



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΚΑΙ ΔΙΚΤΥΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΙΤΛΟΣ: *«Υλοποίηση αλγορίθμου χωροταξικής διευθέτησης προτύπων κελιών με εναισθησία στη θερμοκρασία»*

ΚΟΤΣΙΡΑ ΜΑΡΙΑ

ΕΠΙΒΛΕΠΩΝ:

Σταμούλης Γεώργιος , Καθηγητής

Βόλος 2012

Περιεχόμενα

Ευχαριστίες.....	3
1.Περίληψη.....	4
2.Εισαγωγή	5
3. Διατύπωση της προσέγγισης	7
3.1 Ανάλυση του 1. Τεχνική της πολλαπλής αντανάκλασης.....	7
3.2.1 Ανάλυση του 2. Thermal force model	9
3.2.2. Thermal force placement algorithm.	11
3.3. Αξιοπιστία αλγορίθμου	14
3.4. Πολυπλοκότητα χώρου.	14
3.5. Πολυπλοκότητα χρόνου	14
3.6. Ακρίβεια/ Ορθότητα και όρια του TFPA	15
4. Υλοποίηση αλγορίθμου TFPA σε C.	16
4.1 Επεξήγηση κώδικα	16
4.2 Αποτελέσματα	22
5. Παραδείγματα και αποτελέσματα υπολογισμών.....	29
5.1 Παραδείγματα ενιαίων chips (uniform chips)	30
5.2 Παραδείγματα με chips που έχουν διαφορετικές θερμικές αγωγιμότητες.....	31
5.3 Παραδείγματα με άνισα chips.	33
6. Συμπέρασμα	36
7.Παράρτημα.....	36
7.1. Υπολογισμός του $P_{xc}(\alpha)$	38
7.2. Υπολογισμός του $P_{xr}(\alpha)$	39
7.3. Υπολογισμός του αθροίσματος $ \sum_{a=6} Pz(\alpha)/fx_{i,j} $	41
8. Παραπομπές.....	42

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή και επιβλέποντα της διπλωματικής μου κ. Γεώργιο Σταμούλη για την καθοδήγηση και τη βοήθειά του για την ολοκλήρωση της εργασίας αυτής. Όπως επίσης και τον καθηγητή μου κ. Νέστωρα Ευμορφόπουλο που δέχτηκε να επιβλέψει τη διπλωματική μου εργασία.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τον Αντώνη Δαδαλιάρη για την πολύτιμη βοήθειά του από το ξεκίνημα της διπλωματικής μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την υπομονή και τη στήριξή τους όλα αυτά τα χρόνια αλλά και τους φίλους μου που μου συμπαραστάθηκαν και ήταν και είναι κοντά μου σε όλες τις σημαντικές στιγμές της ζωής μου.

1.Περίληψη

Η μελέτη που ακολουθεί αναφέρεται στη δημοσίευση του κ. Jing Lee με θέμα «Thermal Placement Algorithm Based on Heat Conduction Analogy». Πρόκειται για έναν αλγόριθμο χωροταξικής διευθέτησης κελιών με ευαισθησία στη θερμοκρασία, που ονομάζεται TFPA και προτείνεται για τον σχεδιασμό MCM (Multi-chip modules) [1]. Το MCM συχνά αναφέρεται ως “chip” στον σχεδιασμό, απεικονίζοντας έτσι την ολοκληρωμένη φύση του. Ο αλγόριθμος TFPA αρχίζει με τον μετασχηματισμό του πραγματικού υποστρώματος σε ένα υπόστρωμα με άπειρο αριθμό “chips”. Στη συνέχεια, κάθε chip ωθεί το άλλο chip με δύναμη ανάλογη της θερμικής αγωγιμότητας. Έτσι, κάθε chip θα κινηθεί στην κατεύθυνση της δύναμης, μέχρι στο σύστημα να επέλθει ισορροπία. Ο αλγόριθμος TFPA παράγει αποτελέσματα υψηλής ποιότητας ως προς την τοποθέτηση των κελιών και διατηρεί ένα ψυχρότερο και ομοιόμορφο προφίλ ως προς την θερμοκρασία. Αντίθετα με τους συμβατικούς “force-directed” αλγόριθμους, που ενδέχεται να έχουν προβλήματα επικάλυψης (overlapping problems), ο TFPA με τον τρόπο που τοποθετεί τα chips, δεν προκύπτει κάποια επικάλυψη ή αν προκύψει είναι πραγματικά ελάχιστη κι αυτό επειδή τοποθετεί τις πύλες χωριστά. Στην πράξη, οι αρχικές τοποθετήσεις που γίνονται με βάση τον TFPA, είναι πολύ κοντά στις τελικές θέσεις.

2.Εισαγωγή

Η τάση στη μικροηλεκτρονική ορίζει τα chips να σχεδιάζονται κοντά το ένα στο άλλο στα MCMs, προκειμένου να ελαχιστοποιηθούν οι καθυστερήσεις διάδοσης και ο όγκος. Ωστόσο, αυτή η τάση έχει ως αποτέλεσμα την υψηλότερη ροή θερμότητας στο υπόστρωμα. Επιπρόσθετα, ακόμα υψηλότερες θερμοκρασίες μπορεί να προκύψουν, αν η θερμότητα που διαχέεται δεν εξαλειφθεί σωστά. Μια υψηλότερη θερμοκρασία, δεν επηρεάζει μόνο την απόδοση του κυκλώματος, αλλά μειώνει και την αξιοπιστία του. Αν η ενέργεια διανεμηθεί άνισα, προκύπτουν τα λεγόμενα “hot spots”(Fig.1), τα οποία μπορεί να επιφέρουν θερμικές εντάσεις. Αν αυτές οι εντάσεις είναι σοβαρές και συνεχιστούν για αρκετούς κύκλους, τότε μπορεί να οδηγήσουν σε αποτυχία.

Δεδομένου ότι σε πολλά MCMs, η θερμική διαχείριση είναι ο πιο σημαντικός παράγοντας αξιοπιστίας, θα πρέπει να λαμβάνεται όσο πιο νωρίς γίνεται υπόψη στην διαδικασία σχεδιασμού (packaging process). Το καλύτερο στάδιο, για να γίνει αυτό, είναι αυτό της τοποθέτησης των κελιών, καθώς οι κατανομή της θερμοκρασίας εξαρτάται άμεσα από τα αποτελέσματα της χωροθέτησης. Ένα εργαλείο χωροθέτησης, που δεν έχει λάβει υπόψη τη θερμοκρασία, θα μπορούσε να τοποθετήσει κάποια chips με υψηλή θερμοκρασία πολύ κοντά το ένα στο άλλο και αυτό θα δημιουργούσε ένα “hot spot” στο υπόστρωμα, ακόμα και αν η συνολική κατανάλωση ενέργειας ήταν περιορισμένη.

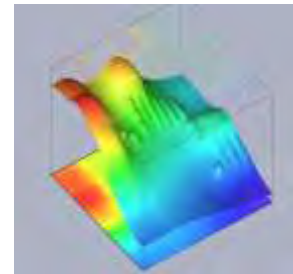


Figure 1-hot spots. The position of power sources and their interaction can create hot spots on the chip. [2]

Ιστορικά, οι τεχνικές χωροταξικής τοποθέτησης κελιών, έχουν αναπτυχθεί κυρίως με βάση τη δρομολόγηση. Οι αλγόριθμοι αυτοί, συνήθως, επικεντρώνονται στην ελαχιστοποίηση του συνολικού καθαρού μήκους, ενώ άλλοι εστιάζουν στην ελαχιστοποίηση των “crossovers” καλωδίων και των vias[3]. Ωστόσο, με την αυξημένη ζήτηση για υψηλής ποιότητας και μακροπρόθεσμη αξιόπιστη απόδοση, έχουν αναπτυχθεί πολλές νέες τεχνικές. Μερικές μελέτες έχουν αναπτυχθεί για την βελτίωση αξιοπιστίας σε κυκλώματα PHC(Power Hybrid Circuits)[4], για PCBs (Printed Circuit Board)[5]-[10], για VLSI [11],[12], και για MCMs[13]-[16]. Επιπλέον, χωροθετήσεις για αξιοπιστία και δρομολόγηση(routability) ταυτόχρονα υπάρχουν για PCBs[17]-[20] και PHCs[21].

Οι force-directed αλγόριθμοι (FDA) έχουν χρησιμοποιηθεί ευρέως για τη βελτιστοποίηση της τοποθέτησης των κελιών ως προς τη δρομολόγηση σε PCBs και VLSI[3],[22],[23]. Οι FDA αλγόριθμοι βασίζονται στο μοντέλο ότι κάθε component ασκεί δυνάμεις έλξης σε άλλα components που είναι συνδεδεμένα με δίκτυα σήματος (signal nets) και στο ότι απωθητικές δυνάμεις χρησιμοποιούνται για να κρατήσουν τα components μακριά για εκείνα που δεν είναι συνδεδεμένα. Οι παραδοσιακοί FDA αλγόριθμοι προφανώς δεν είναι κατάλληλοι για προβλήματα χωροθέτησης κελιών, λαμβάνοντας υπόψη τη θερμοκρασία. Ένας FDA αλγόριθμος βασισμένος σε ένα ασαφές μοντέλο, παρουσιάστηκε για να αντιμετωπίσει προβλήματα θερμοκρασίας για MCMs από τους Huang και Fu[15]. Στο μοντέλο τοποθέτησης που παρουσίασαν, η απωθητική δύναμη που ασκείται επί δύο οποιωνδήποτε chips είναι ανάλογη της ισχύς που παράγεται από τα δύο chips και μειώνεται όταν η απόσταση των δύο chips αυξάνεται.

Εδώ, παρουσιάζεται ένας διαφορετικός αλγόριθμος χωροταξικής διευθέτησης προτύπων κελιών με ευαισθησία στη θερμοκρασία, κατάλληλος για χωροθέτηση κελιών σε MCMs και υβριδικά κυκλώματα.

3. Διατύπωση της προσέγγισης

Καταρχήν, θεωρούμε πως όπου MCM έχουμε ένα απλό chip και όπου chips έχουμε πύλες. Έστω οι πύλες d_1, d_2, \dots, d_m ενός chip και (x_i, y_i) οι καρτεσιανές συντεταγμένες για κάθε πύλη. Το ζητούμενο στο Thermal Placement είναι να αποφασιστεί η θέση των πυλών, έτσι ώστε η αξιοπιστία του συστήματος να είναι βέλτιστη (NP-complete πρόβλημα)[24]-[28]. Ευριστικές μέθοδοι θα χρειαστούν για την επίτευξη μιας βέλτιστης λύσης σε λογικό χρόνο διεκπεραίωσης. Για το πρόβλημά μας υπάρχουν δύο είδη ευριστικών μεθόδων. Η μία είναι μια κατασκευαστική μέθοδος, η οποία βρίσκει μια αρχική λύση χρησιμοποιώντας ευριστικούς κανόνες, συχνά με διαδοχικό, ντετερμινιστικό τρόπο. Η άλλη είναι μια επαναληπτική μέθοδος βελτίωσης, η οποία επιτυγχάνει την βελτιστοποίηση με τοπικούς μετασχηματισμούς.

Στον αλγόριθμο αυτό γίνεται χρήση μιας κατασκευαστικής μεθόδου:

- 1) Το υπόστρωμα με τις πύλες μετατρέπεται σε δίχως όρια υπόστρωμα με άπειρο αριθμό πυλών με την τεχνική της πολλαπλής αντανάκλασης (multiple reflection technique).
- 2) Ο TFPA χρησιμοποιείται για να αποφασιστούν οι τελικές θέσεις των πυλών.

3.1 Ανάλυση του 1. Τεχνική της πολλαπλής αντανάκλασης

Multiple reflection technique:

Τα πλαϊνά του υποστρώματος λειτουργούν ως μονωτικά, καθώς έχουμε ασήμαντη απώλεια θερμότητας από τα πλευρικά τοιχώματα συγκριτικά με τις πλευρές πάνω και κάτω. Συνεπώς, τα πλαϊνά τοιχώματα μπορούν να αντιμετωπιστούν ως τέλεια μονωτικά, καθώς δεν υπάρχει ροή θερμότητας μεταξύ των μονωμένων ορίων και επειδή η ροή της θερμότητας είναι ανάλογη προς τη θερμοκρασία, θεωρούμε ότι έχουν μηδενικές μεταβολές θερμοκρασίας. Σαν αποτέλεσμα, χρησιμοποιείται μια αντανάκλαση σαν εικόνα από καθρέφτη για να αντικατασταθούν τα όρια. (Fig. 2 και 3).

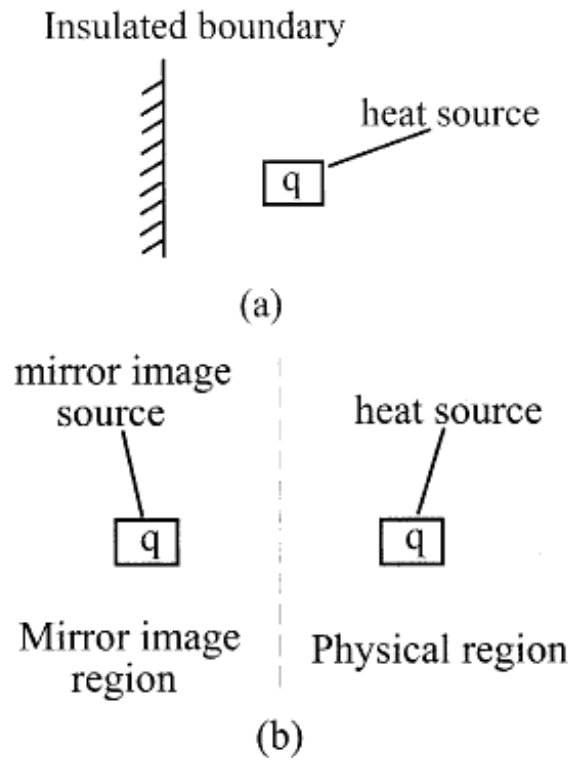


Figure 2.(a) Infinite insulated boundary with a heat source and
 (b) replace the insulated boundary by the use of a reflected mirror image source.

