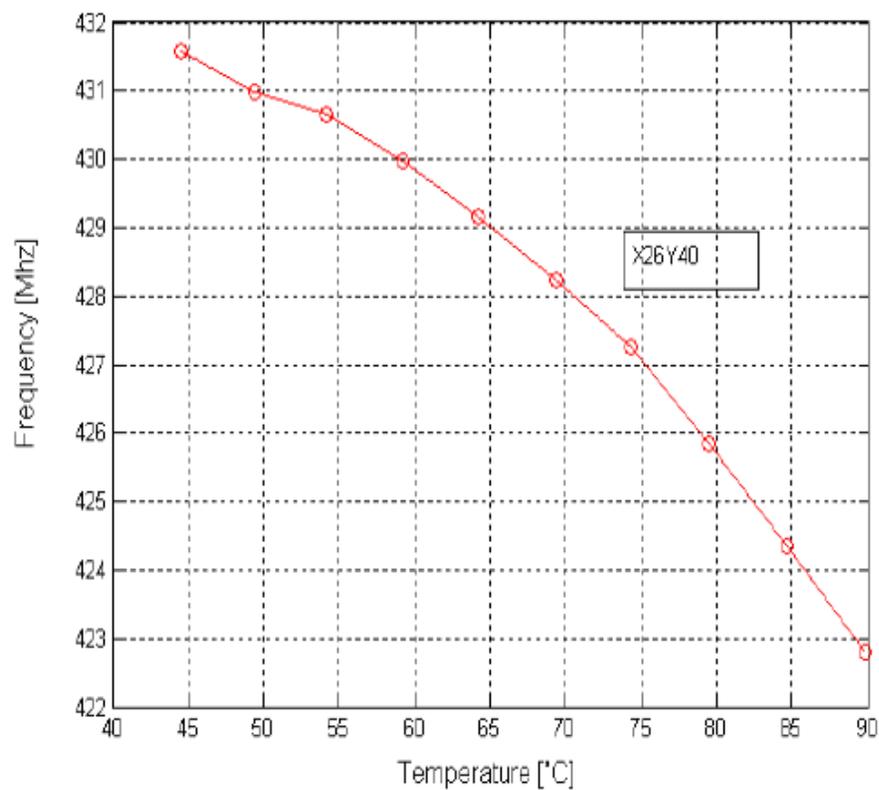


Fig. 4.14 La réponse de sortie d'un capteur de température RO à trois inverseurs en fonction de la température [7].

Un nombre important d'autres études sur des capteurs de température CMOS à base du RO montrent des réponses linéaires [8-12].

Également, [9] montre une bonne linéarité de -50 °C à 120 °C et une légère non-linéarité pour les hautes températures (120 °C à 150 °C) figure 4.15.

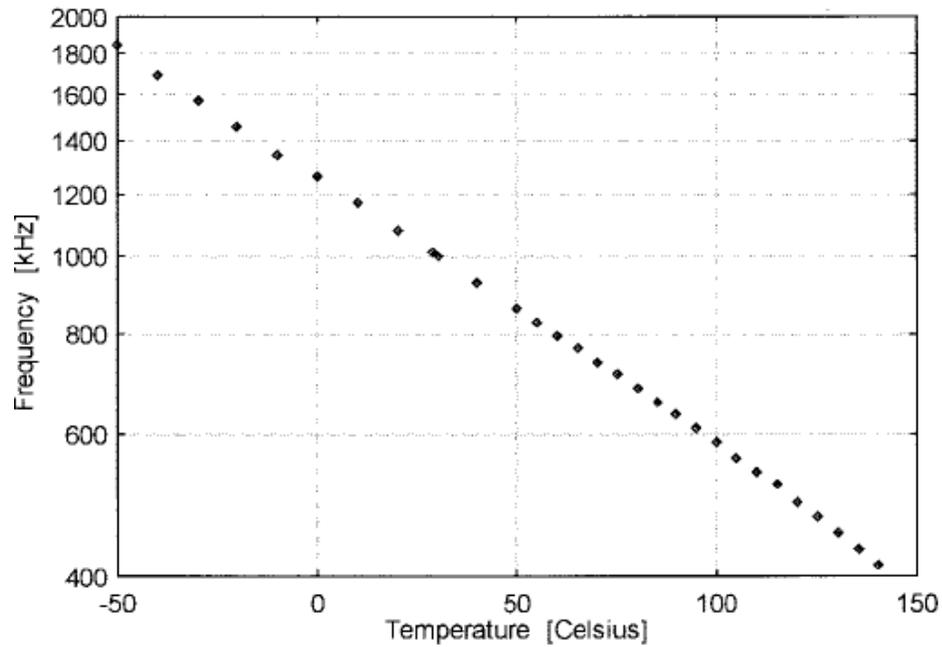


Fig. 4.15. La réponse de sortie d'un capteur de température RO -50 à 120 °C [9].

Alors que [11] utilise un RO pour mesurer une contrainte externe appliquée sur un MOSFET pour le but de détecter une variation dans la capacitance parasite de ce dernier due à une contrainte sur sa structure, et il montre une réponse linéaire (figure 4.16) avec une réponse en fréquence normalisée.

Donc la fréquence de sortie du RO est :

$$\begin{aligned}
f &= \frac{1}{n \times T_D} = \frac{1}{nC(R_n + R_p)} \\
&= \frac{1}{nV_{DD}C\left(\frac{1}{I_{ON_n}} + \frac{1}{I_{ON_P}}\right)} \\
&= \frac{1}{nV_{DD}} \cdot \frac{I_{ON_P}}{1 + \frac{I_{ON_P}}{I_{ON_n}}} \cdot \frac{1}{C}
\end{aligned} \tag{4.29}$$

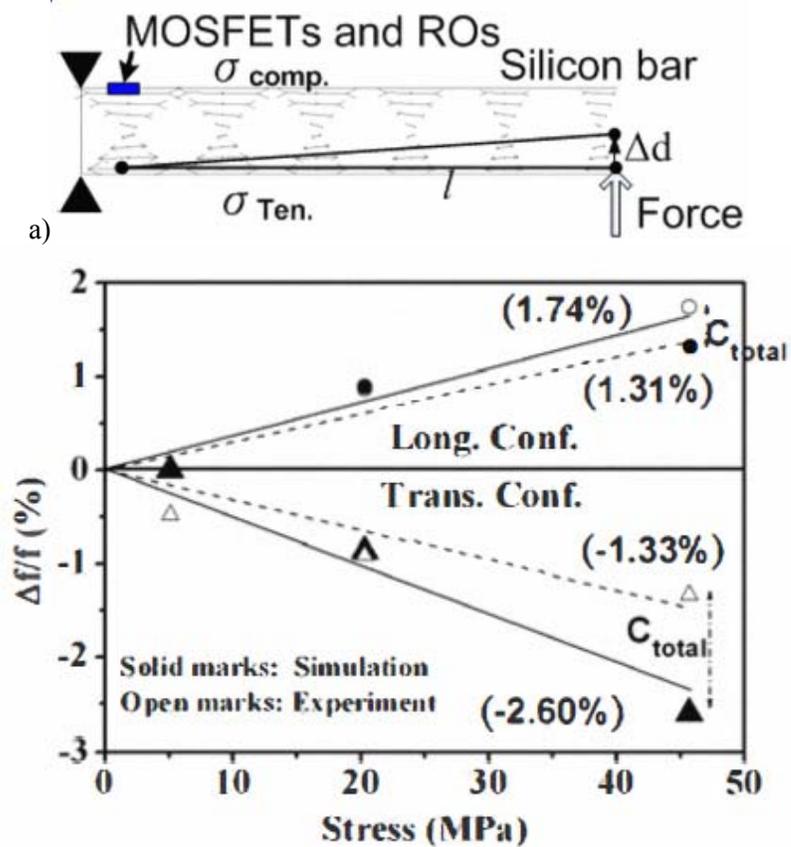
V_{dd} est la tension de polarisation, R_n et R_p sont les résistances de l'état « On » des transistors nMOSFET et pMOSFET respectivement, C est la capacitance totale, I_{ON_n} et I_{ON_p} sont proportionnelles à la mobilité des électrons (μ_n) et des trous (μ_p) respectivement dans nMOSFET et les pMOSFET.

La fréquence normalisée du RO sera résolue par l'équation à dérivée partielle suivante :

$$\frac{\partial f}{f} = \frac{\partial M(\mu_n, \mu_p)}{M(\mu_n, \mu_p)} - \frac{\partial C}{C} \tag{4.30}$$

$\frac{\partial M}{M}$ est une fonction partiellement différentielle :

$$\begin{aligned} \frac{\partial M}{M} &= \frac{\partial \mu_p}{\mu_p} - \frac{\partial(1 + \frac{\mu_p}{\mu_n})}{1 + \frac{\mu_p}{\mu_n}} \\ &= \frac{\partial \mu_n}{\mu_n} - \frac{\partial(1 + \frac{\mu_n}{\mu_p})}{1 + \frac{\mu_n}{\mu_p}} \end{aligned} \quad (4.31)$$



b)
Fig. 4.16. a) Le montage de l'expérience. b) La réponse de sortie normalisée du capteur RO en fonction de la contrainte [11].

[10] Utilise une méthode novatrice pour réduire l'effet de variation de la tension sur la réponse, il utilise deux CMOS RO, le premier est pour mesurer la température avec une réponse linéaire, et le deuxième utilise une structure cascadiée à couplage croisé (figure 2.17) pour rendre ce dernier indépendant de la température.

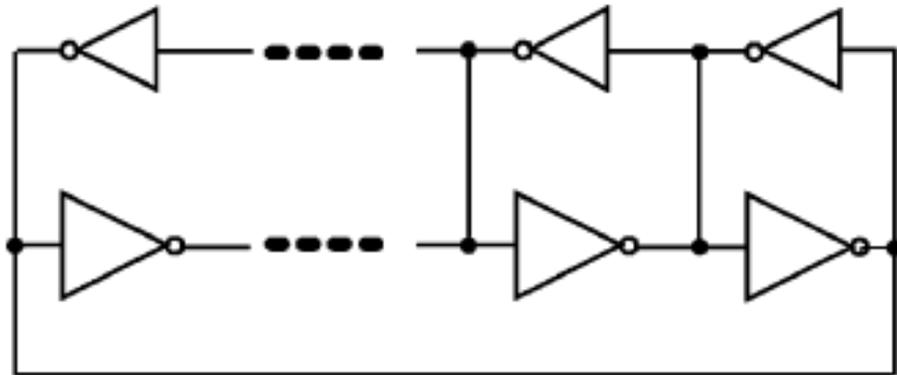


Fig. 4.17. Structure cascadiée à couplage croisé indépendamment de la température [10].

Due à une composition entre les deux RO, la tolérance due à la variation de la tension d'alimentation va être réduite, la figure 4.18 montre le montage du capteur.

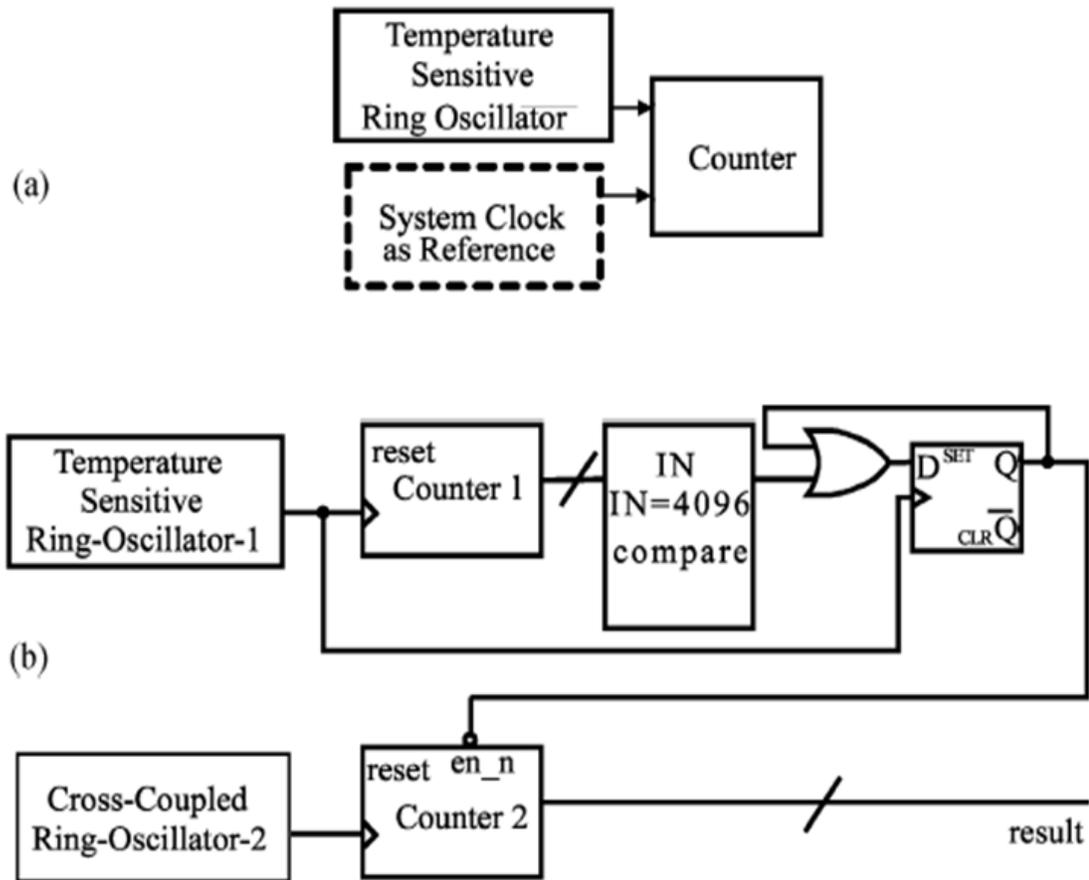


Fig. 4.18. a) Capteur conventionnel; b) Nouveau capteur à deux RO [10].

4.4.2.4. Algorithme de linéarisation de la réponse d'un RO

Il existe plusieurs méthodes de linéarisation de la réponse d'un capteur de température. Comme nous avons vu précédemment lorsque la réponse n'est pas linéaire, elle prendra toujours une forme exponentielle. Alors, ce genre de réponse pourra être linéarisée tout en appliquant à la réponse une équation polynomiale de linéarisation comme celle du J. S. Steinhart and S. R. Hart [13].

$$Y = \frac{1}{T_k} = (C_1 + C_2 \ln(R) + C_3 \ln(R)^3) \quad (4.32)$$

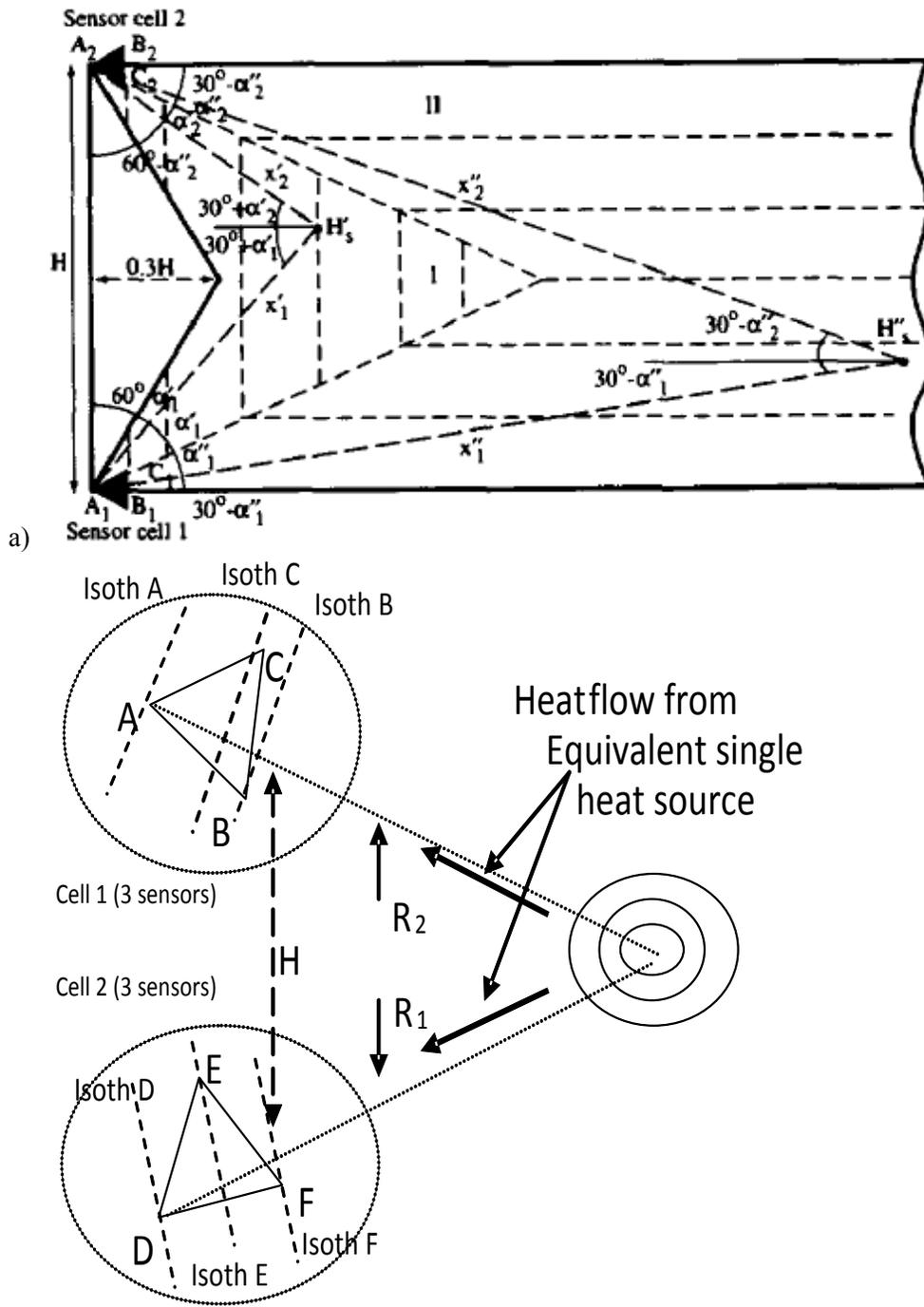
T_k est la température en Kelvin, R est la résistance (ou la fréquence) dépendante de la température T , et C_1 , C_2 et C_3 sont les coefficients spécifiques pour chaque courbe à être linéarisée.

4.5. Les algorithmes et le traitement de données provenant des capteurs de température à base de RO

Il existe plusieurs approches pour aborder ce problème, nous présentons dans cette session quelques méthodes les plus connues. En particulier, nous montrerons les avantages et les inconvénients de chacune de ces méthodes.

4.5.1. Méthode de direction du gradient de température

[14, 15] utilisent la méthode de direction de gradient de la température (Gradient Direction Sensor (GDS)), cette méthode a été prouvée être efficace pour la détection d'une seule source de température sur une surface. Comme la figure 4.19 montre, cette méthode est basée sur la détection des isothermes par deux capteurs, chaque capteur est composé de trois RO qui forment un triangle entre eux.



b)
Fig. 4.19. a) et b) Le principe de détection par GDS d'une source de chaleur.

$$\begin{aligned}
 b &= \overline{AD}, \overline{AD} \xleftrightarrow{\text{corresponds à}} T_B - T_A \xleftrightarrow{\text{corresponds à}} V_B - V_A \\
 b + c &= \overline{AE}, \overline{AE} \xleftrightarrow{\text{corresponds à}} T_C - T_A \xleftrightarrow{\text{corresponds à}} V_C - V_A
 \end{aligned}
 \quad ; \quad (4.34)$$

et $\text{tg } \alpha$ est:

$$\text{tg } \alpha \leftrightarrow \frac{2}{\sqrt{3}} \left(\frac{V_B - V_A}{V_C - V_A} - \frac{1}{2} \right) \quad (4.35)$$

et la température à la source est :

$$T_H \leftrightarrow V_H = \frac{H}{a} (V_C - V_A) \frac{(\text{tg}^2 \alpha_1 + 1)(\sqrt{3} + \text{tg} \alpha_2)}{\sqrt{3}(1 - \text{tg} \alpha_1 \cdot \text{tg} \alpha_2) - (\text{tg} \alpha_1 + \text{tg} \alpha_2)} + V_A \quad (4.36)$$

À cause de la petite distance entre les RO, la différence de température entre les isothermes à chaque RO est très petite, ce qui nécessite une très grande précision ($\geq 0,01^\circ \text{C}$) dans les capteurs pour que cette méthode donne des bonnes résultats.

4.5.2. Méthode avec des capteurs distribués

Comme le montre la figure 4.21, [16] utilise une technique des capteurs distribués sur toute la surface ($x \times y$), cette méthode aide à étudier en détail les caractéristiques thermiques comme les pics thermiques et les gradients de température tout en produisant une carte thermographique de la surface ($x \times y$). La

figure 4.22 montre la carte thermographique d'un FPGA produite avec un tableau (4×10) capteurs.

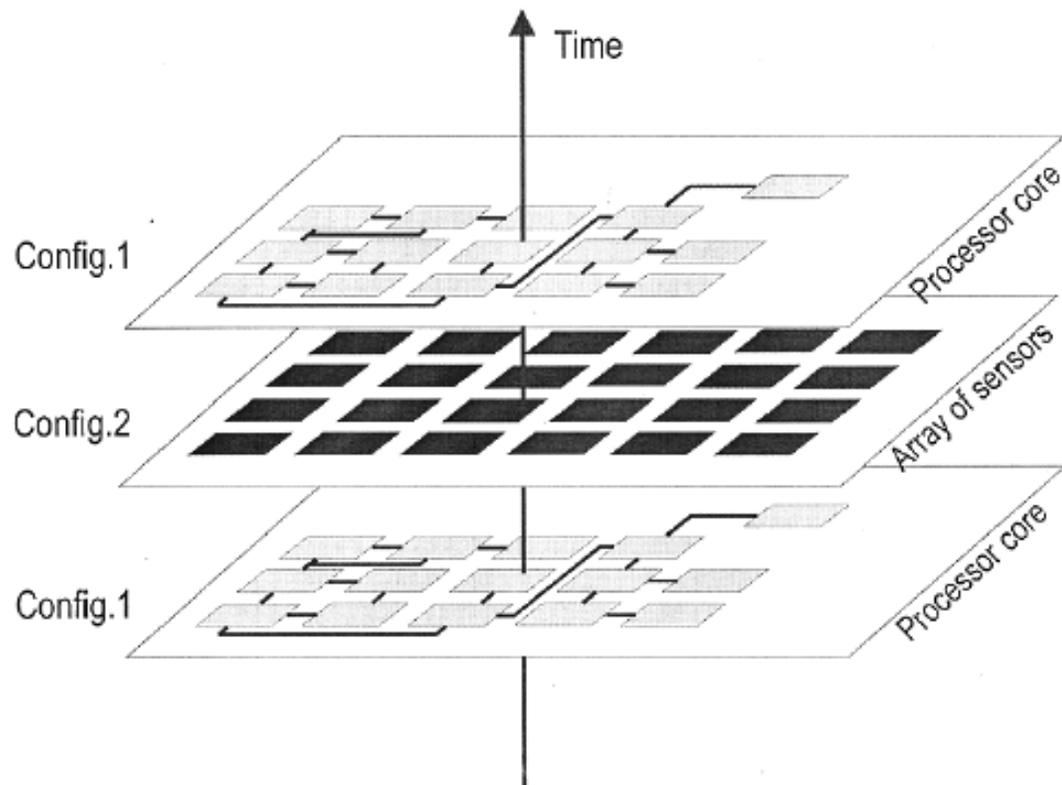


Fig. 4.21 Capteurs distribués de température composés des cellules RO sur toute la surface ($x \times y$) de la puce [16].

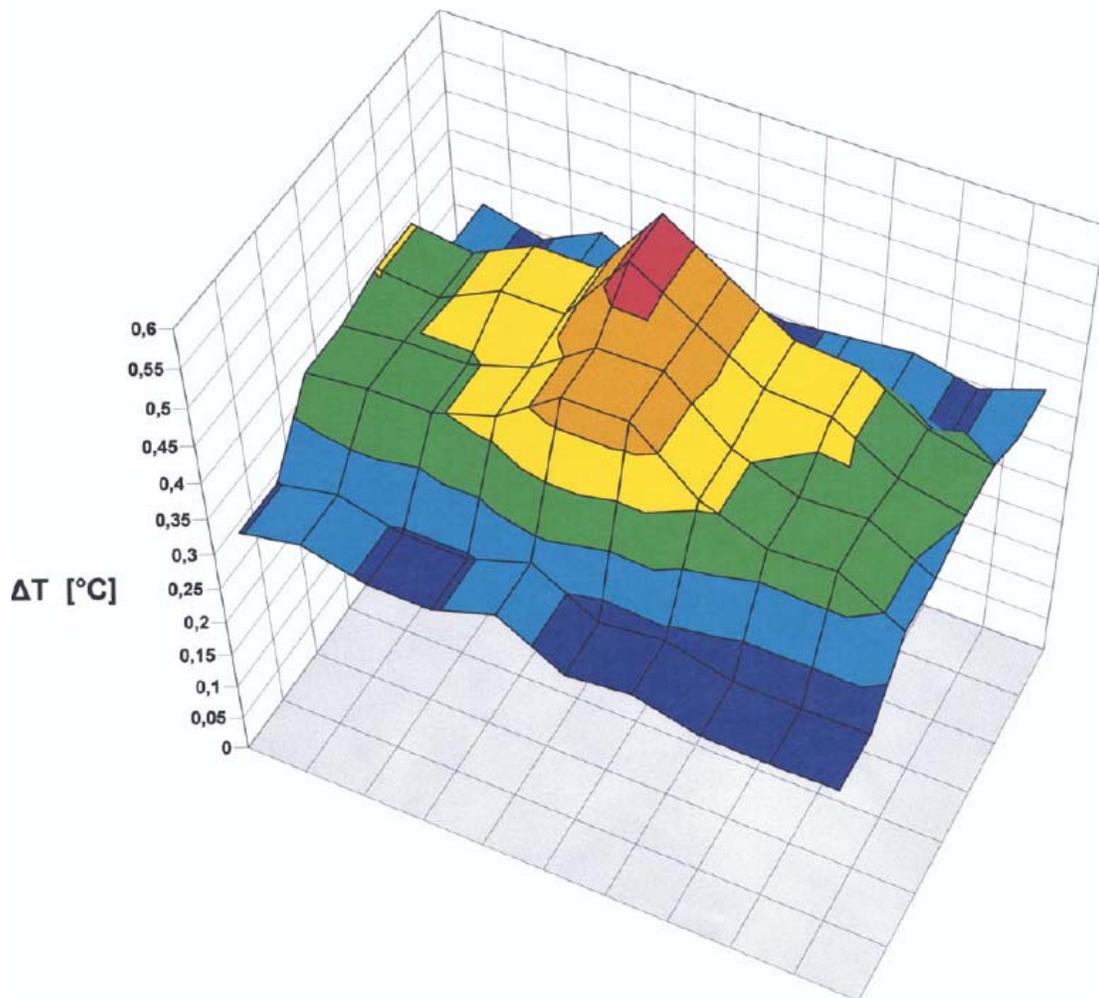


Fig. 4.22. La carte thermographique d'un FPGA produite avec un tableau de (4×10) RO [16].

4.5.3. Méthode de détection des cercles

La référence [17] utilise la transformée de Hough. Chaque point sur le périmètre d'un cercle devient une ligne qui passe par le centre du cercle (les isothermes dans notre cas), la figure 4.23-a montre l'image originale avec des cercles,

la figure 4.23-b applique la transformée de Hough, et la figure 4.230-c montre une modification à la transformée de Hough pour montrer les centres des cercles.

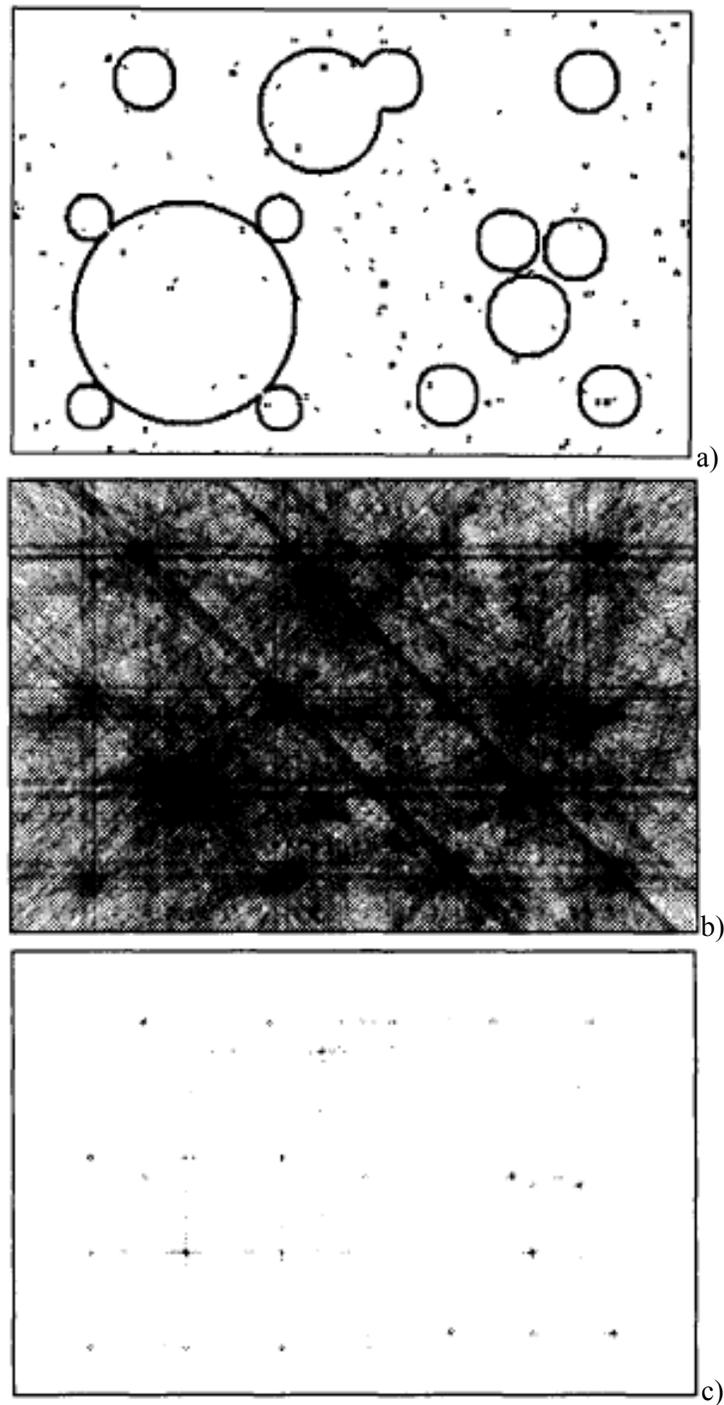


Fig. 4.23. a) Image originale, b) la transformé de Hough sur l'image originale, c) une modification à la transformée de Hough pour montrer les centres des cercles [17].

4.5.4. Méthode avec opérateur spatiotemporel

La référence [18] propose un opérateur spatiotemporel qui utilise le contraste isotopique et l'homogénéité régionale pour la détection d'un mouvement. Cette méthode pourrait être ajustée pour qu'elle soit applicable sur des cartes thermographiques pour la détection des pics thermiques.

5. Méthodologie et conception

Ce projet est composé principalement de deux parties, implémentation matérielle (hardware implementation) des capteurs, et le développement logiciel (Software development) de l'algorithme de détection et de monitoring des contraintes thermomécaniques :

L'implémentation et l'étalonnage d'un capteur de température basé sur le principe d'un RO a été développé sur un FPGA car ni un WaferICTM fonctionnel ni un ``Test Chip``, n'ont été encore prêts à la date de production de ce mémoire;

Le traitement des données provenant des capteurs thermiques est effectué par un algorithme de gestion de la contrainte thermomécanique. Puisqu'un WaferICTM fonctionnel n'a pas encore été disponible, nous avons utilisé, comme alternative, des images thermiques provenant d'une caméra infrarouge avec des sources de chaleurs externes contrôlables placées sous un WaferICTM cobaye.

5.1. Limitations et contraintes

Comme nous avons vu dans le chapitre précédent (état de l'art) de ce document, il existe plusieurs méthodes et techniques pour la détection de température. Toutefois, il faut choisir la meilleure méthode qui répond bien aux exigences du projet, donc à son cahier de charge, il faut respecter les contraintes imposées par ce dernier dans le cadre du projet DreamWaferTM. En général, ces contraintes sont : budgétaires, d'échéancier, du logiciel, d'implémentation physique des capteurs et de communication (entre capteur et algorithme).

5.1.1. Contraintes budgétaires et échéancier

Comme n'importe quel projet R&D industriel et commercial, le projet DreamWafer™ impose des échéanciers très précis à respecter surtout lorsque c'est un travail d'équipe et dont l'avancement global du projet dépend de chaque membre de l'équipe. De plus, il faut garder le coût de développement en dessous d'un certain seuil pour que le produit final soit compétitif et rentable.

5.1.2. Contraintes d'implémentation physique des capteurs, de communication, et de logiciel

La figure 5.1 montre l'architecture du WaferIC™. Sa surface est de 24500 mm² et il est composé de 76 réticules. Chaque réticule est composé de 1024 cellules, donc 77824 cellules au total. De plus, chaque cellule est composée de 16 nanopads, donc 1 245 184 nanopads au total.

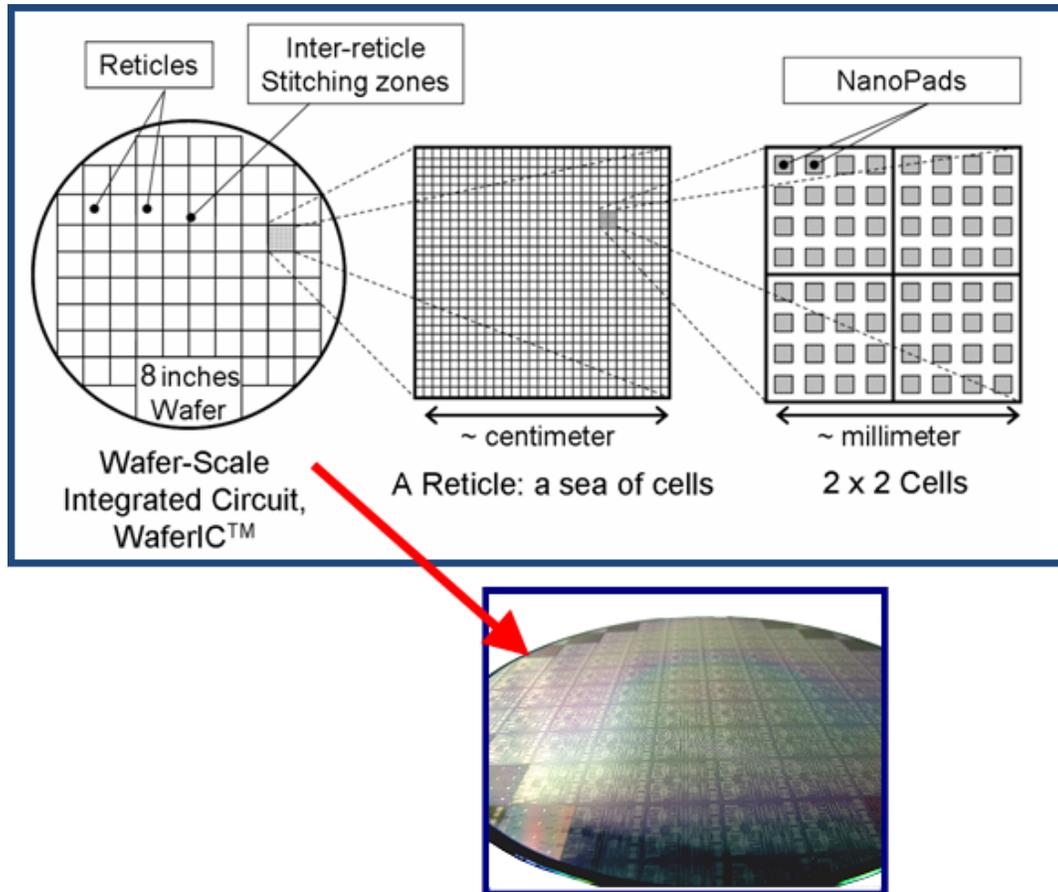


Fig. 5.1. L'architecture du WaferIC™.

Selon le cahier de charge, le capteur de température doit être implémenté au niveau de la cellule. Pour réduire le coût de fabrication, tous les circuits des cellules et des nanopads respectivement doivent être identiques (clone). Le capteur de température peut être implémenté au niveau de la cellule et/ou au niveau du nanopad. Mais les restrictions de surface éliminent la possibilité d'un capteur par nanopad. La limite de surface par cellule pour chaque capteur est de 16 bits maximum. Le même capteur pourrait être utilisé pour mesurer la pression externe exercée sur la surface du WaferIC™ à chaque point. Toutefois, il faut respecter les règles de conception CMOS

0,18 μ m de la fonderie Tower Semiconductor Ltd. avec une orientation unique des MOSFET d'une taille minimale.

Les capteurs doivent utiliser l'interface JTAG déjà existante pour communiquer les données à l'algorithme de calcul. Le capteur doit recevoir les signaux de contrôle à travers le JTAG car une horloge interne de référence dans le WaferICTM n'est pas possible. Le circuit de contrôle du capteur doit avoir la possibilité de multiplexer le signal du RO avec un signal provenant d'un VCO local à la cellule.

L'algorithme de calcul doit effectuer efficacement le traitement nécessaire pour produire les résultats en temps réel tout en affichant les résultats.

5.2. Conception matérielle des capteurs de température

Pour mesurer la température dans le WaferICTM, il faut tout d'abord installer physiquement des capteurs de température dans le WaferICTM lui-même. Les valeurs de températures mesurées localement, doivent être converties localement en valeurs digitales pour ensuite être envoyées sur l'interface JTAG vers l'algorithme de traitement.

5.2.1. Architecture du capteur

Nous avons choisi une architecture distribuée [16] au niveau de la cellule, donc 77824 capteurs discrets distribués sur toute la surface du WaferICTM.

Pour calculer le temps d'accès, la fréquence minimale d'un RO est estimée à 50MHz, le compte maximal d'un compteur de 16 bits est $2^{16} = 65536$ comptes. Donc une constante de temps $t_{RO\ MAX} = 65536/50\ MHz = 1,311\ ms$, et $t_{total} = t_{RO} * 77824 = 102\ s$. Par conséquent, cela veut dire que pour produire une image thermographique de très grande résolution de toute la surface du WaferICTM, nous avons besoin de 2 minutes. Lorsque nous parlons ici de résolution, ce n'est pas la résolution de l'image, car celle-ci est déjà déterminée par le nombre de cellules ou de capteur (77824 capteurs), mais plutôt nous parlons de la résolution thermique qui correspond au niveau de gris qui peuvent être générés pour la gamme ou la bande de température visée (disons 30°C à 100°C). Mais nous pouvons augmenter la vitesse du processus tout en réduisant la constante du temps t_{RO} , mais ça va réduire la résolution. En fin de compte, nous devons faire un compromis entre vitesse et résolution thermique (°C/niveau). Mais avec 16 bits de résolution, nous avons une grande marge de flexibilité pour faire ce choix. Nous devons encore ajouter la fonctionnalité suivante ; chaque RO est sélectivement adressable. Cela veut dire que chaque RO peut avoir sa propre t_{RO} ($t_{RO(n)}$). Ainsi, nous pourrions produire une image thermographique avec de résolutions thermiques et d'images variables, même nous avons la possibilité de surveiller directement une région, même un ou plusieurs groupes de RO n'importe où sur la grandeur du WaferICTM.

5.2.2. Distribution des RO au niveau de la cellule

Le capteur va être utilisé pour mesurer la température et la pression sur la surface du WaferICTM, et comme la pression appliquée sur un point d'une surface supporté est une grandeur scalaire normale à la surface. Alors, un capteur distribué sur toute la surface de la cellule sera un bon choix. La figure 5.2 montre le circuit que nous avons conçu et la distribution des composantes dans une cellule.

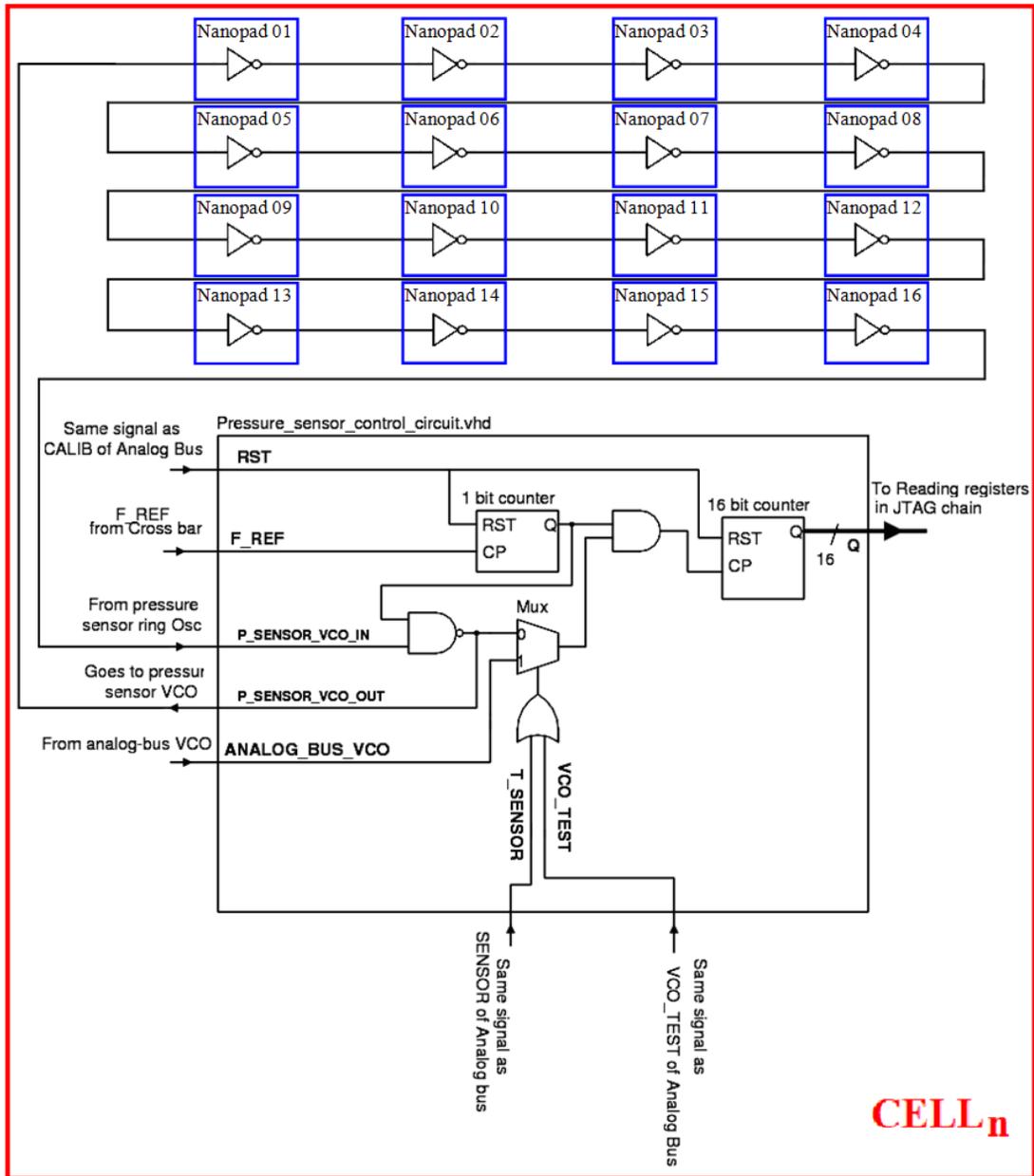


Fig. 5.2. Un circuit d'un seul RO pour une cellule sur le WaferIC™.

Nous avons utilisé un RO de 17 inverseurs, donc un inverseur par nanopad pour un total de 16 inverseurs (au niveau de nanopad) pour les 16 nanopad de la cellule, et le 17^e inverseur avec le circuit de contrôle de 16 bits ont été implémenté au niveau de la cellule. Avec un RO de 17 inverseurs, nous réduisons efficacement la sensibilité de la réponse du capteur à la tension d'alimentation [7]. Ce circuit de contrôle, contrôle encore le signal du VCO de la cellule et la sortie est dirigée vers une chaîne JTAG (comme il a été précisé dans le cahier de charge). Le code VHDL complet avec implémentation sur FPGA et les résultats de validation de fonctionnement sont présentés à l'annexe A.

5.2.3. Définition des paramètres du circuit de contrôle

Pour savoir le nombre de bits requis pour le circuit de contrôle, nous avons établi la relation suivante,

$$2^x = y \Rightarrow x = \log_2 y = \frac{\log_{10} y}{\log_{10} 2} = n \text{ bits} \quad (5.1)$$

y est le compte maximal par période.

Et

$$y = \frac{T_{\max} - T_{\min}}{\text{resolution}} \quad (5.2)$$

Dans notre design de la figure 5.2, nous avons utilisé x=16 bits, qui est le maximum disponible alloué à notre circuit par le cahier de charge, donc y=65536, et une résolution maximale de la température par capteur pour une gamme de 35°C à 125°C de 0,001375°C/compte.

5.2.3.1. Signaux d'entrée et de sortie du capteur

Le circuit d'un RO dans chaque cellule reçoit les signaux suivants :

1. les signaux VCO_TEST et T_SENSOR pour effectuer la sélection de l'un des deux oscillateurs;
2. La constante de temps F_REF (synchronisée sur deux fronts montants) pour activer/désactiver le RO;
3. Et un signal de reset (RST) asynchrone.

Par la suite, il envoie au JTAG la sortie du compteur de fréquence Q (0..15) sur 16 bits.

5.3. Unité de contrôle

Comme nous avons précisé dans la section 5.2.1, la fréquence, la résolution, et l'adressage de chaque capteur sont individuellement sélectionnables. Donc, nous proposons une unité de contrôle qui reçoit des commandes d'un usager ou/et d'un algorithme de traitement de données pour envoyer une requête (individuellement via le JTAG) à un circuit RO d'une cellule spécifique et qui doit contenir les informations suivantes :

- 1- L'adresse du capteur sur 17 bits (0 à 77824);
- 2- un bit pour le choix du capteur (le RO dans notre cas);
- 3- un bit pour la constante du temps t_{RO} (deux fronts montants dans le temps)

4- un bit pour le *reset* du circuit.

Le même programme de contrôle doit saisir via le JTAG la réponse du même circuit RO qui devrait retourner la valeur du compteur de fréquence sur 16 bits;

Après avoir reçu cette information, l'unité de contrôle doit agir en conséquence tout en envoyant l'information au demandeur (usager ou algorithme de traitement de données). Si l'information est une température, peut-être un algorithme de linéarisation [13] sera nécessaire.

5.4. Algorithme de traitement de données

Avec tout le contrôle et toute la flexibilité que nous avons avec notre capteur de température, tous les algorithmes spécifiés dans la section 3 sont applicables. Mais comme c'est spécifié dans le cahier de charge nous devons garder l'algorithme le plus simple et efficace que possible. Notre proposition sera implantée dans le système WaferBoard, par conséquent, c'est très probable d'avoir des modifications majeures durant les phases de développements et de tests de ce dernier.

Finalement, un algorithme de traitement de données va communiquer avec l'unité de contrôle pour avoir la valeur d'une température spécifique sur la surface du WaferICTM. Bien sûr il faut bien gérer cette tâche de lecture pour éviter l'effet d'auto réchauffement du RO [5]. L'algorithme va spécifier au module de contrôle l'adresse et la résolution de la valeur demandée. Selon la valeur reçue, l'algorithme, va probablement faire d'autres requêtes et il doit produire des informations utiles sur

l'état des pics thermiques et la distribution de la chaleur sur toute la surface du WaferICTM. En effet, le programme doit identifier en temps réel tous problèmes thermiques ou bien plus précisément tous problèmes sur le point de devenir dangereux pour le WaferICTM.

Nous proposons un algorithme qui va générer une carte thermographique de la surface du WaferICTM. Dans un temps discret t , l'algorithme proposé va balayer tous (ou sélectivement sur une région d'intérêt) les capteurs sur la surface du WaferICTM, et il va placer ces informations dans un espace de travail spatial et discret $f(x,y)$, qui contient M rangés et N colonnes. L'algorithme va chercher d'autres plans dans un temps discret t_n pour avoir la possibilité d'effectuer un traitement en temps réel dans un plan spatiotemporel $f(x,y,t)$ ou partiellement spatiotemporel selon le cas.

5.4.1. Méthodes possibles pour l'algorithme de traitement de données

Nous avons exposé dans la section 4 plusieurs, méthodes intéressantes qui pourraient être adaptées à notre application.

La méthode de détection des cercles [17] pourrait être utile pour la détection des isothermes et pour identifier les sources des chaleurs, mais pour arriver à obtenir les cercles, plusieurs techniques de traitement d'image devront être appliquées nécessaires comme : le filtrage, la médiane, un seuillage, le gradient pour la détection des cercles, etc..

Encore la méthode d'opérateur spatiotemporel qui utilise le contraste isotopique et l'homogénéité régionale pour la détection d'un mouvement [18], pourrait être simplifiée et adaptée pour qu'elle soit applicable sur des cartes thermographiques pour la détection des pics thermiques et des gradients de température.

5.4.2. Méthodes adaptées pour l'algorithme de traitement de données

De nombreux ouvrages parlent des théories et des techniques de traitement des images et des vidéo [19-26]. Nous nous sommes servis de plusieurs techniques pour développer un algorithme adéquat pour notre application. La méthodologie suivie pour le développement de notre algorithme est la suivante :

1. À un temps t_0 , nous balayons tous les capteurs sur la surface du WaferICTM, pour produire une image thermographique de ce dernier;
2. à partir de cette image, nous pouvons tout de suite identifier les pics thermiques (unité[°C]) dangereux $D(x,y)$ tout en comparant chaque lecture avec un seuil T_s prédéfini,

$$D(x, y) = \begin{cases} 0 & \text{si } f(x, y) \leq T_s \\ 1 & \text{si } f(x, y) > T_s \end{cases}, \quad (5.3)$$

Où $D(x, y)$ identifie chaque pic thermique avec un 1, et T_s est le seuil de température.

3. pour savoir la vitesse avec laquelle une température change localement dans le temps sur chaque cellule du WaferICTM (unité[°C/s]), nous allons appliquer la première dérivée pour chaque cellule

$$g(x, y, t) = \frac{\partial f(x, y, t)}{\partial t} = f(x, y, t_{n+1}) - f(x, y, t_n)$$

(5.4)

Où $g(x, y, t)$ est la vitesse de changement de la température sur chaque capteur (ou cellule).

Encore $g(x, y, t)$ sera comparée avec un seuil T_{s1} qui indique que cette région est sur le point de devenir dangereuse

$$G(x, y) = \begin{cases} 0 & \text{si } g(x, y, t) \leq T_{s1} \\ 1 & \text{si } g(x, y, t) > T_{s1} \end{cases} \quad (5.5)$$

Où $G(x, y)$ représente juste les vitesses de variation de température qui dépassent T_{s1}

4. Une fonction de seuillage d'intensité à pas fixes et à multiniveau (ou sur l'histogramme des intensités) sera nécessaire pour une bonne reproduction des isothermes dans le calcul du Laplacien dans le point 6 suivant. Donc, le seuillage va produire un nombre n_n des régions de gris.

$$g(x, y) = \begin{cases} n_0 & \text{if } f(x, y) \leq T_0 \\ n_1 & \text{if } T_0 < f(x, y) \leq T_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ n_n & \text{if } T_{n-m} < f(x, y) \leq T_n \end{cases}, \quad (5.6)$$

5. Un filtre médian est nécessaire pour réduire le bruit à haute fréquence ce qui va aider à produire (dans point 6) des isothermes distinctes.

$$m(x, y) = \text{median}\{f(x, y)\} \quad (5.7)$$

6. Maintenant nous appliquons le Laplacien sur $f(x,y)$ qui est un opérateur différentiel à coefficients constants et homogènes du second ordre qui a comme avantage d'être isotopique, donc invariant avec l'orientation ou la rotation [19] pour produire les isothermes qui représente une variation de la température par rapport à la distance [$^{\circ}C/mm$].

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (5.8)$$

7. Un seuillage comme 5.3 sera nécessaire sur les résultats de 5.6 pour produire une carte thermographique avec une bonne séparation des niveaux de température.

6. Implémentation et mesures

Une implémentation finale n'est pas encore possible pour l'instant, car une WaferICTM est toujours dans la phase développement. Alternativement, nous allons valider le principe de fonctionnement de la méthode, faire l'implémentation et l'étude de fonctionnement des sous-systèmes tout en utilisant les ressources disponibles dans le laboratoire LIMA et à l'UQO. Donc, nous avons effectué l'implémentation et l'étude préliminaire d'un capteur à base de RO sur un FPGA, avec l'implémentation d'un algorithme de linéarisation sur Simulink. Nous avons utilisé une caméra

infrarouge pour prendre des mesures sur la surface d'une gaufre cobaye (WaferIC™ de développement) avec des sources de chaleurs externes collées sur cette dernière (un scénario qui mimique les données provenant des capteurs, pour tester l'algorithme).

6.1. Implémentation du montage de l'expérience sur l'algorithme de détection de la contrainte thermomécanique avec une caméra infrarouge

La figure 6.1 montre le montage expérimental que nous avons utilisé pour tester l'algorithme. Une caméra infrarouge de type VarioCAM® haute résolution Inspect 768 de la compagnie Jenoptik a été montée verticalement au-dessus du WaferIC™. Pour obtenir un bon rapport résolution/couverture d'image, la distance entre la caméra et le WaferIC™ a été ajustée d'une manière à ce que la fenêtre visible verticalement (l'axe des y) couvre toute la surface de ce dernier. Le WaferIC™ a été placé sur une plateforme à isolation thermique en polystyrène pour réduire la perte thermique avec la surface en dessous. Des résistances électriques avec des puissances différentes et un taux de dégagement (°C/seconde) de chaleur différent ont été placées dans des cavités entre la plateforme isolante et le WaferIC™ pour produire des sources de chaleur sur la surface de ce dernier et qui sont détectables par la caméra infrarouge. La caméra a été ajustée pour produire de la vidéo en échelle grise à 8 bits de résolution, et une bande de température de 30°C à 60°C à pleine échelle. Alors, une lecture de zéro ou couleur noire correspond à une température de 30°C ou moins (dans le cas d'une saturation inférieure), et une lecture de 255 ou couleur blanche correspond à une température de 60°C ou plus (dans le cas d'une saturation supérieure). Avec ces paramètres, la résolution de la lecture maximale sera de 0,117°C/niveau de gris.

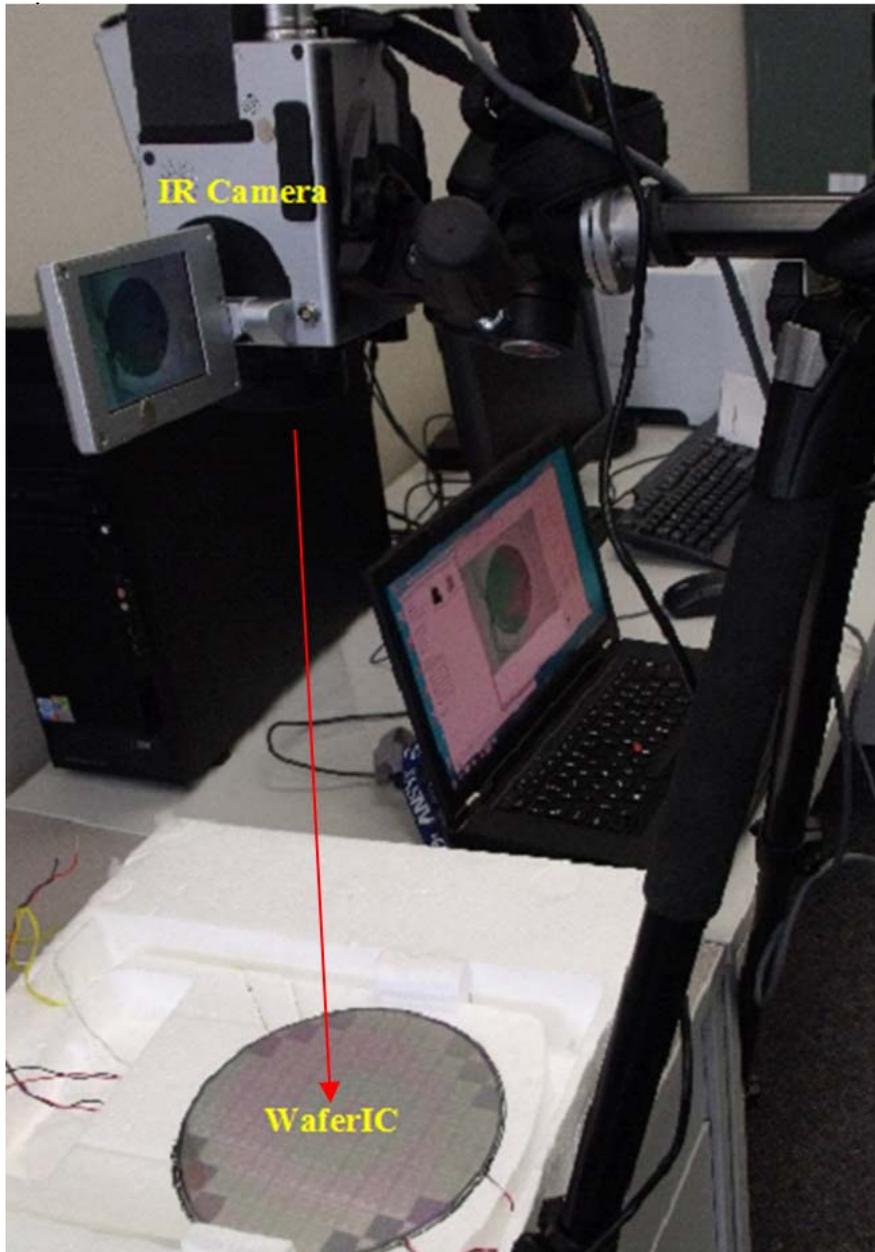


Fig. 6.1. Montage de l'expérience de la détection thermique avec une caméra infrarouge, un WaferIC et des sources de chaleur

6.2. Mesure et traitement de données avec l'algorithme de détection de la contrainte thermomécanique

Pour produire les résultats de cette expérience nous avons utilisé les données de trois trames de vidéo séparées par un dt de 45 secondes et deux sources de chaleur ont été activées (figure 6.2), et nous avons utilisé les logiciels Matlab et XnView pour l'application des algorithmes et la manipulation des images et de vidéo.

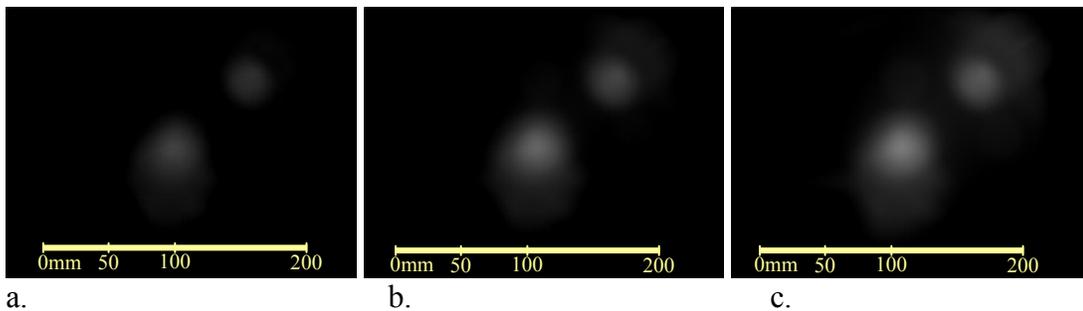


Fig. 6.2. De a. à c. 3 trames de vidéo qui montre la diffusion de la chaleur sur la surface du WaferIC, et qui est séparés par un dt de 45 secondes.

La caméra utilisée dans cette expérience produit de la vidéo avec une résolution de 640x480 pixels. Comme point de départ, il faut réduire cette résolution pour qu'elle soit en accord avec la résolution des vrais capteurs d'images thermiques qui seront implémentés sur la grandeur du WaferIC et qui sont estimés par une matrice de 279x279 capteurs. Donc une transformation géométrique de réduction d'échelle sur les images de la figure 6.2 a été utilisée [19]

$$(x, y) = T\{(v, w)\} \quad (6.1)$$

(v, w) sont les coordonnées de pixel de l'image originale, (x, y) sont les coordonnées de pixel de l'image transformée, et T est la matrice affine avec les éléments suivants,

$$T = \begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Et l'équation de réduction d'échelle sera :

$$(x, y) = (c_x v, c_y w) \quad (6.2)$$

Avec $c_x = (1,72043)^{-1}$ et $c_y = (1,72043)^{-1}$

ce qui donne une réduction d'échelle à 372x279 pixels et une résolution de distance en direction de x et de y de 0,732mm/pixel.

6.2.1. Détection des pics thermiques ou sources de chaleurs dangereuses

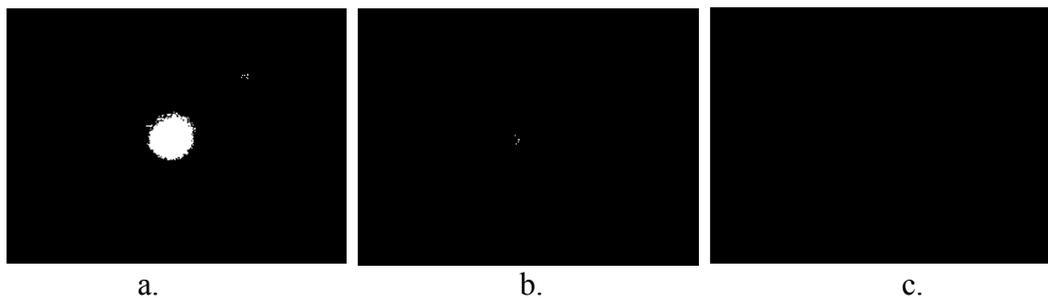
À partir de l'équation de seuillage (5.3), si nous choisissons $T_s = 120$ et avec une résolution de 0,117°C/niveau de gris, nous obtenons un seuillage d'amplitude à 44,04°C, ce qui permettra d'identifier les pics thermiques (ou sources de chaleur) qui dépassent la valeur de ce seuil. La figure 6.3 montre qu'après 135 secondes (tout en partant de la figure 6.3-a vers 6.3-c avec un dt de 45 secondes), un pic thermique qui dépasse le seuil prédéfini (44,04°C) a été détecté dans c.



a. b. c.
Fig. 6.3. Avec un dt de 45 secondes de a. à c. un seuillage à $44,04^{\circ}\text{C}$ détecte un pic thermique dans c.

6.2.2. Détection de la vitesse de changement de la température locale

Une détection de la vitesse d'augmentation de la température locale dans une région est applicable à l'aide de l'équation 5.4, et 5.5. 5.4 va donner la vitesse à chaque point, et 5.5 va produire une image binaire tout en comparant les valeurs obtenues dans 5.4 avec un seuil T_{s1} . La figure 6.4-a montre que dans un temps $t=0$, la température (marquée par le cercle blanc) croît à une vitesse qui dépasse la valeur de T_{s1} , ce qui prédit qu'après un certain temps la température peut attendre une valeur de pic thermique critique, ce qui a été vérifié dans les résultats de la figure 6.3-c.



a. b. c.
Fig. 6.4. a. montre une augmentation brusque de la température qui dépasse une seille T_{s1} , b et c montre que cette augmentation a calmée. a. est le résultant de la dérivée dans le temps de la figure 6.2-a et b. et ainsi de suite.

6.2.3. Production des cartes isothermes

Pour cette partie, nous allons effectuer des analyses plus détaillées que les sections précédentes, car plusieurs manipulations dépendantes sont nécessaires pour l'obtention des résultats acceptables. Le but est de produire des isothermes bien distinctes qui vont faciliter la détection des gradients de température sur la surface du WaferIC.

Le Laplacien de l'équation 5.8, en forme discrète en x et y donne [19],

$$c[\nabla^2 f(x, y)] = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

(6.3) Avec $c = -1$, on peut représenter 6.3 avec le masque h,

$$h = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (6.4)$$

Ce qui donne un résultat isotopique avec résolution de rotation de 90°.

Si nous appliquons ce masque directement (sans seuillage) sur l'image d'origine de la figure 6.5-a, nous obtenons un résultat semblable à la figure 6.5-b qui montre les zones là où le gradient de température est présent, mais comme nous le remarquons, on ne peut pas obtenir des données valables d'un tel résultat. Donc il faut introduire dans l'image des zones homogènes là où la dérivée de deuxième ordre (Laplacien) donne des zéros ou une pente horizontale. Par conséquent, cela se fait à l'aide d'un seuillage multiple de l'équation 5.6

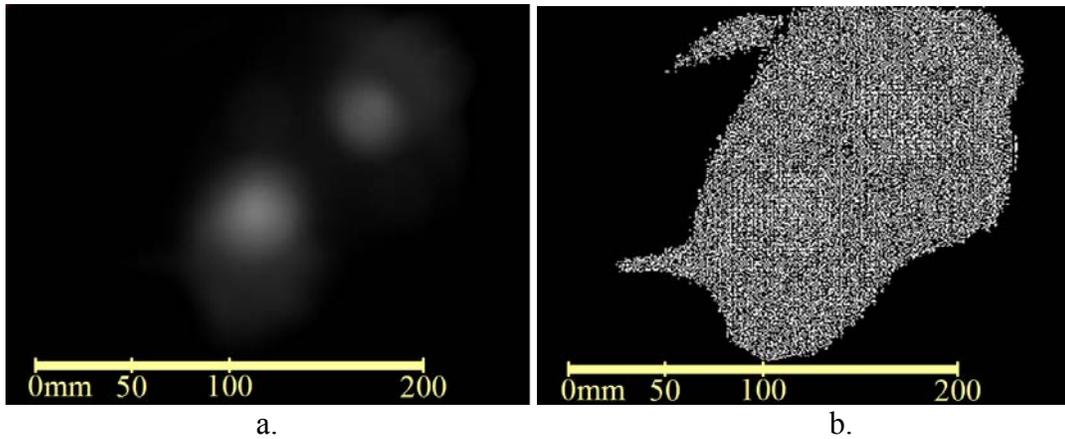


Fig. 6.5. a. Image originale, b. Application du Laplacien sur l'image en a.

Le choix de nombre de niveaux homogène est très délicat. Il faut faire un compromis entre une bonne résolution et un bon résultat des isothermes bien établi.

Maintenant, pour la même image originale (figure 6.5-a), nous allons effectuer un seuillage à 16 niveaux (figure 6.6-a) et appliquer le Laplacien (figure 6.6 b). En observant les résultats, nous trouvons que les isothermes sont bien établis, mais de bruits de haute fréquence est visible autour des courbes ce qui n'est pas souhaitable, car il peut affecter la lecture surtout entre deux isothermes voisines et qui sont très proche l'un de l'autre. Un filtre passe-bas est donc nécessaire pour réduire ce problème de bruit.

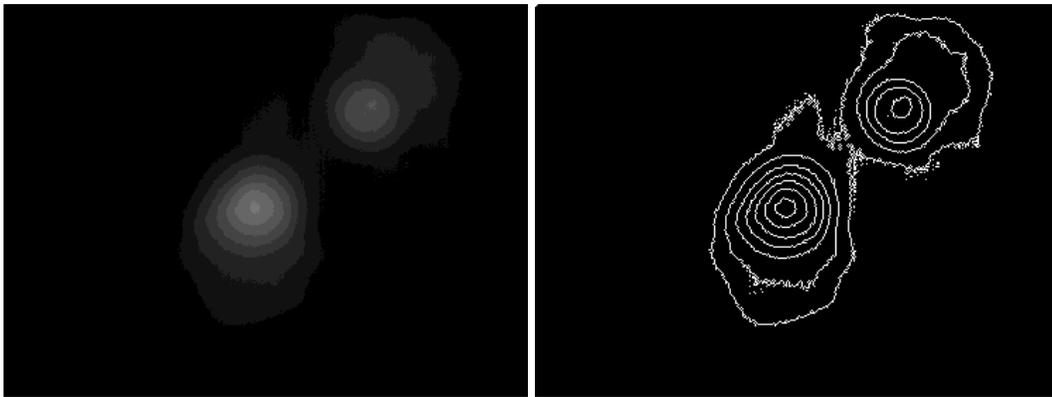


Fig. 6.6. a. Seuillage à 16 niveaux de gris, b. Application de Laplacien sur l'image du a.