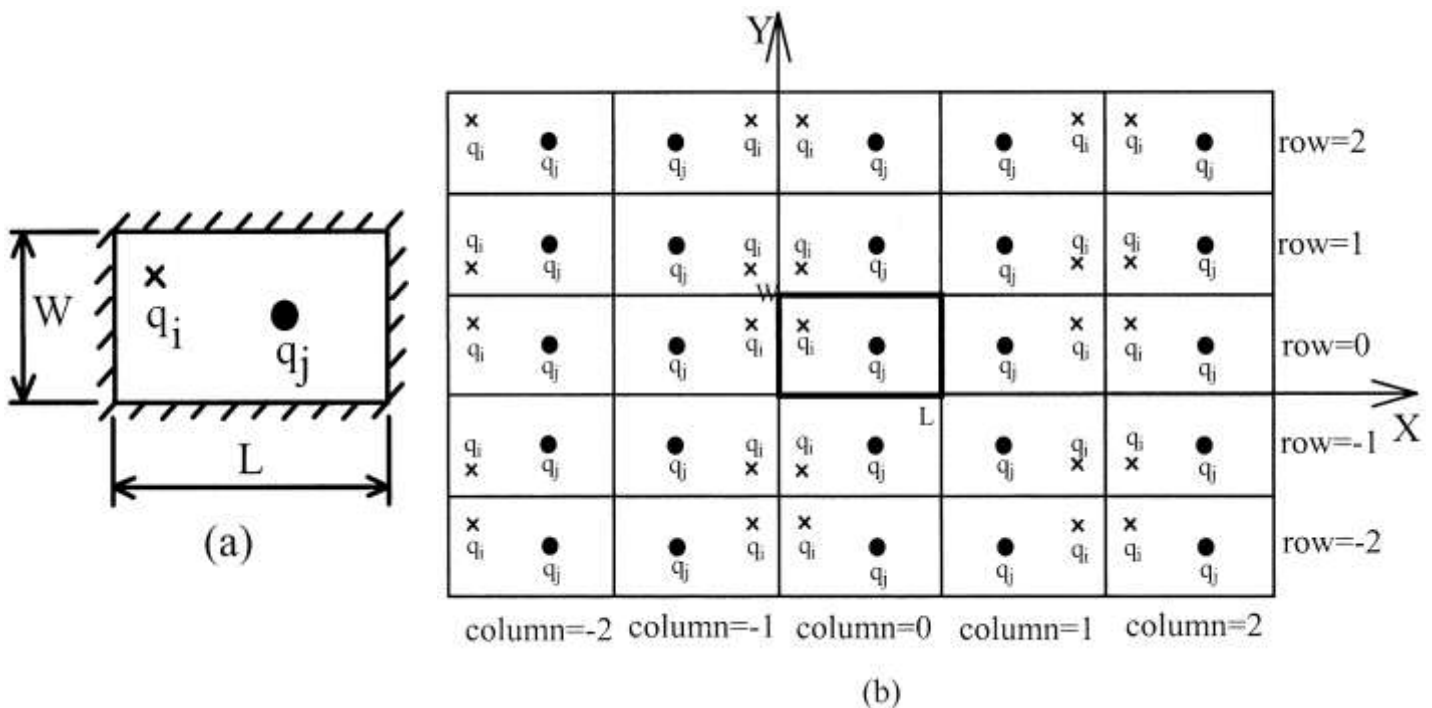


Figure 3. Multiple reflection technique: (a) rectangle region with insulated boundaries and (b) unbounded region with infinite mirror heat sources.

Ένα ορθογώνιο υπόστρωμα με αρκετές διαφορετικές πηγές θερμότητας με 4 μονωμένα όρια, μπορεί να μετατραπεί σε ένα υπόστρωμα χωρίς όρια με άπειρο αριθμό εικόνων αντανάκλασης (mirror images) των πηγών θερμότητας. Η νέα αυτή διαμόρφωση αφήνει την μη αλλαγμένη διανομή θερμοκρασίας εντός της γνήσιας περιοχής. (Παράδειγμα fig. 3.) Η αντανάκλαση του υποστρώματος στην r σειρά και c στήλη (fig.3), καλείται r - c υπόστρωμα. Η θέση της πύλης d_j ορίζεται στο r - c σύστημα στο σημείο $(x_j^{(r,c)}, y_j^{(r,c)})$, όπου

$$x_j^{(r,c)} = \left(c + \frac{1}{2}\right)L + (-1)^c \left(x_j - \frac{L}{2}\right)$$

$$y_j^{(r,c)} = \left(r + \frac{1}{2}\right)W + (-1)^r \left(y_j - \frac{W}{2}\right)$$

Figure 4. Σχέσεις (1) για x και (2) για y .

Και όπου L και W είναι αντίστοιχα το μήκος και το πλάτος του υποστρώματος. Αν μία πύλη βρίσκεται στο πραγματικό υπόστρωμα, δηλαδή $r=0$ και $c=0$, τότε πάντα θεωρούμε πως η τοποθεσία της θα είναι στο (x_i, y_i) .

Από μαθηματική άποψη, οι ασυνέχειες στα σύνορα αποτελούν το μεγαλύτερο εμπόδιο για την πρόβλεψη των θερμοκρασιών. Με τη χρήση της τεχνικής πολλαπλής ανάκλασης, τα όρια τα οποία αποτελούν εμπόδιο για την πρόγνωση θερμοκρασιών, καθώς πρόκειται για ασυνέχειες, εμπεριέχονται στο χωρίς όρια υπόστρωμα και ταυτόχρονα αφαιρούνται από την διαδικασία επίλυσης. Είναι συνήθως ευκολότερο να λύνονται τα προβλήματα της διανομής της θερμοκρασίας σε ένα απείρων διαστάσεων υπόστρωμα. Ορισμένες παλαιότερες μελέτες έχουν χρησιμοποιήσει την τεχνική της πολλαπλής ανάκλασης με την αρχή της επαλληλίας, για να προβλέψουν θερμοκρασίες ηλεκτρονικών συσκευών [29]-[31].

3.2.1 Ανάλυση του 2. Thermal force model

Σε ένα απέραντο σώμα, η θερμότητα ρέει παντού από κάθε πηγή θερμότητας και μειώνεται με το τετράγωνο της απόστασης από την εκάστοτε πηγή στην σταθερή κατάσταση. Το μοντέλο το οποίο μελετάμε είναι ανάλογο του μηχανισμού που μόλις

περιγράψαμε. Κάθε πύλη d_j σπρώχνει κάθε άλλη πύλη d_i με μια δύναμη ανάλογη της απώλειας θερμότητας του d_j και αντιστρόφως ανάλογη του τετραγώνου της απόστασης που χωρίζει τις δύο πύλες:

$$f_{i,j}^{(r,c)} = -\alpha_i \times \frac{q_j}{\left(\Delta r_{i,j}^{(r,c)}\right)^2} \quad (3)$$

Όπου F: δύναμη,

α : Thermal sensitivity factor της d_i . $\alpha_i=1 \Rightarrow$ για κανονικά τσιπάκια. $\alpha_i>1$ για τσιπάκια ευαίσθητα στη θερμοκρασία,

q_j : Απώλεια θερμότητας της d_j .

Και

$$\Delta r_{i,j}^{(r,c)} = \sqrt{\left(x_j^{(r,c)} - x_i\right)^2 + \left(y_j^{(r,c)} - y_i\right)^2} \quad (4)$$

Σπάμε το $f_{i,j}^{(r,c)}$: Σχέσεις (5) και (6).

$$f_{x_{i,j}}^{(r,c)} = -\alpha_i \times \frac{q_j}{\left(\Delta r_{i,j}^{(r,c)}\right)^2} \times \cos\left(\theta_{i,j}^{(r,c)}\right)$$

$$f_{y_{i,j}}^{(r,c)} = -\alpha_i \times \frac{q_j}{\left(\Delta r_{i,j}^{(r,c)}\right)^2} \times \sin\left(\theta_{i,j}^{(r,c)}\right)$$

Όπου

$$\theta_{i,j}^{(r,c)} = \arctan 2 \left(\frac{y_j^{(r,c)} - y_i}{x_j^{(r,c)} - x_i} \right) \quad (7)$$

Γενικός τύπος για όλα τα chips (mirror image) όπου z: x ή y.

$$F_{z_i} = \sum_{j=1}^m \left\{ f_{z_i,j} + \sum_{a=1}^{\infty} \left[\sum_{c=-a}^a \left(f_{z_i,j}^{(-a,-c)} + f_{z_i,j}^{(a,c)} \right) + \sum_{r=-a+1}^{a-1} \left(f_{z_i,j}^{(r,a)} + f_{z_i,j}^{(-r,-a)} \right) \right] \right\} \quad (8)$$

Κανονικά άνω όριο για το a πρέπει να είναι το άπειρο αλλά εδώ και ένας μικρός ακέραιος αρκεί. Π.χ. ο αριθμός 5 για το a είναι επαρκής, καθώς το πραγματικό υπόστρωμα μεταφράζεται σε μια περιοχή με 121 φορές το πραγματικό υπόστρωμα.

3.2.2. Thermal force placement algorithm.

Για να λύσουμε την παραπάνω εξίσωση θέτουμε $F_{z_i}=0$ και λύνουμε ως προς Z_i . Με χρήση της μεθόδου Newton- Raphson(NR). Λύνουμε το σύστημα για να βρούμε την σωστή θέση των πυλών.

$$F'_{z_i} = \sum_{j=1}^m \left\{ f'_{z_i,j} + \sum_{a=1}^{\infty} \left[\sum_{c=-a}^a \left(f'_{z_i,j}^{(-a,-c)} + f'_{z_i,j}^{(a,c)} \right) + \sum_{r=-a+1}^{a-1} \left(f'_{z_i,j}^{(r,a)} + f'_{z_i,j}^{(-r,-a)} \right) \right] \right\} \quad (9)$$

Όπου: (σχέσεις (10) και (11)).

$$f'_{x_i,j}^{(r,c)} = \frac{df_{x_i,j}^{(r,c)}}{dx_i}$$

$$= -\alpha_i \times \frac{q_j}{(\Delta r_{i,j}^{(r,c)})^3} \times [3 \cos^2(\theta_{i,j}^{(r,c)}) - 1]$$

$$f'_{y_i,j}^{(r,c)} = \frac{df_{y_i,j}^{(r,c)}}{dx_i}$$

$$= -\alpha_i \times \frac{q_j}{(\Delta r_{i,j}^{(r,c)})^3} \times [3 \sin^2(\theta_{i,j}^{(r,c)}) - 1],$$

Εν τέλει, η απόσταση που κινείται η d_i ορίζεται ως:

$$\Delta z_i = 0.5 \times \frac{F_{z_i}}{F'_{z_i}}. \quad (12)$$

Εδώ ορίζουμε 0,5 γιατί κάθε δύο πύλες σπρώχνονται μεταξύ τους και γι' αυτό μόνο τις 0,5 φορές θα χρειαστεί να αλλάξει η καθορισμένη απόσταση. Η νέα θέση μετά από κάθε επανάληψη είναι $z_i^{(new)} = z_i + \Delta z_i$ (13) και ελέγχουμε κάθε φορά μήπως βγούμε εκτός ορίων. Η διαδικασία επαναλαμβάνεται μέχρι $z_i + \Delta z_i$ να είναι στο πραγματικό υπόστρωμα.

Επεξήγηση αλγορίθμου (fig.4):

```

procedure TFPA(Q, A, X, Y, L, W,  $\epsilon_1$ ,  $\epsilon_2$ )
// Q={ $q_i$  |  $1 \leq i \leq m$ }, A = { $\alpha_i$  |  $1 \leq i \leq m$  }, (X, Y)={( $x_i, y_i$ ) |  $1 \leq i \leq m$  } //
// L and W are the length and the width of the substrate. //
//  $\epsilon_1$  and  $\epsilon_2$  are two stopping criterion values. //
begin
  do
    Random(X, Y) //  $0 < x_i < L, 0 < y_i < W$  //
    Norm ← a large value
  do
    Old_Norm ← Norm
    Norm ← 0
    for  $i \leftarrow 1$  to  $m$ 
      ( $F_{x_i}, F_{y_i}$ ) ← Equation (8)
      ( $F'_{x_i}, F'_{y_i}$ ) ← Equation (9)
      ( $\Delta x_i, \Delta y_i$ ) ←  $0.5 \times (\frac{F_{x_i}}{F'_{x_i}}, \frac{F_{y_i}}{F'_{y_i}})$ 
      while  $x_i + \Delta x_i \geq L$  or  $x_i + \Delta x_i \leq 0$  do  $\Delta x_i \leftarrow \Delta x_i / 2$  repeat
      while  $y_i + \Delta y_i \geq W$  or  $y_i + \Delta y_i \leq 0$  do  $\Delta y_i \leftarrow \Delta y_i / 2$  repeat
      ( $x_i, y_i$ ) ← ( $x_i + \Delta x_i, y_i + \Delta y_i$ )
      Norm ←  $|F_{x_i}| + |F_{y_i}|$ 
    repeat
      while  $\frac{|Old\_Norm - Norm|}{Norm} > \epsilon_1$ 
      while Norm  $> \epsilon_2$ 
    output(X, Y)
end

```

Επεξήγηση:

- Αρχικά ο αλγόριθμος παράγει ένα τυχαίο αρχικό placement. Νέα placements υπολογίζονται με την NR μέθοδο μέχρι να επιτευχθεί μια συγκλίνουσα λύση.
- Επίσης, θέτουμε το παρακάτω:

$$\text{Norm} = \sum_{i=1}^m (|F x_i| + F y_i) \quad (14)$$

- Στη μεταβλητή Old_Norm κρατάμε τη Norm του προηγούμενου placement.
- Τα ε_1 , ε_2 αποτελούν τα κριτήρια τερματισμού του αλγορίθμου. Αποτελούν δυο μικρές ακέραιες τιμές που καθορίζονται από τους χρήστες. Για την βέλτιστη επιλογή των ε_1 , ε_2 χρειάζεται να κάνουμε δοκιμές. Αν, όμως, θέσουμε ε_1 και $\varepsilon_2 \leq 0,01$, τα placements που προκύπτουν έχουν ελάχιστες διαφορές.
- Στην περίπτωση που $|Old_Norm - Norm|/Norm > \varepsilon_1$, ο αλγόριθμος δε συγκλίνει ακόμα και ο αλγόριθμος επιδιώκει να υπολογίσει νέο placement.
- Στην περίπτωση που ισχύει $|Old_Norm - Norm|/Norm \leq \varepsilon_1$ αλλά $Norm > \varepsilon_2$ τότε το εκάστοτε placement παγιδεύεται σε ένα τοπικό βέλτιστο και δεν προχωράει ο αλγόριθμος. Διαφορετικά, ο αλγόριθμος παράγει νέο placement και ξαναεκτελεί τη διαδικασία από την αρχή.
- Όταν όμως ισχύει $|Old_Norm - Norm|/Norm \leq \varepsilon_1$ και $Norm \leq \varepsilon_2$, τότε ο αλγόριθμος τερματίζει.

3.3. Αξιοπιστία αλγορίθμου

Στον TFPA, η θερμική δύναμη είναι ανάλογη της ροής θερμότητας σε μια περιοχή χωρίς όρια και σύνορα. Ο αλγόριθμος πάντα κινεί τις πύλες σε περιοχή χαμηλότερης θερμοκρασίας, οπότε πάντα καταλήγει σε συγκλίνον αποτέλεσμα και ομοιόμορφη κατανομή θερμοκρασίας.

3.4. Πολυπλοκότητα χώρου.

Ο TFPA απαιτεί $10m$ κελιά για αποθηκευτικό χώρο για τα $\{ a_i, q_i, x_i, y_i, Fx_i, Fy_i, F'x_i, F'y_i, \Delta x_i, \Delta y_i \mid 1 \leq i \leq m \}$. Βασικά, η πολυπλοκότητα χώρου για τον TFPA είναι $O(m)$.