Alors juste après la segmentation, nous avons appliqué un filtre médian qui est décrit dans l'équation 5.7, et qui a été appliqué avec un masque de 7×7 , ce qui a donné les résultats de la figure 6.7 qui donne un résultat avec des isothermes propres de bruit et des courbes bien connectées.

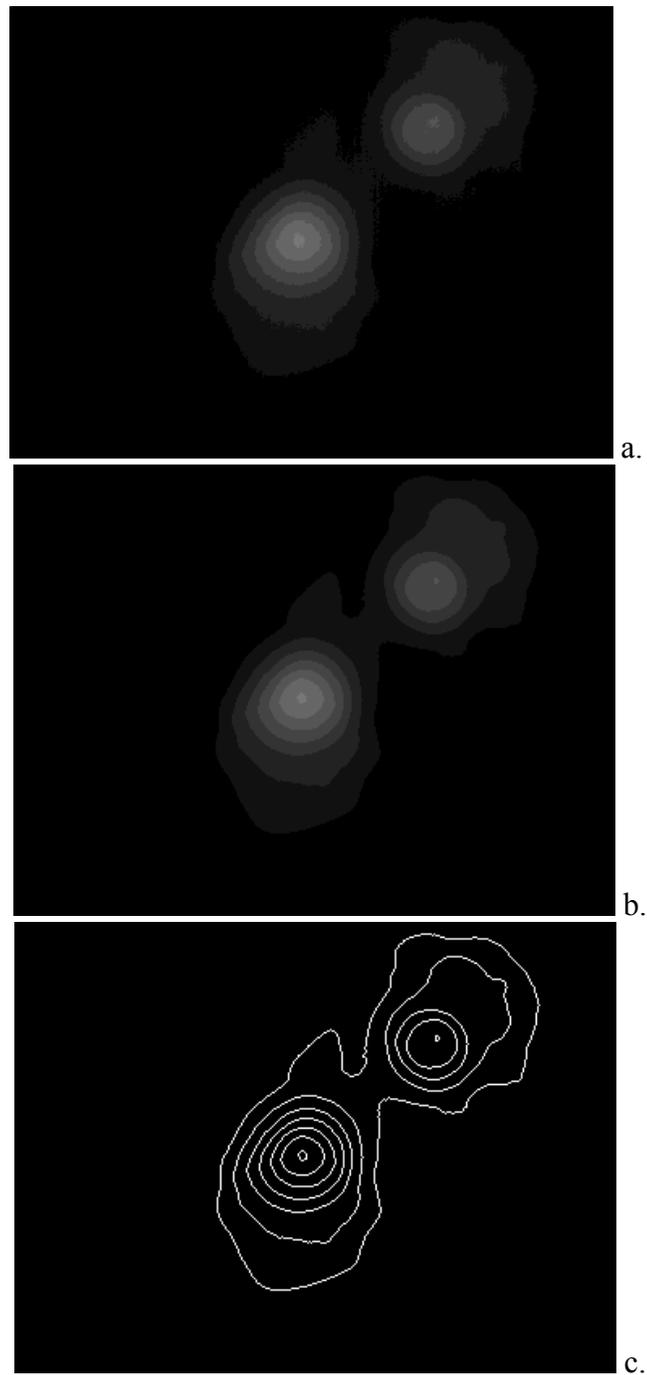


Fig. 6.7. a. Seuillage à 16 niveaux de gris, b. Application d'un filtre médian de 7x7, c. Laplacien sur l'image du b.

Maintenant, lorsque les résultats obtenus à la figure 6.7 sont satisfaisants, nous avons augmenté le nombre de niveau de gris graduellement pour définir la résolution maximale et nous avons obtenu 32 niveaux comme le montre la figure 6.8. Au-delà de 32 niveaux, les isothermes commencent à se coller ensemble. Donc 32 niveaux pour une variation de 30°C (30°C à 60°C), ce qui donne une résolution thermique de 0,9375°C par isotherme (ou niveau de gris)

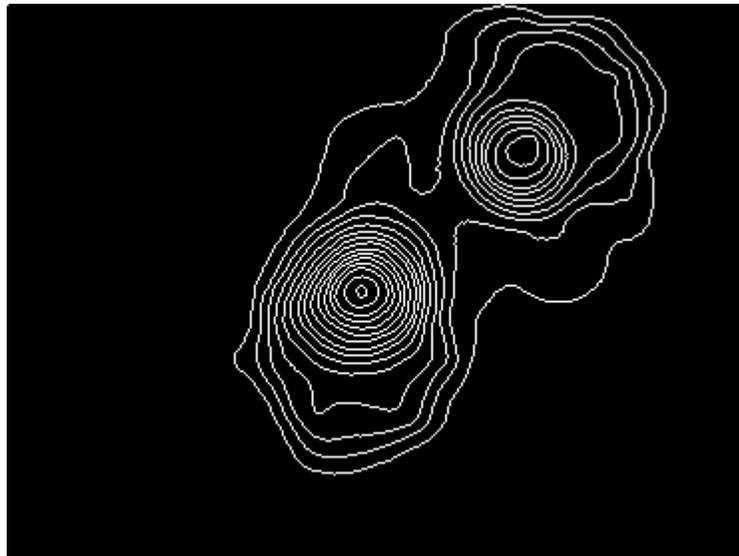


Fig. 6.8. Isotherme à résolution maximale, dont une résolution thermique de 0,9375°C/isotherme.

6.2.4. Détection de la contrainte associée au choc thermique

Dans la section précédente, nous avons défini la méthode de détection avec le gradient de température. Dans cette section nous allons appliquer cette méthode pour la détection du gradient de température sur les images de départs de la figure 6.2, donc les résultats sont montrés dans la figure 6.9.

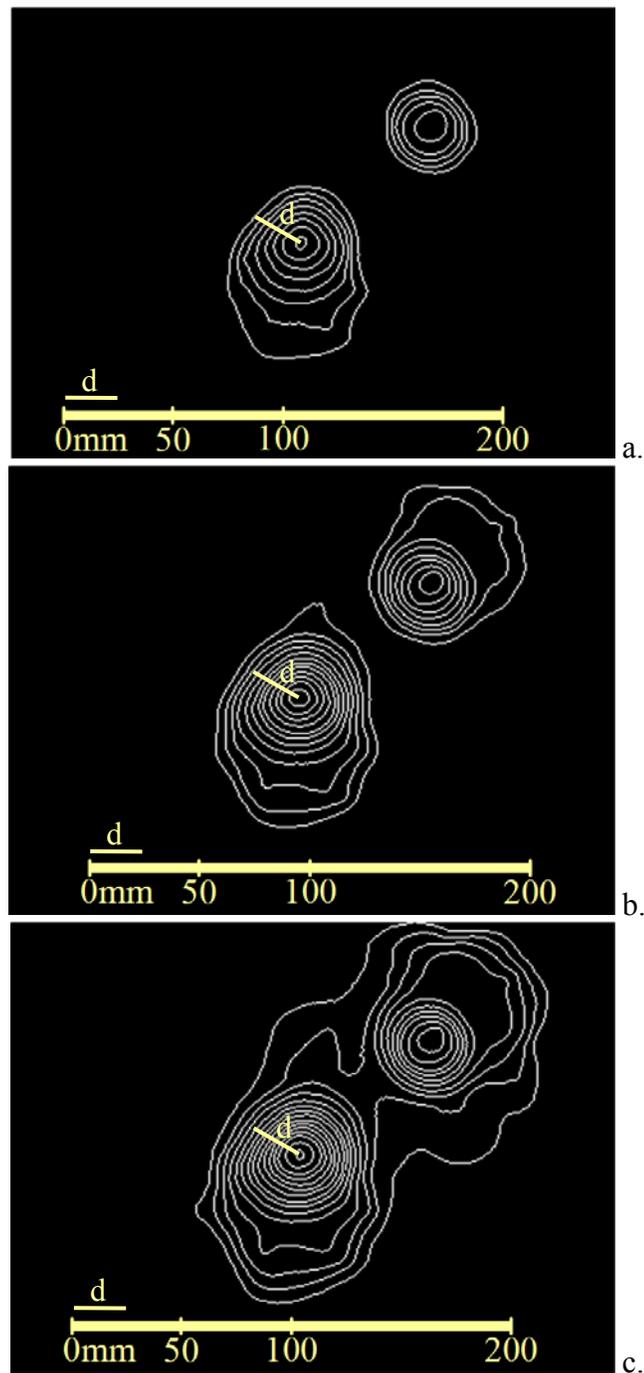


Fig. 6.9. Évolution de la température sur la surface du WaferIC. De a. à c. les images sont séparées d'un dt de 45 secondes et une résolution de $0,9375^{\circ}\text{C}/\text{isotherme}$.

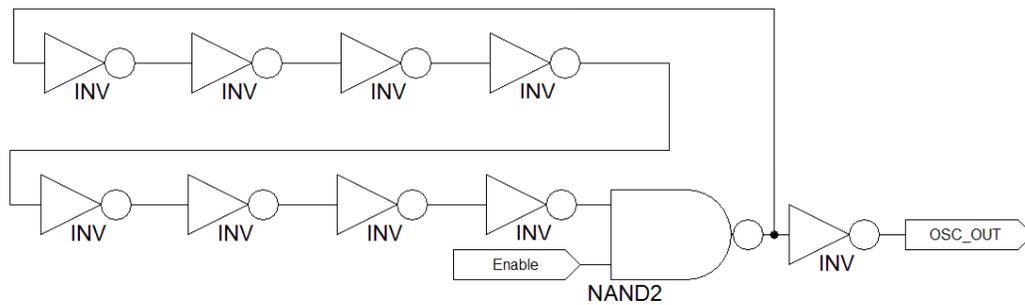
Une façon d'interpréter les données mesurées dans la figure 6.9 est de définir une constante de temps d . Dans la figure nous avons montré $d = 24\text{mm}$. Maintenant disons-nous que la limite maximale permise d'un gradient thermique est de $10^\circ\text{C}/d$, au-delà de cette valeur, un risque d'une contrainte associée au choc thermique dans les couches minces est grand. Tout en effectuant un balayage à partir du centre de chaque source en déplaçant vers l'extérieur du cercle avec des pas de 45° (donc 8 mesures autour de chaque source). Nous pouvons identifier (ligne jaune sur la figure) le (ou les) gradient maximal pour chaque source. Dans notre cas, nous montrons juste un seul pour chaque image pour la source la plus puissante. Pour la même direction dans les trois images de la figure 6.8. Dans a. nous avons 8 isothermes dans 24mm de distance, donc un gradient de $7,5^\circ\text{C}/24\text{mm}$. Dans b. nous avons 10 isothermes ce qui donne $9,4^\circ\text{C}/24\text{mm}$. Dans c nous avons mesuré 12 isothermes/ d ce qui donne un gradient de $11,25^\circ\text{C}/24\text{mm}$ ce qui dépasse le seuil, alors une contrainte associée au choc thermique est détectée.

La plupart du code Matlab de cette partie de traitement d'image est présenté en annexe C.

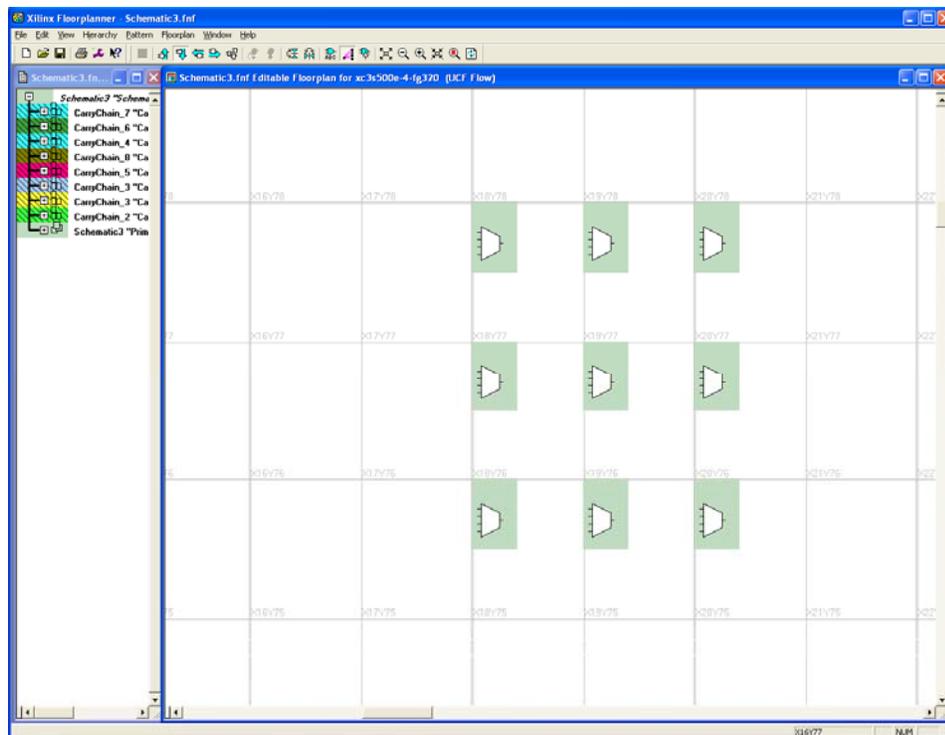
6.3. Implémentation et caractérisation des capteurs de température à base de RO sur un FPGA

Pour cette expérience, nous avons utilisé une carte FPGA Spartan-3E (XILINX xc3s500e-4fg320 FPGA), le logiciel ISE Project Navigator du Xilinx, et un capteur de température linéaire de type LM35DZ pour avoir une référence de température. Nous avons ensuite monté le capteur de température à base de RO de la

figure 6.10 qui comporte jusqu'à 9 inverseurs et un signal Enable pour activer/désactiver le capteur, et nous l'avons implémenté sur le FPGA.



a.



b.

Fig. 6.10. a. Capteur de température à 9 RO, b. Emplacement physique du capteur sur le FPGA.

6.3.1. Définition des paramètres du test

Avant de lancer le test sur le capteur, nous avons en premier considéré les limites de température de fonctionnement du FPGA. Le CI utilisé est le xc3s500e-4fg320 FPGA, et le tableau 6.I nous montre les données thermiques extraites de la fiche technique de ce dernier.

TABLEAU. 6.I

DONNÉES THERMIQUES DE LA FICHE TECHNIQUE DU FPGA XC3S500E-4FG320 FPGA

Résistances thermiques:	
Junction-à-Boîte:	$\theta_{JC} = \frac{9,8^{\circ}C}{Watt}$
Junction-à-Carte:	$\theta_{JB} = \frac{15,6^{\circ}C}{Watt}$
Junction-à-ambiante:	$\theta_{JA} = \frac{26,1^{\circ}C}{Watt}$ - 0 LFM (air immobile)
	$\theta_{JA} = \frac{20,6^{\circ}C}{Watt}$ - 250 LFM
	$\theta_{JA} = \frac{19,4^{\circ}C}{Watt}$ - 500 LFM
	$\theta_{JA} = \frac{18,6^{\circ}C}{Watt}$ - 750 LFM
Température de jonction maximale: $T_J = 125^{\circ}C$	
Température de jonction opérationnelle recommandée : $T_{JMIN} = 0^{\circ}C$ et	
$T_{JMAX} = 85^{\circ}C$	

Selon nos estimations de conception de pire cas, le RO ne va dissiper pas plus que quelques mW de puissance. Alors, en effectuant cette expérience dans un four avec une bande de température de 25°C à 65°C sans convection ne devrait pas endommager le FPGA.

6.3.2. Test préliminaire du capteur dans un four

Pour cette expérience, nous avons alimenté le capteur de température de référence (LM35) avec 12V, et nous l'avons collé directement à la surface du FPGA XILINX xc3s500e-4fg320, ce qui va nous donner une approximation de la température du capteur à base de RO qui est sur le FPGA.

Une première mesure nous a donné une lecture de 50MHz à 25°C, ensuite, avec un placement manuel avec l'outil 'Floor Planner' de l'ISE Project Navigator', nous avons changé l'emplacement du RO sur le FPGA horizontalement (sur x) et nous avons obtenu la même lecture (50MHz à 25°C). Ensuite, nous avons répété la même approche, mais cette fois-ci verticalement (sur y) et nous n'avons pas obtenu la même fréquence. Une variation de $\pm 10\%$ maximale pour la même température. Nous avons chauffé le FPGA dans un four de 25°C à 60°C, et nous avons obtenu un décrement de la fréquence d'oscillation autour 2,5% quel que soit le placement ou la fréquence de référence (à 25°C). Donc, nous avons conclu que cette variation initiale de la fréquence causée par le changement de placement est strictement liée à la structure physique interne du FPGA et pas à une variation de température sur la surface de ce dernier. Donc, il faut prendre ça en considération dans notre design dans la phase d'étalonnage du capteur.

Maintenant, nous avons ajouté le circuit de contrôle de la figure 6.11 au capteur. La figure 6.11 montre le diagramme bloc du circuit de contrôle, et le capteur de température de la figure 6.10 est marqué par le bloc RO. Dans ce circuit, nous avons utilisé un oscillateur à cristal de 50MHz comme une source de fréquence de référence. Un convertisseur de fréquence donne une sortie de 25Hz. Le décodeur 3 à 8 donne 8 phases d'opérations avec des périodes P égaux. Chaque période $P=1/25\text{Hz}=40\text{ms}$. À la phase six, nous activons le RO et le compteur de fréquence de 25 bits commence à compter à partir de zéro. Comme le montre la figure 6.11, le RO alimente l'entrée horloge de ce compteur donc la fréquence de compte est relative à la température. À la phase sept, nous désactivons le RO et le registre de 25 bits enregistre le dernier compte du compteur. À la phase 5 nous effectuons la remise à zéro du compteur de 25 bits et le cycle se recommence. Pour réduire l'effet d'auto réchauffement du RO [5], ce dernier sera activé pour une seule phase et désactivé pour 7 phases, donc 40ms ON et 280ms OFF.

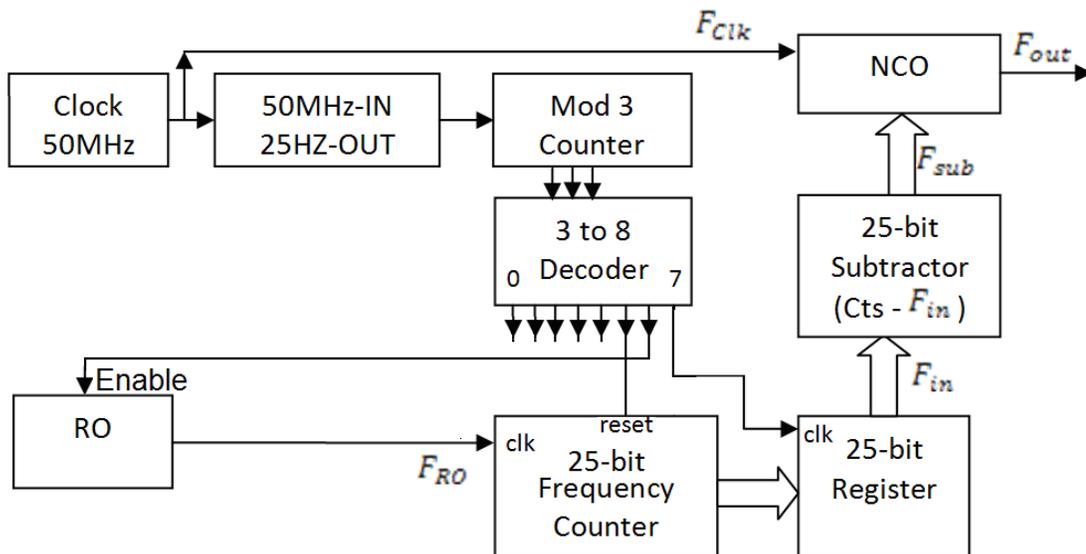


Fig. 6.11. Le capteur de température à 5 RO, avec le circuit de contrôle.

Le soustracteur à la sortie du registre de 25 bits est utilisé pour soustraire une grande partie de la fréquence de la porteuse (le 50MHz) qui est la fréquence la plus élevée qui représente la température la plus basse. Donc, la constante Cts représente une fréquence un peu plus grande de 50MHz donc F_{sub} et toujours positive et qui représente des températures à partir de 25°C et plus. Donc nous avons,

$$F_{in} = F_{RO} \times P \quad (6.5)$$

F_{in} est la fréquence d'entrée du soustracteur [Hz], F_{RO} est la fréquence de sortie du RO [Hz] et P est la période [Secondes].

$$F_{sub} = cts \times F_{in} \quad (6.6)$$

F_{sub} est le compte relatif à la température mesurée par le RO, cts est une constante qui représente la plus petite température ou la plus haute fréquence.

$$F_{out} = \frac{F_{Clk}}{2F_{sub}} [Hz] \quad (6.7)$$

F_{out} est la fréquence de sortie qui correspond à la température mesurée par le RO mais elle est à basse fréquence et facile à mesurer avec précision sur un simple oscilloscope.

6.3.2.1. Étude de la stabilité du circuit du contrôle par rapport à la variation de la température

Pour étudier la stabilité du circuit de contrôle par rapport à la variation de la température, nous avons remplacé le RO par un oscillateur à cristal de 50,000KHz

(figure 6.12). En utilisant une fréquence stable à l'entrée du circuit, et en variant la température, nous pouvons mesurer à la sortie F_{out} la contribution du circuit à la mesure.

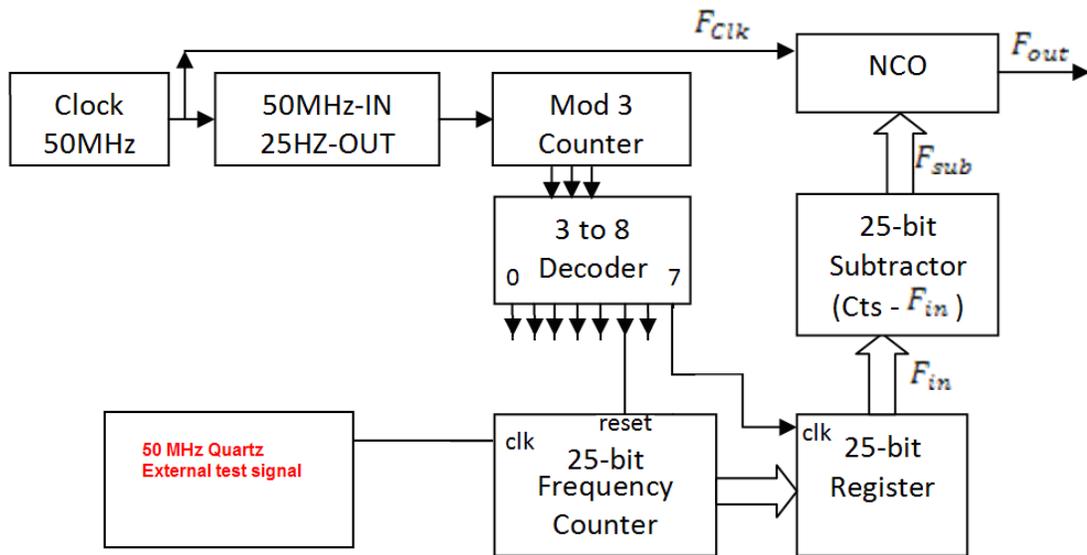


Fig. 6.12. Le capteur de température à 5 RO, avec le circuit de contrôle.

La figure 6.13 montre le montage de l'expérience. La carte FPGA est placée dans le four et un oscilloscope est branchée à la sortie F_{out} pour mesurer la fréquence.

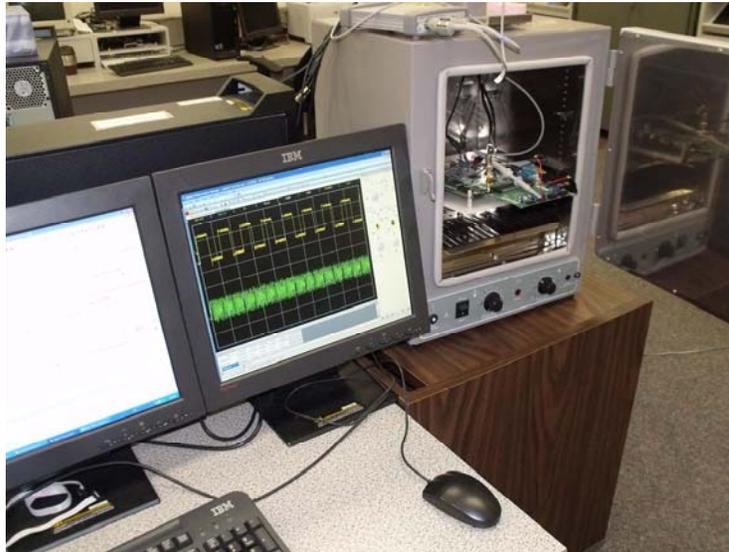


Fig. 6.13. Montage expérimental.

L'oscillateur de fréquence de 50MHz qui remplace le RO a été gardé à la température de la pièce (à l'extérieur du four), mais l'horloge de référence a été gardée à l'intérieur du four.

À 25°C nous avons mesuré $F_{\text{out}}=6000\text{Hz}$ (figure 6.14),

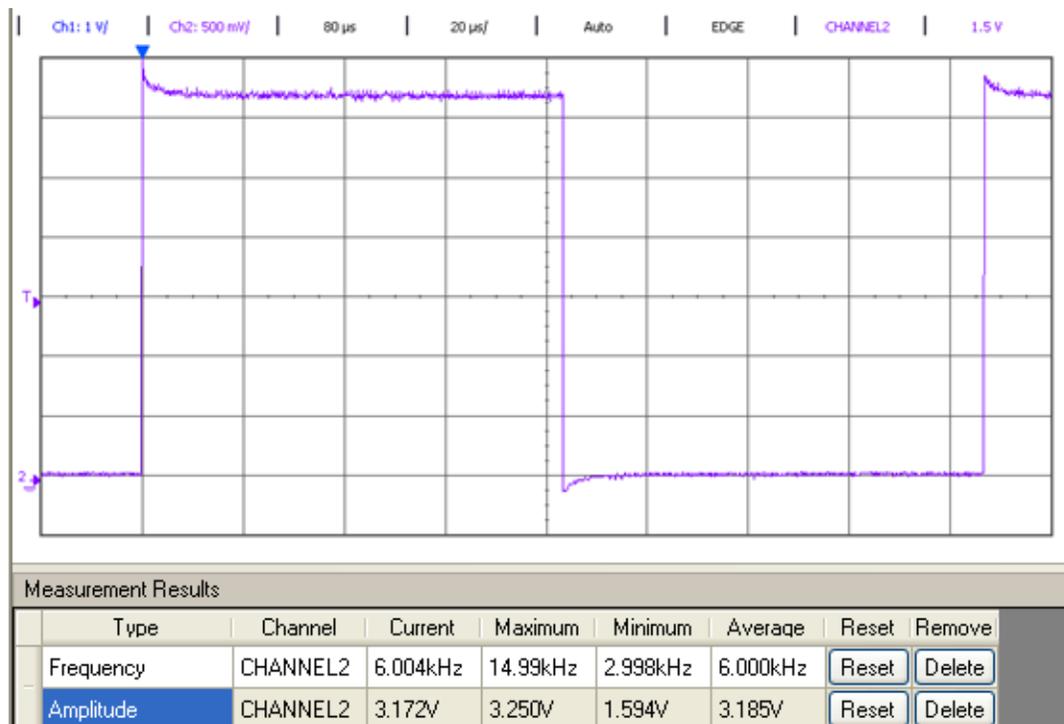


Fig. 6.14. Mesure de la contribution du circuit de contrôle sur la réponse du capteur; F_{out} à 25°C est égale à 6000Hz.

Par la suite nous avons chauffé le four à 60°C et nous avons mesuré $F_{out}=6010\text{Hz}$.

Ensuite, nous avons gardé les deux oscillateurs à l'extérieur du four, et nous avons répété l'expérience. Mais cette fois la fréquence de sortie est restée stable à 6000Hz, donc nous concluons que le circuit ne contribue pas à la mesure, car il est synchrone, mais plutôt cette petite variation sur la réponse dans le premier test est causée par l'oscillateur à cristal.

Dans notre cas cette contribution est négligeable, et n'affecte pas l'analyse de la contrainte. Mais pour d'autres applications où une très grande précision est

demandée, il faut considérer tous les paramètres qui peuvent affecter la réponse; comme la stabilité de la tension d'alimentation, la stabilité de la fréquence de référence et dans certains cas la pression externe contre le capteur qui peut encore affecter la lecture, car la réponse du RO est dépendante de deux variables, la pression et la température.

6.3.2.2. Mesure de la variation de fréquence du RO par rapport à une variation de température

Dans cette expérience, nous avons placé le FPGA qui contient le circuit de la figure 6.9 dans le four, mais pour plus de précision, nous avons gardé l'oscillateur de référence de 50MHz à la température ambiante et nous avons varié la température du four de 30°C à 62°C et nous avons mesuré la sortie F_{out} . À partir de l'équation 6.7 ($F_{sub} = \frac{F_{clk}}{2F_{out}} [Hz]$), nous avons calculé F_{sub} (tab. 6.II), car F_{out} est simple à mesurer sur un oscilloscope. Ensuite nous avons tracé la courbe de la fig. 6.15 qui représente la variation de la fréquence du capteur par rapport à la température.

TABLEAU. 6.II
MESURE DE LA FRÉQUENCE DE SORTIE F_{out} PAR RAPPORT À LA TEMPÉRATURE

F_{out} [HZ]	F_{sub} [Hz]	T[°C]
7954	3143	30
4650	5376	31
3300	7575	32
2580	9690	33
2170	11520	34
1880	13298	35
1570	15924	36
1460	17123	37
1300	19231	38
1170	21366	39

1091	22915	40
993	25176	41
923	27086	42
850	29412	43
774	32300	44
740	33784	45
696	35920	46
656	38110	47
625	40000	48
601	41597	49
566	44170	50
545	45872	51
515	48544	52
495	50505	53
477	52411	54
460	54348	55
440	56818	56
423	59102	57
405	61728	58
395	63291	59
382	65445	60
371	67385	61
359	69637	62

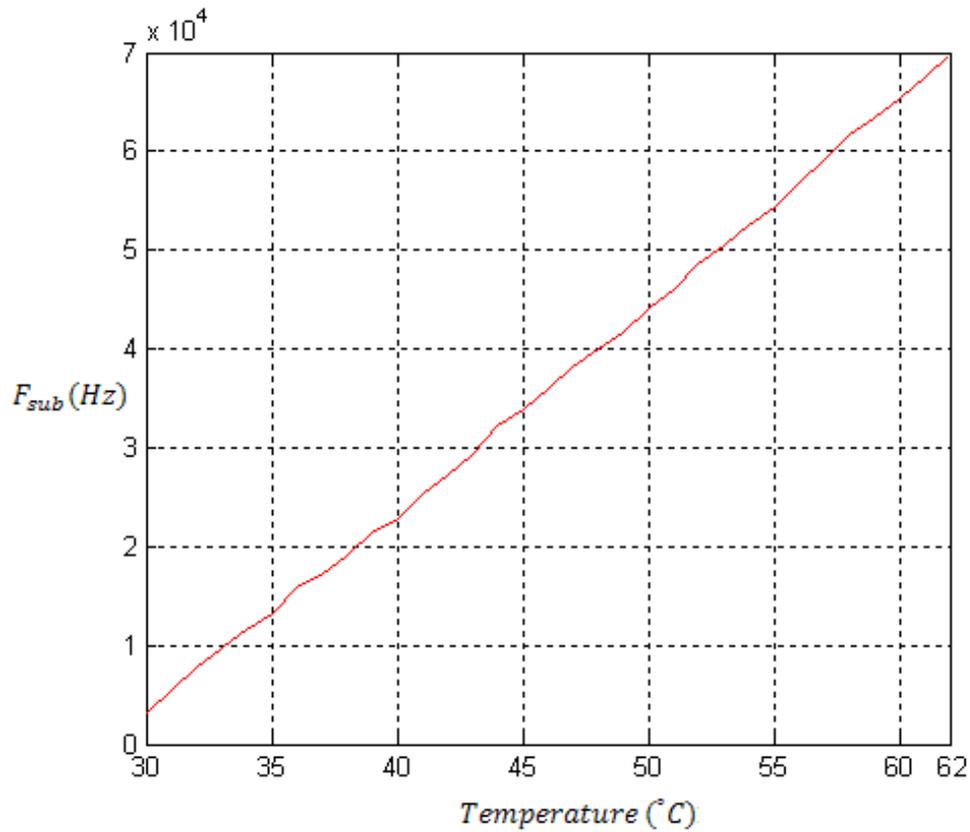


Fig. 6.15. Réponse de la fréquence du RO par rapport à la température.

La courbe de la fig. 6.15, montre que la réponse fréquentielle du RO monté sur un FPGA XILINX xc3s500e-4fg320 par rapport à la température, sur un intervalle de 30°C à 62°C est linéaire. Mais selon [6] et [7] ce n'est pas toujours le cas. [7] montre une linéarité partielle. Selon notre étude dans ce mémoire, plusieurs variables peuvent affecter la réponse donc affecter la linéarité de cette dernière. Ainsi, comme nous avons précisé dans 4.3.2.4, un bon algorithme de linéarisation pourrait être nécessaire pour une bonne précision. Nous traitons ce sujet dans la prochaine section.

6.4. Amélioration de la réponse du capteur et l'étude de la méthode de linéarisation

Le but principal de cette partie est d'étudier la méthode de linéarisation de J. S. Steinhart and S. R. Hart [13] sur la réponse que nous avons obtenue dans la section précédente. Mais avant, nous allons tout d'abord améliorer cette réponse afin d'obtenir une courbe idéale, ce qui va éliminer la possibilité que les résultats de linéarisations soient influencés par des erreurs de mesure ou par des bruits.

6.4.1. Amélioration de la réponse de Fsub

La figure 6.15 dans section précédente montre la réponse mesurée de Fsub. La réponse montre des fluctuations causées soit par du bruit ou par des erreurs de mesures. Mais dans les deux cas nous remarquons qu'ils s'agissent de bruit de hautes fréquences. Ainsi, nous avons appliqué un filtre passe-bas (la moyenne) pour aligner ou redistribuer les fréquences avec des pas égaux sur tous les intervalles de températures.

$$F_{sub_mean} = \frac{\sum_{i=2}^{i \leq n+1} F_{sub}(i) - F_{sub}(i-1)}{n} [Hz] \quad (6.8)$$

Et nous avons appliqué la réponse de 6.8 sur Fsub :

$$\begin{cases} F_{sub_balanced}(1) = F_{sub}(1) \\ F_{sub_balanced}(i) = \sum_{i=2}^{i \leq n} F_{sub_balanced}(i-1) + F_{sub_mean} \end{cases} \quad (6.9)$$

La figure 6.16 compare la réponse avant et après l'opération. Nous remarquons que la réponse en rouge ($F_{sub_balanced}$) est une ligne parfaitement droite et bien alignée avec F_{sub} .

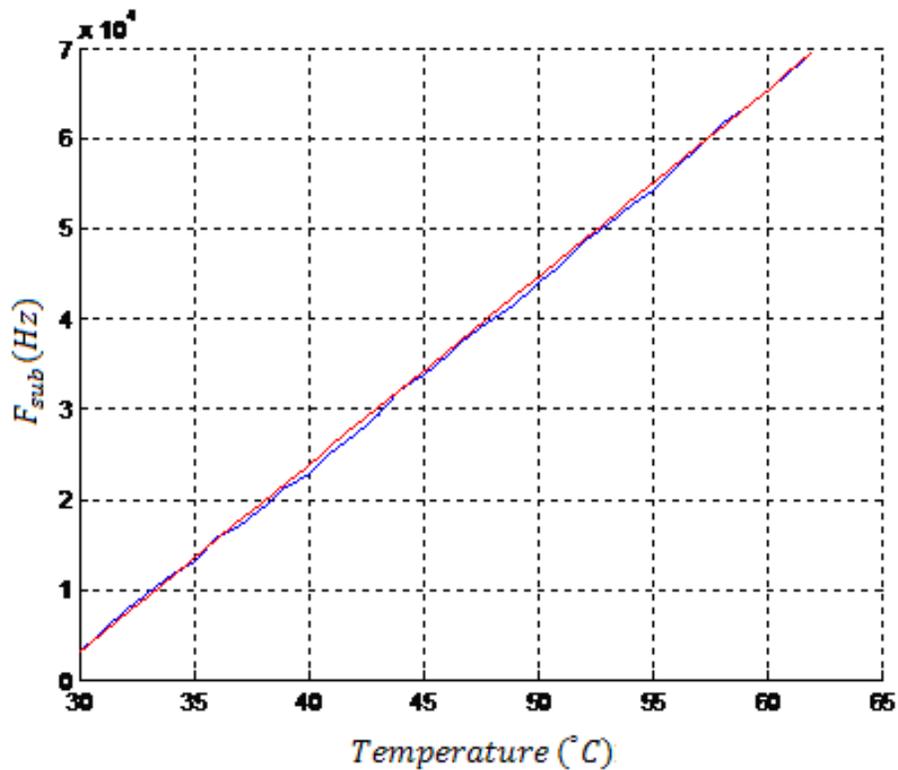


Fig. 6.16. Comparaison de la réponse de la fréquence du capteur (F_{sub}) par rapport à la température, avant (bleu) et après (rouge) l'amélioration.

Maintenant nous avons effectué un calcul inverse pour améliorer F_{out} à partir de ces nouvelles données de F_{sub} avec l'équation de 6.7 ($F_{out} = \frac{F_{clk}}{2F_{sub}} [Hz]$), et les résultats sont comparés à la figure 6.17.

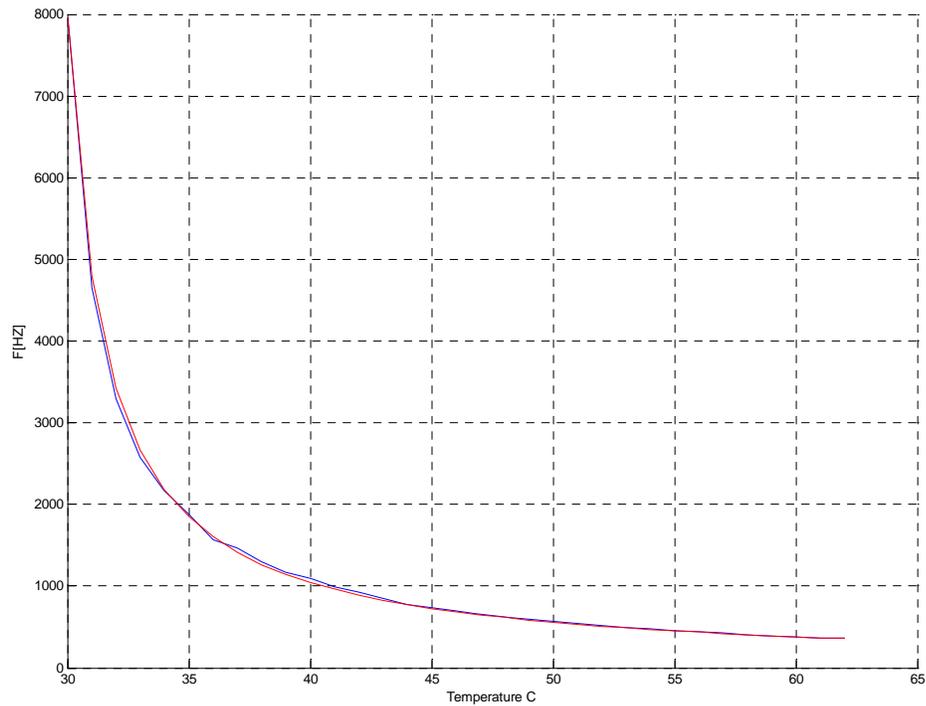


Fig. 6.17. Comparaison de F_{out} par rapport à la température, avant (bleu) et après (rouge) l'amélioration.

6.4.2. Étude de la méthode de linéarisation sur la réponse de F_{out} corrigée

À partir de la valeur F_{out} que nous l'avons amélioré dans la section précédente, nous avons appliqué l'algorithme de linéarisation

($Y = \frac{1}{T_k} = C1 + C2 \ln(R) + C3 \ln(R)^3$ (4.20)) de J. S. Steinhart and S. R. Hart [13].

Cependant, pour une variation de fréquence au lieu de résistance, nous allons modifier cette équation tout en remplaçant la résistance R par la fréquence F,

$$Y = \frac{1}{T_k} = (a + b \ln(F) + c \ln(F)^3) \quad (6.10)$$

Pour trouver les coefficients a, b et c, nous avons besoin de déterminer 3 valeurs de fréquence et leurs correspondances en températures.

Si par exemple nous prenons les résultats du tableau 6.I, nous remarquons que les valeurs F_{out} ne sont pas linéaires puisque $F_{out} = \frac{F_{Clk}}{2F_{sub}}$ (6.7) n'est pas linéaire.

Donc pour l'intervalle de température de 30°C à 62°C (Tab. 6.I) nous avons besoin de trois échantillons : un à la température la plus basse, un au centre et un à la température la plus élevée.

Donc à :

$T_l = 30^\circ\text{C}$, $F_l = 7954\text{Hz}$; $T_m = 46^\circ\text{C}$, $F_m = 696\text{Hz}$; $T_h = 62^\circ\text{C}$, $F_h = 359\text{Hz}$

Alors,

$$\frac{1}{303.15} = (a + b \ln(7954) + c \ln(7954)^3)$$

$$\frac{1}{308.15} = (a + b \ln(1890) + c \ln(1890)^3)$$

$$\frac{1}{335.15} = (a + b \ln(359) + c \ln(359)^3) .$$

Pour ce système de trois équations à trois inconnus pour a, b et c (voir code Matlab en Annexe C), nous trouvons, $a=0.000513906$, $b=0.000502285$ et $c=-0.000002383$. Enfin, nous obtenons le résultat de la figure 6.18.

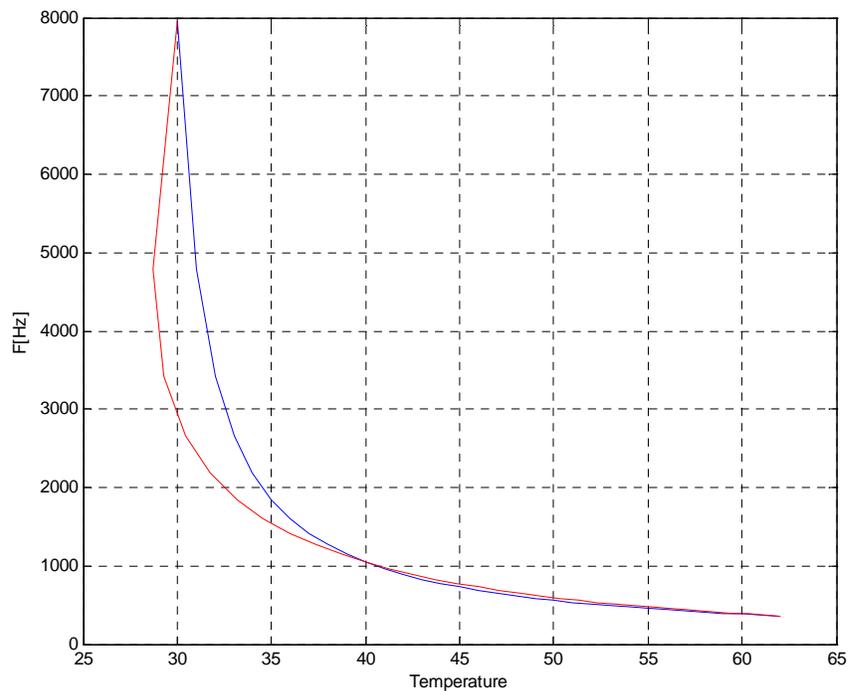


Fig. 6.18. La courbe rouge est la sortie de l'algorithme de linéarisation, en bleu est la sortie Fout améliorée.

D'après la courbe obtenue (en rouge) la réponse n'a pas suivi la courbe originale dans la zone des basses températures ($<40^{\circ}\text{C}$). Tout en investiguant la raison, nous avons remarqué que le choix de la constante de soustraction c_{ts} (6.6) aide la réponse à converger très vite vers la fréquence de 50MHz, donc perte de résolution et perte de signal à partir de 28°C et plus bas, ce qui implique que la pente de la courbe de F_{out} pour les basses fréquences est très grande ce qui semble pas assez de résolution avec juste trois coefficients. Pour vérifier cette théorie, et lorsque l'intervalle des fréquences/Température est connu pour F_{sub} , nous avons éliminé la bande de fréquence de 30 à 39°C est nous avons modifié l'intervalle de mesure pour couvrir la zone de 40°C à 72°C ce qui va nous éloigner de 12°C de la limite de fonctionnement. Par la suite, nous avons appliqué l'algorithme de nouveau pour obtenir la réponse de la figure 6.19 suivante. Maintenant, nous constatons que la réponse est beaucoup plus satisfaisante, puisque la réponse de l'algorithme (en rouge) a bien suivi la courbe originale (en bleu) ce qui valide notre théorie expliquée un peu plus haut.

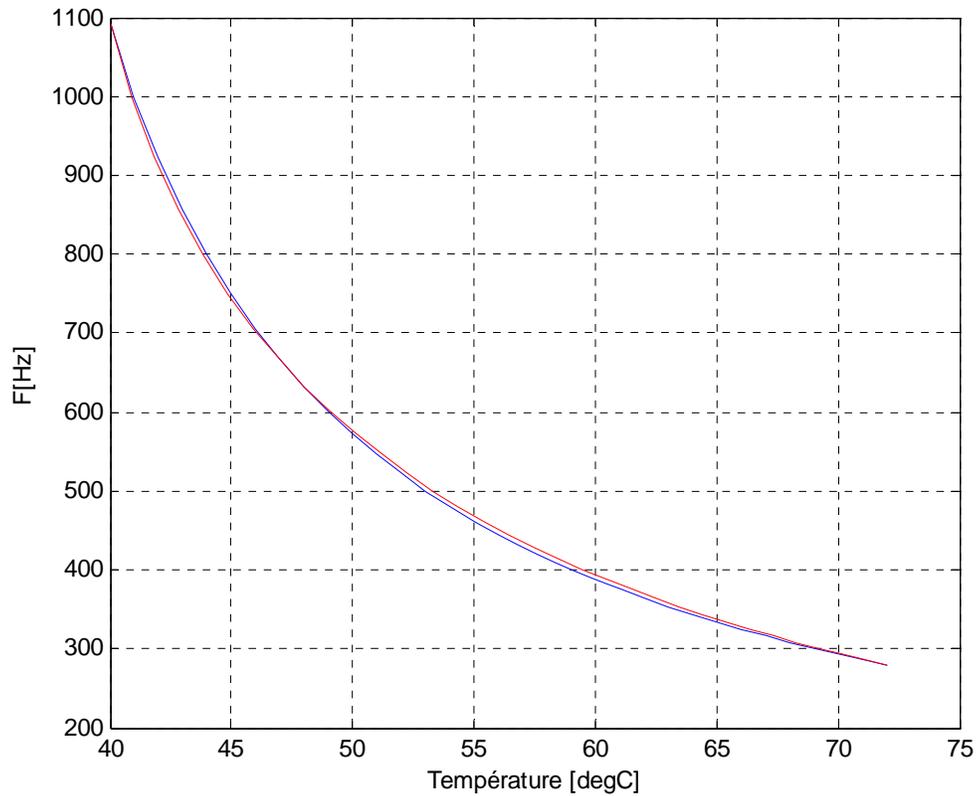


Fig. 6.19. La courbe rouge est la sortie de l'algorithme de linéarisation, en bleu est la sortie Fout améliorée.

Mais pour améliorer encore la résolution nous pouvons augmenter le nombre de coefficients par exemple à quatre ce qui donne,

$$Y = \frac{1}{T_k} = a + b \cdot \ln(F) + c \cdot \ln(F)^2 + d \cdot \ln(F)^3 \quad (6.11)$$

Cependant, cela va compliquer un peu le calcul, puisque nous devons maintenant résoudre quatre équations pour quatre inconnus.

7. Conclusion

Nous avons présenté dans ce mémoire le projet du DreamWafer™. Nous avons montré la problématique à résoudre spécifiquement pour ce projet et en général au problème de contrainte thermomécanique dans les LAIC. Nous avons montré les résultats des recherches que nous avons effectuées qui couvrent tous les aspects de la problématique comme la théorie, la diffusion de la température et la contrainte dans les couches minces. Nous avons exposé plusieurs travaux sur l'implémentation et la caractérisation des RO dans le but de mesurer la température et la contrainte dans les circuits CMOS. De plus, nous avons montré plusieurs algorithmes qui pourraient être adaptés pour la résolution de notre problème.

Nous avons également montré la méthodologie et les techniques que nous avons adaptées pour développer une méthode efficace qui traite la problématique des pics thermiques et du gradient de température sur la surface d'un LAIC et qui sont les principales causes de la contrainte thermomécanique dans ces derniers. Ainsi, comme un WaferIC™ n'a pas encore été produit pour l'implémentation du capteur, nous avons adopté comme alternative des implémentations préliminaires des capteurs sur un FPGA, pour valider le fonctionnement et effectuer les étalonnages nécessaires pour notre application. Nous avons étudié et adapté un algorithme de linéarisation pour cette application, nous avons exposé la problématique liée à l'utilisation d'un tel algorithme et nous avons validé les solutions proposées.

Nous avons utilisé un montage d'une caméra infrarouge et une gaufre cobaye (WaferIC de développement) avec des sources de chaleurs externes collées sur cette dernière pour tester un algorithme de détection. Les mesures effectuées sur la surface du WaferIC montrent que la propagation de la température dans un temps transitoire de d'environ 160 secondes, autour d'une source de chaleur reste relativement locale,

ce qui montre que l'effet du gradient thermique peut être significatif sur les LAIC et doit être surveillé en tout temps pour assurer qu'elle reste dans les intervalles permis. Toutefois, les résultats de cette étude ne sont que des analyses préliminaires. Contrairement, à des capteurs intégrés, l'utilisation d'une caméra infrarouge dans cette expérience exige que la surface du WaferIC reste exposée. À cette étape, on ne sait pas précisément c'est quoi la contribution de la perte de chaleur par convection entre la surface exposée de la carte et l'air ambiant sur les résultats des mesures. Ce qui est vraiment important de savoir c'est l'effet de ce gradient sur les couches minces sur la surface de la carte. Des capteurs implémentés directement dans ces couches minces donnent des résultats plus précis et plus fiables qu'une caméra infrarouge, car l'environnement de mesure est plus contrôlable. Enfin, une étude plus détaillée et plus approfondie est nécessaire pour bien couvrir ce problème intéressant dans le but de concevoir un système embarqué de capteurs distribués permettant de faire le monitoring thermomécanique dans les circuits LAIC. C'est justement le sujet de notre thèse de doctorat entamée à l'hiver 2013.

**Annexe A Conception du circuit du capteur et mesure sur une
carte spartan-E3 de Xilinx**



**Final System Hardware Desing for Thermal and
Physical Stress Monitoring System to be
Implemented on theTest Chip 5.**

By

Michel Saydé

Ahmed Lakhssassi

Report No 63

UQO-POLY-UQÀM-TechnoCap

three_inv.vhd

Note : C'est le même circuit de mémoire mais avec 49 inverseur au lieu de 17, et flag pour counter overflow et MUX selection

```

-- file: three_inv.vhd
-- Three serial inverters
-- by: Michel Saydé
-- 6 April 2012
-- to be implemented under each nanopad of each cell of the
-- wafer-IC, except the middle of the cell

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY three_inv IS
    PORT (
        inv_in: IN STD_LOGIC;
        inv_out: OUT STD_LOGIC);
END three_inv;

ARCHITECTURE intern OF three_inv IS
    SIGNAL a: STD_LOGIC_VECTOR(1 DOWNTO 0); -- for interconnection

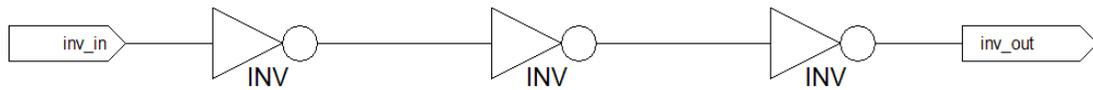
    -- Do not optimize the design
    ATTRIBUTE KEEP: BOOLEAN;
    ATTRIBUTE KEEP OF a: SIGNAL IS TRUE;

BEGIN

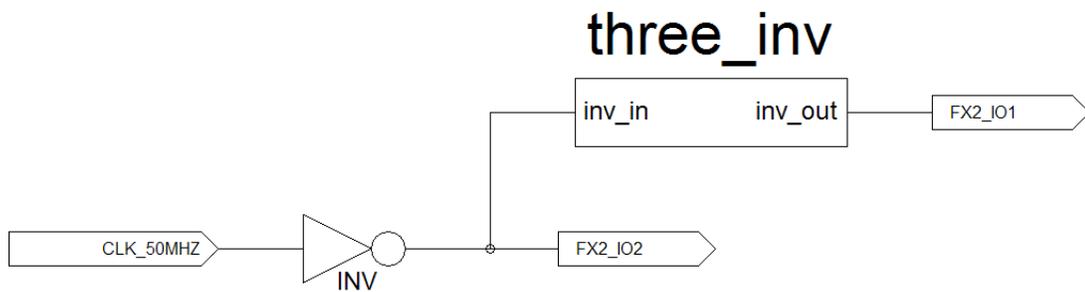
    a(0)    <= NOT inv_in;
    a(1)    <= NOT a(0);
    inv_out <= a(1);

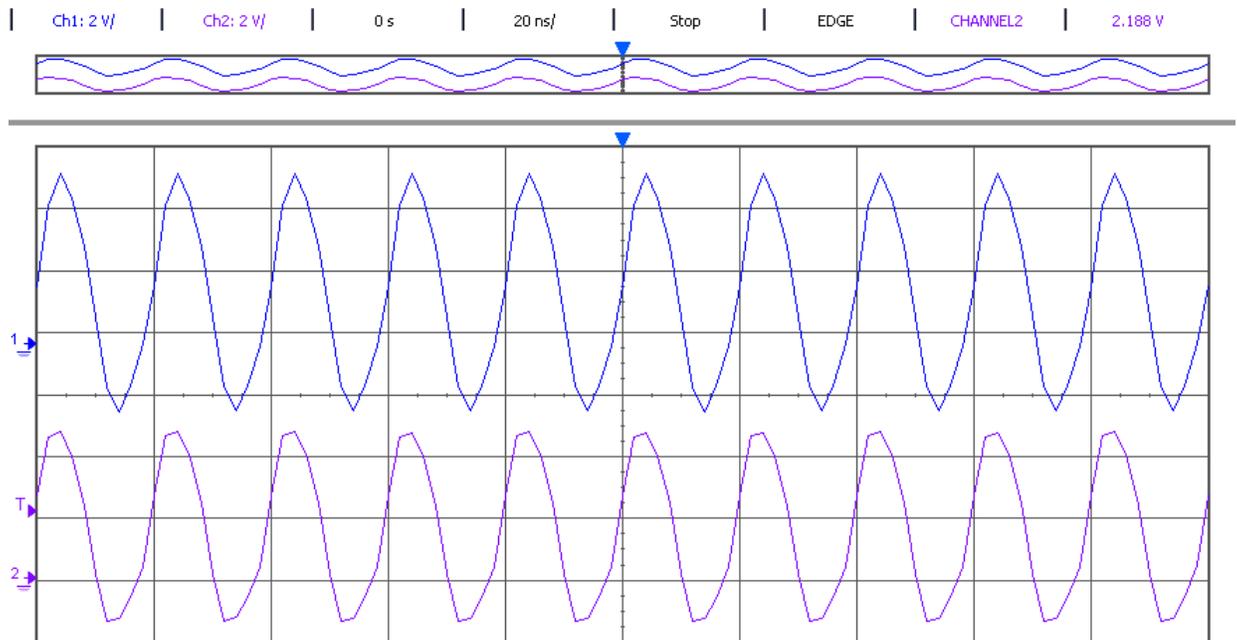
END intern;

```

Function of the circuit:

three_inv

**Test circuit physically implemented on a Spartan-3E FPGA:****Oscilloscope measurement:**



Blue (1) = inv_out

Violet (2) = inv_in

and_inv.vhd

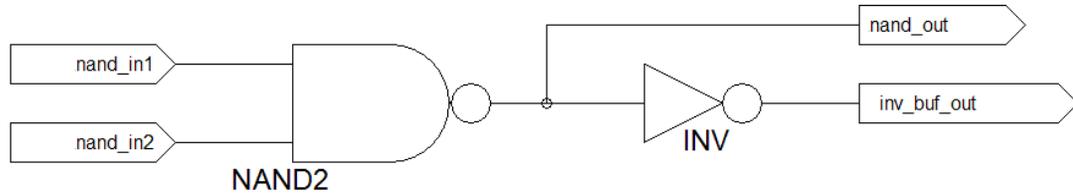
```
-- file: nand_inv.vhd
-- NAND gate serial with NOT gate
-- by: Michel Saydé
-- 6 April 2012
-- to be implemented at the center of each cell of the
-- wafer-IC, provides enable and buffer output for each
-- cell ring oscillator.

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

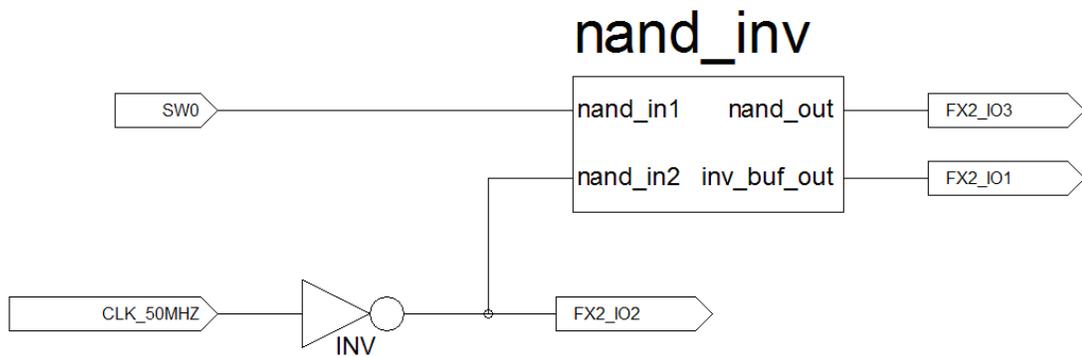
ENTITY nand_inv IS
  PORT (
```

```
        nand_in1, nand_in2: IN STD_LOGIC;  
        nand_out, inv_buf_out: OUT STD_LOGIC);  
END nand_inv;  
  
ARCHITECTURE internal OF nand_inv IS  
    SIGNAL inter: STD_LOGIC; -- for interconnection  
BEGIN  
  
    inter <= nand_in1 NAND and_in2;  
    nand_out <= inter;  
    inv_buf_out <= NOT inter;  
  
END internal;
```

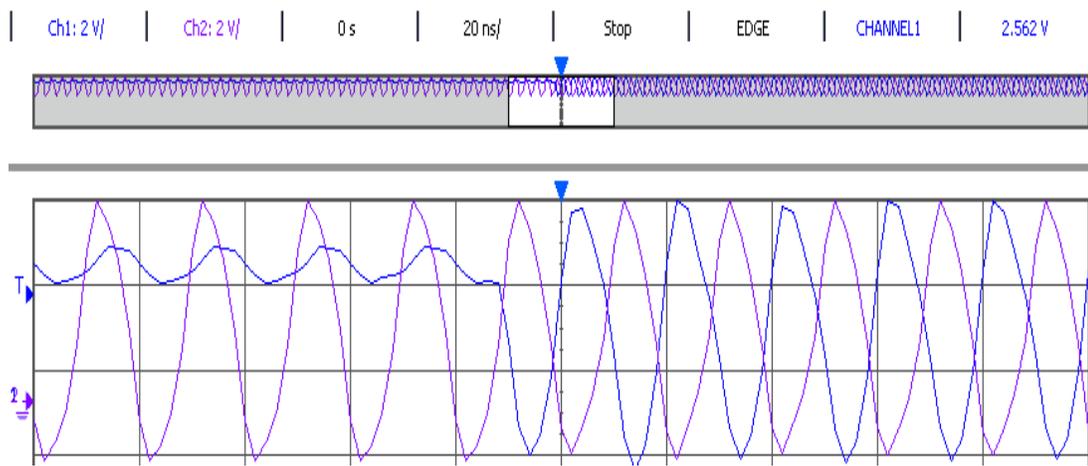
Function of the circuit:



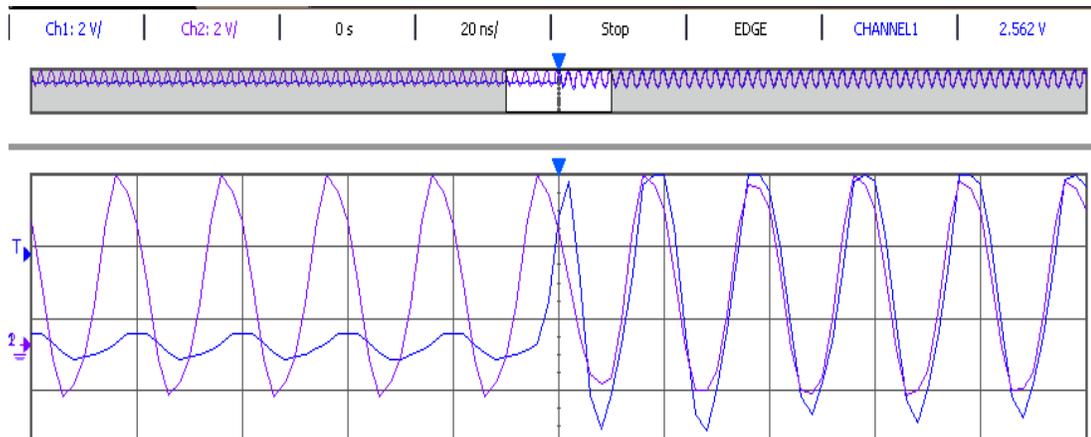
Test circuit physically implemented on a Spartan-3E FPGA:



nand_in2 nand inv_buf_out with nand_in1 switches from '0' to '1':



nand_in2 nand nand_out with nand_in 1 switches from '0' to '1':



sixteen_bit_count.vhd

```
-- file: sixteen_bit_count.vhd
-- by: Michel Saydé
-- 6 April 2012
-- 16-bit up counter with asynchronous reset, sync. rising edge
-- count and overflow flag output. If maximum count is reached
-- the count will stop and the overflow flag will flip to '1' till
-- reset ('0') is applied.
-- Each cell of the wafer_IC will have one counter to be used to
-- count the RO sensor and the VCO frequencies.

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY sixteen_bit_count IS
    PORT (
        rst, clk: IN STD_LOGIC;
        o_flow: OUT STD_LOGIC;
        q: OUT INTEGER RANGE 0 TO 65535);
END sixteen_bit_count;

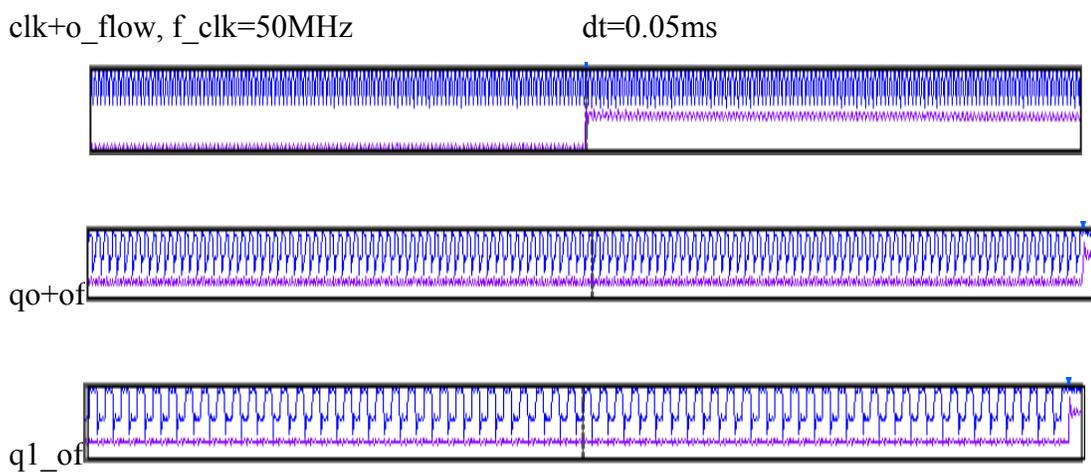
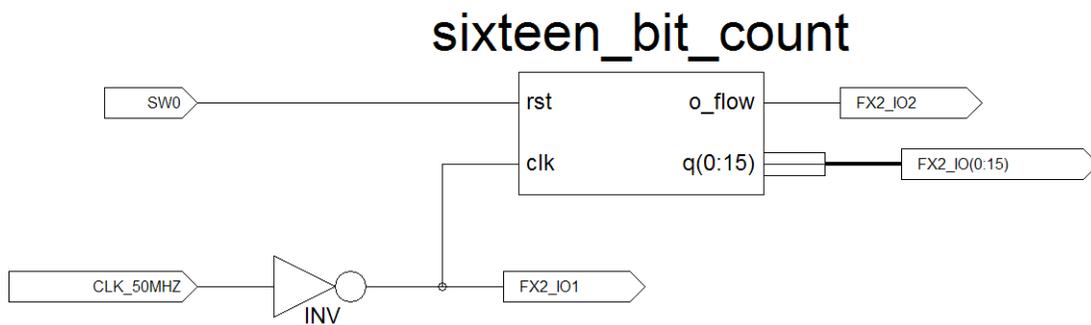
ARCHITECTURE sixteen_bit_count_internal OF sixteen_bit_count IS
BEGIN
    PROCESS(rst, clk)
        VARIABLE inter_q : INTEGER RANGE 0 TO 65535;
    BEGIN
        IF(rst='0') THEN
            inter_q := 0;
```