3.5. Πολυπλοκότητα χρόνου

Ο αλγόριθμος εφαρμόζει τη μονοδιάστατη μέθοδο NR $2m$ φορές, μία για κάθε πύλη. Ο χρόνος υπολογίζεται από την εκτέλεση των πράξεων που υπολογίζουν τα Fz_i και $F'z_i$ και ο χρόνος που απαιτείται για τα υπόλοιπα βήματα δεν υπολογίζεται καθόλου. Στους τύπους αυτούς, ανά βήμα, η συχνότητα για την αξιολόγηση των αντίστοιχων F για r, c είναι ίση με:

$$\sum_{j=1}^m \left\{ 1 + \sum_{a=1}^5 \left[\sum_{c=-a}^a 2 + \sum_{r=-a+1}^{a-1} 2 \right] \right\} = 121m.$$

Αφήνουμε το χρόνο που χρειάζονται οι f και f' , για αξιολόγηση/ εκτίμηση, να είναι η μονάδα του χρόνου. Τότε, απαιτεί 242 μονάδες χρόνου ανά επανάληψη του TFPA. Καταλήγουμε για τον αλγόριθμο στην πολυπλοκότητα $O(m^2)$ για κάθε επανάληψη. Ο αριθμός των επαναλήψεων που απαιτούνται εξαρτάται από το μέγεθος (m) του προβλήματος, την απώλεια θερμότητας από τις πύλες και τις τιμές των $\varepsilon_1, \varepsilon_2$.

3.6. Ακρίβεια/ Ορθότητα και όρια του TFPA

Η ακρίβεια μιας τέτοιας λύσης εξαρτάται τουλάχιστον από τους παρακάτω τρεις παράγοντες:

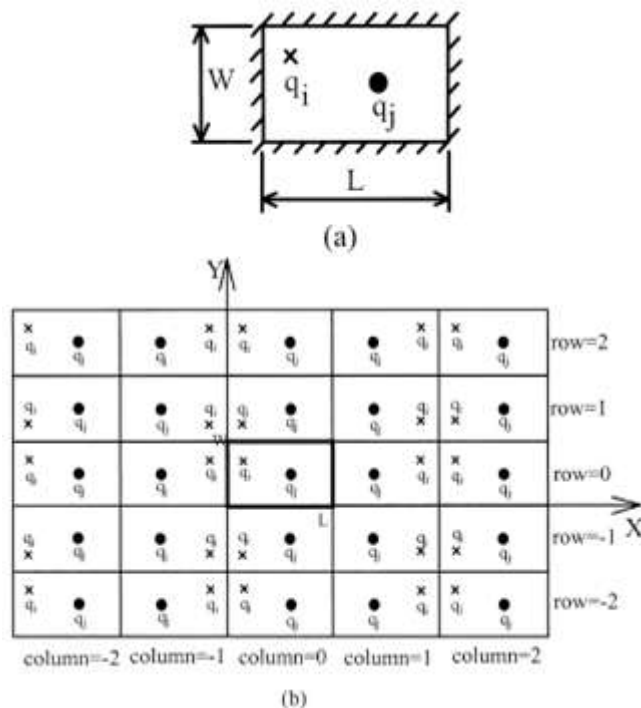
- ✓ Ο βαθμός στον οποίο το φυσικό πρόβλημα και το μοντέλο TFPA θα βρίσκονται σε αντιστοιχία (υλικό, μονωμένες άκρες κ.λ.π.) . Η δομή του packaging απλοποιείται ως ένα ορθογώνιο επίπεδο, το υλικό του υποστρώματος θεωρείται ιστροπικό[*] και ομοιογενές ως προς την θερμική αγωγιμότητα και οι άκρες του είναι τέλεια μονωμένες. Όλες αυτές οι απλοποιήσεις και παραδοχές εμπεριέχουν κάποιο σφάλμα. Ωστόσο, από τη στιγμή που ο TFPA τοποθετεί τα chips ξεχωριστά, έτσι ώστε ακόμη και αν ένα package δεν πληροί τις παραπάνω προϋποθέσεις, ο αλγόριθμος μπορεί να χρησιμοποιηθεί και να βγάλει ένα “καλό” αποτέλεσμα.
- ✓ Ο TFPA, για να απλοποιήσει το πρόβλημα, παραμελεί τις διαστάσεις των πυλών/chips και θεωρεί την κάθε πύλη/chip σαν σημείο. Αυτό μπορεί να οδηγήσει σε overlapping, κάτι το οποίο, όμως, δεν είναι τόσο σοβαρό πρόβλημα, όπως θα δείτε παρακάτω στα παραδείγματα.
- ✓ Τέλος, η ακρίβεια επηρεάζεται επίσης από την περικοπή των υψηλής διάταξης όρων στη σχέση (8). Ωστόσο, το σφάλμα αυτό δεν έχει σημαντική επίδραση στην ακρίβεια, όπως θα δείτε και στο παράρτημα.

***Ισοτροπία:** η ιδιότητα των σωμάτων σύμφωνα με την οποία οι φυσικές τους ιδιότητες είναι οι ίδιες προς όλες τις κατευθύνσεις

4. Υλοποίηση αλγορίθμου TFPA σε C.

4.1 Επεξήγηση κώδικα

Γενικά, με βάση τη θεωρία του αλγορίθμου, έχουμε στο μυαλό μας ότι έχουμε πολλές εικόνες του αρχικού chip σύμφωνα με το παράδειγμα (b) της εικόνας, στην οποία $a=2$, ενώ εμείς στον αλγόριθμό μας ορίσαμε $a=5$.



Αρχείο main.c :

Στην αρχή ορίζονται οι βιβλιοθήκες που χρησιμοποιούνται στον κώδικα της main, κάνοντας include τα αντίστοιχα αρχεία και ορίζουμε και τις σταθερές X και Y που θα μας χρειαστούν. Στη συνέχεια, γίνεται η δήλωση των μεταβλητών. Πρώτα δηλώνονται οι εξωτερικές μεταβλητές(extern), οι οποίες ορίζονται μεν στη main αλλά υπολογίζονται από συναρτήσεις άλλων αρχείων.

Στην αρχή της main ορίζουμε τις μεταβλητές που χρειαζόμαστε:

```
float aa = 1.5; /*thermal sensitivity factor */
int L = 100; /*chip length */
int W = 100; /*chip width */
int totalCells; /* αριθμός των πυλών*/
double e1 = 0.01; /*stopping criterion value */
```



```

double e2 = 0.01; /*stopping criterion value */
long double Norm;
long double Old_Norm, check_norm, total_norm;
long double temp1, temp2;
int i=0; /*counter */
int initCellPositions; /*Define after we calculate the initial coordinates for our cells */
long double F1x, F1y, F2x, F2y;
int yy=0;
int xx=0;
double *q; /*πίνακας με τις τιμές των q */

```

```

double dx, dy; // για τη μετακίνηση των πυλών στους άξονες x,y

```

```

double *x; //Array used to store the final x coordinates of the cells
double *y; //Array used to store the final y coordinates of the cells
int g; //counter

```

Στη συνέχεια διαβάζουμε τα αρχεία που θέλουμε για να βρούμε τον αριθμό των πυλών του chip και ελέγχουμε και αν όντως υπάρχουν. Μετά καλούμε τη συνάρτηση `comp_num_of_gates` δίνοντας το κατάλληλο αρχείο ως παράμετρο. Έχουμε δημιουργήσει μια βιβλιοθήκη `lib.h` και το αντίστοιχο αρχείο `lib2.c` όπου γίνεται η υλοποίηση των συναρτήσεων.

```

int comp_num_of_gates( FILE * input_file_ptr )
{
    int total_cells=0;
    int i; αρχικοποίηση του counter

```

Για τις 12 πρώτες γραμμές του αρχείου καλείται η συνάρτηση **`void read_line(FILE * file_ptr)`** η οποία βρίσκεται κι αυτή στο ίδιο αρχείο. Η συνάρτηση αυτή χρησιμοποιείται για να διαβαστεί μια σειρά. Στο εσωτερικό της συνάρτησης αρχικοποιείται ένας μετρητής `bytes_read` και αρχικοποιείται και η μεταβλητή `max_bytes` η οποία ορίζει τον μέγιστο αριθμό των bytes που μπορούν να διαβαστούν χωρίς να ξαναδεσμευτεί (`realloc`) μνήμη. Επιπρόσθετα, ορίζεται μια μεταβλητή `cur_char` για τον εκάστοτε χαρακτήρα που θα διαβαστεί. Εν συνεχεία, τσεκάρουμε τον file pointer και αν είναι null το τυπώνουμε στην οθόνη (`printf("read_line: file pointer is NULL\n"`)) και επιστρέφουμε χωρίς να τρέξουμε τον υπόλοιπο κώδικα της συνάρτησης αυτής. Διαφορετικά, συνεχίζει διαβάζοντας τον πρώτο χαρακτήρα της σειράς από το αρχείο και το αποθηκεύει στην μεταβλητή `cur_char`. Μετά από αυτό, με τη `while` διαβάζουμε και τους υπόλοιπους χαρακτήρες. Εάν ο `cur_char` δεν είναι η αλλαγή γραμμής και δεν έχουμε φτάσει στο τέλος του αρχείου, τότε, αν `bytes_read=max_bytes`, επαναδεσμεύεται μνήμη για επιπλέον 100 θέσεις, καθώς η μνήμη που αρχικά δεσμεύσαμε για την `input_line` δεν ήταν τελικά αρκετή για τις ανάγκες μας. Αφού εκτελεστεί η `realloc`, αν επιστρέψει έναν νέο δείκτη που θα δείχνει στο νέο κομμάτι μνήμης, η εκτέλεση της επανάληψης `while` θα συνεχιστεί κανονικά. Αν όμως επιστρέψει null, τότε

σημαίνει πως έχει αποτύχει να βρει την απαιτούμενη μνήμη και έτσι κάνει exit αφού δεν γίνεται να διαβαστούν άλλοι χαρακτήρες, αφού πρώτα τυπώσει σχετικό ενημερωτικό μήνυμα. Σε γενικές γραμμές, στη while της συνάρτησης αυτής, για να διαβάσουμε κάθε επόμενο χαρακτήρα της σειράς, ελέγχουμε πρώτα αν ο μετρητής έχει ξεπεράσει τον μέγιστο αριθμό που ορίσαμε. Ανάλογα με τα αποτελέσματα των ελέγχων που κάνουμε, πραγματοποιούμε τις απαιτούμενες ενέργειες από αυτές που μόλις αναφέρθηκαν και προχωράμε στον επόμενο χαρακτήρα που θα αποθηκευτεί στην επόμενη θέση του πίνακα. Αφού τελειώσει η διαδικασία αυτή, αποθηκεύεται ο χαρακτήρας τερματισμού “\0” στο input_line[bytes_read].

Με τον παραπάνω τρόπο το πρόγραμμα διαβάζει τις πρώτες 12 γραμμές του αρχείου.

```
for( i=0; i<12; i++ )
{
    read_line( input_file_ptr );
}
```

Αφού γίνει αυτό και για όσο δεν είμαστε στο τέλος του αρχείου, αν ο πρώτος χαρακτήρας της input_line είναι “C” αυξάνει κατά 1 τον συνολικό αριθμό των κελιών, επιστρέφει στη main_test, αποθηκεύει τον αριθμό που βρήκε στην totalCells και τον τυπώνει.

```
printf( "Total Gates=%d\n", totalCells );
printf( "\n" );
```

Στη συνέχεια δεσμεύουμε μνήμη για τον πίνακα των q με μια malloc, γεμίζουμε τις θέσεις του πίνακα με τυχαίες τιμές και τις τυπώνουμε.

Παρακάτω, δεσμεύουμε μνήμη για τους πίνακες x, y τους οποίους χρειαζόμαστε για να αποθηκεύσουμε τις συντεταγμένες της θέσης κάθε πύλης. Για να ορίσουμε τις συντεταγμένες αυτές, τις οποίες και ορίζουμε τυχαία (αρχικό τυχαίο placement) καλούμε την initialize_cell_coordinates(totalCells) της οποίας η υλοποίηση βρίσκεται στο αρχείο lib2.c και κάπως έτσι ξεκινάει το πρώτο do...while. Εκεί δεσμεύουμε μνήμη για έναν πίνακα cells_array και του δίνουμε τιμές για x και για y. Έχουμε ορίσει στην αρχή του αρχείου τον πίνακα: Cell ** cells_array; Η struct Cell έχει οριστεί στην αρχή του αρχείου lib.h. Επιστρέφουμε στη main όπου και τυπώνουμε τις συντεταγμένες των πυλών. Νέα τυχαία placements υπολογίζονται μέχρι να επιτευχθεί μια συγκλίνουσα λύση.

Στη συνέχεια, δίνουμε στην μεταβλητή Norm μια μεγάλη τιμή(έστω 10000) και μπαίνουμε στο δεύτερο do...while. Ορίζουμε Old_Norm = Norm και Norm = 0. Ακολουθεί ένα for loop, το οποίο τρέχει από 1 μέχρι και τον συνολικό αριθμό των πυλών. Στη for αυτή αρχικά υπολογίζουμε το σύνολο των δυνάμεων που ασκείται στην πύλη από τις υπόλοιπες πύλες του chip, διασπώντας σύμφωνα με τη θεωρία του αλγορίθμου τη δύναμη αυτή σε Fx και Fy.

Αυτά τα υπολογίζουμε καλώντας δύο φορές τη συνάρτηση **equation8** η οποία υπολογίζει το:

$$Fz_i = \sum_{j=1}^m \left\{ fz_{i,j} + \sum_{a=1}^{\infty} \left[\sum_{c=-a}^a \left(fz_{i,j}^{(-a,-c)} + fz_{i,j}^{(a,c)} \right) + \sum_{r=1}^{a-1} \left(fz_{i,j}^{(r,a)} + fz_{i,j}^{(-r,-a)} \right) \right] \right\}$$

Μία για z=X και μία για z=Y. Πηγαίνουμε στο αρχείο lib2.c για να δούμε πώς υλοποιείται η συνάρτηση. Η συνάρτηση έχει ως παραμέτρους: το i, το οποίο δείχνει τον αριθμό της εκάστοτε πύλης, το σύνολο των πυλών του chip, την ΧοrΥ με την οποία ελέγχουμε αν θέλουμε την Fx ή την Fy δίνοντας στην κλήση X ή Y αντίστοιχα και τους πίνακες x,y,q. Στη συνάρτηση αρχικά ορίζουμε τις μεταβλητές που θα χρειαστούμε:

```
int aa=1.5; //thermal sensitivity factor
int j; //counter
long double temp=0.0;
int f1=0;
int a, c, r; //c is for column, r is for row (unbound region with infinite mirror heat sources
long double temp0, temp1, temp2, temp3=0.0;
double z,w,n,o; // comp_f1,comp_f2
```

Μετά υλοποιούμε μία for η οποία τρέχει για το σύνολο των πυλών και μέσα σε αυτήν υπολογίζουμε τα f που χρειαζόμαστε για την παραπάνω σχέση καλώντας κάθε φορά τη συνάρτηση comp_f1 με τις αντίστοιχες παραμέτρους ανάλογα με την περίπτωση.