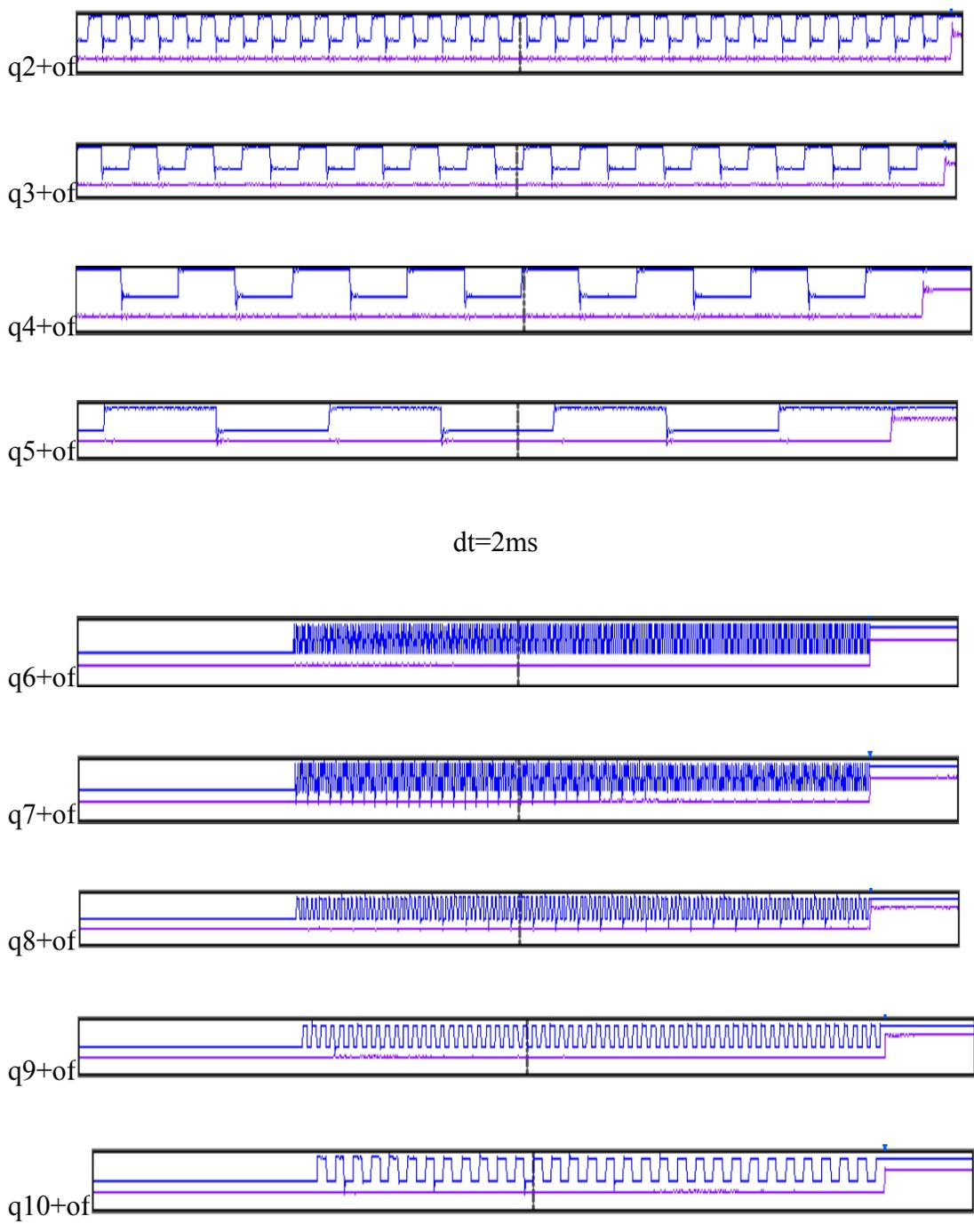
```

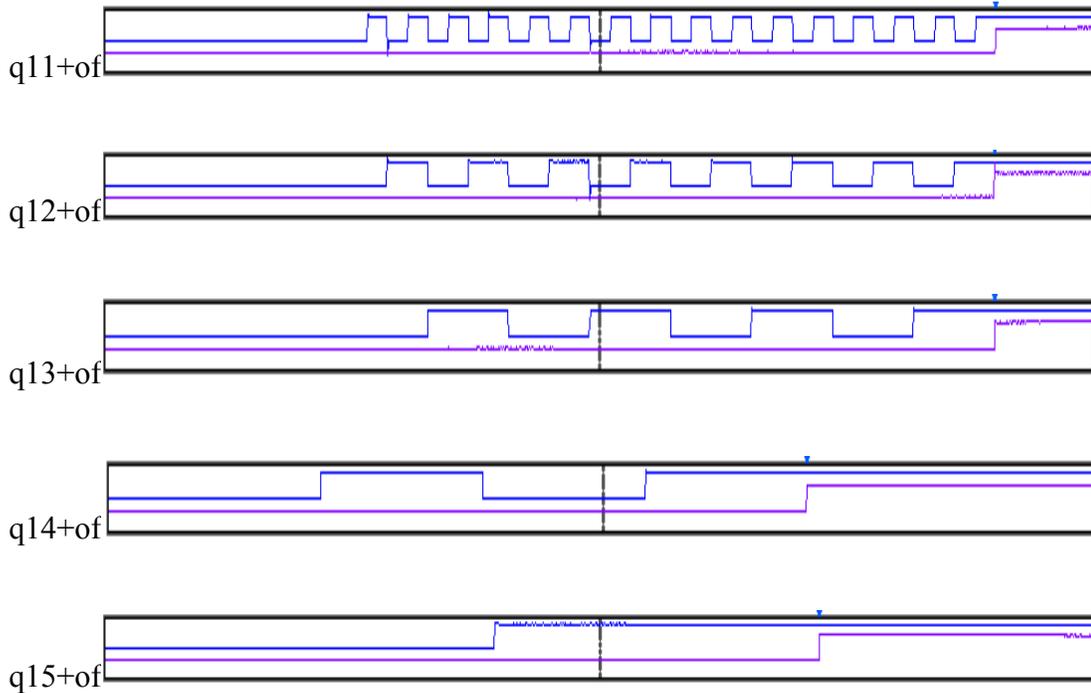
        o_flow <= '0';
    ELSIF(clk'event AND clk='1') THEN
        IF(inter_q = 65535) THEN
            o_flow <= '1';
        ELSE
            inter_q := inter_q + 1;
        END IF;
    END IF;
    q <= inter_q;
END PROCESS;
END sixteen_bit_count_internal;

```

Test circuit physically implemented on a Spartan-3E FPGA:







q15+of when reset is applied (1 to 0):



one_bit_count.vhd

```
-- file: one_bit_count.vhd
-- by: Michel Saydé
-- 6 April 2012
-- 1-bit up counter with asynchronous reset and sync. rising edge
-- count. reset is active low.
-- Each cell of the wafer_IC will have two of this counter, one to
-- be used to turn on/off the RO sensor unit, and the other to
```

```

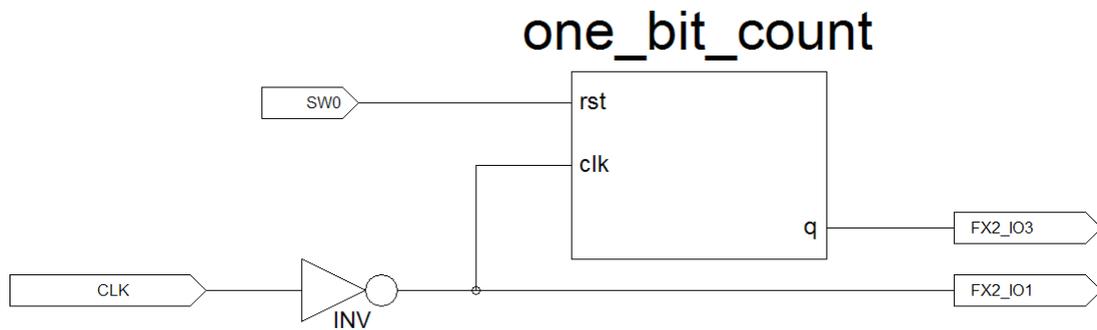
--      select between RO sensor unit and the VCO.

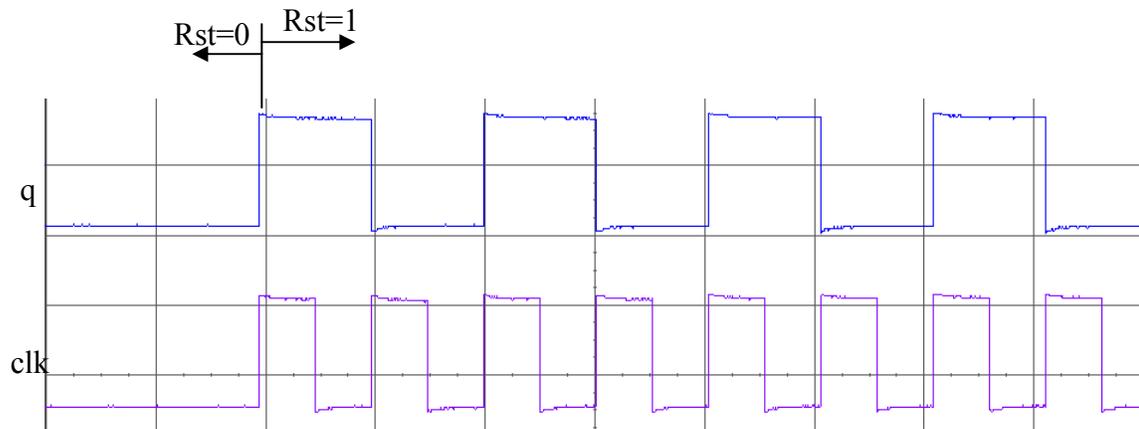
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY one_bit_count IS
    PORT (
        rst, clk: IN STD_LOGIC;
        q: OUT STD_LOGIC);
END one_bit_count;

ARCHITECTURE one_bit_count_internal OF one_bit_count IS
BEGIN
    PROCESS(rst, clk)
        variable inter_q : STD_LOGIC;
    BEGIN
        IF(rst='0') THEN
            inter_q := '0';
        ELSIF(clk'event and clk='1') THEN
            inter_q := not inter_q;
        END IF;
        q <= inter_q;
    END PROCESS;
END one_bit_count_internal;

```

Test circuit physically implemented on a Spartan-3E FPGA:





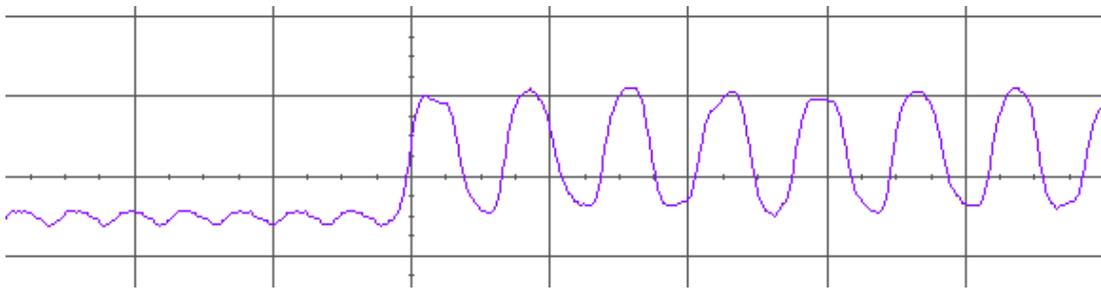
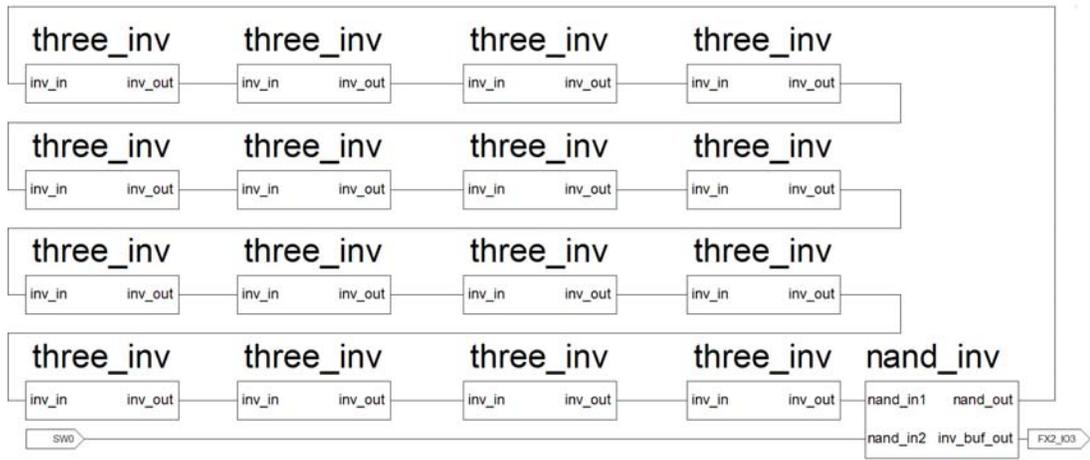
two_to_one_mux.vhd

```
-- file: two_to_one_mux.vhd
-- by: Michel Saydé
-- 27 April 2012
-- two to one multiplexer
-- Each cell of the wafer_IC will have one MUX.
-- It will be used to select between RO sensor
-- unit and the VCO.

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY two_to_one_mux IS
    PORT (
        mux_in_ro, mux_in_vco, s: IN STD_LOGIC;
        mux_out: OUT STD_LOGIC);
END two_to_one_mux;

ARCHITECTURE two_to_one_mux_internal OF two_to_one_mux IS
BEGIN
    WITH s SELECT
        mux_out <= mux_in_vco WHEN '1',
        mux_in_ro WHEN OTHERS;
END two_to_one_mux_internal;
```

Test circuit physically implemented on a Spartan-3E FPGA:

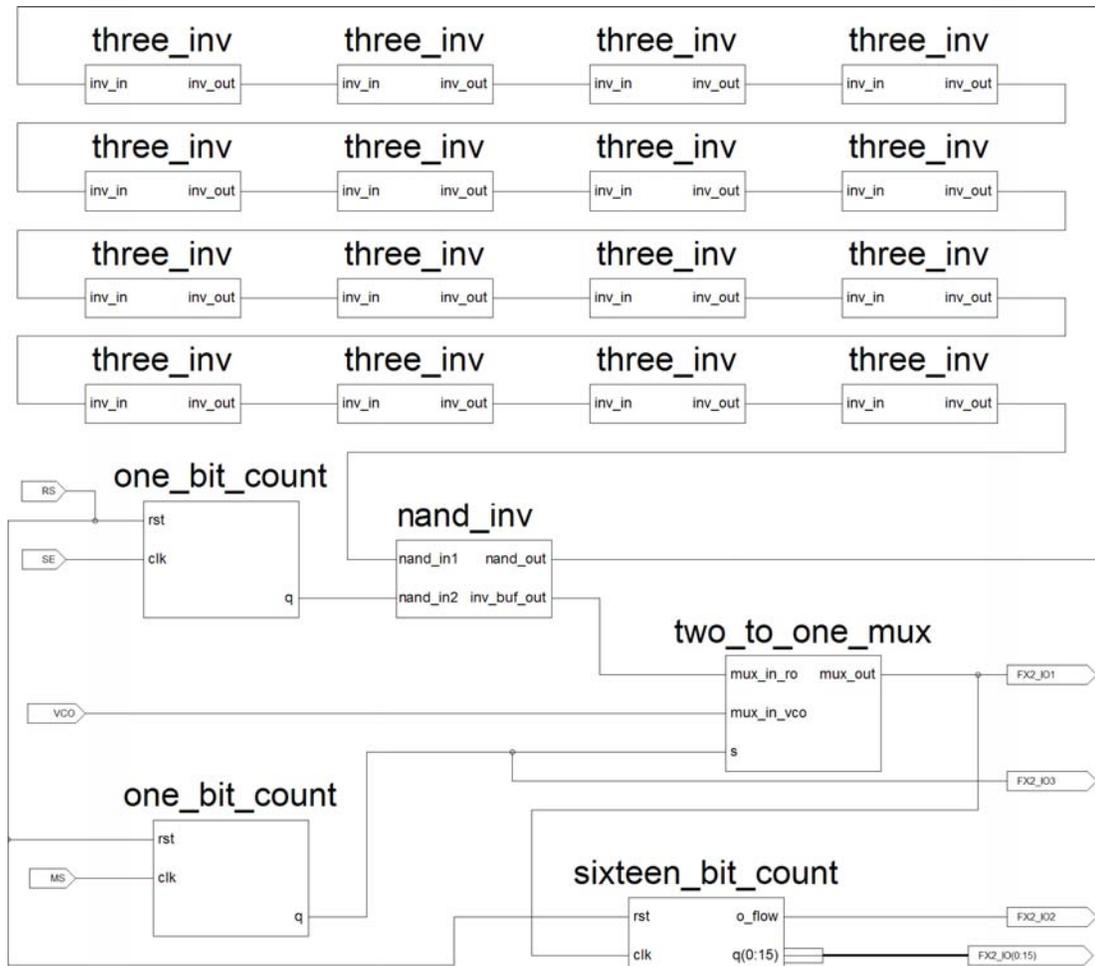


Type	Channel	Current	Maximum	Minimum	Average
Frequency	CHANNEL2		51.95MHz	27.99MHz	28.70MHz

Amplitude	CHANNEL2	3.438V	3.500V	62.50mV	2.620V
-----------	----------	--------	--------	---------	--------

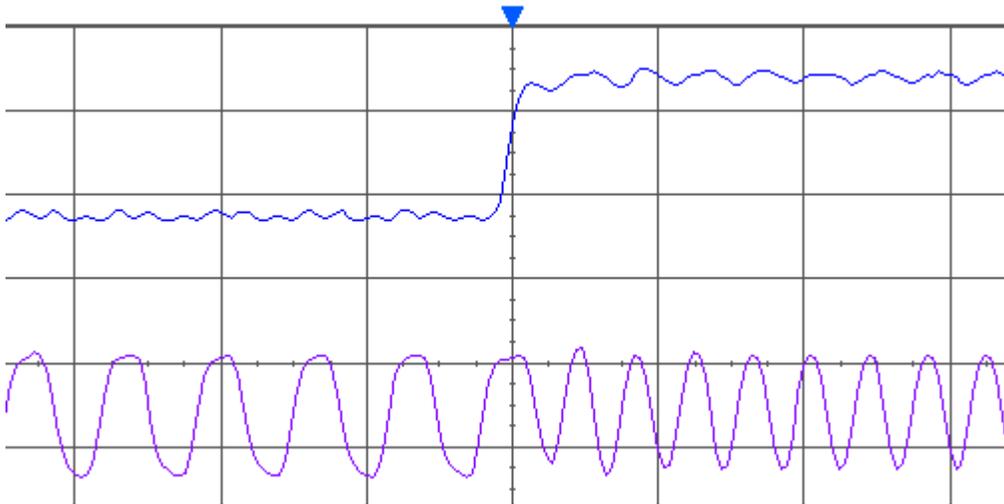
When Enable Input of the and_inv change from '0' to '1', "inv_buf_out" oscillate at 28.7MHz at room temperature. ()

One-Cell System

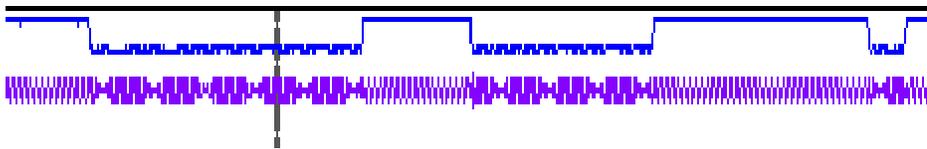


This is the complete system for one cell of the wafer IC. It has 4 inputs: Reset (RS), Sensor Enable (SE), VCO, and Mux Select (MS).

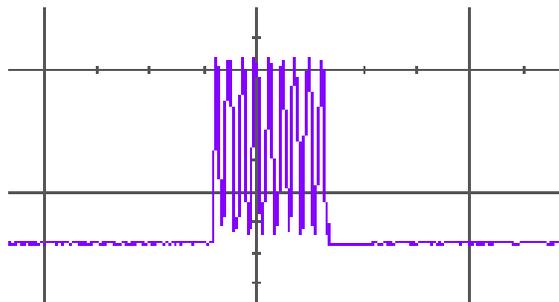
Test circuit physically implemented on a Spartan-3E FPGA and we measured:



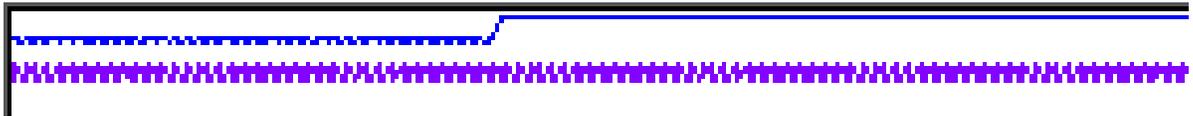
When MS changes from '0' to '1', mux_out switches from RO sensor (28.7MHz) to VCO (50MHz).



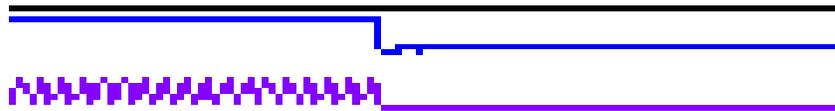
Also here we can see switching between RO sensor and VCO at each rising edge of MS.



With MS = '0' we measured the RO sensor at mux_out. We notice that at the first rising edge of SE, we activate the Oscillator, and at the second one, we deactivate it.



We test here the overflow flag of the 16-bit counter. It changes to '1', at maximum count.

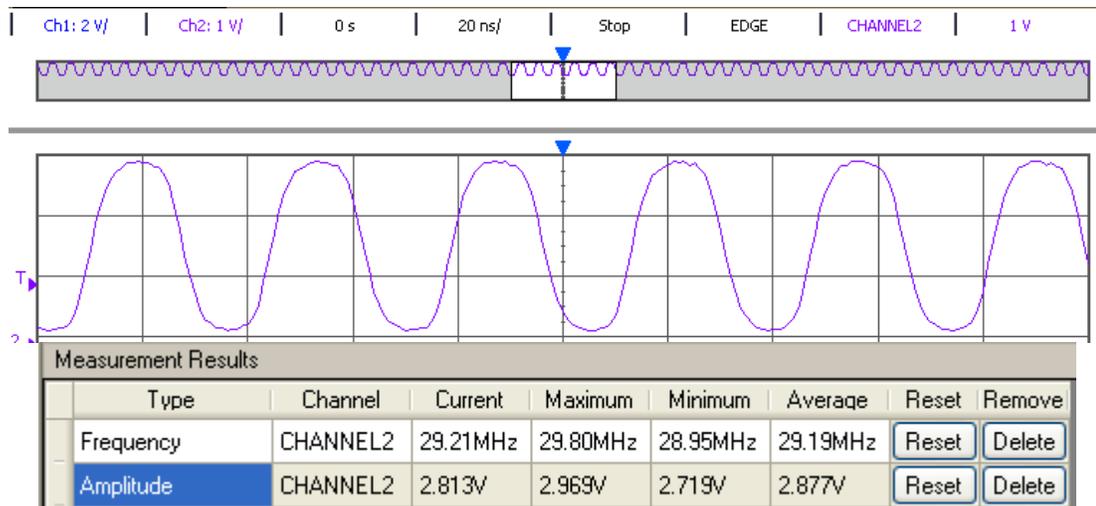


When reset is applied, we notice that the flag overflow fall to '0' and the mux switch back to the RO sensor. (We notice here that the RS will clear also the SE stored '1', and as result, the RO is disabled). And next cycle of measurement initialization will be ready when RS rise back to '1'.

VHDL as top level design

We created same project but this time with VHDL file as the top level design. This top level file is used only to interconnect the design in order to produce same functionality as the schematic design.

To test the RO Sensor we physically implemented it on a Spartan-3E FPGA and we temporarily routed the output of the sensor to an output pin and we measured on an oscilloscope the following signal:



These results confirm the proper functioning of the sensor.

Now we used a VHDL Top-Level source instead of schematic and the code is as follow:

```
-- file: one_cell_top
-- Top-Level Source file
-- by: Michel Saydé
-- 6 May 2012
-- to interconnect different modules related to temperature and pressure
-- sensor of one cell of the Wafer IC
-----

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY one_cell_top IS
  PORT (
```

```

        rs : IN STD_LOGIC;
        se : IN STD_LOGIC;
        vco : IN STD_LOGIC;
        m_s : IN STD_LOGIC;
    o_flow_flag : OUT STD_LOGIC;
    sens_indic: OUT STD_LOGIC;
    test_f_out: OUT STD_LOGIC;
    sensor_out: OUT INTEGER RANGE 0 TO 65535);

END one_cell_top;

architecture Behavioral of one_cell_top is

-- components
-- three_inv component
COMPONENT three_inv
    PORT (
        inv_in: IN STD_LOGIC;
        inv_out: OUT STD_LOGIC);
END COMPONENT;

-- nand_inv component
COMPONENT nand_inv
    PORT (
        nand_in1, nand_in2: IN STD_LOGIC;
        nand_out, inv_buf_out: OUT STD_LOGIC);
END COMPONENT;

-- one_bit_count component
COMPONENT one_bit_count
    PORT (
        rst, clk: IN STD_LOGIC;
        q: OUT STD_LOGIC);
END COMPONENT;

-- sixteen_bit_count component
COMPONENT sixteen_bit_count
    PORT (
        rst, clk: IN STD_LOGIC;
        o_flow: OUT STD_LOGIC;
        q: OUT INTEGER RANGE 0 TO 65535);
END COMPONENT;

-- two_to_one_mux component
COMPONENT two_to_one_mux
    PORT (
        mux_in_ro, mux_in_vco, s: IN STD_LOGIC;
        mux_out: OUT STD_LOGIC);
END COMPONENT;

```

```

-- signals
-- three_inv signals
SIGNAL inv_in  : STD_LOGIC_VECTOR(15 DOWNT0 0);
SIGNAL inv_out : STD_LOGIC_VECTOR(15 DOWNT0 0);

-- and_inv signals
SIGNAL nand_in1 : STD_LOGIC;
SIGNAL nand_in2 : STD_LOGIC;
SIGNAL nand_out  : STD_LOGIC;
SIGNAL inv_buf_out : STD_LOGIC;

-- one_bit_count signals
SIGNAL rst : STD_LOGIC_VECTOR(1 DOWNT0 0);
SIGNAL clk : STD_LOGIC_VECTOR(1 DOWNT0 0);
SIGNAL q   : STD_LOGIC_VECTOR(1 DOWNT0 0);

-- sixteen_bit_count signals
SIGNAL rst1 : STD_LOGIC;
SIGNAL clk1 : STD_LOGIC;
SIGNAL o_flow : STD_LOGIC;
SIGNAL q1 : INTEGER RANGE 0 TO 65535;

-- two_to_one_mux signals
SIGNAL mux_in_ro : STD_LOGIC;
SIGNAL mux_in_vco : STD_LOGIC;
SIGNAL mux_out : STD_LOGIC;
SIGNAL s : STD_LOGIC;

-- Start of circuit description
BEGIN

    -- three_inv using logic fabric
    three_inv_0 : three_inv
    PORT MAP (
        inv_in => inv_in(0),
        inv_out => inv_out(0));

    three_inv_1 : three_inv
    PORT MAP (
        inv_in => inv_in(1),
        inv_out => inv_out(1));

    three_inv_2 : three_inv
    PORT MAP (
        inv_in => inv_in(2),
        inv_out => inv_out(2));

    three_inv_3 : three_inv
    PORT MAP (
        inv_in => inv_in(3),
        inv_out => inv_out(3));

```

```
three_inv_4 : three_inv
PORT MAP (      inv_in => inv_in(4),
             inv_out => inv_out(4));

three_inv_5 : three_inv
PORT MAP (      inv_in => inv_in(5),
             inv_out => inv_out(5));

three_inv_6 : three_inv
PORT MAP (      inv_in => inv_in(6),
             inv_out => inv_out(6));

three_inv_7 : three_inv
PORT MAP (      inv_in => inv_in(7),
             inv_out => inv_out(7));

three_inv_8 : three_inv
PORT MAP (      inv_in => inv_in(8),
             inv_out => inv_out(8));

three_inv_9 : three_inv
PORT MAP (      inv_in => inv_in(9),
             inv_out => inv_out(9));

three_inv_10 : three_inv
PORT MAP (      inv_in => inv_in(10),
             inv_out => inv_out(10));

three_inv_11 : three_inv
PORT MAP (      inv_in => inv_in(11),
             inv_out => inv_out(11));

three_inv_12 : three_inv
PORT MAP (      inv_in => inv_in(12),
             inv_out => inv_out(12));

three_inv_13 : three_inv
PORT MAP (      inv_in => inv_in(13),
             inv_out => inv_out(13));

three_inv_14 : three_inv
PORT MAP (      inv_in => inv_in(14),
             inv_out => inv_out(14));

three_inv_15 : three_inv
PORT MAP (      inv_in => inv_in(15),
             inv_out => inv_out(15));

-- nand_inv using logic fabric
nand_inv_0 : nand_inv
PORT MAP (
```

```

        nand_in1 => nand_in1,
        nand_in2 => nand_in2,
        nand_out => nand_out,
        inv_buf_out => inv_buf_out);

-- one_bit_count using logic fabric
one_bit_count_0 : one_bit_count
PORT MAP (
    rst => rst(0),
    clk => clk(0),
    q => q(0));

one_bit_count_1 : one_bit_count
PORT MAP (
    rst => rst(1),
    clk => clk(1),
    q => q(1));

-- sixteen_bit_count using logic fabric
sixteen_bit_count_0 : sixteen_bit_count
PORT MAP (
    rst => rst1,
    clk => clk1,
    o_flow => o_flow,
    q => q1);

two_to_one_mux_0 : two_to_one_mux
PORT MAP (
    mux_in_ro => mux_in_ro,
    mux_in_vco => mux_in_vco,
    mux_out => mux_out,
    s => s);

-- Signals routing
-- RO sensor loop routing
inv_in(1) <= inv_out(0);
inv_in(2) <= inv_out(1);
inv_in(3) <= inv_out(2);
inv_in(4) <= inv_out(3);
inv_in(5) <= inv_out(4);
inv_in(6) <= inv_out(5);
inv_in(7) <= inv_out(6);
inv_in(8) <= inv_out(7);
inv_in(9) <= inv_out(8);
inv_in(10) <= inv_out(9);
inv_in(11) <= inv_out(10);
inv_in(12) <= inv_out(11);
inv_in(13) <= inv_out(12);
inv_in(14) <= inv_out(13);
inv_in(15) <= inv_out(14);
nand_in1 <= inv_out(15);

```

```

inv_in(0) <= nand_out;

-- nand_inv routing
mux_in_ro <= inv_buf_out;
nand_in2 <= q(0);

-- one_bit_count_0 routing
rst(0) <= rs;      --1->>>>> rs is an EXTERNAL input SIGNAL <<<<<--
clk(0) <= se;      --2->>>>> se is an EXTERNAL input SIGNAL <<<<<--

-- two_to_one_mux routing

mux_in_vco <= vco; --3->>>>> vco is an EXTERNAL input SIGNAL <<<<<--
clk1 <= mux_out; -- to 16_bit_cout clk
s <= q(1);

-----
test_f_out <= mux_out; -->>> Temporary frequency test output SIGNAL <<<<--
-----

-- one_bit_count_1 routing
rst(1) <= rs;      --1->>>>> rs is an EXTERNAL input SIGNAL <<<<<--
clk(1) <= m_s;     --4->>>>> ms is an EXTERNAL input SIGNAL <<<<<--
sens_indic <= q(1); --5->>> sens_indic is an EXTERNAL output SIGNAL <<<--

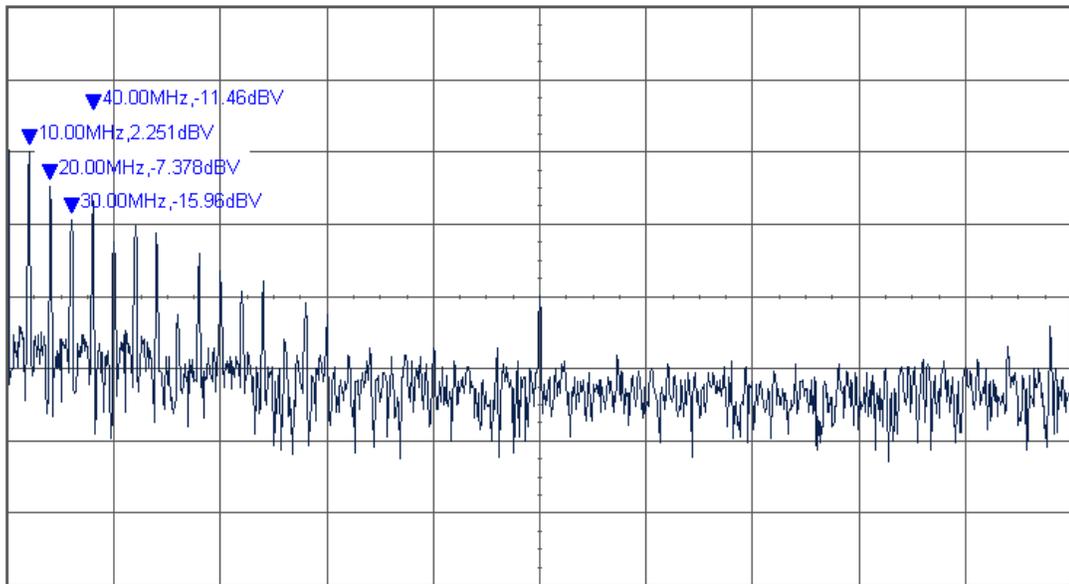
-- sixteen_bit_count_0 routing
sensor_out <= q1;   --6->>> sensor_out is an EXTERNAL output SIGNAL <<<--
o_flow_flag <= o_flow; --7->>> o_flow_flag is an EXTERNAL output SIGNAL <<--
rst1 <= rs;        --1->>>>> rs is an EXTERNAL input SIGNAL <<<<<--

END Behavioral;

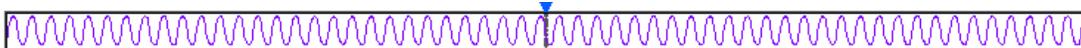
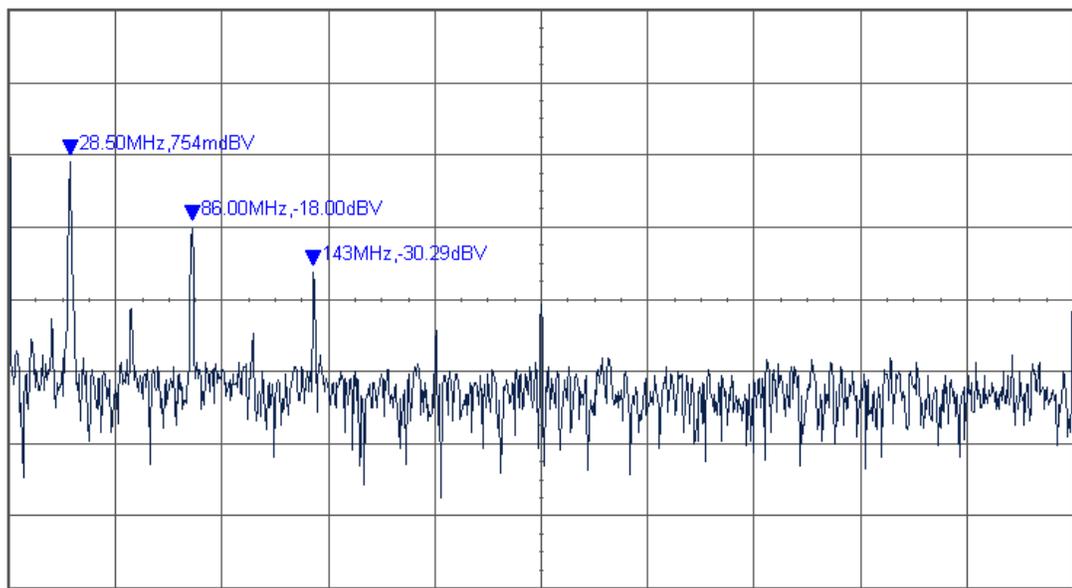
```

The Circuit has been tested and results same as the schematic (previous section)

Spectrum analysis of the output signal of the oscillator:



Rectangular wave from function generator.