Ας δούμε πώς δουλεύει η **comp_f1**. Ο κώδικας που την υλοποιεί βρίσκεται στο ίδιο αρχείο. Η συνάρτηση έχει τις εξής παραμέτρους: τη μεταβλητή aa(thermal sensitivity factor), τον πίνακα q, τη γραμμή r και τη στήλη c, τα i και j που είναι οι counters που έχουμε συναντήσει για τις πύλες, την ΧοrΥ για να δούμε ποια συνιστώσα της δύναμης ψάχνουμε(x,y), και τους πίνακες x,y. Δηλώνουμε τις μεταβλητές:

```
Long double f1=0.0; // για το αποτέλεσμα
double xj, yj; //x,y θέσεις των πυλών στα r,c
double u, dr; // γωνία u και dr απόσταση πυλών
```

Αν τα r,c είναι μηδέν, δηλαδή βρισκόμαστε στο πραγματικό chip και όχι σε μια εικόνα του που έχουμε δημιουργήσει (σύμφωνα με την εικόνα που είδαμε προηγουμένως), τότε δεν χρειάζεται να βρούμε τη συντεταγμένη x ή y που χρειαζόμαστε καθώς την τιμή της την έχουμε ήδη και είναι η αρχική τιμή που δόθηκε με το τυχαίο placement που κάναμε στην αρχή του αλγορίθμου. Αν τα r, c είναι διάφορα του μηδενός, τότε η πύλη βρίσκεται σε μια εικόνα του chip και όχι στο πραγματικό και έτσι πρέπει να υπολογίσουμε τις συντεταγμένες της θέσης στην οποία βρίσκεται. Οι θέσεις αυτές υπολογίζονται με τους τύπους:

$$x_j^{(r,c)} = \left(c + \frac{1}{2}\right) L + (-1)^c \left(x_j - \frac{L}{2}\right)$$

$$y_j^{(r,c)} = \left(r + \frac{1}{2}\right) W + (-1)^r \left(y_j - \frac{W}{2}\right)$$

Επίσης, στη συνάρτηση αυτή υπολογίζουμε και τη γωνία θ και το dr σύμφωνα με τους τύπους:

$$\theta_{i,j}^{(r,c)} = a \tan 2 \left(\frac{y_j^{(r,c)} - y_i}{x_j^{(r,c)} - x_i} \right)$$

και

$$\Delta r_{i,j}^{(r,c)} = \sqrt{\left(x_j^{(r,c)} - x_i\right)^2 + \left(y_j^{(r,c)} - y_i\right)^2}$$

Αν το dr είναι μηδέν, σημαίνει ότι η απόσταση μεταξύ των πυλών είναι μηδέν, δηλαδή οι πύλες των οποίων ψάχνουμε την dr , στην ουσία αποτελούν την ίδια πύλη και γι' αυτό και το αποτέλεσμα είναι μηδενικό. Σε αυτήν την περίπτωση δεν υφίσταται δύναμη f , καθώς η πύλη δεν ασκεί δύναμη στον εαυτό της και γι' αυτό ορίζουμε την δύναμη ίση με μηδέν ($f_1=0$). Διαφορετικά υπολογίζουμε το f_1 σύμφωνα με τον τύπο που δίνεται στη θεωρία και ανάλογα με το αν ψάχνουμε τη συνιστώσα y ή τη συνιστώσα x .

$$f_{x_{i,j}}^{(r,c)} = -\alpha_i \times \frac{q_j}{\left(\Delta r_{i,j}^{(r,c)}\right)^2} \times \cos\left(\theta_{i,j}^{(r,c)}\right)$$

$$f_{y_{i,j}}^{(r,c)} = -\alpha_i \times \frac{q_j}{\left(\Delta r_{i,j}^{(r,c)}\right)^2} \times \sin\left(\theta_{i,j}^{(r,c)}\right)$$

όπου $\alpha_i = \alpha = 1.5$

Είμαστε στη `for` που εκτελείται για $j=0$ και όσο το j είναι μικρότερο του συνόλου των κελιών και βρίσκεται στη συνάρτηση `equation8`. Αρχικά, υπολογίζουμε το f_1 ανάλογα με το αν δίνουμε X ή Y και για $r=0$ και $c=0$;

Μετά ορίζουμε μία `temp0` στην οποία θέλουμε να αποθηκεύσουμε το άθροισμα που τρέχει από $\alpha=1$ μέχρι $\alpha=5$.

Αρχίζει η αντίστοιχη `for`, στο εσωτερικό της οποίας ορίζουμε μία `temp1`, στην οποία θέλουμε να αποθηκεύσουμε το άθροισμα της σχέσης από $c=-a$ και όσο το $c \leq a$. Εκεί μέσα υπολογίζουμε τα δύο f_1 που ζητούνται, ανάλογα με το ποια συνιστώσα της δύναμης θέλουμε και τα αποθηκεύουμε στις z, w . Στη συνέχεια τα προσθέτουμε και μεταξύ τους και συνολικά για τις επαναλήψεις με την εντολή:

```
temp1 += ( z + w );
```

Κλείνει το εσωτερικό αυτό `for loop` και ενώ βρισκόμαστε στο εσωτερικό της `for` : `for(a=1; a<upper_bound_a; a++) {` ορίζουμε άλλο ένα `temp2` και εκεί κάνουμε την ίδια διαδικασία με τη `for` που μόλις έκλεισε, αλλά αυτή τη φορά εκτελούμε για : `(r = (-a + 1); r <= (a-1); r++)` όπου r είναι οι γραμμές στις οποίες βρίσκονται οι εικόνες του `chip` που υποθέτουμε πως υπάρχουν με βάση τη θεωρία του αλγορίθμου.

Αφού βγούμε και από αυτή τη for προθέτουμε τα temp1 και temp2 μεταξύ τους και συνολικά για όλες τις επαναλήψεις με την εντολή: temp0 += (temp1 + temp2).

Και τέλος, για το εξωτερικό for loop προσθέτουμε το f1 που βρήκαμε στην αρχή του loop και το σύνολο temp0 μεταξύ τους και συνολικά για όλα τα j με την εντολή temp += (temp0 + f1 = temp3). Η συνάρτηση επιστρέφει το περιεχόμενο της temp.

Επιστρέφουμε, λοιπόν, στη main. Εκεί έχοντας καλέσει δύο φορές την equation8 και έχοντας βρει τα F1x, F1y που ψάχνουμε, συνεχίζουμε βρίσκοντας τα F2x, F2y σύμφωνα με τους τύπους:

$$F'_{Z_i} = \sum_{j=1}^m \left\{ f'_{Z_{i,j}} + \sum_{a=1}^{\infty} \left[\sum_{c=-a}^a \left(f'_{Z_{i,j}}^{(-a, -c)} + f'_{Z_{i,j}}^{(a, c)} \right) + \sum_{r=-a+1}^{a-1} \left(f'_{Z_{i,j}}^{(r, a)} + f'_{Z_{i,j}}^{(-r, -a)} \right) \right] \right\}$$

$$f'_{X_{i,j}}^{(r, c)} = \frac{df_{i,j}^{(r, c)}}{dx_i} = -\alpha_i \times \frac{q_j}{\left(\Delta r_{i,j}^{(r, c)}\right)^3} \times \left[3 \cos^2 \left(\theta_{i,j}^{(r, c)} \right) - 1 \right]$$

$$f'_{Y_{i,j}}^{(r, c)} = \frac{df_{i,j}^{(r, c)}}{dx_i} = -\alpha_i \times \frac{q_j}{\left(\Delta r_{i,j}^{(r, c)}\right)^3} \times \left[3 \sin^2 \left(\theta_{i,j}^{(r, c)} \right) - 1 \right],$$

Για να βρούμε τα παραπάνω, καλούμε δύο φορές, μία για X και μία για Y, την equation9 η οποία με τη σειρά της καλεί όπου χρειαστεί την comp_f2. Και οι δύο συναρτήσεις λειτουργούν με τον ίδιο ακριβώς τρόπο με τις equation8 και comp_f1, απλά αλλάζουν λίγο οι τύποι.

Αφού τα έχουμε υπολογίσει αυτά για το εκάστοτε i, υπολογίζουμε καλώντας δύο φορές, μία για X και μία για Y, την double move(double F1X, double F1Y, double F2X, double F2Y, int XorY);, η οποία υλοποιείται στο αρχείο lib2.c και ανάλογα με το αν δίνουμε X ή Y, κάνει την αντίστοιχη διαίρεση, το αποτέλεσμα της οποίας πολλαπλασιάζει με 0.5 σύμφωνα με τον τύπο :

$$\Delta z_i = 0.5 \times \frac{Fz_i}{F'z_i}.$$

Με αυτόν τον τύπο βρίσκουμε πόσο θα μετακινηθεί η πύλη στον άξονα των x και πόσο στον άξονα των y και αποθηκεύουμε τα αντίστοιχα αποτελέσματα στις μεταβλητές dx, dy. Αν η θέση στον άξονα των x συν το dx μας δίνει κάποιο νέο x το οποίο όμως είναι εκτός ορίων του chip, τότε διαρούμε το dx με το 2 μέχρι το άθροισμά τους να είναι εντός ορίων του chip. Τον ίδιο έλεγχο κάνουμε και για το y+dy.

Τυπώνουμε τις νέες θέσεις των πυλών και υπολογίζουμε το άθροισμα των απόλυτων τιμών των F1x, F1y μεταξύ τους αλλά τη Norm = F1x + F1y,, τα οποία υπολογίσαμε με την equation8. Όλα αυτά τα κάνουμε για το σύνολο των πυλών. Τυπώνουμε για κάθε πύλη και την τιμή της Norm που έχει υπολογιστεί και αφού τελειώσουν όλες οι επαναλήψεις τυπώνουμε και την τελική τιμή της Norm.

Στη συνέχεια κάνουμε τις πράξεις:

```
temp1 = Old_Norm - Norm ;
temp2 = fabs(temp1) / Norm ;
```

και αν το temp2 είναι μεγαλύτερο της e1 που έχουμε ορίσει, το εσωτερικό do...while συνεχίζει τις επαναλήψεις του μέχρι η τιμή temp2 να γίνει μικρότερη της e1. Αν η τιμή της Norm είναι μικρότερη της e2 που έχουμε ορίσει, σταματάει τις επαναλήψεις του και το εξωτερικό do...while.

Τα e1, e2 αποτελούν τα κριτήρια τερματισμού του αλγορίθμου. Αποτελούν δυο μικρές τιμές οι οποίες καθορίζονται από τους χρήστες. Για την βέλτιστη επιλογή των e1, e2 χρειάζεται να κάνουμε δοκιμές.

Γενικά, στην περίπτωση που temp2>e1 , το placement δεν συγκλίνει ακόμα και ο αλγόριθμος επιστρέφει για να υπολογίσει ένα νέο placement. Στην περίπτωση που temp2<e1 αλλά τότε ο αλγόριθμος εκτελεί τη διαδικασία από την αρχή. Αν όμως emp2<e1 και Norm<e2, τότε ο αλγόριθμος τερματίζει και τυπώνουμε τις νέες και τελικές πλέον θέσεις των πυλών.

4.2 Αποτελέσματα

Ακολουθούν τα αποτελέσματα της παραπάνω υλοποίησης. Τα e1,e2, σύμφωνα με τη μελέτη για τον TFPA,όπως είδαμε και προηγουμένως, αποτελούν τα κριτήρια τερματισμού του αλγορίθμου. Συγκεκριμένα, πρόκειται για δυο μικρές ακέραιες τιμές

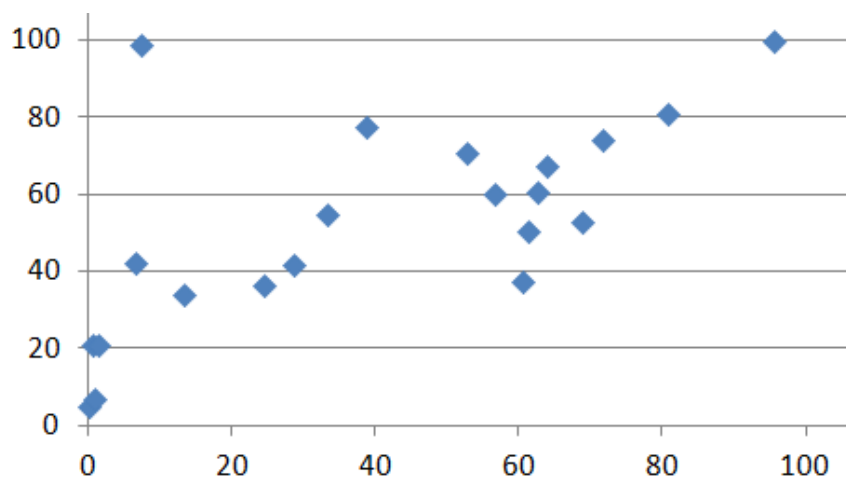
που καθορίζονται από τους χρήστες. Για την βέλτιστη επιλογή των ϵ_1, ϵ_2 χρειάζεται να κάνουμε δοκιμές. Αν, όμως, θέσουμε ϵ_1 και $\epsilon_2 \leq 0,01$, τα placements που προκύπτουν έχουν ελάχιστες διαφορές από τα τελικά. Εδώ παρατηρούμε πως ενώ η μελέτη αναφέρεται στα ϵ_1, ϵ_2 ως μικρούς ακεραίους τα θέτει ίσα ή μικρότερα από 0.01. Στα παραδείγματα τρεξίματος του κώδικα που ακολουθούν, έχουμε πειραματιστεί με τις τιμές των ϵ_1, ϵ_2 . Επίσης, σύμφωνα με τον ψευδοκώδικα, που επίσης είδαμε παραπάνω, κάποια στιγμή εκτελεί την πράξη $\text{Norm} = |F_{x_i}| + |F_{y_i}|$. Όταν ακολουθούμε πιστά τον ψευδοκώδικα και εκτελούμε τη συγκεκριμένη πράξη, ο αλγόριθμος τερματίζει κανονικά. Παρ'όλα αυτά, στο θεωρητικό κομμάτι της μελέτης που ακολουθεί τον ψευδοκώδικα, προτείνεται για πιο αποτελεσματική συγκλίνουσα λύση να πάρουμε το άθροισμα των F_{x_i}, F_{y_i} για τις γραμμές από 1 έως m που είναι και ο αριθμός των πυλών. Ωστόσο, σε αυτήν την περίπτωση ο αλγόριθμος δεν συγκλίνει και δεν τερματίζει.

✚ Ακολουθούν παραδείγματα στα οποία έχουμε διαφορετική τιμή heat dissipation για κάθε πύλη. Οι τιμές αυτές υπολογίζονται με μία εντολή random και κυμαίνονται μεταξύ 0.1 και 1.5, όπως συμβαίνει και στα παραδείγματα της μελέτης που θα δείτε σε επόμενη ενότητα.

- Για $\epsilon_1 = \epsilon_2 = 0.1$ και διαφορετικές τιμές για τα q (heat dissipation).

Τελικό placement με βάση τον TFPA.