Measurement Results								
Type	Channel	Current	Maximum	Minimum	Average	Reset	Remove	
Frequency	CHANNEL2	28.94MHz	29.25MHz	28.45MHz	28.67MHz	<input type="button" value="Reset"/>	<input type="button" value="Delete"/>	
Amplitude	CHANNEL2	2.984V	2.984V	2.875V	2.942V	<input type="button" value="Reset"/>	<input type="button" value="Delete"/>	

RO output (with 49 inverter) from Spartan E-3 Board.

As we can see from the spectrum of the rectangular wave, we have the fundamental at 28.5MHz and the harmonics of the rectangular wave. There is noise at 57MHz but the amplitude is very low.

Annexe B Code VHDL du circuit de test du chapitre 6

```

--
-- 3-bit (000 to 111) unsigned up counter with an asynchronous
reset.
-- By Michel Sayde 20 Jan 2013

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity Compt_3_bits is
port(Clock, Reset : in std_logic;
      Q : out std_logic_vector(2 downto 0));
end Compt_3_bits;

architecture Compt_3_bits of Compt_3_bits is
    signal tmp: std_logic_vector(2 downto 0);

    begin
        process (Clock, Reset)
        begin
            if (Reset='1') then
                tmp <= "000";
            elsif (Clock'event and Clock='1') then
                tmp <= tmp + 1;
            end if;
        end process;
        Q <= tmp;
    end Compt_3_bits;

```

```

----
-- 10 bit frequency counter with an asynchronous reset.
-- With Count window of 00ms Fmax is around 00Mhz
-- By Michel Sayde 20 Jan 2013

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity Compt_10_bits is
port(Clock, Reset : in std_logic;
      Q : out std_logic_vector(9 downto 0));
end Compt_10_bits;

```

```

architecture Compt_10_bits of Compt_10_bits is
    signal tmp: std_logic_vector(9 downto 0);

    begin
        process (Clock, Reset)
            begin
                if (Reset='1') then
                    tmp <= "0000000000";
                elsif (Clock'event and Clock='1') then
                    tmp <= tmp + 1;
                end if;
            end process;
        Q <= tmp;
    end Compt_10_bits;

```

```

-----
-- 25 bit frequency counter with an asynchronous reset.
-- With Count window of 320ms Fmax is around 104.8Mhz
-- By Michel Sayde 21 Jan 2013

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity Compt_25_bits is
port(Clock, Reset : in std_logic;
      Q : out std_logic_vector(24 downto 0));
end Compt_25_bits;

architecture Compt_25_bits of Compt_25_bits is
    signal tmp: std_logic_vector(24 downto 0);

    begin
        process (Clock, Reset)
            begin
                if (Reset='1') then
                    tmp <= "000000000000000000000000";
                elsif (Clock'event and Clock='1') then
                    tmp <= tmp + 1;
                end if;
            end process;
        Q <= tmp;
    end Compt_25_bits;

```

```

----
-- output clock controled by 25-bit input counter and clock input
-- Clock input must be >> that the 25-bit count variation
-- By Michel Sayde 9 August 2012

library IEEE;
  use IEEE.STD_LOGIC_1164.all;
  use IEEE.STD_LOGIC_ARITH.all;
  use IEEE.STD_LOGIC_UNSIGNED.all;

entity convert_frequence is
  port
    (
      inputs : in STD_LOGIC_VECTOR(24 downto 0); --
      horloge_entre : in STD_LOGIC;
      horloge_sortie : out STD_LOGIC
    );
end convert_frequence;

architecture convert_bihav of convert_frequence is

  constant zero : STD_LOGIC_VECTOR(19 downto 0)
:= "00000000000000000000";
  signal compteur: STD_LOGIC_VECTOR(24 downto 0) :=
"000000000000000000000000";
  signal horlogeBasseFrequenceInterne: STD_LOGIC;
begin

  process (horloge_entre)
  begin
    if (horloge_entre'event and horloge_entre = '1') then
      compteur <= compteur + 1;
      --
      if (compteur >= inputs and horlogeBasseFrequenceInterne
>= '1') then
        horlogeBasseFrequenceInterne <= '0';
        compteur <= (others =>'0' );
      elsif (compteur >= inputs and
horlogeBasseFrequenceInterne <= '0') then
        horlogeBasseFrequenceInterne <= '1';
        compteur <= (others =>'0' );
      end if;
    end if;
  end process;
  horloge_sortie<= horlogeBasseFrequenceInterne;
end convert_bihav;

```

```

----
-- 4-bit Signed Up Counter with Asynchronous Reset and Modulo
--Maximum
--Michel Saydé/12 Jan 2013
--

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;

entity Count_mod_6 is
    generic (MAX : integer := 6);
    port(Clock, Reset : in std_logic;
         Q : out integer range 0 to MAX-1);
end Count_mod_6;

architecture archi of count_Mod_6 is
    signal i : integer range 0 to MAX-1;
begin
    process (Clock, Reset)
    begin
        if (Reset='1') then
            i <= 0;
        elsif (rising_edge(Clock)) then
            i <= (i + 1) mod MAX ;
        end if;
    end process;

    Q <= i;

end archi;

```

```

---- 8-Phase generator (Decoder)
-- By Michel Sayde, 6 Jan. 2013
--
library ieee;
use ieee.std_logic_1164.all;
entity decoder_8 is
port (sel: in std_logic_vector (2 downto 0));
Q : out std_logic_vector (7 downto 0));
end decoder_8;
architecture archi of decoder_8 is
begin
Q <= "00000001" when sel = "000" else
     "00000010" when sel = "001" else
     "00000100" when sel = "010" else
     "00001000" when sel = "011" else

```

```

    "00010000" when sel = "100" else
    "00100000" when sel = "101" else
    "01000000" when sel = "110" else
    "10000000";
end archi;

```

```

--Diviseur d'horloge
--By Michel Sayde

library IEEE;
  use IEEE.STD_LOGIC_1164.all;
  use IEEE.STD_LOGIC_ARITH.all;
  use IEEE.STD_LOGIC_UNSIGNED.all;

entity diviseurHorloge is
  port
  (
    horlogeHauteFrequence : in  STD_LOGIC;
    horlogeBasseFrequence : out STD_LOGIC
  );
end diviseurHorloge;

architecture comportHorloge of diviseurHorloge is
  constant dix : STD_LOGIC_VECTOR(6 downto 0) := "1111111";
  constant zero : STD_LOGIC_VECTOR(6 downto 0) := "0000000";
  signal compteur: STD_LOGIC_VECTOR(6 downto 0);
  signal horlogeBasseFrequenceInterne: STD_LOGIC;

begin
  process (horlogeHauteFrequence) begin
    if (horlogeHauteFrequence'event and horlogeHauteFrequence = '1')
    then
      compteur <= compteur + 1;

      if (compteur = dix and horlogeBasseFrequenceInterne = '1')
      then
        horlogeBasseFrequenceInterne <= '0';
        compteur <= (others =>'0' );
      elsif(compteur = dix and horlogeBasseFrequenceInterne <= '0')
      then
        horlogeBasseFrequenceInterne <= '1';
        compteur <= (others =>'0' );
      end if;
    end if;
  end process;
  horlogeBasseFrequence<= horlogeBasseFrequenceInterne;
end comportHorloge;

```

```

-- By Michel Saydé
-- 29 Jan. 2013
-- Convert 50Mhz input clock to 25Hz Output clock 50% duty cycle

library IEEE;
  use IEEE.STD_LOGIC_1164.all;
  use IEEE.STD_LOGIC_ARITH.all;
  use IEEE.STD_LOGIC_UNSIGNED.all;

entity Mhz_to_Hz is
  port(
    Input_50Mhz : in  STD_LOGIC;
    Output_25Hz : out STD_LOGIC
  );
end Mhz_to_Hz;

architecture comportHorloge of Mhz_to_Hz is
  constant dix      : STD_LOGIC_VECTOR(19 downto 0) :=
"11110100001001000000";
  constant zero     : STD_LOGIC_VECTOR(19 downto 0)
:= "00000000000000000000";
  signal compteur: STD_LOGIC_VECTOR(19 downto 0);
  signal horlogeBasseFrequenceInterne: STD_LOGIC;

begin
  process (Input_50Mhz) begin
    if (Input_50Mhz'event and Input_50Mhz = '1') then
      compteur <= compteur + 1;

      if (compteur >= dix and horlogeBasseFrequenceInterne >= '1')
then
        horlogeBasseFrequenceInterne <= '0';
        compteur <= (others =>'0' );
      elsif(compteur >= dix and horlogeBasseFrequenceInterne <= '0')
then
        horlogeBasseFrequenceInterne <= '1';
        compteur <= (others =>'0' );
      end if;
    end if;
  end process;
  Output_25Hz<= horlogeBasseFrequenceInterne;
end comportHorloge;

```

```

-----
-- 25-bits parallel in parallel out shift register
-- with Positive-Edge Clock and Asynchronous Reset.

```

```
-- Michel Sayde, 30 Jan. 2012
--

library ieee;
use ieee.std_logic_1164.all;

entity reg_25_Bts is
  port(Clock, Reset : in std_logic;
        D : in std_logic_vector(24 downto 0);
        Q : out std_logic_vector(24 downto 0));
end reg_25_Bts;

architecture archi of reg_25_Bts is
begin
  process (Clock, Reset)
  begin
    if (Reset = '1')then
      Q <= "000000000000000000000000";
    elsif (Clock'event and Clock='1')then
      Q <= D;
    end if;
  end process;
end archi;
```

Annexe C Code Matlab de l'amélioration des résultats de sortie du RO et de l'algorithme de linéarisation

```

%Mémoire Michel Sayde
%Le 20 avril 2013
% Linearisation et amélioration de la réponse de Fout
%-----

%Intervales des températures
t=[30 : 1: 62];

%valeurs mesurées de Fout sur l'oscilloscope
Fout = [7954 4650 3300 2580 2170 1880 1570 1460 1300 1170 1091 993
923 ...
      850 774 740 696 656 625 601 566 545 515 495 477 460 440 423 405
...
      395 382 371 359];
plot(t,Fout);
grid on;

Xtal=50000000; %Cristal Oscillator frequency (50Mhz)

%Calcule de Fsub
Fsub=0;
for (i=1 :1 : 33)
    Fsub(i)=(Xtal/(2*Fout(i))); %de 6.7
end

%Calcule de la moyenne de Fsub
Fsub_sum=0;
for (i=2 :1 : 33)
    Fsub_sum=Fsub_sum+(Fsub(i)-Fsub(i-1));
end
Fsub_mean=Fsub_sum/32;

%Re-distribution des intervales pour balancer la réponse
Fsub_balanced = [Fsub(1) : Fsub_mean : Fsub_mean*34]; %30 to 62
hold on;
plot(t,Fsub);
plot(t,Fsub_balanced, '-r');
grid on;

% Generating Balanced Fout
for (i=1 :1 : 33)
    Fout_balanced(i)=50000000/(2*Fsub_balanced(i));
end
hold on;
plot(t,Fout);

```

```

plot(t,Fout_balanced, '-r');
grid on;

% Calcul des coefficients a, b, et c
tl=30; tm=40; th=62; %les intervalles de température de la courbe
T1k=tl+273.15; T2k=tm+273.15; T3k=th+273.15; %t en Kelvin

F1= Fout_balanced(tl-(tl-1));
F2= Fout_balanced(tm-(tl-1));
F3= Fout_balanced(th-(tl-1)); %Les 3 intervalles de fréquences

%Calcule de a,b et c (3 équations 3 inconnus)
Y1=1/T1k; Y2=1/T2k; Y3=1/T3k;
L1=log(F1); L2=log(F2); L3=log(F3);
y2=(Y2-Y1)/(L2-L1); y3=(Y3-Y1)/(L3-L1);
c=((y3-y2)/(L3-L2))*(L1+L2+L3)^-1;
b=y2-(c*(L1^2+(L1*L2)+L2^2));
a=Y1-(b+L1^2*c)*L1;

% function [Tc1] = Linearization(Fout)
f = log(Fout_balanced);
ex_a = a+(b*(f));
ex_b = c*(f).^3;
ny = ex_a+ex_b;
Ti = 1./ny;
Tc1 = Ti-273.15;

%les résultats
plot(t,Fout_balanced);
grid on;
hold on;
plot(Tc1,Fout_balanced, '-r');
hold off;

%-----
%%40 to 72 deg
%-----
t72=[40 : 1: 72];
Fsub_bal_72 = [Fsub(11) : Fsub_mean : Fsub_mean*44]; %40 to 72

% Generating Balanced Fout
for (i=1 :1 : 33)
Fout_bal_72(i)=50000000/(2*Fsub_bal_72(i));
end

hold on;
plot(t72,Fout_bal_72, '-r');
grid on;

```

```

% Calcul des coefficients a, b, et c
t1=40; tm=48; th=72; %les intervalles de température de la courbe
T1k=t1+273.15; T2k=tm+273.15; T3k=th+273.15; %t en Kelvin

F1= Fout_bal_72(t1-(t1-1));
F2= Fout_bal_72(tm-(t1-1));
F3= Fout_bal_72(th-(t1-1)); %Les 3 intervalles de fréquences

%Calcule de a,b et c (3 équations 3 inconnus)
Y1=1/T1k; Y2=1/T2k; Y3=1/T3k;
L1=log(F1); L2=log(F2); L3=log(F3);
y2=(Y2-Y1)/(L2-L1); y3=(Y3-Y1)/(L3-L1);
c=((y3-y2)/(L3-L2))*(L1+L2+L3)^-1;
b=y2-(c*(L1^2+(L1*L2)+L2^2));
a=Y1-(b+L1^2*c)*L1;

% function [Tc1] = Linearization(Fout)
f = log(Fout_bal_72);
ex_a = a+(b*(f));
ex_b = c*(f).^3;
ny = ex_a+ex_b;
Ti = 1./ny;
Tc1 = Ti-273.15;

%les résultats
plot(t72,Fout_bal_72);
grid on;
hold on;
plot(Tc1,Fout_bal_72,'-r');
hold off;

```

Annexe D Code Matlab de quelques fonctions de traitement d'image utilisés dans ce mémoire (Ch.6)

```

    Thermal1 = imread('Reduced_2.png'); %Placer l'image dans la
variable
    Thermal2 = imread('Reduced_3.png'); %Placer l'image dans la
variable
    imshow(uint8(Thermal1)); % Afficher l'image
    Thermal1 = rgb2gray(Thermal1);
    Thermal1 =double(Thermal1);
    Thermal2 = rgb2gray(Thermal2);
    Thermal2 =double(Thermal2);

    %Thermal = im2double(Thermal); %convertir en double f(x,y)/255
    [x,y] = size(Thermal1); % saisir grandeur de l'image
    TThermal = x*y; %Taille de de l'image
    %min et max de chaque images
    max_Thermal1 = max(max(Thermal1));
    min_Thermal1 = min(min(Thermal1));
    max_Thermal2 = max(max(Thermal2));
    min_Thermal2 = min(min(Thermal2));

    %First derivative 2 images
    seuil=23;
    test1 = Thermal1;
    test2 = Thermal2;
    for i = 1:x
        for j = 1:y
            if (test2(i,j)-test1(i,j) >= seuil)
                test(i,j) = (test2(i,j)-test1(i,j))*12;
            else
                test(i,j) = 0;
            end
        end
    end
    imshow(uint8(test)); % Afficher l'image

    %% 1- seuillage DES IMAGES
    seuil=100;
    test = Thermal1;
    for i = 1:x
        for j = 1:y
            if (test(i,j)>=seuil)
                test(i,j) = 255;
            else
                test(i,j) = 0;
            end
        end
    end

```

```

        end
    end
end
imshow(uint8(test)); % Afficher l'image

%% 1- NEGATIVE DES IMAGES

for i = 1:x
    for j = 1:y
        Thermal_negative(i,j)= 255-Thermal1(i,j);
    end
end
imshow(uint8(Thermal_negative)); % Afficher l'image

%% 1- La fonction 'bruit_reduction'
%Entrées: - une image 'im_in'
%         - genre de filtre 'genre':
%           0 = moyen
%           1 = Gaussien
%           2 = médian
%         - 'taille' est la taille du filtre
%
%Sortie:
%   'im_out' reçoit l'image filtrée

%Paramètres d'entrés de la fonction
im_in = Thermal_negative; %Image d'entrée
genre = 2; %0=moyen, 1=Gaussien ou 2=médian
taille = 19; %une valeur impaire pour le masque

%Appel de la fonction avec les paramètres d'entrés
im_out = bruit_reduction(im_in,genre,taille);
imshow(uint8(im_out));%Image de la sortie
im_out=0; %reset de la variable 'im_out'
im_in=0; %reset de la variable 'im_in'
imshow(uint8(im_in));

%% 2- Laplacien
%Paramètres d'entrés de la fonction
im_in = Thermal_negative; %Image d'entrée

%Fonction call
im_out = Laplacien(im_in);
imshow(uint8(im_out));%Image de la sortie
im_out=0; %reset de la variable 'im_out'
im_in=0; %reset de la variable 'im_in'

```

```

%% 3- La fonction 'contours_detection'

%Paramètres d'entrés de la fonction
im_in = Thermal_negative; %Image d'entrée
seuil = 10; %seuil de separation pour image binaire

%On applique un filtre Gaussien 5x5
im_out = bruit_reduction(im_in,1,5);
im_in=im_out;
im_out = bruit_reduction(im_in,2,3);%et un filtre median
im_in=im_out;
%Detection de contours
im_out = contours_detection(im_in,seuil);
imshow(uint8(im_out));%Image de la sortie
im_out=0; %reset de la variable 'im_out2'
im_in=0; %reset de la variable 'im_in'
imshow(uint8(im_in));

```

```

% La fontion Laplacien
% Par Michel Saydé

function [im_out]=laplacien(im_in)

% Entrées:
% - im_in est l'image d'entrée
%
% Sortie:
% 'im_out' est l'image filtrée

[y,x] = size(im_in);
co=0;
taille=3;
%h = fspecial('laplacian', 0) %Masque a
%h=[-1,-1,-1;
% -1, 8,-1;
% -1,-1,-1]; %masque d

h= [0 -1 0;
-1 4 -1;
0 -1 0];

```

```

        for i = 1+(taille-1)/2:y-(taille-1)/2
            for j = 1+(taille-1)/2:x-(taille-1)/2
                for k=i-(taille-1)/2:i+(taille-1)/2
                    for l=j-(taille-1)/2:j+(taille-1)/2
                        convo=(h(1+k-i+(taille-1)/2,1+l-j+(taille-
1)/2)*im_in(k,l));
                        co=co+convo;
                    end
                end
            end
            im_out(i,j)= co+im_in(i,j)*1.4; %avec amelioration de la
luminositee
            %im_out(i,j)= co-im_in(i,j);
            co=0;
        end
    end
im_in=0;

```

```

% La fonction de filtrage de bruit d'une image
% Par Michel Saydé
% le 1 mars 2013

function [im_out]=bruit_reduct(im_in,genre,taille)

% Entrées:
% - im_in est l'image d'entrée
% - 'genre' est le type de filtre: 0 = moyen
%           1 = Gaussien
%           2 = médian
% - 'taille' est la taille du filtre

% Sortie:
% 'im_out' est l'image filtrée

[y,x] = size(im_in);
co=0;
if genre < 2
    if genre < 1
        h = fspecial('average', taille)
        %h=[ 1, 2, 1; 2, 4, 2; 1, 2, 1] %masque d

    else
        h = fspecial('gaussian', taille, 1)
    end
    sumh=sum(sum(h))
    for i = 1+(taille-1)/2:y-(taille-1)/2
        for j = 1+(taille-1)/2:x-(taille-1)/2

```

```

        for k=i-(taille-1)/2:i+(taille-1)/2
            for l=j-(taille-1)/2:j+(taille-1)/2
                convo=(h(1+k-i+(taille-1)/2,1+l-j+(taille-
1)/2)*im_in(k,l));
                co=co+(convo/sumh);
            end
        end
        im_out(i,j)= co;
        co=0;
    end
end
else
%Median
    im_out=0;
    comp_med=1;
    vect_med=0;
    for i = 1+(taille-1)/2:y-(taille-1)/2
        for j = 1+(taille-1)/2:x-(taille-1)/2
            for k=i-(taille-1)/2:i+(taille-1)/2
                for l=j-(taille-1)/2:j+(taille-1)/2
                    vect_med(comp_med)=(im_in(k,l));
                    comp_med=comp_med+1;
                end
            end
            co=median(vect_med);
            comp_med=1;
            im_out(i,j)= co;
            co=0;
        end
    end
end %end if
im_in=0;

```

SoC Systems Thermal Monitoring Using Embedded Sensor Cells Unit

Michel Saydé¹ Ahmed Lakhssassi¹ Mohammed Bougataya¹ Omar Terkawi² Yves Blaquière³
 Saym01@uqo.ca ahmed.lakhssassi@uqo.ca Mohammed.Bougataya@uqo.ca Omar.Terkawi@gmail.com blaquiere.yves@uqam.ca

⁽¹⁾Computer Sc. & Eng. Department,
 Université du Québec en Outaouais
 Gatineau (PQ), Canada.

⁽²⁾Electrical Engineering Department,
 École Polytechnique de Montréal
 Montréal (PQ), Canada.

⁽³⁾Computer Sc. Department,
 Université du Québec à Montréal
 Montréal (PQ), Canada.

Abstract— During the development of SoC (System on Chip) Systems, the thermal design aspects have proved crucial to its reliable operation. Reducing thermally induced stress and preventing local overheating remain major concerns when optimizing the capabilities of the LAIC (Large Area Integrated Circuits) based technology. Thermal monitoring is essential in integrated circuit (IC) and VLSI chip which are a multilayer structure and a stack of different materials. Thermal phenomena research activities on micro-scale level are essential for SoC and MEMS-based applications. However, various measurement techniques are needed to understand the thermal behavior of VLSI chip. In particular, measurement techniques for surface temperature distributions of large VLSI systems are a highly challenging research topic. This paper presents an approach and the experimental result of silicon-die thermal monitoring using embedded sensor cells unit. Sensor implementation results and analysis are also presented.

Index Terms- Silicon Sensors, VLSI, SoC, FPGA, Thermal Analysis, GDS technique.

I. INTRODUCTION

The SoC (System on Chip) has become/became more common in the electronic industry during the last years. Complex systems are integrated on a single SoC [1]. This trend is motivated by the needs of rendering the system high efficient with high performance and more compact with less power consumption. The high integration in the SoC systems increase the power density of power consumed and dissipated in the chip. However, 50% of the failure in the electronic device is due to the increase of the internal temperature of the chip [2].

Due to aggressive technology scaling, VLSI integration density as well as power density increases drastically. For example, the power density of high performance microprocessors has already reached $50\text{W}/\text{cm}^2$ at 100nm technology and $100\text{W}/\text{cm}^2$ at 50nm technology [3]. It has been reported in [4] that temperature variations of 30°C can occur in a high performance microprocessor design. The magnitude of thermal gradients and associated thermo-mechanical stress is expected to increase further as VLSI designs move into nanometer processes and multi-GHz frequencies. In [5], a low-power ring oscillator-based temperature sensor designed for U-healthcare was presented. The proposed sensor exploits the frequency of the ring

oscillator (RO) that increases as temperature increases as a way of converting temperature to a digital output. However, the sensor exhibits nonlinear characteristic in its temperature range of 5 to 100°C .

One of the important questions in the field of thermal issues of VLSI systems and micro-systems is how to perform the thermal monitoring, in order to indicate the overheating situations, without complicated control circuits. Traditional approach consists of placement of many sensors everywhere on the chip, and then their output can be read simultaneously and compared with the reference voltage recognized as the overheating level.

In this paper the idea of the proposed method is to predict the local temperature and gradient along the given distance in a few places only on the monitored surface, and evaluate obtained information in order to predict the temperature of the heat source. Therefore, in the case of a high density SoC devices, there is no place on the layout for the complicated unit performing computations, but there is also no need for it, as we only want to detect the overheating situations. These peaks are essentials during thermal die monitoring to avoid a critical induced thermo-mechanical stress. Moreover, in most cases, the overheating occurs only in one place.

Hence we will presents an approach and the experimental result of silicon-die thermal monitoring method using embedded silicon sensor cells unit. The proposed method using only two sensor cells will be formulated in a manner to facilitate the development of modular architectures using minimum silicon space in regards of their implementation in LAIC systems. The architecture selected will be modeled in high level languages, simulated in order to evaluate its performances, and then implemented on a FPGA (Field-Programmable Gate Array). A closed loop of simulation is used in order to evaluate the performances of the architectures proposed at each stage. The proposed architecture will be designed in a modular perspective after separation of the different elementary functions of the monitoring system. Hence the design of surface peaks thermal monitoring approach will be presented. The architecture is designed in high level languages such as

Matlab™-Simulink®, simulated and tested using VHDL and synthesized using Xilinx™ ISE [6] and Altera™ DSP [7] tools. The simulation and hardware implementation results obtained will be compared to a finite element method (FEM) temperature prediction of the entire GDS (Gradient Direction Sensor) method [8] configuration cells.

II. THERMAL MONITORING BY GDS METHOD

The Thermal Monitoring System (TMS) is a Surface Peaks Thermal Detector Algorithm (SPTDA) which is based on the GDS method for evaluating a single heat source on the chip surface. This method principle of work is explained in details in [8] and [9]. The GDS method has been studied and analyzed for its applicability as inverse engineering problem which is capable to detect the thermal peaks and associated thermo-mechanical stress in the critical surface areas of large VLSI devices [9]. In order to obtain information about heat-source parameters we only need to know from where the phenomenon is distributed and sometimes how fast it is changing. The geometrical coordinates and an estimation of the investigated heat source can be obtained by applying the GDS method. For two sensors, A and C placed in the distance “a” (Fig. 1), the difference between their output voltages or frequency is proportional to the changes of the temperature value along the distance “a” [8]. This is true only when the heat source is directly on the line AC for any other cases the values of the angle α has to be taken into account for the proper calculation of ΔT (1). Correspondence between V, F to temperature T will depend on sensor type used (V refer to sensor voltage and F to Ring Oscillator (RO) frequency:

$$\frac{\Delta T}{\Delta r} = \frac{T_C - T_A}{a \cdot \cos \alpha} \leftrightarrow \frac{V_C - V_A}{a \cdot \cos \alpha} \quad (1)$$

Where r is the distance from the heat source.

In figure 1, we have:

$$b = \overline{AD}, \overline{AD} \leftrightarrow T_B - T_A \leftrightarrow V_B - V_A \quad (2)$$

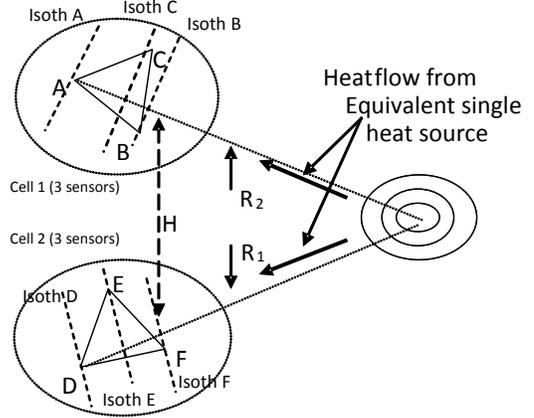
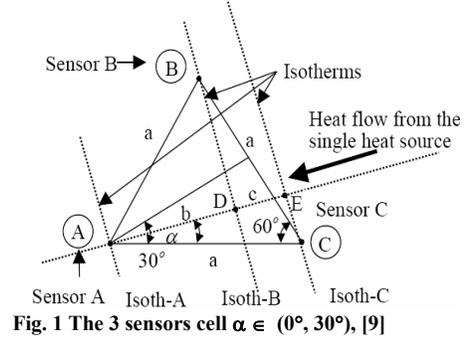
$$b + c = \overline{AE}, \overline{AE} \leftrightarrow T_C - T_A \leftrightarrow V_C - V_A \quad (3)$$

In order to obtain the information about angle α we should apply the third sensor. In the simplest case the GDS contains only three temperature sensors placed in distance a (fig. 1).

On the basis of equations (2) and (3) and the geometrical dependencies from (fig. 1), eq. (4) is obtained:

$$\tan \alpha \leftrightarrow \frac{2}{\sqrt{3}} \cdot \left(\frac{V_B - V_A}{V_C - V_A} - \frac{1}{2} \right) \quad (4)$$

Using the cell we can obtain information on the temperature distribution and partly on the position of the heat source. In order to obtain the temperature value of a single punctual heat source we have to calculate the distance between the sensor and this source. Two sensor cells are required for this purpose (fig. 2). The cells are placed at a given distance (H) and each of them gives information about angle α (α_1 and α_2) in the direction of the heat source. Under the consideration that “a” is relatively small, we can assume that the heat source and the center of the cells form a triangle in which the length of one side and values of the angles adjacent to this side are known.



Therefore, the third angle can be deduced. This means that we can calculate the distances R_1 and R_2 between the heat source and the sensors. Now we can calculate the temperature gradient along the known distance. By adding it to the temperature of the sensor we obtain the temperature of the heat source. Two sensor cells A, B, C and D, E, F are placed in two corners of a monitored die in the distance H. Hence, the temperature of the heat source can be obtained by equation (5).

$$T_s \leftrightarrow V_s = \frac{H}{a} \cdot (V_{C1} - V_{A1}) \cdot \frac{(\tan^2 \alpha_1 + 1) \cdot (\sqrt{3} + \tan \alpha_2)}{\sqrt{3} \cdot (1 - \tan \alpha_1 \cdot \tan \alpha_2) - (\tan \alpha_1 + \tan \alpha_2)} + V_{A1} \quad (5)$$

Figure 2 show the proposed distribution of the 6 sensors divided into 2 cells located implemented on the chip.

III. ALGORITHMIC DESIGN

The SPTDA algorithm flowchart is presented in figure 3. The algorithm is based on parallelism computation between the internal variables represented by $\tan \alpha_1$ and $\tan \alpha_2$. This parallelism will be used later in the architectural design in order to optimize the speed of the running implementation permitting reuse of the unit for the whole chip sequential thermal monitoring. To this end, the internal architecture of the algorithm is divided into modular functions. Each module is responsible for a certain set of operations and sequentially passes the results to another module. The architecture has been divided in 3 principal modules. Each module has been assigned a set of calculation tasks formulated in a way to maintain a high degree of parallelism.

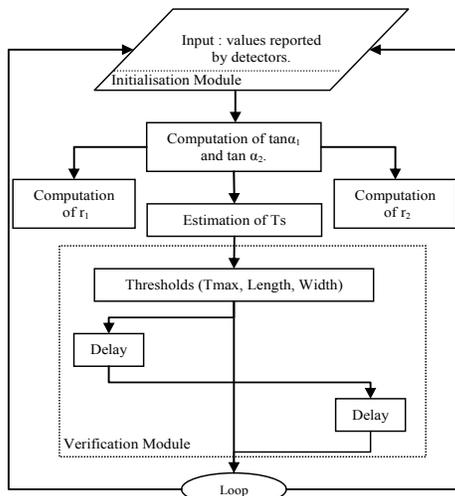


Fig. 3 SPTDA flowchart

Figure 4 represents the internal architecture of module 1 and module 2 for the computation of the first and second slopes. Elementary processes from E1 to E4 execute specific computation tasks in parallel between module 1 and 2, but these modules output produce different results. During this computation, some temporary variables are passed in parallel to module 3 which computes the temperature of the thermal peak. However module 3 works in sequence with module 1 and 2, because the final results delivered by those modules are used as input to the third module.

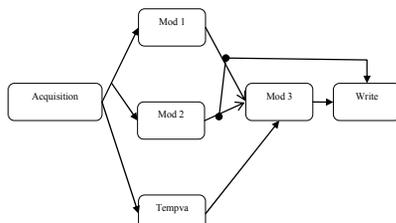


Fig. 4 SPTDA Algorithm graph

IV. SYSTEM DESIGN AND IMPLEMENTATION

According to our measurements, a Ring Oscillator (RO) with 9 inverters implemented on XILINX xc3s500e-4fg320 FPGA, will oscillate at an average frequency of 50 MHz at 25 °C.

As figure 5 shows. We have implemented the TMS on a Xilinx xc3s500e-4fg320 FPGA. Each of the 2 cells is composed of 3 RO sensors and each RO sensor is composed of 9-cell inverters. The TSM is placed manually on the FPGA at a specific locations and we had measured an average of 50MHz as output frequency of each RO sensor.

Figure 6 shows the TSM signal acquisition block diagram. The frequency variation due to temperature change is very small and difficult to measure.

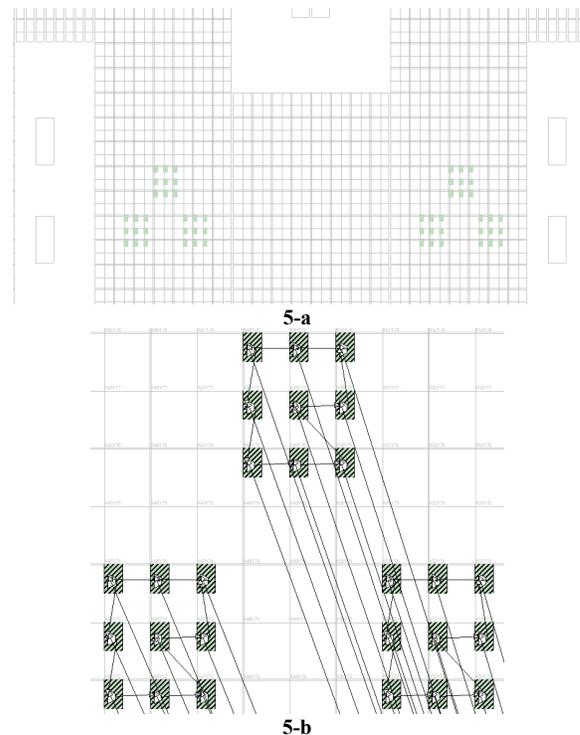


Fig. 5 a- A general "Floorplanner" view from Xilinx ISE shows the TMS physical implementation on a XILINX xc3s500e-4fg320 FPGA. b- A close look on one cell. We can see that each cell is composed of 3 RO sensors and each RO sensor is composed of 9 inverters.