X	Y	q
0.050134	5.06086	0.3
0.895241	6.894175	1.5
0.583484	21.147013	0.5
1.382285	21.068655	1.3
68.946075	52.799679	0.3
24.397801	36.423445	1.1
13.208669	33.974403	0.2
28.552275	41.899779	0.8
38.717673	77.577388	1.1
60.543483	37.257906	1.1
33.297982	54.765791	0.1
6.489686	41.999598	0.5
61.329306	50.566206	0.2
62.660209	60.790806	1.2
63.869476	67.361638	0.7
52.789561	70.800456	0.7
80.757592	81.019097	1.5
56.627933	60.170233	0.2
95.643073	99.822229	0.4
71.755731	74.313012	1.4
7.311519	98.62181	0.8

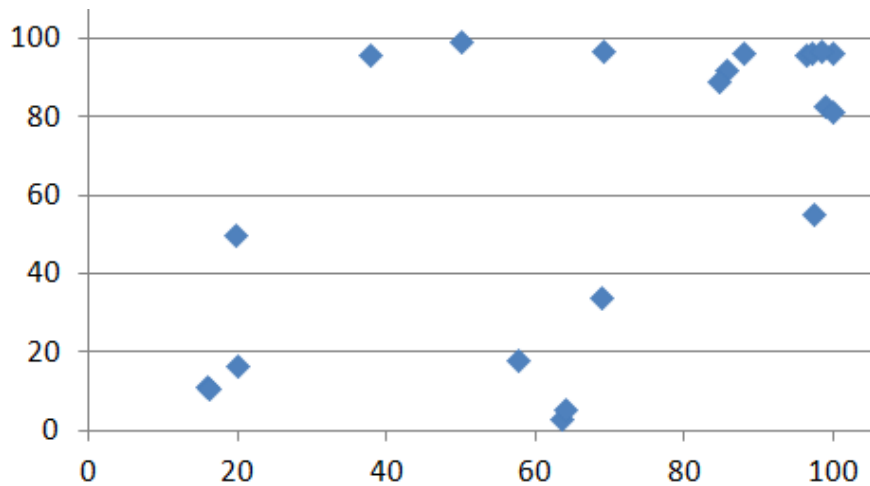


Στο παραπάνω σχήμα, βλέπουμε την τελική τοποθέτηση των πυλών. Παρατηρούμε πως υπάρχουν ελάχιστες επικαλύψεις και γενικά οι πύλες απέχουν και έχουν τοποθετηθεί ξεχωριστά. Επίσης, παρατηρούμε πως οι πύλες που είναι σχετικά κοντά δεν παρουσιάζουν υψηλό q .

- Για $e_1, e_2 = 0.05$.

Το placement που προτείνει ο TFPA.

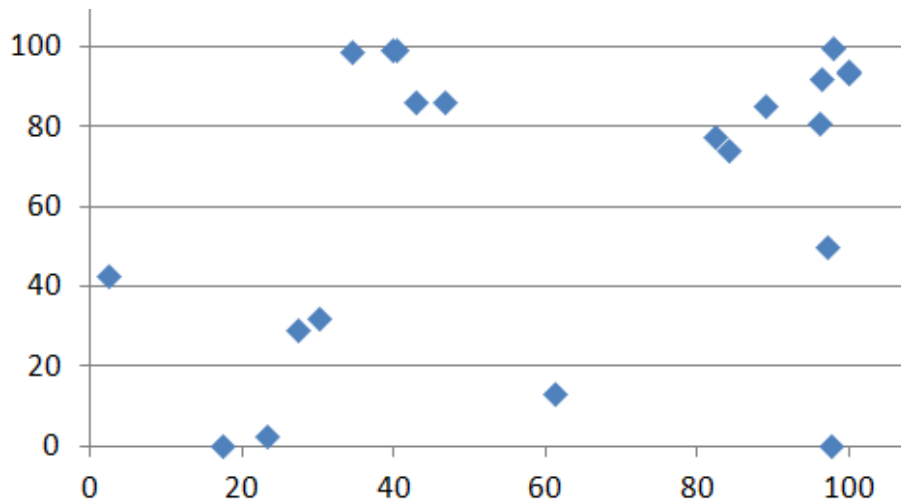
x	y	q
96.274765	95.668949	1.3
63.408894	2.909203	1.4
19.693725	50.138053	1.4
15.944147	11.168537	0.6
19.828681	16.747549	1.4
16.058928	10.707347	0.9
37.841328	95.858442	0.8
98.919142	82.996739	0.6
63.991484	5.322448	1.3
97.167037	96.590457	1.4
99.982666	81.138558	1.4
85.514287	92.191256	1.2
87.87815	96.342379	1.3
49.871669	99.440814	0.6
57.500561	17.79702	0.6
84.470329	89.245395	1.3
68.744953	34.019059	0.3
99.965281	96.360719	1
69.092687	97.008207	1.2
98.332293	96.664204	0.7
97.293614	55.28476	0.7



Εδώ παρατηρούμε πως κάποιες πύλες είναι πολύ κοντά με κίνδυνο τη δημιουργία κάποιου “hot spot”, αλλά μην ξεχνάμε πως τρέχουμε τον αλγόριθμο για κριτήρια σύγκλισης 0.05 και όχι 0.01, όπως προτείνεται από την μελέτη για καλύτερα και πιο κοντά στο τελικό placement αποτελέσματα. Ένα μείον είναι πως συσσωρεύονται κάποιες πύλες στα άκρα, όπου αυτό επιφέρει χειρότερα αποτελέσματα ως προς την ψύξη.

- Για $e_1=e_2=0.04$.

x	y	q
42.835267	86.405614	0.9
97.614888	0.134573	0.9
30.220663	32.030603	0.7
97.002872	50.138071	1
61.047209	13.172935	0.6
96.368625	91.80324	0.8
99.888414	93.584661	0.2
39.716528	99.415566	0.5
23.323869	2.73294	0.2
82.238178	77.492934	1.2
34.532953	98.90558	0.9
83.955114	74.145972	1.3
40.342769	99.290466	0.3
46.756072	86.032422	0.8
99.972559	93.785338	1.3
27.199272	29.355174	0.7
88.862504	85.145385	0.7
96.148158	81.051369	1.1
97.931692	99.515959	0.3
17.464732	0.056132	0.1
2.327959	42.596039	0.2

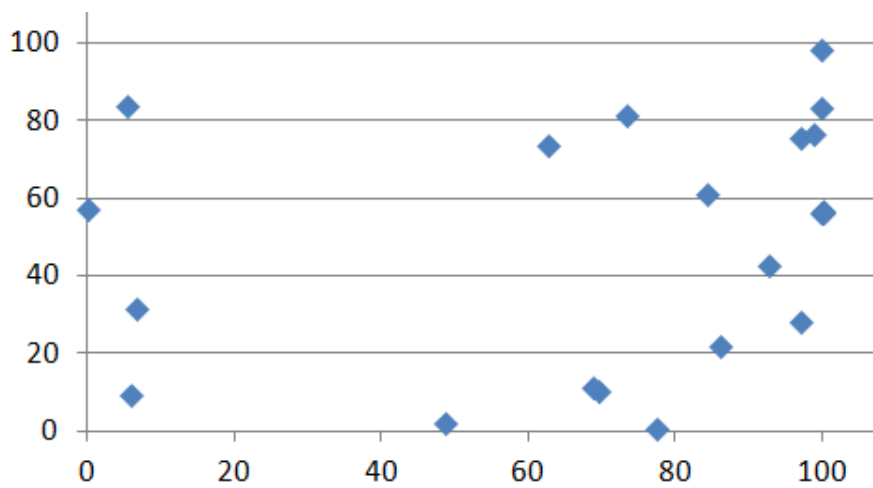


Και εδώ παρατηρούμε, ελάχιστες βέβαια, επικαλύψεις. Γενικά, οι πύλες έχουν απόσταση η μία από την άλλη. Και εδώ βλέπουμε πως συσσωρεύονται κάποιες πύλες στα άκρα, με κίνδυνο να έχουμε χειρότερα αποτελέσματα ως προς την ψύξη.

✚ Ακολουθούν παραδείγματα στα οποία έχουμε διαφορετική τιμή heat dissipation για κάθε πύλη ίση με 1 w.

- Για $e_1=e_2=0.01$.

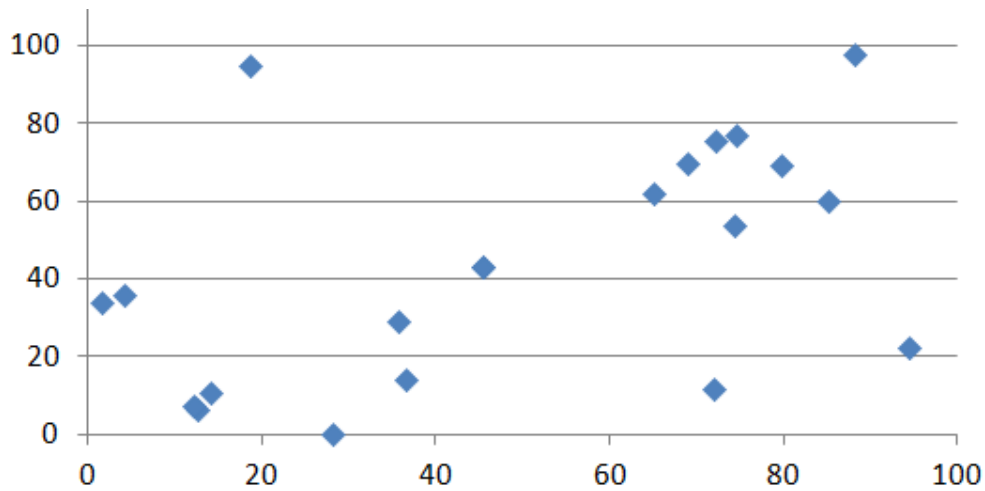
x	y
69.517219	10.419365
68.795212	11.004353
5.921356	9.545946
6.746929	31.727759
99.906112	98.229426
73.305484	81.503719
99.918379	56.304855
96.938666	75.600717
99.928463	83.131228
84.206956	61.235864
99.998792	56.57406
77.378257	0.78075
92.692818	42.471964
97.076807	28.127803
62.776819	73.644877
0.148183	56.958397
48.575338	1.967042
99.999959	56.167815
86.082033	21.679816
98.93774	76.390639
5.497346	83.603063



Αντίστοιχα, στο γράφημα αυτό και με $e_1=e_2=0.01$, βλέπουμε πως οι πύλες έχουν διαμοιραστεί ελαφρώς πιο ομοιόμορφα με σχεδόν μηδενικές επικαλύψεις, με κλινδουνο βέβαια οι πύλες στα άκρα να επιβαρύνουν την απόδοση ως προς την ψύξη.

- Για $e_1=e_2=0.05$.

x5	y5
28.197439	0.020332
14.206601	10.64255
94.426078	22.127229
4.14412	35.991503
12.539222	6.515202
12.230234	7.35079
35.725357	29.08139
1.508611	34.19808
72.045411	11.711093
45.438849	43.355358
45.41226	42.992665
18.700194	94.945253
65.007888	61.957542
74.267984	53.633422
68.893286	69.722227
72.111611	75.341215
74.500674	77.089479
85.142226	60.314553
79.66966	69.318016
88.235211	97.671339
36.52227	14.021454



Στο παραπάνω γράφημα παρατηρούμε πως δεν προκύπτει overlapping, πέρα από ένα σημείο . Είναι πιο ομοιόμορφο από άλλα παραδείγματα και οι περισσότερες πύλες είναι τοποθετημένες σε απόσταση.

Από το σύνολο των αποτελεσμάτων του αλγορίθμου, παρατηρούμε πως το φαινόμενο της επικάλυψης είναι ελάχιστο και οι πύλες σε γενικές γραμμές απέχουν μεταξύ τους για την αποφυγή δημιουργίας κάποιου hot spot.

5. Παραδείγματα και αποτελέσματα υπολογισμών.

Ο αλγόριθμος έχει υλοποιηθεί σε γλώσσα C και για τα παρακάτω παραδείγματα έχει τρέξει σε έναν 300 MHz Pentium II προσωπικό υπολογιστή. Δεδομένου ότι δεν υπάρχουν καθιερωμένα κυκλώματα αναφοράς MCM, η παρούσα μέθοδος εφαρμόζεται σε αρκετά παραδείγματα που έχουν δημοσιευθεί. Για να απλοποιηθεί η ανάλυση, όλα τα παραδείγματα που ακολουθούν θεωρούμε ότι έχουν το ίδιο packaging module και τις ίδιες συνθήκες ψύξης, αλλά μπορεί να έχουν διαφορετικές γεωμετρικές διαστάσεις. Ένα πρόγραμμα TAM [32],[33], βασισμένο σε επίλυση με σειρά Fourier, χρησιμοποιείται για να προβλέψει τις κατανομές της θερμοκρασίας του chip για το υπόστρωμα. Η παρακάτω εικόνα (fig.5) δείχνει ένα μοντέλο TAM ενός MCM. Τα διάφορα μέρη της δομής ενός MCM είναι τα: το πολυστρωματικό υπόστρωμα, το εποξειδικό στρώμα και το κανάλι θερμότητας.

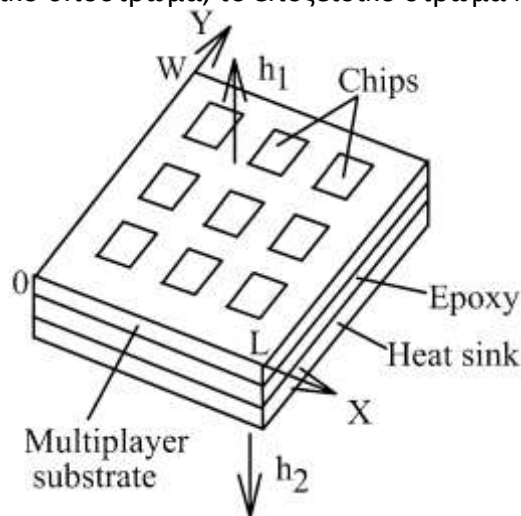


Figure 5. Schematic representation of an MCM model used here.

Σε κάθε στρώση, το υλικό θεωρείται ότι είναι γραμμικό, ισότροπο και ομοιογενές. Η ροή θερμότητας και θερμοκρασίας είναι συνεχής ανάμεσα στις στρώσεις. Η αγωγιμότητα του αέρα στον κενό χώρο ανάμεσα στην κεραμική επιφάνεια του MCM και του καλύμματος δεν λαμβάνεται υπόψη (η χειρότερη περίπτωση επίδρασης στα αποτελέσματα). Οι θερμικές αγωγιμότητες του πολυστρωματικού υποστρώματος, το εποξειδικό στρώμα και το κανάλι θερμότητας έχουν αντίστοιχα τις τιμές: $2.5 \cdot 10 \text{ W/mm C}$, $1.17 \cdot 10 \text{ W/mm C}$, και $1.95 \cdot 10 \text{ W/mm C}$. Επειδή η μέση ροή θερμότητας είναι πολύ υψηλή στα παραδείγματά μας, επιλέγουμε διάδοση θερμότητας με ταχύτητα 4 m/s για τη βάση, και 0.3 m/s για τις πλευρές. Οι μέσοι συντελεστές μεταφοράς θερμότητας είναι $3.26 \cdot 10 \text{ W/mm C}$ και $1.94 \cdot 10 \text{ W/mm C}$. Το ποσοστό αποτυχίας ενός chip υπολογίζεται με τον τύπο Arrhenius.

$$\lambda_T(i) = \lambda_o \times \exp \left[\frac{E_a}{k} \times \left(\frac{1}{298} - \frac{1}{T} \right) \right] \quad (15)$$

Όπου λ_T και λ_o είναι τα ποσοστά αποτυχίας σε $T \text{ K}$ και 298 K , αντίστοιχα.