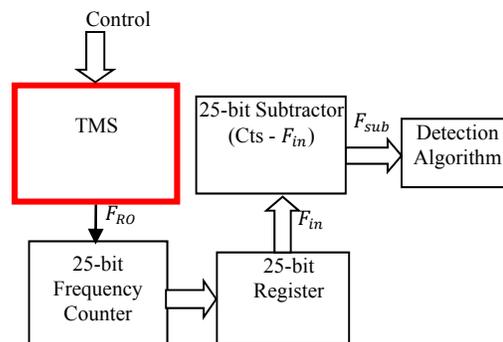


Fig. 6 Block diagram of the system circuitry

$$F_{sub} = cts - F_{in} \quad (6)$$

F_{sub} = Count reading relative to temperature measured by RO;

cts = Subtraction constant (specific to each RO sensor).

The constant cts should be always greater than F_{in} , so we select the value of cts after knowing the largest value of F_{RO} which represent the lowest temperature to be measured.

V. EXPERIMENTAL RESULTS

Oven calibration test:

In order to study the RO response to local temperature variation, we placed the FPGA board into a controlled oven temperature (figure 7) and changed gradually the temperature from ambient to around 60 °C.

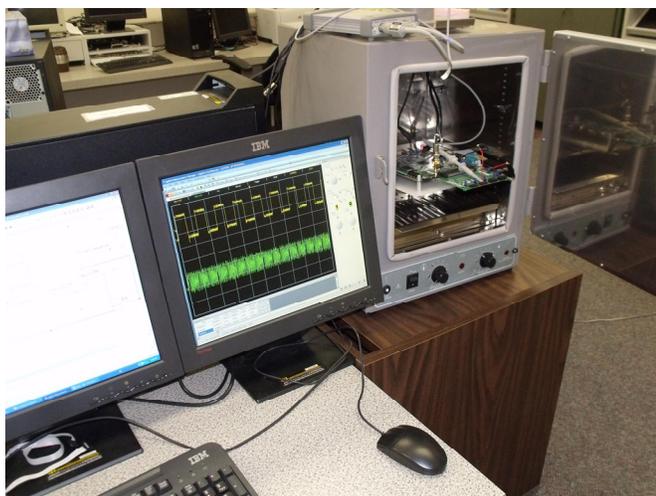


Figure 7- Oven calibration test

Then we've plotted the relation between temperature variation and frequency. Figure 8 shows the plot of 1 RO sensor from 30 to 62°C. In order to the algorithm to work properly, it requires a minimum reading resolution. For smaller areas where the sensor and the source distances are restively small we require a higher resolution. By simulation (not shown in this paper), the desired resolution for our system is defined to be 100 values per 1 degree of temperature changes. So for a temperature variation from 25 to 65 °C we need a minimum of 4000 values (12-bit). So the 25-bit used is more than enough. Figure 8 shows the measured variation of frequency of one RO sensor with temperature. We can see a linear relation at least for the desired interval (25 to 65°C). This linearity cannot be garneted for RO sensors implemented on different hardware or for larger intervals. So a linearization method should be required if a correct and precise temperature reading is desired. A good linearization method could be Steinhart and Hart [10] method. But by analogy we need to replace resistivity variation in the equation by frequency variation.

VI. DISCUSSION

In this paper, a methodology to evaluate and predict a thermal peak of large VLSI circuit using embedded sensor was presented. The important factors contributing to the device's thermal heating were presented. The monitoring approach reported in this paper can be applied to predict the peak thermal stress of multilevel structures. Surface peaks thermal detection is necessary in modern VLSI circuits and LAIC systems; their internal stress due to packaging combined with local self-heating becomes serious and may result in large performance variation, circuit malfunction and even chip cracking.

VII. CONCLUSION

This paper has presented an approach and the experimental result of silicon-die thermal monitoring using embedded sensor cells unit into the SoC systems. We have presented a method to calibrate the sensor and to adapt it to our need. As

prototype, only one cell unit test results has been presented. Therefore this study is part of ongoing research toward the development of a complete stress monitoring system to be implemented on a SoC based system technology in order to increase the lifetime cycle of the product and to insure a safe and reliable operation.

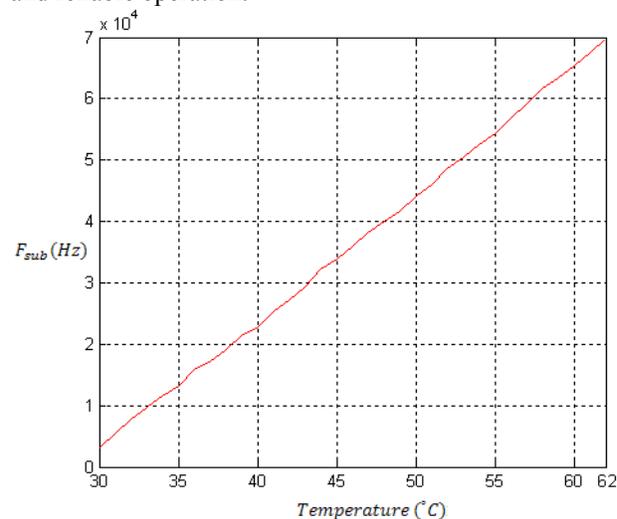


Figure 8- Variation of F_{sub} with temperature for one RO sensor

ACKNOWLEDGMENTS

The authors thank the Natural Sciences and Engineering Research Council of Canada (NSERC), Le Regroupement Stratégique en Microsystèmes du Québec (ReSMIQ) and CMC Microsystems for providing design tools, support and associated technologies. The authors thank the MITACS and Gestion TechnoCap Inc. for their financial support.

REFERENCES

- [1] B. Al-Hashimi "System-on-chip: next generation electronics", Institution of Electrical Engineers, 2006, pages 64-70.
- [2] Yeh, L. T., 1995 "Review of Heat Transfer Technologies in Electronic Equipment," ASME J. Electronic Packaging, Vol.117, pp. 333-339.
- [3] ITRS, International Technology Roadmap for Semiconductors (ITRS), 2003.
- [4] P. Gronowski et al., "High performance microprocessor design," IEEE J. Solid-State Circuits, vol. 33, pp. 676-686, May 1998.
- [5] Sung-Sik Woo; Jung-Hyup Lee; SeongHwan Cho, A ring oscillator-based temperature sensor for U-healthcare in 0.13μ CMOS, SoC Design Conference (ISOCC), pp. 548 - 551, 2009.
- [6] Xilinx corp (<http://www.xilinx.com>), January 2011
- [7] Altera corp (<http://www.altera.com>), accessed Jan 2011
- [8] Lakhssassi, A., M. Bougataya. VLSI Thermal Analysis and Monitoring. VLSI. Vienna : In-Tech, Kirchengasse 43/3, A-1070 Vienna, Austria, pp. 441-456, 2009.
- [9] Sergio Lopez-Buedo, Javier Garrido, and Eduardo I. Boemo: 'Dynamically Inserting, Operating, and Eliminating Thermal Sensors of FPGA-Based Systems', IEEE TRAN ON COM AND PACK TECHNOLOGIES, VOL. 25, NO. 4, DECEMBER 2002, pp. 561-566.
- [10] John S. Steinhart and Stanley R. Hart. Calibration curves for thermistors. Deep Sea Research and Oceanographic Abstracts, Volume 15, Issue 4, August 1968, Pages 497-503.

Silicon-Die Thermal Monitoring Using Embedded Sensor Cells Unit

Michel Saydé¹ Oussama Berriah^{1,2} Ahmed Lakhssassi¹ Mohammed Bougataya¹ Emmanuel Kengne¹ Larbi Talbi¹
 Saym01@uqo.ca oussama.berriah@uqtr.ca ahmed.lakhssassi@uqo.ca Mohammed.Bougataya@uqo.ca ekengne6@yahoo.fr Larbi.Talbi@uqo.ca

⁽¹⁾Comp. Sci. & Eng. Dept. U. du Québec en Outaouais, Canada.

⁽²⁾Elec. & Comp. Eng. Dep. UQTR, Canada.

Abstract— Thermal monitoring is essential in integrated circuit (IC) and VLSI chip which are a multilayer structure and a stack of different materials. The increase of the internal temperature of the VLSI circuits can conduct to serious thermal and also thermo-mechanical problems. Due to aggressive technology scaling, VLSI integration density as well as power density increases drastically. Thermal phenomena research activities on micro-scale level are essential for SoC and MEMS-based applications. However, various measurement techniques are needed to understand the thermal behavior of VLSI chip. In particular, measurement techniques for surface temperature distributions of large VLSI systems are a highly challenging research topic. This paper presents an algorithm and the experimental result of silicon-die thermal monitoring method using embedded sensor cells unit. Sensor implementation results and analysis are also presented.

Index Terms- Silicon Sensors, VLSI, SoC, FPGA, Thermal Analysis, GDS technique.

I. INTRODUCTION

The SoC (System on Chip) has become/became more common in the electronic industry during the last years. Complex systems are integrated on a single SoC [1]. This trend is motivated by the needs of rendering the system high efficient with high performance and more compact with less power consumption. The high integration in the SoC systems increase the power density of power consumed and dissipated in the chip. This leads to the increase of the internal temperature into the chip which can significantly reduce its performance. The decrease in the chip performance is essentially caused by the increase of the power consumption in the chip circuit and the latency. In addition, 50% of the failure in the electronic device is due to the increase of the internal temperature of the chip [2].

Due to aggressive technology scaling, VLSI integration density as well as power density increases drastically. For example, the power density of high performance microprocessors has already reached 50W/cm² at 100nm technology and it will reach 100W/cm² at 50nm technology [3]. This evolution towards higher integration levels is motivated by the needs of advanced high performance, lighter and more compact systems with less power consumption. Meanwhile, to mitigate the overall power consumption, many low power techniques such as dynamic power management [4], clock gating [5], voltage islands [6], dual V_{dd}/V_{th} [7] and power gating [8], [9] have been proposed recently. These

techniques, though helpful to reduce the overall power consumption, may cause significant on-chip thermal gradients and local hot spots due to different clock/power gating activities and varying voltage scaling. It has been reported in [10] that temperature variations of 30°C can occur in a high performance microprocessor design. The magnitude of thermal gradients and associated thermo-mechanical stress is expected to increase further as VLSI designs move into nanometer processes and multi-GHz frequencies.

Nevertheless, the growth of power density dissipated brought a number of critical thermo-mechanical problems. Thus the heat produced in the structure of the VLSI devices is directed towards its edges where it is dissipated by radiation, conduction, or convection. The principal effect of the absence of a good dynamic thermal management is the gradual and continuous degradation of the quality of performance as well as some other direct effects on the life cycle of the electronic systems [11]. Thus, an algorithm for the detection and the localization of the thermal peaks is extremely important in order to manage the thermal peak stress on high density semiconductor devices. These peaks are essentials during thermal die monitoring to avoid a critical induced thermo-mechanical stress.

One of the important questions in the field of thermal issues of VLSI systems and micro-systems is how to perform the thermal monitoring, in order to indicate the overheating situations, without complicated control circuits. Traditional approach consists of placement of many sensors everywhere on the chip, and then their output can be read simultaneously and compared with the reference voltage recognized as the overheating level.

In this paper the idea of the proposed method is to predict the local temperature and gradient along the given distance in a few places only on the monitored surface, and evaluate obtained information in order to predict the temperature of the heat source. Therefore, in the case of an high density SoC devices, there is no place on the layout for the complicated unit performing computations, but there is also no need for it, as we only want to detect the overheating situations. These peaks are essentials during thermal die monitoring to avoid a critical induced thermo-mechanical stress. Moreover, in most cases, the overheating occurs only in one place.

Hence we will presents an algorithm and the experimental result of silicon-die thermal monitoring method using embedded silicon sensor cells unit. The proposed algorithm using only two sensor cells will be formulated in a manner to facilitate the development of modular architectures using minimum silicon space in regards of their implementation in VLSI. The architecture selected will be modeled in high level languages, simulated in order to evaluate its performances, and then implemented on a FPGA (Field-Programmable Gate Array). A closed loop of simulation is used in order to evaluate the performances of the architectures proposed at each stage. The proposed architecture of the algorithm will be designed in a modular perspective after the separation of the different elementary functions of the algorithm. Hence the design of surface peaks thermal detector algorithm (SPTDA) with flexible modular-based architecture will be presented. The architecture is designed in high level languages such as MatlabTM-Simulink[®], simulated, tested using VHDL and synthesized using XilinxTM ISE [12] and AlteraTM DSP [13] tools. The simulation and hardware implementation results obtained will be compared to a finite element method (FEM) temperature prediction of the entire GDS method configuration cells.

II. THERMAL MONITORING BY GDS METHOD

SPTDA algorithm is based on the GDS method for evaluating a single heat source on the chip surface. This method principle of work is explained in details in [4] and [7]. The GDS method has been studied and analyzed for its applicability as inverse engineering problem which is capable to detect the thermal peaks and associated thermo-mechanical stress in the critical surface areas of large VLSI devices [14]. In order to obtain information about heat-source parameters we only need to know from where the phenomenon is distributed and sometimes how fast it is changing. The geometrical coordinates and an estimation of the investigated heat source can be obtained by applying the GDS method. For two sensors, A and C placed in the distance "a" (Fig. 1), the difference between their output voltages is proportional to the changes of the temperature value along the distance "a" [4]. This is true only when the heat source is directly on the line AC for any other cases the values of the angle α has to be taken into account for the proper calculation of ΔT (1). Correspondence between V, F to temperature T will depend on sensor type used (V refer to sensor voltage and F to Ring Oscillator (RO) frequency:

$$\frac{\Delta T}{\Delta r} = \frac{T_C - T_A}{a \cdot \cos \alpha} \leftrightarrow \frac{V_C - V_A}{a \cdot \cos \alpha} \quad (1)$$

Were r is the distance from the heat source.

In figure 1, we have:

$$b = \overline{AD}, \overline{AD} \leftrightarrow T_B - T_A \leftrightarrow V_B - V_A \quad (2)$$

$$b + c = \overline{AE}, \overline{AE} \leftrightarrow T_C - T_A \leftrightarrow V_C - V_A \quad (3)$$

In order to obtain the information about angle α we should apply the third sensor. In the simplest case the GDS contains only three temperature sensors placed in distance a (fig. 1).

On the basis of equations (2) and (3) and the geometrical dependencies from (fig. 1), eq. (4) is obtained:

$$\tan \alpha \leftrightarrow \frac{2}{\sqrt{3}} \cdot \left(\frac{V_B - V_A}{V_C - V_A} - \frac{1}{2} \right) \quad (4)$$

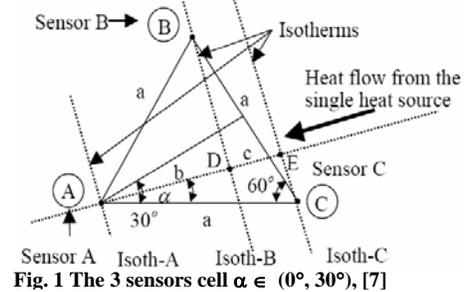


Fig. 1 The 3 sensors cell $\alpha \in (0^\circ, 30^\circ)$, [7]

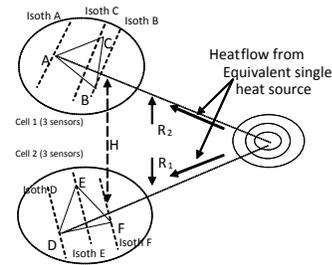


Fig. 2 Problem description and the distribution of the sensors cells

Using the cell we can obtain information on the temperature distribution and partly on the position of the heat source. In order to obtain the temperature value of a single punctual heat source we have to calculate the distance between the sensor and this source. Two sensor cells are required for this purpose (fig. 2). The cells are placed at a given distance (H) and each of them gives information about angle α (α_1 and α_2) in the direction of the heat source. Under the consideration that "a" is relatively small, we can assume that the heat source and the centre of the cells form a triangle in which the length of one side and values of the angles adjacent to this side are known. Therefore, the third angle can be deducted. This means that we can calculate the distances R1 and R2 between the heat source and the sensors. Now we can calculate the temperature gradient along the known distance. By adding it to the temperature of the sensor we obtain the temperature of the heat source. Two sensor cells A1, B1, C1 and A2, B2, C2 are placed in two corners of a monitored layout in the distance H. Hence, the temperature of the heat source can be obtained by equation (5).

$$T_s \leftrightarrow V_s = \frac{H}{a} \cdot (V_{C1} - V_{A1}) \cdot \frac{(\tan^2 \alpha_1 + 1) \cdot (\sqrt{3} + \tan \alpha_2)}{\sqrt{3} \cdot (1 - \tan \alpha_1 \cdot \tan \alpha_2) - (\tan \alpha_1 + \tan \alpha_2)} + V_{A1} \quad (5)$$

Figure 2 show the proposed distribution of the 6 sensors divided into 2 cells located implemented on the chip.

III. ALGORITHMIC DESIGN

The SPTDA algorithm flowchart is presented in figure 3. The algorithm is based on parallelism computation between the internal variables represented by $\tan \alpha_1$ and $\tan \alpha_2$. This parallelism will be used later in the architectural design in order to optimize the speed of the running implementation permuting reuse of the unite for the whole chip sequential thermal monitoring. To this end, the internal architecture of

the algorithm is divided into modular functions. Each module is responsible for a certain set of operations and sequentially passes the results to another module. The architecture has been divided in 3 principal modules. Each module has been assigned a set of calculation tasks formulated in a way to maintain a high degree of parallelism.

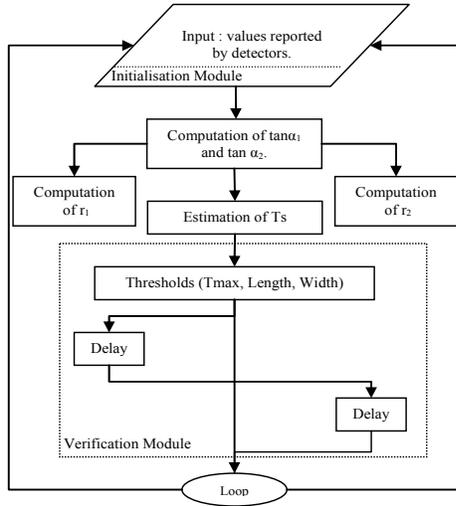


Fig. 3 SPTDA flowchart

Figure 4 represents the internal architecture of module 1 and module 2 for the computation of the first and second slopes. Elementary processes from E1 to E4 execute specific computation tasks in parallel between module 1 and 2, but these modules output produce different results. During this computation, some temporary variables are passed in parallel to module 3 which computes the temperature of the thermal peak. However module 3 works in sequence with module 1 and 2, because the final results delivered by those modules are used as input to the third module.

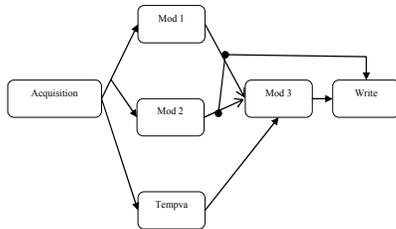


Fig. 4 SPTDA Algorithm graph

Before modeling the system with FPGA, we have to perform a quantification analysis in order to determine the combination of fixed bits that will lead to the minimum quantification error. Due to the fixed number of bits in the fixed point representation, overflow and underflow problems are serious and common. Thus, a set of simulations is required to determine the best fixed point representation. Figure 6 show that a quantification combination of a signed 11.4 and 12.4 bits had great differences with the main function; 13.4 was the turning point, and increasing the number of bits presented no great enhancement to the result. However, it is important to note that the quantification process induce a certain loss of precision and can turn on

negative temperature values. The fixed point specification has to be applied to every operation and even every constant block in a model.

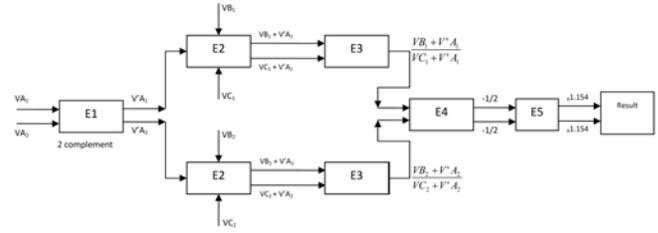
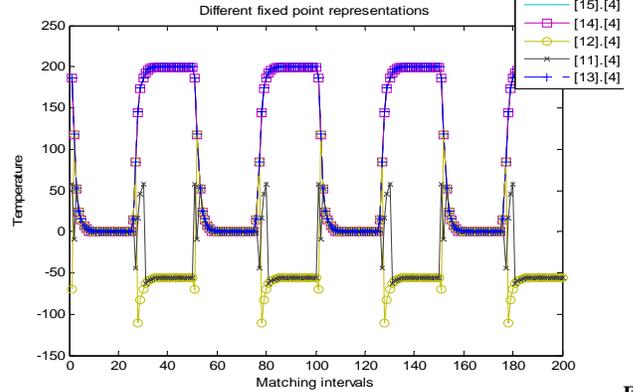


Fig. 5 Partial internal architecture (modules 1 and 2)



Fi

g. 6 Results generated using different fixed-point representations

IV. RESULTS AND DISCUSSION

In this paper, investigations are done for the simplest case, only six temperature sensors (A,B,C,D,E and F, Figure 2) in the form of two sensor cells and one single power heating source, in order to validate prediction with the 3-D FEM model. The simulations have been carried out for a one source placed at the junction surface level. As expected, the peak temperature profile is located at the centre of the heat source. The sensor cells can be placed in any way out of the monitored area (different distance H between cell 1 and cell 2), but in some cases adequate placement can simplify the thermal control unit design. In this part the results of thermal peaks can be very useful for indicating overheating situations and critical thermo-mechanical stress occurring in the device structure. Hence, table 1 display a comparison between the temperatures peaks on surface with different implementation under the same conditions. Thus, the FEM results obtained are in full agreement with the GDS predictions.

Experimental results after placement of six sensors on the FPGA chip are presented on table 2. Numerical technique, linearization and filtering were used to extract ring oscillator sensor frequency output. Furthermore, RO frequency oscillation from 50 MHz to 25 Hz reduction was used to achieve local temperature measurement in the range of 25°C to 65 °C. Appropriate range to be implemented on the module (MHz to Hz) was found. Hence, the following bits configuration (11110100001001000000) was used for calibration of frequency output (F_{out}) that can vary from 2Hz to 25MHz. Therefore configuration bits are carefully selected

to keep F_{out} away from those 2 upper and lower boundaries to avoid frequency saturation.

TABLE I: Temperature comparisons

Detectors Values			
Detector	Set 1	Set 2	Set 3
VA1	68.655	98.655	75.324
VB1	70.248	100.248	76.324
VC1	71.325	101.325	77.055
VA2	69.365	99.365	75.603
VB2	70.325	100.325	76.540
VC2	72.318	102.318	78.649
Temperature peaks estimated on surface			
Estimated by	Set 1	Set 2	Set 3
FEM	82.115	115.213	84.632
SPTDA Float	85.320	115.300	85.730
SPTDA Fix	84.060	114.100	84.190
SPTDA Altera	85.630	115.600	85.310

Table II show experimental temperature results extracted from each RO sensors for different chip heat sources.

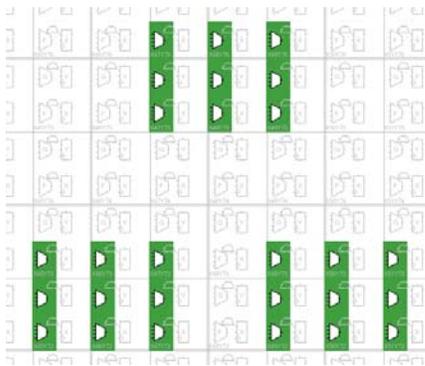


Fig. 7 RO sensors cells placement on FPGA Configurable Logic Blocs.

TABLE II: Experimental temperature results extracted from RO sensors for different chip heat sources.

RO sensor number	Chip temp. 24 °C	Chip temp. 35 °C	Chip temp. 45 °C	Chip temp. 55 °C	Chip temp. 65 °C
RO1	42.675	42.140	41.886	41.557	41.114
RO2	42.645	42.148	41.880	41.490	41.114
RO3	37.892	37.430	37.160	36.791	36.456
RO4	39.140	38.643	38.362	38.037	37.704
RO5	45.580	45.064	44.745	44.358	43.995
RO6	48.622	48.091	47.775	47.295	46.983

V. CONCLUSION

In this paper an approach to thermal and thermo-mechanical stress monitoring of VLSI chip by using embedded sensor cells unit was presented. The method for evaluating thermal peaks and temperature distribution all over the monitored area by using GDS has been shown.

The thermal peaks of investigated source can be obtained by applying the gradient direction sensors method. The adequate placement of sensors cells can accurately evaluate the thermal peaks and simplify the thermo-mechanical control unit design. That will enable the chip designer to establish the most homogeneous thermo-mechanical cartography during SoC operation. The cost of the thermal management of SoC device depends heavily upon the efficiency of the chip design. However, heat sources spatial distribution has a significant effect on the WSI, LAIC and SoC devices operation. Hence, in-situ thermo-mechanical control unite must be implemented to prevent unexpected pitfalls.

ACKNOWLEDGMENTS

The authors thank the Natural Sciences and Engineering Research Council of Canada (NSERC), Le Regroupement Stratégique en Microsystèmes du Québec (ReSMI) and CMC Microsystems for providing design tools, support and associated technologies.

REFERENCES

- [1] B. Al-Hashimi "System-on-chip: next generation electronics", Institution of Electrical Engineers, 2006, pages 64-70.
- [2] Yeh, L. T., 1995 "Review of Heat Transfer Technologies in Electronic Equipment," ASME J. Electronic Packaging, Vol.117, pp. 333-339.
- [3] ITRS, International Technology Roadmap for Semiconductors (ITRS), 2003.
- [4] E. Sergent "Thermal management handbook for electronic assemblies", 1998, Chapter 1, 1.1.
- [5] J. Oh and M. Pedram, "Gated clock routing for low-power microprocessor design," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 20, pp. 715-722, Jun 2001.
- [6] R. Puri, L. Stok, J. Cohn, D. Kung, D. Pan, D. Sylvester, A. Srivastava, and S. H. Kulkarni, "Pushing ASIC Performance in a Power Envelope," in Proc. Design Automation Conf., 2003.
- [7] A. Srivastava, D. Sylvester, and D. Blaauw, "Concurrent Sizing, Vdd and Vth Assignment for Low-Power Design," in Proc. Design, Automation and Test in Europe, vol. 1, Feb 2004, pp. 718-719.
- [8] J. Kao, A. Chandrakasan, and D. Antoniadis, "Transistor sizing issues and tool for multi-threshold CMOS technology," in Proc. Design Automation Conf., 1997.
- [9] C. Long and L. He, "Distributed sleep transistor network for power reduction," in Proc. Design Automation Conf., 2003.
- [10] P. Gronowski et al., "High performance microprocessor design," IEEE J. Solid-State Circuits, vol. 33, pp. 676-686, May 1998.
- [11] Sergio Lopez-Buedo, Javier Garrido, and Eduardo I. Boemo: "Dynamically Inserting, Operating, and Eliminating Thermal Sensors of FPGA-Based Systems", IEEE TRAN ON COM AND PACK TECHNOLOGIES, VOL. 25, NO. 4, DECEMBER 2002, pp. 561-566.
- [12] Xilinx corp (<http://www.xilinx.com>), January 2011
- [13] Altera corp (<http://www.altera.com>), accessed Jan 2011
- [14] Lakhssassi, Ahmed, Mohammed Bougataya. VLSI Thermal Analysis and Monitoring. VLSI. Vienna : In-Tech, Kirchengasse 43/3, A-1070 Vienna, Austria, pp. 441-456, 2009.

Visual Attention Mechanisms for Data-Driven Multi-Scale Descriptor Extraction in Images

Ahmed Lakhssassi, Roman Palenichka, Michel Sayde and Marek Zaremba

Université du Québec en Outaouais
Gatineau, QC, Canada,

Abstract. Novel visual attention mechanisms are described in this paper to time-efficiently extract image descriptors for object detection (retrieval) in images. It is proposed to extract multi-location image descriptors of objects of interest at maxima points of a multi-scale attention operator, which indicates locations with high intensity contrast, regional homogeneity and shape saliency. In order to reduce computational expenses while keeping a reasonable accuracy of image descriptors, a hierarchical object-driven feature-extraction procedure is developed. Experiments on the accuracy of attention-point detection have proved the operator consistency and its high potential for visual object mining in images databases.

Keywords: descriptor extraction, image mining, visual attention, attention operator, object detection.

1 Introduction

Image data mining is usually preceded by a descriptor extraction since the image matching and decision-making processes are based on object-relevant image descriptors also called object features [1]. In this paper we mostly consider the use of visual attention to extract image descriptors with the application task of object detection and retrieval. There are two major ways of image feature extraction for this particular case of image retrieval: global approach with a single vector of descriptors [1, 2] and multi-location approach with a set or a relationship graph of local descriptor vectors [3-5]. In the global, single vector approach to feature extraction, a vector of image descriptors has to contain object-relevant descriptors, which concisely represent the entire image or its fragment of interest. This approach is typical for the early stage of research on content-based image retrieval. The second approach is preferred in practice because of its high descriptive power, relational description of image objects, stability to distortions, and resistance to partial occlusions. On the other hand, the single vector of simple scalar descriptors has the advantage of direct use of metric vector distances during feature-based matching. The local descriptor vectors are usually extracted in particular image points called feature points, where objects of interest or image distinctive features are located. Therefore, the multi-location approach requires prior detection of such points based on some local image properties such as the image saliency [6-8]. The multi-location feature extraction

methods based on image local saliency can be subdivided into two categories: point-based descriptors and region-based descriptors. In the point-based descriptors, a neighborhood of fixed size is considered and descriptors are estimated within that neighborhood. The region-based approach usually supposes a regional detector is first applied to find homogeneous regions [9-11].

The use of a visual attention operator is a time-efficient solution of object recognition tasks by feature extraction in multiple image locations containing potential objects of interest [11-18]. The greater part of the attention operators is based on the image spatial differentiation and integration [6-8], while some of them involve measurements of local symmetry [17, 18]. In this paper, we consider a visual attention operator as a computational core of visual attention mechanisms (VAM) – computational strategies for time-efficient and robust image analysis to detect and recognize objects of interest. Since the object detection is mostly performed through the extraction of transformation-invariant image descriptors, we also consider VAMs as an effective method for descriptor extraction.

Some of the existing attention operators as well as our visual attention mechanisms were inspired by numerous research results and findings on the human visual system (HVS) [19-21]. Visual attention is vital for effective feature extraction, fast recognition of objects of interest and temporal events by the HVS in the conditions of distracting stimuli, occlusions, and constantly changing background of the objects of interest. The remarkable performance of the HVS and its visual learning capability makes it attractive to imitate some basic functions of the HVS in the design of computationally-efficient algorithms for object detection and content-based image retrieval.

An important characteristic of many image operators is the local scale concept and its selection method in each image point to achieve invariance to the changes in region (object) size [7, 8, 14]. The existing attention operators mostly perform the classical multi-scale image analysis based on multiple Gaussian differentiation filters at various scales as the filtering window sizes [7,14].

The prime objective in designing our VAMs is to overcome difficulties in time-efficient detection of large-scale homogeneous regions when applying the existing attention operators [12-17]. Another major drawback that we want to eliminate is the low stability of attention-point detection and hence descriptor extraction at the attention points. The VAMs proposed in this paper consist in the use of the multi-scale isotropic operator to transformation-invariant and robust descriptor extraction. The proposed operator computes the image saliency map, which is proportional to the regional contrast, homogeneity and shape saliency of image locations in the neighborhoods of attention points. It performs multi-scale image analysis based on the local scale estimation in each image point. Here, a completely different scale selection principle is adopted for image segmentation, which is based on region homogeneity and its local contrast with the adjacent regions.

Following this introductory section, Section 2 describes application of the multi-scale visual attention mechanisms to time-efficient extraction of image descriptors. The computational core of the proposed attention algorithms is the multi-component multi-scale attention (MMIA) operator, which is disclosed in Section 3. Section 4 briefly describes the descriptor extraction based on the concept of salient image disks. A descriptor-matching algorithm using the proposed feature-driven visual attention is

described in Section 5. Some experimental results of attention-point detection are presented in Section 6 and concluding remarks are given in Section 7.

2 Visual Attention Mechanisms for Descriptor Extraction and Description Learning

The VAMs we propose for descriptor extraction consist of a visual attention operator coupled with particular algorithms of multi-scale hierarchical selection of attention points and feature learning. All this is considered as a strategy for descriptor extraction in the context of object detection. Similarly to other attention operators, the object descriptors are estimated at the operator's maximum points called attention points [7-9]. Moreover, the operator values are proportional to the relevance or saliency of image fragments centered at the maximum points. The fragment size is determined by the local scale value at its center. We have adopted the scale morphological definition which is similar to the one given in the context of shape representation [22]. Here, the scale estimation consists in selecting the greatest structuring element S_r centered at (i,j) and inscribed into the central homogeneous region of the fragment. The scale value is the diameter r of the structuring element in pixels. By the descriptor-extraction problem statement, the attention operator has to perform a data-driven multi-scale attention. The data-driven definition of the visual attention as well as the required stability to distortions and invariance to eventual image transformations imply the following basic conditions in the design of this operator. Sensitivity to local regional contrast and homogeneity is necessary for a reliable feature extraction in variable conditions of image acquisition. This requirement is equivalent to the high signal-to-noise ratio condition for the stability of signal detection. Selectivity to high radial symmetry of a regional shape allows object-relevant details to be distinguished from other less relevant locations with the same high contrast and homogeneity. It also determines the local uniqueness for the attention points. The maxima of the operator have to be located on the median lines of homogeneous regions including corners as opposed to simple edges.