E_a η ενέργεια ενεργοποίησης (eV).

k είναι η σταθερά Boltzmann.

Για να προσδιοριστεί ο ρυθμός αποτυχίας συστήματος ενός MCM, χρειάζεται να διευκρινιστούν διάφορες λειτουργικές παράμετροι. Χωρίς απώλεια της γενίκευσης, όλα τα chips θεωρείται ότι έχουν τα ίδια λ_0 και E_a , τα οποία είναι $1 \text{ Fit}(10^{-9}/\text{ώρα})$ και 1eV , αντίστοιχα. Ο ρυθμός αποτυχίας λ_S ενός MCM με m chips υπολογίζεται ως το άθροισμα των μεμονωμένων ρυθμών αποτυχίας των chips.

$$\lambda_S = \sum_{i=1}^m \lambda_T(i). \quad (16)$$

5.1 Παραδείγματα ενιαίων chips (uniform chips)

Τα πρώτα δύο παραδείγματα αποτελούνται από 12 και 16 chips αντίστοιχα. Είναι όλα από 1 watt, 5.1 mm τετράγωνα chips. Το υπόστρωμα και για τις δύο περιπτώσεις έχει μήκος, πλάτος και πάχος 30.5 mm, 30.5 mm και 5 mm, αντίστοιχα. Το αποτέλεσμα τοποθέτησης με βάση τον TFPA αλγόριθμο φαίνεται στην παρακάτω εικόνα και για τις δύο περιπτώσεις.

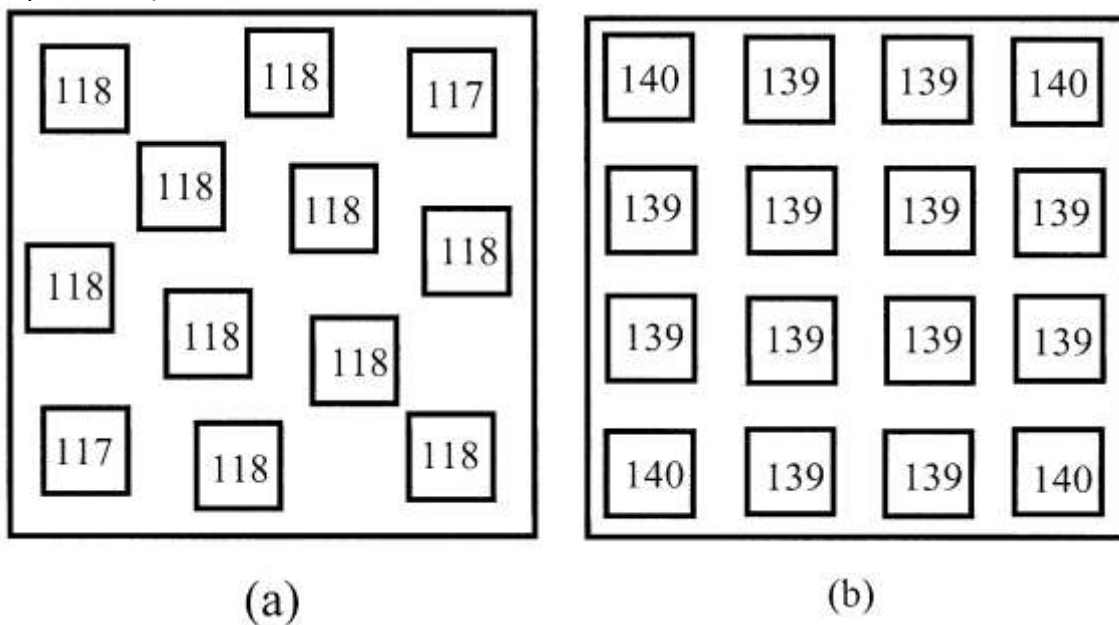


Figure 6. Placements with temperature ($^{\circ}\text{C}$) distribution obtained by TFPA: (a) 12 chips example and (b) 16 chips example.

Από τα αποτελέσματα, μπορεί κάποιος να δει ότι με τον τρόπο που τα chips είναι τοποθετημένα, οι θερμοκρασίες των chips είναι ομοιόμορφα κατανομημένες και δεν αποτελεί έκπληξη το γεγονός ότι τα αποτελέσματα και στις δύο περιπτώσεις είναι βέλτιστα.

5.2 Παραδείγματα με chips που έχουν διαφορετικές θερμικές αγωγιμότητες.

Στα επόμενα τέσσερα παραδείγματα, όλα τα chips θα έχουν το ίδιο μέγεθος υποστρώματος και αριθμό chips, αλλά διαφορετικά chips θα έχουν διαφορετικές θερμικές αγωγιμότητες σε κάθε περίπτωση. Θα χρησιμοποιηθεί ένα υπόστρωμα 50 mm^2 , με πάχος 5 mm. Τα chips και πάλι είναι τετράγωνα με άκρη 5 mm. Στον πίνακα I φαίνονται οι θερμικές αγωγιμότητες για κάθε chip.

Cases	Number of chips	Power dissipation value (W)
A	5	0.8, 1.3, 1.5, 1.5, 1.7
B	6	0.5, 0.8, 0.9, 1.1, 1.3, 1.5
C	8	0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
D	10	0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2

Figure 7. TABLE I - FOUR CASES WITH VARIOUS CHIP NUMBER AND POWER DISSIPATION.

Τα αποτελέσματα του TFPA συγκρίνονται με αυτά των Huang και Fu [15]. Τα αποτελέσματα της μέγιστης, ελάχιστης, μέσης θερμοκρασίας, μέγιστης διαφοράς θερμοκρασίας και το ποσοστό αποτυχίας συστήματος (T_{\max} , T_{\min} , T_{av} , ΔT_{\max} , λ_s), που λαμβάνονται από τους TFPA και FDMA για διάφορες περιπτώσεις, αναφέρονται συγκεντρωτικά στον πίνακα II.

Cases	A		B		C		D	
	FMFDA	TFPA	FMFDA	TFPA	FMFDA	TFPA	FMFDA	TFPA
$T_{\max}(\text{°C})$	119	94	108	86	82	68	81	78
$T_{\min}(\text{°C})$	64	64	52	52	44	44	47	47
$\Delta T_{\max} = T_{\max} - T_{\min}$	55	30	56	34	38	24	34	30
$T_{\text{av}}(\text{°C})$	100.2	82.4	82.2	69.5	62	55.8	64.6	63
λ_s (Fit)	25432	3830	8755	1610	1324	415	1550	1252

Figure 8. TABLE II - COMPARISONS OF TFPA AND FMFDA FOR THE CASES IN TABLE I

Οι θέσεις των chips που αναφέρονται στον FMFDA λήφθηκαν απευθείας από το [15], αλλά οι κατανομές της θερμοκρασίας αναλύονται κάτω από το συγκεκριμένο packaging και τις συνθήκες ψύξης. Όπως φαίνεται στον πίνακα II, τα αποτελέσματα του TFPA είναι

χαμηλότερα ή ίσα με αυτά του FMFDA. Συνεπώς, και οι τιμές T_{av} , ΔT_{max} και λ_s του TFPA, είναι επίσης χαμηλότερες από αυτές του FDMA. Ειδικά, η τιμή λ_s στον TFPA είναι μόνο 15% και 18% της αντίστοιχης τιμής στον FMFDA, για κάθε περίπτωση αντίστοιχα. Για πιο λεπτομερείς συγκρίσεις, δείτε τις παρακάτω εικόνες.

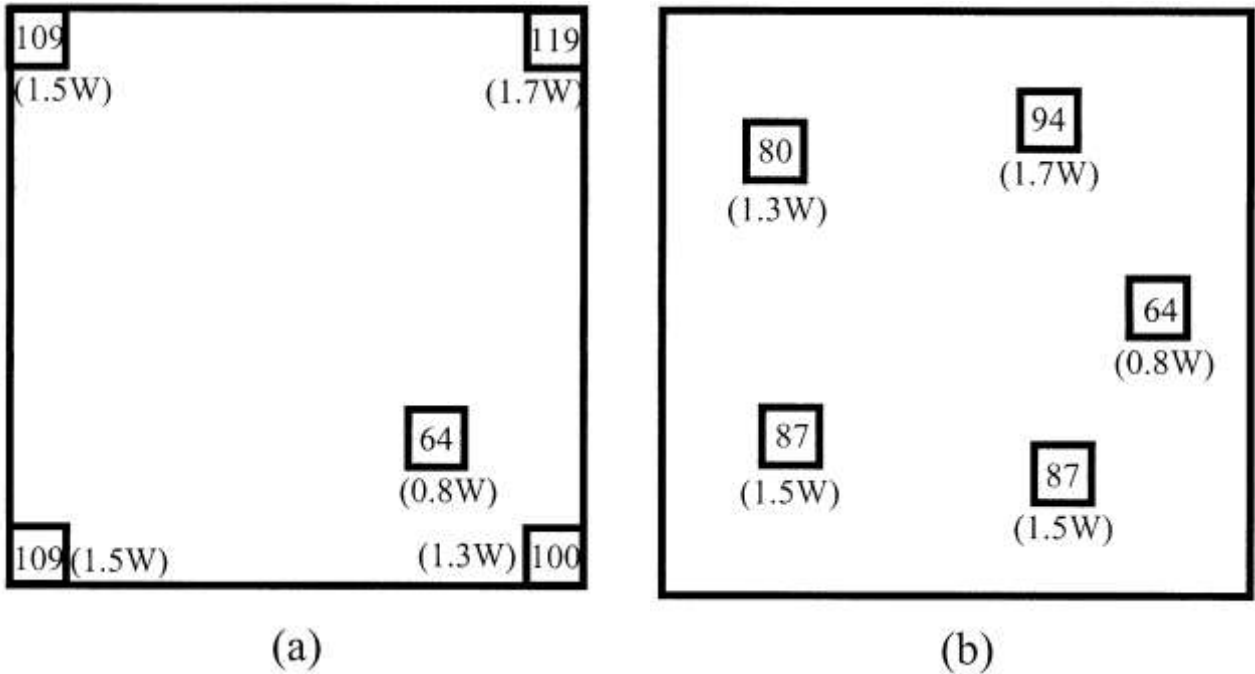


Figure 9.Placements with temperature (C) distribution and heat dissipations for case A: (a) placement result by FMFDA and (b) placement result by TFPA.

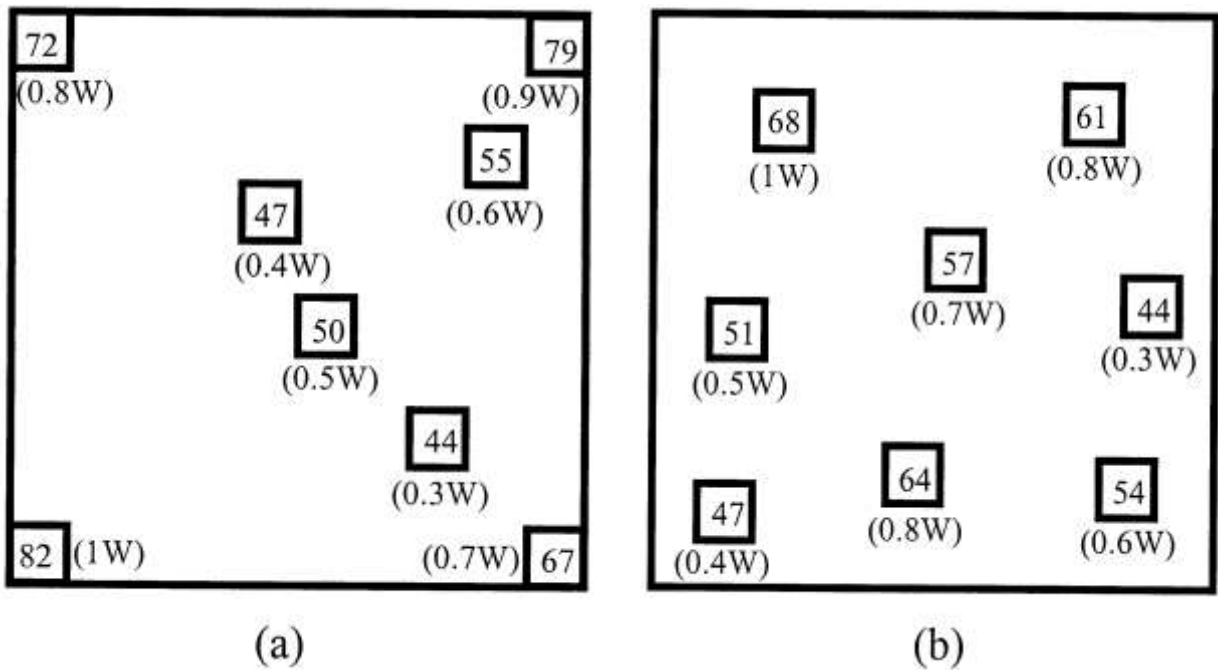


Figure 10.Placements with temperature (C) distribution and heat dissipations for case C: (a) placement result by FMFDA and (b) placement result by TFPA.

Παρατηρούμε στα σχήματα αυτά πως ο FMFDA τοποθετεί τα chips με τις μεγαλύτερες θερμικές αγωγιμότητες στις γωνίες του υποστρώματος για να αποφευχθεί η δημιουργία κάποιου hot spot στην κεντρική περιοχή του υποστρώματος. Ωστόσο, τα chips στις γωνίες συνήθως έχουν χειρότερα αποτελέσματα ψύξης, δεδομένου ότι δεν υπάρχει αρκετός χώρος για τη μεταφορά της θερμότητας που παράγεται. Ως αποτέλεσμα, δημιουργούνται placements με τέσσερα “hot spots” στις τέσσερις γωνίες των υποστρωμάτων, τα οποία παρουσιάζονται στο σχήμα 9. Αυτό μπορεί να προκαλέσει σοβαρή θερμική καταπόνηση στις γωνίες και μείωση της αξιοπιστίας του MCM. Από την άλλη πλευρά, ο TFPA τοποθετεί τα chips ξεχωριστά, έτσι ώστε κάθε chip να έχει αρκετό χώρο για να μεταφέρει τη θερμότητα που διαχέεται. Ως εκ τούτου, μπορεί να επιτευχθεί η τοποθέτηση με ένα ψυχρότερο και ομοιόμορφο θερμικό προφίλ. Παρ’ όλα αυτά, σε περιπτώσεις που η θερμότητα δεν συγκεντρώνεται στις γωνίες, ο FMFDA ενδέχεται να παράγει καλύτερα αποτελέσματα.

5.3 Παραδείγματα με άνισα chips.

Τα τελευταία δύο παραδείγματα είναι για τους Power PCs 603 και 604. Ο πίνακας III δείχνει το μέγεθος του chip και τις πληροφορίες για την ενέργεια και για τα δύο MCMs[34].

MCM		Processor	Bridge controller	Bridge buffer	Clock Distr.	SRAM	Tag RAM	Bus Buffer
PowerPC 603	Chip size (mm)	7.4×11.4	6.7×7.1	7.8×7.4	4.1×4.1	5.7×12.1	9.8×4.1	3.2×1.3
	Max. chip power (W)	3	0.75	0.5	0.9	1	0.7	0.4
PowerPC 604	Chip size (mm)	12.4×15.8	6.7×7.1	7.8×7.4	4.1×4.1	5.7×12.1	9.8×4.1	3.2×1.3
	Max. chip power (W)	13	0.75	0.5	0.9	1	0.7	0.4

Figure 11. TABLE III - CHIP SIZE AND POWER INFORMATION FOR POWERPC 603 AND 604 MCM

Τα chips έχουν διαφορετικά ύψη, πλάτη και απώλειες θερμότητας. Ο PowerPc 604 θεωρείται ένα χαρακτηριστικό παράδειγμα, καθώς εξαλείφει 13 watts, που είναι σχετικά υψηλότερα από των άλλων chips. Οι εικόνες που ακολουθούν δείχνουν τα layouts των MCMs και από τον PowerPc 603 αλλά και από τον 604, σε 44mm, 44 mm υπόστρωμα. Οι κατανομές της θερμοκρασίας, στις εικόνες αυτές, αναλύονται με βάση το παρόν packaging και τις παρούσες συνθήκες ψύξης.

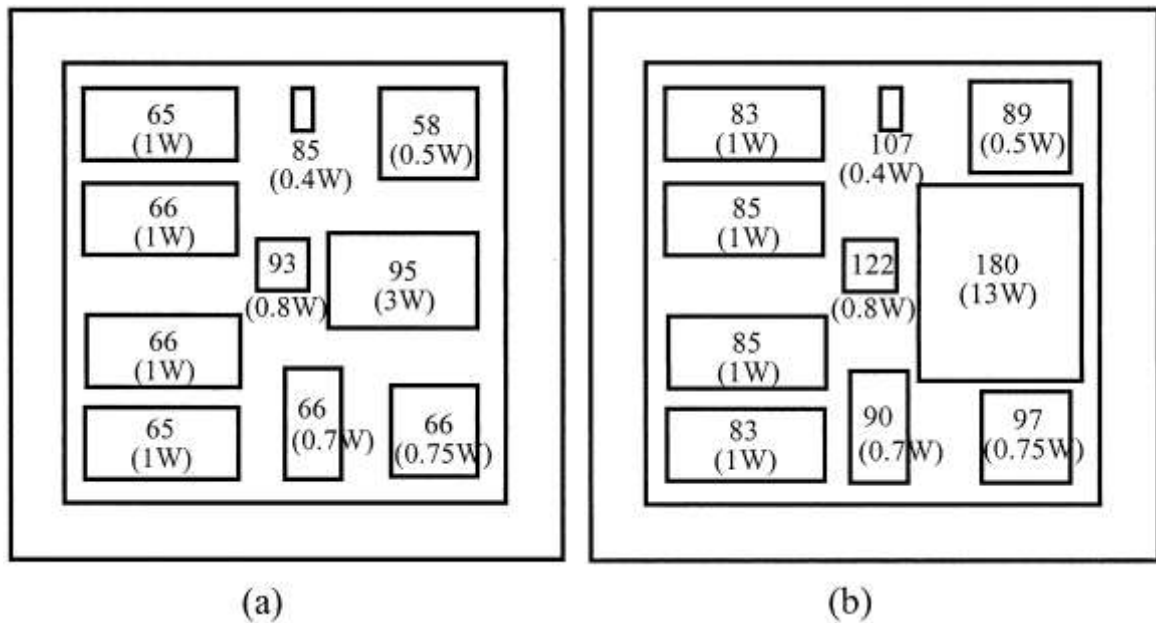


Figure 12. Chip layout with temperature (C) distribution and heat dissipations for PowerPC 603 and 604 in [31]: (a) PowerPC 603 and (b) PowerPC 604.

Οι αρχικές τοποθετήσεις για τους PowerPC 603 και 604, από τον TFPA φαίνονται στις παρακάτω εικόνες:

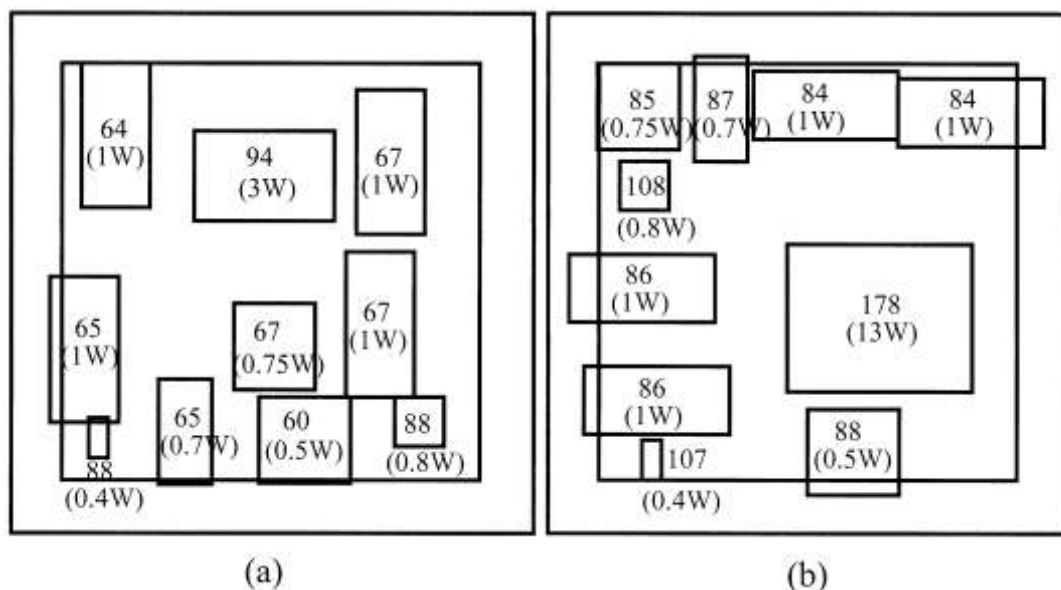


Figure 13. Initial placements obtained by TFPA: (a) PowerPC 603 and (b) PowerPC 604.

Προφανώς, και τα δύο placements των εικόνων (13a, 13b) είναι υψηλής ποιότητας, καθώς τα chips είναι τοποθετημένα ξεχωριστά και με ελάχιστες επικαλύψεις. Οπότε, μπορεί εύκολα κάποιος εξαλείφοντας τις επικαλύψεις να καταλήξει στην τελική τοποθέτηση των κελιών, σύμφωνα με τους κανόνες σχεδίασης. Μπορούμε να δούμε την τελική τοποθέτηση στις επόμενες εικόνες.

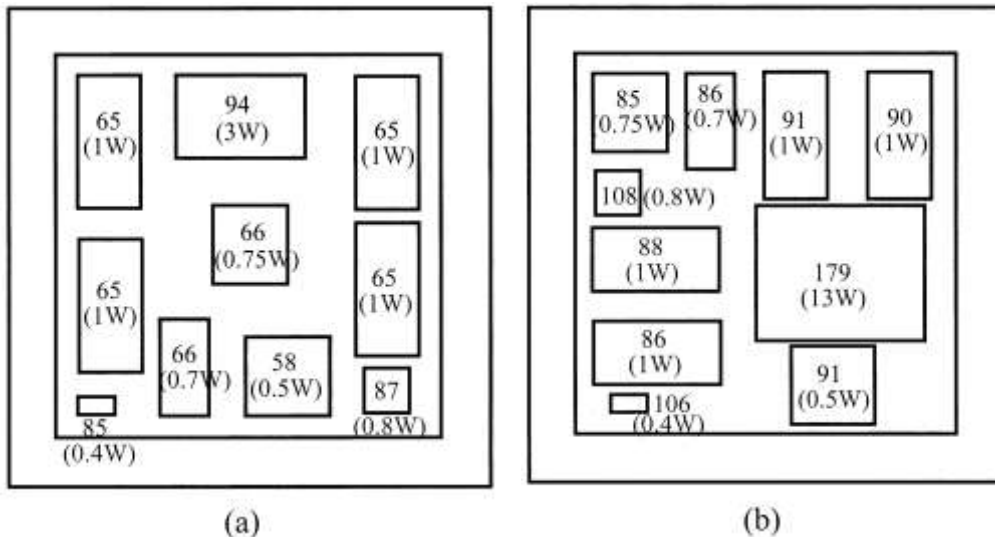


Figure 14. Final placement after design rules checking: (a) PowerPC 603 and b) PowerPC 604.

T_{max} , T_{min} , T_{av} , ΔT_{max} , λ_s για την χωροταξική τοποθέτηση στα σχήματα 12 και 14 συνοψίζονται στον πίνακα IV.

Cases	PowerPC 603		PowerPC 604	
	Original	TFPA	Original	TFPA
$T_{max}(^{\circ}C)$	94	94	180	179
$T_{min}(^{\circ}C)$	59	58	83	85
$\Delta T_{max} = T_{max} - T_{min}$	35	36	97	94
$T_{av}(^{\circ}C)$	72.5	71.7	102.1	101.1
λ_s (Fit)	4445	3788	659875	605712

Figure 15. TABLE IV - COMPARISONS OF THE PLACEMENTS OBTAINED BY TFPA.

Όπως βλέπουμε και στον πίνακα, τα αποτελέσματα του TFPA για τον PowerPC 603 είναι ελαφρώς καλύτερα. Το ποσοστό αποτυχίας του συστήματος έχει βελτιωθεί από το αρχικό κατά 15%. Για τον PowerPC 604, η τοποθέτηση με βάση τον TFPA έχει χαμηλότερες τιμές από τα πραγματικά αποτελέσματα κατά 8% περίπου.

6. Συμπέρασμα

Ένας θερμικός αλγόριθμος τοποθέτησης, βασιζόμενος στην αναλογία της θερμικής αγωγιμότητας, προτείνεται για τη σχεδίαση τοποθέτησης ενός MCM. Ο αλγόριθμος TFPA έχει αποδειχθεί πως είναι ένας αποτελεσματικός και αποδοτικός αλγόριθμος για το σκοπό αυτό. Οι πολυπλοκότητες του TFPA είναι $O(m)$ και $O(m^2)$, αντίστοιχα. Στον TFPA τα chips με επαναληπτικές διαδικασίες μετακινούνται σε περιοχές χαμηλότερης θερμοκρασίας. Σαν αποτέλεσμα, οδηγούμαστε σε ψυχρότερα και ομοιόμορφα προφίλ χωροταξικών τοποθετήσεων. Στις προηγούμενες ενότητες, είδαμε οχτώ παραδείγματα με τριών ειδών διαφορετικά chips και τα αποτελέσματα που έβγαλε για αυτά ο αλγόριθμος TFPA. Για τα ομοιόμορφα chips ο αλγόριθμος παράγει βέλτιστες τοποθετήσεις όσον αφορά στην αξιοπιστία. Για τον τύπο των chips ίδιου μεγέθους αλλά διαφορετικής απώλειας θερμότητας, ο TFPA παράγει τοποθετήσεις με ακόμα πιο ομοιόμορφη κατανομή θερμοκρασίας από τη μέθοδο με την οποία τον συγκρίναμε. Για τους τύπους των άνισων chips, ο TFPA παράγει τοποθετήσεις με μεγαλύτερη αξιοπιστία.

7. Παράρτημα

Στην ενότητα 2, η (8) δίνει την δύναμη που ασκείται στην d_i από την d_j .

$$\begin{aligned} Fz_{i,j} &= fz_{i,j} + \sum_{a=1}^{\infty} P_z(a) \\ &= fz_{i,j} + \sum_{a=1}^{\infty} [P_{zc}(a) + P_{zr}(a)] \end{aligned}$$

Όπου $P_{zc}(a) = \sum_{c=-a}^a (fz_{i,j}^{(-a,-c)} + fz_{i,j}^{(a,c)})$, $P_{zr}(a) = \sum_{r=-a+1}^{a-1} (fz_{i,j}^{(r,a)} + fz_{i,j}^{(-r,-a)})$, and $P_z(a) = P_{zc}(a) + P_{zr}(a)$ and z is either x or y .

Εδώ αναλύουμε το σφάλμα αποκοπής λόγω του $\sum_{a=6}^{\infty} P_z(a)$ στη σχέση (8).

Ακολουθως, η παραγωγή στην y κατεύθυνση είναι ίδια με αυτήν στην x , λαμβάνουμε υπόψη μόνο ως προς την x . Για διευκόλυνση, θεωρούμε το υπόστρωμα ως ένα τετράγωνο και το d_i ως το κεντρικό σημείο. Η διαφορά ως προς τις συντεταγμένες για τα d_i, d_j υπολογίζεται ως:

$$\Delta x_{i,j}^{(r,c)} = c + \delta x^{(c)} \quad (17a)$$

$$\Delta y_{i,j}^{(r,c)} = r + \delta y^{(a)} \quad (17b)$$

Όπου $\delta x^{(c)} = (-1)^c(x_j - x_i)$ και $\delta y^{(r)} = (-1)^r(y_j - y_i)$.

Προφανώς: $|\delta x^{(c)}| \leq 0.5, |\delta y^{(r)}| \leq 0.5$.

Το τετράγωνο της απόστασης μεταξύ των d_i και του d_j στο $(-a, -c)$ υπολογίζεται ως εξής:

$$\begin{aligned} & \left[\Delta r_{i,j}^{(-a,-c)} \right]^2 \\ &= \left(\Delta x_{i,j}^{(-a,-c)} \right)^2 + \left(\Delta y_{i,j}^{(-a,-c)} \right)^2 \\ &= \left(-c + \delta x^{(c)} \right)^2 + \left(-a + \delta y^{(a)} \right)^2 \\ &= (a^2 + c^2) \\ & \times \left[1 - 2 \frac{a\delta y^{(a)} + c\delta x^{(c)}}{a^2 + c^2} + \frac{(\delta x^{(c)})^2 + (\delta y^{(a)})^2}{a^2 + c^2} \right]. \quad (18) \end{aligned}$$

Since

$$\begin{aligned} \frac{(\delta x^{(c)})^2 + (\delta y^{(a)})^2}{a^2 + c^2} &< \frac{1}{2(a^2 + c^2)} \\ &\leq \frac{1}{2a^2} < 0.014 \ll 1 \quad \text{as } a \geq 6. \end{aligned}$$

Η σχέση (18) προσεγγίζεται ως:

$$\left[\Delta r_{i,j}^{(-a,-c)} \right]^2 \approx (a^2 + c^2)[1 - 2(h_1 + h_2)] \quad \text{as } a \geq 6 \quad (19a)$$

Όπου $h_1 = a\delta y^{(a)} / (a^2 + c^2)$ and $h_2 = c\delta x^{(c)} / (a^2 + c^2)$.