The flowchart of the proposed visual attention mechanisms in Fig. 1 illustrates the basic steps of descriptor extraction using the proposed attention operator. The intermediate computation results are explained on the left side of the flowchart. The multi-resolution sampling of the input image is implemented first. The objective is to select next image fragment centered at the current focus of attention (FA) point and to perform sub-sampling of original image data if necessary. The previously extracted attention points determine the fragment location and size. In particular, the attention-point coordinates at the previous (low-resolution) level provide the fragment center and the local scale determines the fragment size. The image analysis starts at the lowest resolution and the current FA point is set to the center of the entire image in order to capture its content at minimal possible number of descriptor vectors.

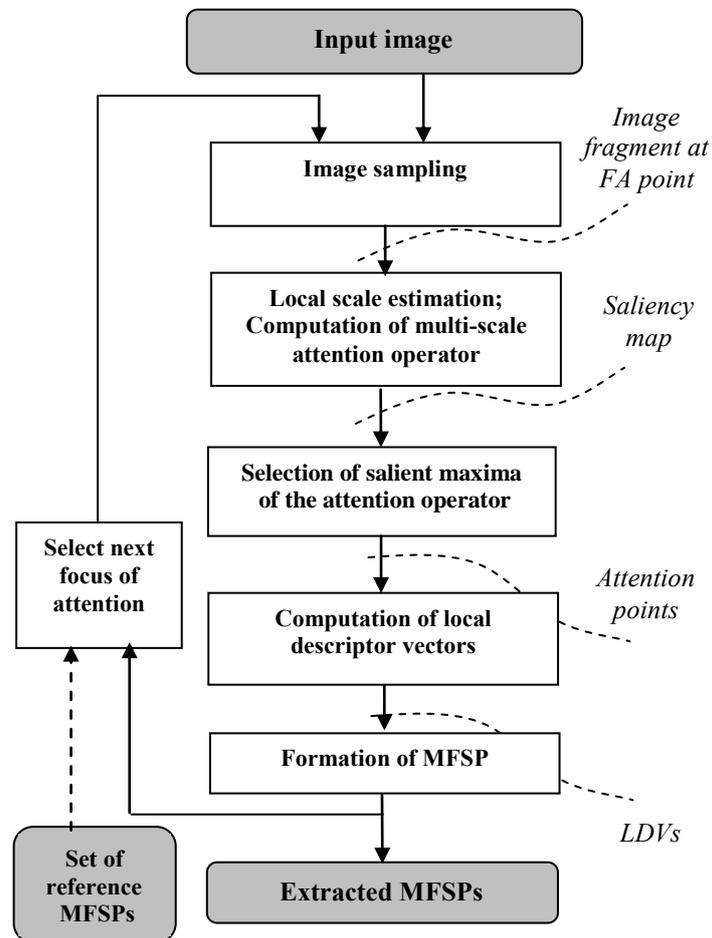


Fig. 1. Flow-chart of image descriptor extraction using the approach of multi-scale visual attention.

An important characteristic of the proposed operator is the local scale concept. Although the multi-scale descriptor extraction is related to image resolution concept, it is different from the multi-resolution approach to image analysis [22, 23]. The main difference between the multi-resolution image analysis and the multi-scale approach is that the image is sequentially analyzed on various resolution levels (scales) starting from the coarse one. On the contrary, the multi-scale analysis is performed once but at different local scales simultaneously, which depend on the image locations, i.e. the local scale is a function of the image point. In our method, the local scale is determined for each pixel of the image fragment by maximizing the local isotropic

contrast with regional homogeneity constraint (Section 3). An obvious advantage of the multi-scale image analysis for object detection is that it operates at the highest resolution. On the contrary, the existing attention operators mostly perform the classical multi-scale image analysis based on multiple Gaussian filters at various scales as the filtering window sizes [7, 8, 14].

The intermediate results of the local scale estimation are used to time-efficiently compute the attention operator (Section 3). The consecutive local maxima of the MMIA operator determine the locations of candidate attention points. The candidate attention points are further tested on their saliency level. If the image fragment centered at the candidate attention point passes the testing, the local descriptors are computed based on intensity properties of the current fragment (Section 4). They form a local descriptor vector (LDV) associated with the current attention point, which also is the feature point. Finally, all the LDVs of a current fragment are associated into a multi-location feature-saliency pattern (MFSP) at a given hierarchical level of image analysis. The association includes the descriptor transformation for each LDV in order to make them transformation invariant (Section 4). After the MFSP extraction at the current level, the same procedure is implemented for all the attention points of the current level through the sequential selection of the next FA point (Fig. 1). The analysis is terminated after the total number of MFSP exceeds a threshold for the description length. This is a data-driven VAM since no prior information on the objects of interest is involved in the descriptor extraction process. However, an object-drive version of this VAM can also be used to optimize the selection of attention points and object-relevant descriptors. In particular, we propose to compare the extracted description with a reference one in order to determine ambiguity in the MFSP description for image objects belonging to different classes (Fig. 1).

3 Multi-Scale Isotropic Attention Operator

The multi-scale isotropic operator of the VAMs proposed in this article is a generic attention operator, which assumes local maxima in image fragments of a variable size at the center of homogeneous regions with a high contrast and regional shape saliency. It represents a modified and improved version of another similar operator [25]. The main improvement is the possibility to handle color and, eventually, multi-component images. This implies a new selection procedure for local scale determination. Computationally, the proposed operator called MMIA (multi-component multi-scale isotropic attention) operator is the most complex part of the attention algorithm in Fig. 1 and consists of a weighted sum of P local contrast-minus-variance terms with the weight coefficients $\{\beta_p\}$:

$$\Phi[f(i, j), \rho(i, j)] = \sum_{p=1}^P \beta_p \cdot [c(i, j, p, \rho) - \alpha \cdot d(i, j, p, \rho)] \quad (3.1)$$

where $f(i,j)$ is the input image fragment, $c(i,j,p,\rho)$ the local isotropic contrast, $d(i,j,p,\rho)$ is the local mean deviation as the measure of non-homogeneity for the p th component and local scale ρ , and α is the homogeneity coefficient [25]. The first term in Eq. (3.1) is defined in such a way that it is proportional to both local regional contrast and regional shape symmetry [25]. The second term is a measure of local non-homogeneity [25]. The local scale $\rho(i,j)$ in Eq. (3.1) is differentially estimated in each image point:

$$\rho(i,j) = \arg \max_{1 \leq r \leq R} \max_{1 \leq p \leq P} \{c(i,j,p,r) - \alpha \cdot d(i,j,p,r)\} \quad (3.2)$$

The initial (non-optimized) values for the coefficients in Eq. (3.1) are: $\{\beta_p=1, \text{ for } \forall p\}$ and $\alpha=1$. The optimized value for the non-homogeneity coefficient α can be determined by applying the maximum likelihood rule with respect to the two terms in Eq. (3.2) while assuming they follow exponential distributions with different parameters [25]. Coefficients $\{\beta_p\}$ can also be optimized in order to increase the relevance of descriptor extraction relatively to a given type of regions and objects of interest. This can be made by assigning large weight coefficients to the components, which have high contrast and homogeneity for the objects of interest. The linear perceptron (back-propagation) algorithm can be used for this purpose since the MMIA operator is a linear function, which maxima represent the attention points for a given type of the regions of interest [1].

As a result of the MMIA computation, a level of saliency w_k is assigned to each image point, which is defined as the product of the local scale value and contrast-to-homogeneity ratio at the attention point $(i,j)_k$:

$$w_k(i,j) = \rho_k(i,j) \cdot c_k(i,j) / d_k(i,j), \quad (3.3)$$

In this equation, the level of saliency is directly proportional to the local scale, isotropic contrast and inversely proportional to the local variance. Since the value of the local isotropic contrast is proportional to both the local contrast and radial symmetry, the saliency level will also be proportional to the radial symmetry. It can be easily proved that the definition of the saliency level in Eq. (3.3) is also invariant to a linear (affine) transformation of the image intensity.

The first K consecutive maxima points $\{(i,j)_k, k=1, \dots, K\}$ of the function $\Phi[f(i,j), \rho(i,j)]$ in Eq. (3.1), which saliency level in Eq. (3.3) exceeds a given saliency threshold, are chosen as the attention points for subsequent descriptor extraction.

4 Extraction of Transformation-Invariant Descriptors

In this section, we explain how the proposed VAMs can effectively be used for image descriptor extraction. The detail description of particular descriptors and their computational analysis is out of scope for the present paper since we concentrate here on the VAM implementation rather than on particular descriptors. The underlying

idea for the MFSP method of image description is to represent the entire image or its particular fragment by a set of K LDVs $\{\xi_k\}$, where $\xi_{k,l}$ denotes the l th descriptor in the k th vector related to k th attention point $(i,j)_k$. Each vector ξ_k consists of L heterogeneous scalar descriptors, which concisely describe one salient image fragment that is a circular neighborhood of $(i,j)_k$ with the diameter $\rho_k(i,j)$. The descriptor extraction is based on the concept salient image disks (SIDs) introduced for the task of control point extraction to perform image matching and registration [25]. Our concept of SIDs is somewhat similar to the SIFT method of image description in a neighbourhood of key-points [14]. However, the SID definition is a more general concept as a model-based approximation of image intensity, which is applicable to multi-component images. It is also a constructive method since it provides the way to extract SIDs from gray-level images without the image pre-segmentation. In this context, it should be noted that the locality of a SID fragment description by the LDV is variable and in the general case can even be comparable with the image size.

Assuming the image model of high-contrast homogeneous regions, the local maxima of the MMIA operator correspond to the SID locations. The constraint of homogeneity keeps the circular SID fragment inscribed into a current homogeneous region. Therefore, the proposed MMIA operator can be applied to detect SID location and extract the LDVs, which are associated into MFSPs by their spatial proximity.

Formally, the MFSP is defined as a set of LDVs corresponding to the SIDs located in a neighborhood of a current attention point, which is also the feature point. Although the MFSP is quantitatively defined by all its LDVs, it has some integral characteristics, which are involved in the proposed VAMs. The MFSP is characterized by the *centroid* (X,Y) , *dominant scale* $\rho(X,Y)$ and *rotation angle* $\theta(X,Y)$. We select the MFSP centroid among the center coordinates of all the K LDVs by the maximum of the saliency level:

$$(X,Y) = \arg \max_{(i,j)_k} \{w_k(i,j)\} \quad (4.1)$$

In this case, the center of the SID under consideration is selected as the centroid and the surrounding SIDs in a neighborhood of it form the current MFSP. The dominant scale $\rho(X,Y)$ of the MFSP is the largest local scale among all the SIDs. In order to estimate the rotation angle, the saliency levels of all the SIDs are considered without the central SID. Their centroid called the MFSP *dominance point* (x_d, y_d) is determined as follows:

$$x_d = \frac{\sum_k w_k \cdot |i_k - X| \cdot i_k}{\sum_k w_k \cdot |i_k - X|} \quad \text{and} \quad y_d = \frac{\sum_k w_k \cdot |j_k - X| \cdot j_k}{\sum_k w_k \cdot |j_k - X|}, \quad (4.2)$$

where $w_k \cdot |i_k - X|$ is the saliency moment for the k th SID centered at $(i,j)_k$ with respect to (X,Y) . The dominance point indicates the direction with the most concentration of saliency due to the use of the weight coefficients in Eq. (4.2). The MFSP rotation

angle is determined by the angle of the line connecting the MFSP centroid (X, Y) and the dominance point (x_d, y_d) .

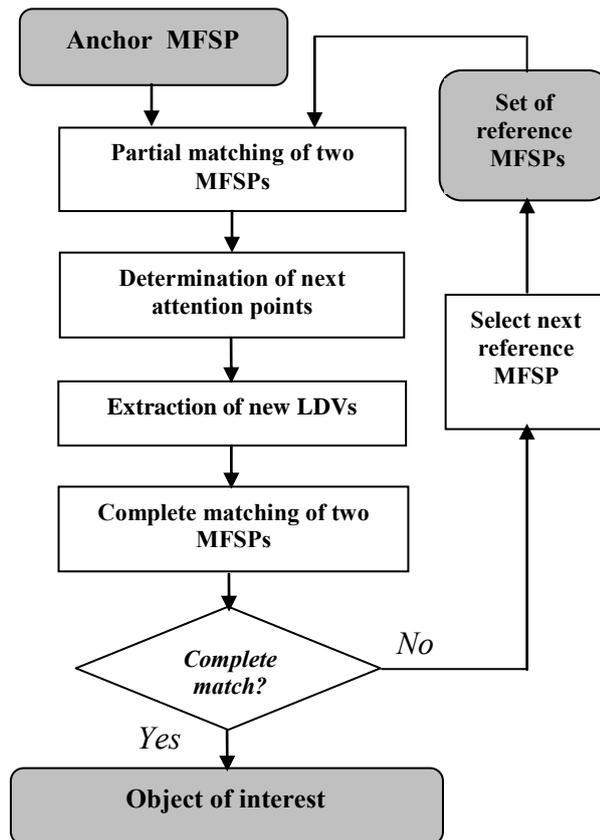


Fig. 2. Flow-chart of image object detection using the object-driven VAM.

The next step in the formation of the MFSP is the invariance transformation of the LDV pose descriptors in order to make them invariant to image similarity transformations (translation, scaling and rotation). The transformation-invariance is achieved by the rotation of the initial coordinates on the MFSP on the rotation angle $\theta(X, Y)$ with their shift to the center (X, Y) and followed by the division on the maximum of the shifted coordinates. Finally, the intensity descriptors for each LDV are linearly transformed to make them invariant to the intensity affine transformations. Similarly to the pose coordinates, the intensity descriptors are shifted on the intensity value for the central LDV and divided by the maximum shift over all the LDVs. The regional shape features of individual LDVs are not included into the

MFSP for the purpose of description conciseness since they are somewhat redundant. In particular, the LDVs located in a close neighbourhood of the central SID also determine the regional shape since they are adjacent to the central LDV.

5 Object-driven VAMs using feature-saliency patterns

A disadvantage of the data-driven VAM is the non-optimal number of attention (feature) points and hence the total number of LDVs to describes different objects in images for object detection and content-based image retrieval. This can be overcome in the case when the reference MFSPs already exist and are associated with particular object classes. The object-driven attention is a computationally efficient method of feature extraction, which also is based on the proposed MMIA operator. Since each MFSP contains information of the attention point coordinates and the local scale (Section 4), it determines the next attention points provided the task of object detection of a particular class have to be executed. For this purpose, a MFSP of a small size – also called the *anchor sub-pattern* of a given MFSP – is extracted by taking into account only first M ($M < K$) attention points (Fig. 2). According to the MFSP definition, the MFSP of minimal size contains only two LDVs associated with two most salient attention points. However, it is recommended to use more than two attention points in order to avoid ambiguities. These attention points are determined by a data-driven VAM described in Section 2. Matching the anchor MFSP with the reference MFSP using the IMD measure of similarity gives the coordinates of next attention points [25]. The result of the matching is a sub-pattern of the current reference MFSP and new feature points, which are not included in the sub-pattern. These non-included feature points determine the centers of image fragments where the search for the salient maxima of the MMIA operator is performed on the next step (Fig. 2). The LDVs corresponding to the new attention points are sequentially estimated and the complete matching of the two MFSPs is performed. The object-driven VAM in Fig. 2 is applied to all the reference MFSPs and final decision is made by the most similar reference MFSP to the extracted one, which contains the anchor sub-pattern.

6 Experimental Results

6.1 Stability and invariance of attention-point extraction

The goal of the experiments was to verify the stability of the proposed VAMs and to validate them while solving the descriptor extraction task (Section 4). The stability and transformation-invariance of the attention-point extraction was tested by the accuracy of the attention-point localization and the local scale estimation when images were subjected to different transformations of scaling and rotation. The

operator invariance to image translations is obvious. Since real image sequences cannot provide reference positions for attention-point coordinates and local scales, we have used semi-synthetic image sequences. Consecutive images of the test sequence are created by scaling or rotating one initial real image as the first image of the sequence. This method provides relative reference values for the attention-point coordinates and local scales, which allowed us to estimate the error in the attention-point extraction.

The first type of experiments was testing the local scale estimation in a scaling test sequence with the same image object. An example of a semi-synthetic sequence of scaling transformations is shown in Fig. 3 with the attention points extracted by the MMIA operator. Each image of the sequence is formed by a sub-sampling of the previous image with a constant scaling factor to model image scaling transformations. In this way, we were able to evaluate the accuracy of the MMIA operator with regards to local scale estimation and the scaling adaptation as its main stability characteristic. The first six attention points were taken into account for the error estimation (Fig. 3). The results of this testing is the average error of the scaling factor estimation (Fig. 4). Obviously, the error estimation was performed only for the same attention locations since some images in the sequence may have significantly different attention points. The following weight coefficients have been set in the MMIA operator in Eqs. 3.1 and 3.2: $\alpha=1.0$ and $\{\beta_p=1 \text{ for } \forall p\}$. The three normalized color components have been involved in the experiments, i.e., $P=3$ in Eq. (3.1). In this experiment, all the coefficients $\{\beta_p\}$ are equal because there are no color preferences for the objects of interest in the data-driven VAMs. The maximal scale size was approximately equal to $1/3$ of the image size in order to detect large homogeneous regions and 22 scales have been involved in the detection.

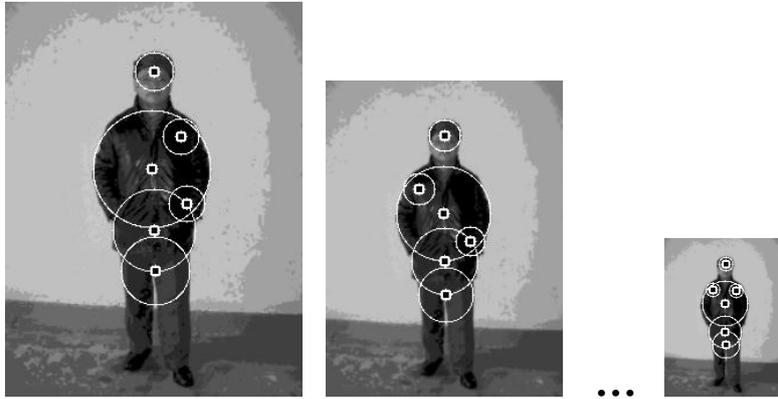


Fig. 3. Example of a test image sequence with scaling transformations.

For comparison, the existing DoG operator in the SIFT method of image descriptor extraction [14] was implemented and tested for its accuracy on the same

semi-synthetic image sequence. The spatial local scale selection in the SIFT method is different by definition from that used in the MMIA operator, therefore only the stability of local scale selection of the two methods can be compared (Fig. 4). In particular, the MMIA method uses two basic tokens – isotropic contrast and regional homogeneity – in the local scale estimation that provides better stability as compared to the DoG operator performance [14]. The DoG operator used only one token – spatial differentiation as a difference of two Gaussians.

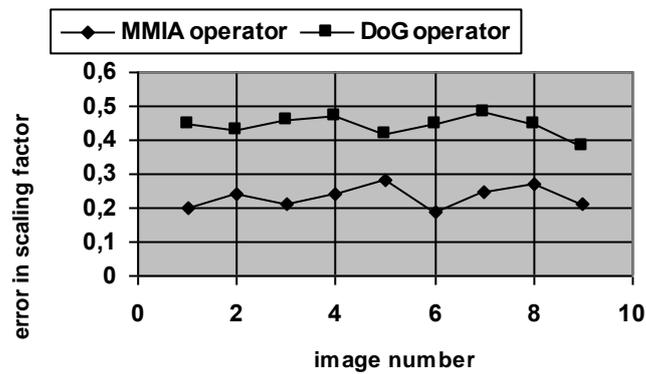


Fig. 4. Accuracy of the local scale estimation in the attention-point extraction.

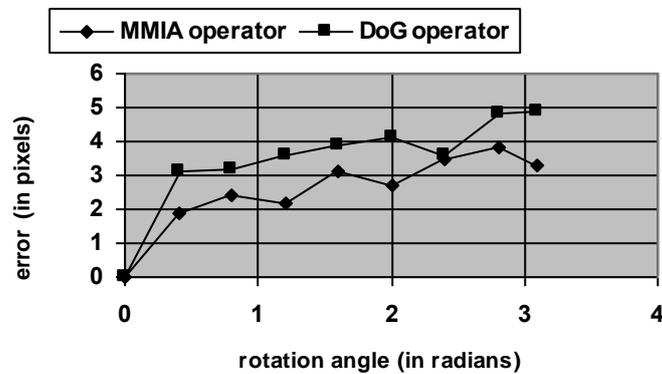


Fig. 5. Coordinate accuracy in the attention-point extraction.

Similar experiments on the accuracy of coordinate determination and rotation-invariance of attention points have been carried out by generating a sequence of rotated images from a single initial image. Each image of the sequence is obtained by

rotating the previous one on a constant rotation angle to model image rotation transformations. The results of the coordinate accuracy testing in the attention-point extraction are given in Fig. 5. The parameters of the MMIA operator are the same as in the previous example of scaling transformations.

Another testing of the proposed VAMs is their stability with respect to random intensity changes using the first image of the test sequence in Fig. 3. The level of random distortions is characterized by the contrast-to-noise ratio where the noise level is its mean square deviation. The final error estimate in the attention-point coordinates is averaged over single-point errors from 6 first local maxima of the MMIA operator (Fig. 6).

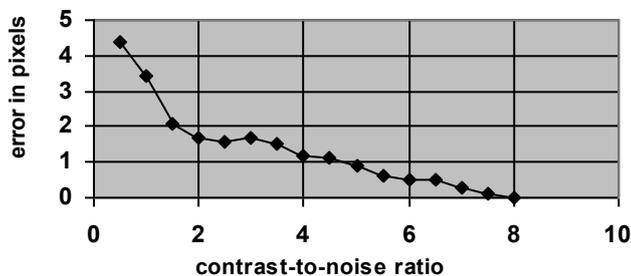


Fig. 6. Evaluation of the MMIA operator performance in the localization of attention points.

6.2 Application to object retrieval in image databases

The goal of the experiments was the validation of the descriptor extraction in the task of object detection. A search for a given hand image in a database of hand images of different persons has been performed using the object-driven VAM algorithm described in Section 5.

Fig. 7 illustrates the object matching by the object-driven VAM algorithm of an observed image with a reference one from a database of hand images. The observed image is obtained by the same scanner as the reference one, but with different hand positions, scaling factors, rotation angles and eventual non-rigid transformations because flexible finger positions are allowed. Positions and local scales of the LDVs on the initial images are indicated by white circles (Fig. 7 a and b). The MMIA operator was implemented with the same parameters as described in Section 6.1. In this experiment, each LDV contained $L=15$ descriptors. Normalized mean intensity, local contrast and local variance are the intensity descriptors in the LDVs for the three color components. Fig. 7 shows the pair of best-matched anchor MFSPs from the two images found by the IMD algorithm [25]. The result of image matching while supposing a similarity transformation between the two images is shown by their four-quadrant superposition in Fig. 7c.

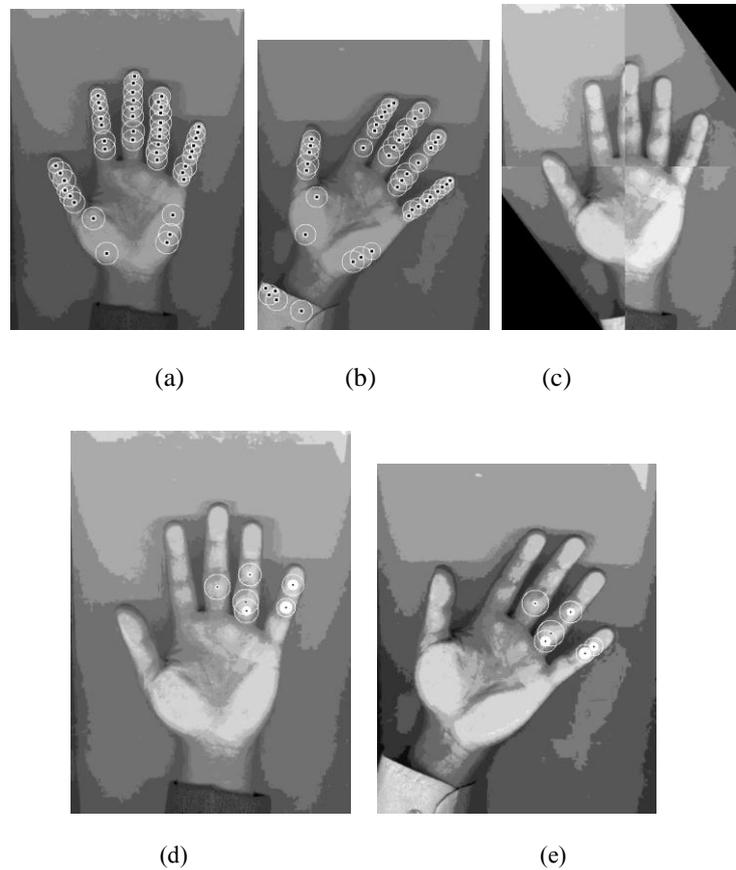


Fig. 7. Matching hand images using the proposed object-driven VAM algorithm: (a) and (b) different hand images of the same person; (c) result of object matching; (d) and (e) two best-matched anchor MFSPs.

An appropriate measure of the VAM performance will be the object detection (retrieval) precision, Q_p , conventionally defined as the number of correct object detection divided by the total number of objects detected by the VAM (includes also false matches) [2]. Another, complementary characteristic of the performance in object-relevant image mining is recall Q_r , which is a ratio of the number of correct object detection (retrieval) divided by the total number of the present objects of interest (includes also false rejections). The conducted experiments for the given database of hand images gave the following values for these two measures: $Q_p=0.93$ and $Q_r=0.91$.

7 Conclusions

A novel computational model of VAMs is proposed to effectively extract image descriptors for object detection in images. The computational core of the VAMs is the multi-scale attention operator MMIA, the local maxima of which indicate image fragments of variable size with high intensity contrast, regional homogeneity and shape saliency. The conducted experiments confirmed the stability of attention-point extraction by the MMIA operator and their invariance to image transformations of translation, scaling and rotation.

We have also explained how the proposed VAMs can effectively be used to provide concise image descriptors for object-relevant image mining. An object-driven attention is also proposed as an extension of the data-driven VAMs to enhance the image descriptor relevance and reduce the descriptive length at the same time. The conducted experiments on the accuracy of descriptor extraction have proved the operator reliability and its high potential in image mining.

References

1. R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, New York: John Wiley & Sons, 2001.
2. T. Evgeniou, *et al.*, "Image representations and feature selection for multimedia database search", *IEEE Trans. KDE*, Vol. 15(4), pp. 911-920, 2003.
3. C. Schmid and R. Mohr, "Local gray-value invariants for image retrieval", *IEEE Trans. PAMI*, Vol. 19(5), pp. 530-535, 1997.
4. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L.V. Gool, "A comparison of affine region detectors", *Int. J. Computer. Vision*, Kluwer Academic Publishers, 65, pp. 43-72, 2005.
5. X. Jiang, A. Munger, and H. Bunke, "On median graphs: properties, algorithms and applications", *IEEE Trans. PAMI*, 23(10), pp. 1144-1151, 2001.
6. N. Sebe and M.S. Lew, "Comparing salient point detectors", *Pattern Recognition Letters*, 24(1-3), pp. 89-96, 2003.
7. T. Lindeberg, "Detecting salient blob-like image structures and their scale with a scale-space primal sketch: A method for focus of attention", *Int. J. of Computer Vision*, Vol. 11, pp. 283-318, 1993.
8. T. Kadir and M. Brady. "Saliency, scale and image description", *Int. J. of Comp. Vision*, Vol. 45(2), pp. 85-103, 2001.
9. R. Palenichka, A. Lakhssassi and M. Zaremba, "Transformation-invariant extraction of multi-location image features from remote sensing imagery", *Proc. IEEE Symposium IGARSS 2010 (Honolulu, USA)*, pp. 2471-2474, 2010.
10. J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," *Proc.13th British Machine Vision Conf.*, pp. 384-393, 2002.
11. L. Itti, C. Koch and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis", *IEEE Trans. PAMI*, Vol. 20(11), pp. 1254-1259, 1998.
12. T. Lindeberg, "Detecting salient blob-like image structures and their scale with a scale-space primal sketch: A method for focus of attention", *Int. J. of Comp. Vision*, Vol. 11, pp. 283-318, 1993.

13. C. Harris and M. Stephens, "A combined corner and edge detector", *Proc. 4th ALVEY Vision Conference*, pp. 147-151, 1988.
14. D.G. Lowe, "Distinctive image features from scale-invariant key-points", *Int. J. of Comp. Vision*, Vol. 60(2), pp. 91-110, 2004.
15. H. D. Tagare, K. Toyama, and J.G. Wang, "A maximum-likelihood strategy for directing attention during visual search", *IEEE Trans. PAMI*, Vol. 23(5), pp. 490-500, 2001.
16. P.J. Burt, "Attention mechanisms for vision in a dynamic world", *Proc. 9th International Conference on Pattern Recognition*, Vol. 2, pp. 977-987, 1988.
17. D. Reisfeld, H. Wolfson and Y. Yeshurun, "Context-free attentional operators: the generalized symmetry transform", *Int. J. of Comp. Vis.*, Vol. 14, pp. 119-130, 1995.
18. G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Trans. PAMI*, Vol. 25(8), pp. 959-973, 2003.
19. A. L. Yarbus, *Eye Movements and Vision*, Premium Press, New York, 1967.
20. A.M. Treisman and G.A. Gelade, "A Feature-integration Theory of Attention", *Cognitive Psychology*, Vol. 12, pp.97-136, 1980.
21. P. Maragos, "Pattern spectrum and multi-scale shape representation", *IEEE Trans. PAMI*, Vol. 11(7), pp. 701-717, 1989.
22. P.J. Burt, *The Pyramid as Structure for Efficient Computation*. In: *Multiresolution Image Processing and Analysis*, Springer-Verlag, pp. 6-35, 1984.
23. V. Conception and H. Wechsler, "Detection and localization of objects in time-varying imagery using attention, representation and memory pyramids", *Pattern Recognition*, Vol. 29(9), pp. 1543-1557, 1996.
24. R. Palenichka and M. Zaremba, "Automatic extraction of control points for the registration of optical satellite and LiDAR images", *IEEE Trans GRS*, 48 (7), pp. 2864-2879, 2010.

Références

- [1] S. M. Hu, "Stress-related problems in silicon technology," *Journal of Applied Physics*, vol. 70, pp. R53-R80, 1991.
- [2] S. D. Senturia, *Microsystem Design*: Springer, 2000.
- [3] C. Liu, *Foundations of MEMS*: Pearson Prentice Hall, 2006.
- [4] (23 Avril 2013). Available: <http://tel.archives-ouvertes.fr/tel-00158019>
- [5] E. Boemo and S. Lopez-Buedo, "Thermal verification on FPGAs," in *NORCHIP Conference, 2005. 23rd*, 2005, pp. 48-53.
- [6] W. Sung-Sik, L. Jung-Hyup, and C. SeongHwan, "A ring oscillator-based temperature sensor for U-healthcare in 0.13µm CMOS," in *SoC Design Conference (ISOCC), 2009 International*, 2009, pp. 548-551.
- [7] J. J. L. Franco, E. Boemo, E. Castillo, and L. Parrilla, "Ring oscillators as thermal sensors in FPGAs: Experiments in low voltage," in *Programmable Logic Conference (SPL), 2010 VI Southern*, 2010, pp. 133-137.
- [8] S. R. Boyle and R. A. Heald, "A CMOS circuit for real-time chip temperature measurement," in *Compcon Spring '94, Digest of Papers.*, 1994, pp. 286-291.
- [9] V. Szckely, M. Rencz, A. Pahi, and B. Courtois, "Thermal monitoring and testing of electronic systems," *Components and Packaging Technologies, IEEE Transactions on*, vol. 22, pp. 231-237, 1999.
- [10] M. Tie and X. Cheng, "A cross-coupled-structure-based temperature sensor with reduced process variation sensitivity," *Journal of Semiconductors*, vol. 30, p. 045002, 2009.
- [11] C. Wen-Teng, L. Jian-An, and L. Ming-Feng, "Stress-induced capacitance approximation using ring oscillator delay," in *Next-Generation Electronics (ISNE), 2010 International Symposium on*, 2010, pp. 124-127.

- [12] F. Zito, F. Aquilino, L. Fragomeni, M. Merenda, and F. G. D. Corte, "CMOS wireless temperature sensor with integrated radiating element," *Sensors and Actuators A: Physical*, vol. 158, pp. 169-175, 2010.
- [13] J. S. Steinhart and S. R. Hart, "Calibration curves for thermistors," *Deep Sea Research and Oceanographic Abstracts*, vol. 15, pp. 497-503, 1968.
- [14] M. Bougataya, A. Lakhssassi, and D. Massicotte, "Steady State Thermo-mechanical Stress Prediction for Large VLSI circuits using GDS Method," in *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on*, 2006, pp. 1205-1209.
- [15] 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.
- [16] 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.
- [17] X. Cao and F. Deravi, "An efficient method for multiple-circle detection," in *Computer Vision, 1990. Proceedings, Third International Conference on*, 1990, pp. 744-747.
- [18] R. Palenichka, A. Lakhssassi, and M. Zaremba, "Spatiotemporal attention operator using isotropic contrast and regional homogeneity," *Journal of Electronic Imaging*, vol. 20, pp. 023018-023018, 2011.
- [19] P. H. D. Rafael C Gonzalez and R. E. Woods, *Digital Image Processing, Third Edition*: Prentice Hall, 2008.
- [20] R. C. A. GONZALEZ, R. E. A. WOODS, and S. L. A. EDDINS, *Digital image processing: using Matlab*: Prentice Hall, 2004.
- [21] C. Watkins, A. Sadun, and S. Marenka, *Modern image processing: warping, morphing, and classical techniques*: Academic Press, 1993.
- [22] C. G. Relf, *LabVIEW*: Taylor & Francis Group, 2004.
- [23] H. Maître, J. Zinn-Justin, and U. J. Fourier, *Les progrès du traitement des images*: Elsevier, 1996.

- [24] G. Blanchet and M. Charbit, *Digital signal and image processing using Matlab*: ISTE Ltd., 2006.
- [25] W. K. Pratt, *Digital Image Processing: PIKS Scientific Inside*: Wiley, 2007.
- [26] A. C. Bovik, *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*: Academic Press, Inc., 2005.