Επίσης,

$$\left[\Delta r_{i,j}^{(a,c)} \right]^2 \approx (a^2 + c^2)[1 + 2(h_1 + h_2)] \quad \text{as } a \geq 6. \quad (19b)$$

7.1. Υπολογισμός του $P_{xc}(\alpha)$

Σύμφωνα με τη σχέση (5), βρίσκουμε

$$f_{x_{i,j}}^{(-a,-c)} + f_{x_{i,j}}^{(a,c)} = -\alpha_i q_j \left[\frac{\Delta x_{i,j}^{(-a,-c)}}{(\Delta r_{i,j}^{(-a,-c)})^3} + \frac{\Delta x_{i,j}^{(a,c)}}{(\Delta r_{i,j}^{(a,c)})^3} \right]. \quad (20)$$

Substituting (17a), (17b), (19a) and (19b) into (20), yield (21)

$$\begin{aligned} f_{x_{i,j}}^{(-a,-c)} + f_{x_{i,j}}^{(a,c)} &= \frac{-\alpha_i q_j}{(a^2 + c^2)^{3/2}} \left\{ \frac{-c + \delta x^{(c)}}{[1 - 2(h_1 + h_2)]^{3/2}} + \frac{c + \delta x^{(c)}}{[1 + 2(h_1 + h_2)]^{3/2}} \right\} \\ &= \frac{-\alpha_i q_j}{(a^2 + c^2)^{3/2}} \left\{ \frac{(-c + \delta x^{(c)}) [1 + 2(h_1 + h_2)]^{3/2} + (c + \delta x^{(c)}) [1 - 2(h_1 + h_2)]^{3/2}}{[1 - 4(h_1 + h_2)^2]^{3/2}} \right\} \end{aligned} \quad (21)$$

Εφαρμόζοντας σειρές Taylor:

$$\begin{aligned} [1 \pm 2(h_1 + h_2)]^{3/2} &= 1 \pm 3(h_1 + h_2) \\ &\quad + \frac{3}{2}(h_1 + h_2)^2 \pm \dots \end{aligned} \quad (22)$$

$$[1 - 4(h_1 + h_2)^2]^{3/2} = 1 - 6(h_1 + h_2)^2 + \dots, \quad (23)$$

Οι υψηλότερης διάταξης όροι μπορούν να παραλειφθούν σύμφωνα με την παρακάτω ανάλυση:

$$\begin{aligned} \frac{3}{2}(h_1 + h_2)^2 &= \frac{3}{2} \left(\frac{c\delta x^{(c)} + a\delta y^{(a)}}{a^2 + c^2} \right)^2 \\ &< \frac{3}{8} \left(\frac{a + c}{a^2 + c^2} \right)^2 \leq \frac{3}{8} \left(\frac{1 + \sqrt{2}}{2a} \right)^2 \\ &\leq 0.015 \ll 1 \quad \text{as } a \geq 6 \end{aligned}$$

$$\text{και} \quad 6(h_1 + h_2)^2 \leq 0.06 \ll 1 \quad \text{as } a \geq 6.$$

$$(22), (23) \rightarrow [1 \pm 2(h_1 + h_2)]^{3/2} \approx 1 \pm 3(h_1 + h_2) \quad \text{as } a \geq 6 \quad (24)$$

$$[1 - 4(h_1 + h_2)^2]^{3/2} \approx 1 \quad \text{as } a \geq 6. \quad (25)$$

Βάζοντας τις (24) και (25) στην (21) έχουμε:

$$f_{i,j}^{(-a, -c)} + f_{i,j}^{(a, c)} \approx \frac{-\alpha_i q_j [2\delta x^{(c)} - 6c(h_1 + h_2)]}{(a^2 + c^2)^{3/2}}. \quad (26)$$

$$\begin{aligned} P_{xc}(a) &\approx \sum_{c=-a}^a \frac{-\alpha_i q_j [2\delta x^{(c)} - 6c(h_1 + h_2)]}{(a^2 + c^2)^{3/2}} \\ &= -\alpha_i q_j \left\{ \frac{2\delta x^{(0)}}{a^3} + \sum_{c=1}^a \left[\frac{2\delta x^{(c)} - 6c(h_1 + h_2)}{(a^2 + c^2)^{3/2}} \right. \right. \\ &\quad \left. \left. + \frac{2\delta x^{(c)} + 6c(h_1 - h_2)}{(a^2 + c^2)^{3/2}} \right] \right\} \\ &= -\alpha_i q_j \left[\frac{2\delta x^{(0)}}{a^3} + \sum_{c=1}^a \frac{4\delta x^{(c)} - 12ch_2}{(a^2 + c^2)^{3/2}} \right] \\ &= -\alpha_i q_j \left[\frac{2\delta x^{(0)}}{a^3} + 4 \times \sum_{c=1}^a \delta x^{(c)} \times \frac{a^2 - 2c^2}{(a^2 + c^2)^{5/2}} \right]. \end{aligned} \quad (27)$$

7.2. Υπολογισμός του $P_{xr}(\alpha)$.

$$\begin{aligned} P_{xr}(a) &= \sum_{r=-a}^a \left(f_{i,j}^{(-r, -a)} + f_{i,j}^{(r, a)} \right) \\ &\quad - \left(f_{i,j}^{(-a, -a)} + f_{i,j}^{(-a, a)} + f_{i,j}^{(a, -a)} + f_{i,j}^{(a, a)} \right). \end{aligned} \quad (28)$$

Το πρώτο άθροισμα της σχέσης (28):

$$\begin{aligned} &\sum_{r=-a}^a \left(f_{i,j}^{(-r, -a)} + f_{i,j}^{(r, a)} \right) \\ &\approx -\alpha_i q_j \left[\frac{2\delta x^{(0)}}{a^3} + 4 \times \sum_{r=1}^a \delta x^{(r)} \times \frac{a^2 - 2r^2}{(a^2 + r^2)^{5/2}} \right]. \end{aligned} \quad (29)$$

Ο δεύτερος όρος του δεξιού μέλους της (28):

$$\begin{aligned}
& f_{X_{i,j}}^{(-a,-a)} + f_{X_{i,j}}^{(-a,a)} + f_{X_{i,j}}^{(a,-a)} + f_{X_{i,j}}^{(a,a)} \\
&= -\alpha_i q_j \left[\frac{\Delta x_{i,j}^{(-a,-a)}}{\left(\Delta r_{i,j}^{(-a,-a)}\right)^3} + \frac{\Delta x_{i,j}^{(-a,a)}}{\left(\Delta r_{i,j}^{(-a,a)}\right)^3} \right. \\
&\quad \left. + \frac{\Delta x_{i,j}^{(a,-a)}}{\left(\Delta r_{i,j}^{(a,-a)}\right)^3} + \frac{\Delta x_{i,j}^{(a,a)}}{\left(\Delta r_{i,j}^{(a,a)}\right)^3} \right] \\
&\approx \frac{-\alpha_i q_j}{2\sqrt{2}a^3} \left(\frac{-a + \delta x^{(a)}}{1 - \frac{3}{2} \times \frac{\delta x^{(a)} + \delta y^{(a)}}{a}} + \frac{a + \delta x^{(a)}}{1 + \frac{3}{2} \times \frac{\delta x^{(a)} - \delta y^{(a)}}{a}} \right. \\
&\quad \left. + \frac{-a + \delta x^{(a)}}{1 - \frac{3}{2} \times \frac{\delta x^{(a)} - \delta y^{(a)}}{a}} + \frac{a + \delta x^{(a)}}{1 + \frac{3}{2} \times \frac{\delta x^{(a)} + \delta y^{(a)}}{a}} \right) \\
&= \frac{-\alpha_i q_j}{2\sqrt{2}a^3} \left[\frac{-\delta x^{(a)} - 3\delta y^{(a)}}{1 - \frac{9}{4} \times \left(\frac{\delta x^{(a)} + \delta y^{(a)}}{a}\right)^2} \right] \\
&\approx \frac{\alpha_i q_j \delta x^{(a)}}{\sqrt{2}a^3} \quad \text{as } a \geq 6. \tag{30}
\end{aligned}$$

Substituting (29) and (30) into (28), we find

$$\begin{aligned}
P_{xr}(a) \approx & -\alpha_i q_j \left[\frac{2\delta x^{(0)}}{a^3} - \frac{\delta x^{(a)}}{\sqrt{2}a^3} + 4 \right. \\
& \left. \times \sum_{r=1}^a \frac{\delta x^{(r)} \times (a^2 - 2r^2)}{(a^2 + r^2)^{5/2}} \right] \quad \text{as } a \geq 6. \tag{31}
\end{aligned}$$

7.3. Υπολογισμός του αθροίσματος $|\sum_{a=6}^{\infty} P_z(a)/f_{x_{i,j}}|$

$$|(\sum_{a=6}^{\infty} P_z(a))/f_{x_{i,j}}|$$

(27) and (31) into (8),

$$P_x(a) \approx -\alpha_i q_j \left[\frac{4\delta x^{(0)}}{a^3} - \frac{\delta x^{(a)}}{\sqrt{2}a^3} + 8 \times \sum_{c=1}^a \frac{\delta x^{(c)} \times (a^2 - 2c^2)}{(a^2 + c^2)^{5/2}} \right] \quad \text{as } a \geq 6. \quad (32)$$

Προφανώς, οι παραπάνω σειρές συγκλίνουν απόλυτα. Συνοψίζοντας, το $P_x(a)$ για $a=6$ έως 1000 στο Matlab μας δίνει:

$$\sum_{a=6}^{\infty} P_x(a) \approx 0.0043\alpha_i q_j \delta x^{(0)}. \quad (33)$$

Διαιρώντας με $f_{x_{i,j}}$ παίρνουμε:

$$\frac{\sum_{a=6}^{\infty} P_z(a)}{f_{x_{i,j}}} = \frac{0.0043\alpha_i q_j \delta x^{(0)}}{-\alpha_i q_j \times \frac{\delta x^{(0)}}{(\Delta r_{i,j})^3}} = -0.0043 (\Delta r_{i,j})^3. \quad (34)$$

Αφού

$$(\Delta r_{i,j})^3 < (0.5^2 + 0.5^2)^{3/2} = 0.3536$$

Τότε

$$\left| \frac{\sum_{a=6}^{\infty} P_z(a)}{f_{x_{i,j}}} \right| < 0.0015. \quad (35)$$

Καταλήγουμε, λοιπόν, στο ότι το σφάλμα αποκοπής λόγω του $\sum_{a=6}^{\infty} P_z(a)$ δίνει σφάλμα μικρότερο του 0.15% σε σχέση με το $f_{x_{i,j}}$.

8. Παραπομπές

- [1]MCM : http://en.wikipedia.org/wiki/Multi-chip_module
- [2]figure 1 - hot spots : <http://chipdesignmag.com/display.php?articleId=256>
- [3] G. Wippler, M. Wiesel, and D. Mlynski, "A combined force and cut algorithm for hierarchical VLSI layout," in Proc. 22th Design Automat. Conf., 1982, pp. 671–677.
- [4] W. Maly and A. P. Piotrowski, "Heat exchange optimization technique for high-power hybrid ICs," IEEE Trans. Comp., Hybrids, Manufact. Technol., vol. 2, pp. 226–231, June 1979.
- [5] A. H. Mayer, "Computer-aided thermal design of avionics for optimum reliability and minimum life cycle cost," Air Force Flight Dynamics Laboratory, Tech. Rep. AFFDL-TR-78-48, 1978.
- [6] M. Pecht and J. Naft, "Thermal reliability management in PCB design," in Proc. 1987 Annu. Rel. Maintainability Symp., 1987, pp. 27–29.
- [7] D. Dancer and M. Pecht, "Reliability optimization technique for convectively electronics," IEEE Trans. Reliability, vol. 38, pp. 199–205, June 1989.
- [8] M. Osterman and M. Pecht, "Component placement for reliability on conductivity cooled printed wiring boards," Trans. ASME J. Electron. Packag., vol. 111, pp. 149–156, 1989.
- [9] R. Eliasi, T. Elperin, and A. Bar-Cohen, "Monte Carlo thermal optimization of populated printed circuit board," IEEE Trans. Comp, Hybrids, Manufact. Technol., vol. 13, pp. 953–960, Dec. 1990.
- [10] M. D. Osterman, "A physics of failure approach to component placement," Trans. ASME J. Electron. Packag., vol. 114, pp. 305–309, 1992.
- [11] C. N. Chu and D. F. Wong, "A matrix synthesis approach to thermal placement," IEEE Trans. Computer-Aided Design, vol. 17, pp. 1166–1174, Nov. 1998.
- [12] C.-H. Tsai and S.-M. Kang, "Cell-level placement for improving substrate thermal distribution," IEEE Trans. Computer-Aided Design, vol. 19, pp. 253–266, Feb. 2000
- [13] K. Y. Chao and D. F. Wong, "Thermal placement for high-performance multi-chip modules," in Proc. Int. Conf. Comput. Design, Austin, TX, 1995, pp. 218–223.

- [14] M. C. Tang and J. D. Carothers, "Consideration of thermal constraints during multichip module placement," *Electron. Lett.*, vol. 33, no. 12, pp. 1043–1045, 1997.
- [15] Y.-J. Huang and S.-L. Fu, "Thermal placement design for MCM applications," *Trans. ASME J. Electron. Packag.*, vol. 122, pp. 115–120, 2000.
- [16] C. Beebe, J. D. Carothers, and A. Ortega, "MCM placement using a realistic thermal model," in *Proc. 10th Great Lakes Symp. VLSI, 2000*, pp. 189–192.
- [17] J. Lee, J. H. Chou, and S. L. Fu, "Reliability and wireability optimizations for module placement on convectively cooled printed wiring board," *Integration, VLSI J.*, vol. 18, no. 2&3, pp. 173–186, 1995.
- [18] M. D. Osterman and M. Pecht, "Placement for reliability and routability of convectively cooled PWBs," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 734–744, July 1990.
- [19] N. V. Queipo, J. A. C. Humphrey, and A. Ortega, "Multiobjective optimal placement of convectively cooled electronic components on printed wiring boards," *IEEE Trans. Comp., Packag., Manufact. Technol. A*, vol. 21, pp. 142–153, Mar. 1998.
- [20] N. V. Queipo and G. F. Gil, "Multiobjective optimal placement of convectively and conductively cooled electronic components on printed wiring boards," *Trans. ASME J. Electron. Packag.*, vol. 122, pp. 152–159, 2000.
- [21] J. Lee and J. H. Chou, "Hierarchical placement for power hybrid circuits under reliability and wireability constraints," *IEEE Trans. Rel.*, vol. 45, pp. 200–207, June 1996.
- [22] N. Quinn and M. Breuer, "A forced directed component placement procedure for printed circuit boards," *IEEE Trans. Circuits Syst.*, vol. 26, pp. 377–388, June 1979.
- [23] F. Mo, A. Tabbara, and R. K. Brayton, "A force-directed macro-cell placer," in *Proc. Int. Conf. Computer-Aided Design, 2000*, pp. 177–180.
- [24] http://www.astro.auth.gr/~kokkotas/lesson/na_book/node23.html
- [25] <http://en.wikipedia.org/wiki/NP-complete>
- [26] http://en.wikipedia.org/wiki/Karp's_21_NP-complete_problems
- [27] http://en.wikipedia.org/wiki/Cook%E2%80%93Levin_theorem

- [28] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. New York: Plenum, 1972.
- [29] D. J. Dean, *Thermal Design of Electronic Circuit Boards and Packages*. Edinburgh, Scotland: Electrochemical Publications, 1985.
- [30] A. L. Palisoc and C. C. Lee, "Exact thermal representation of multilayer rectangular structures by infinite plate structures using the method of images," *J. Appl. Phys.*, vol. 12, no. 64, pp. 6851–6857, 1988.
- [31] Y.-K. Cheng and S.-M. Kang, "A temperature-aware simulation environment for reliable ULSI chip design," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 1211–1220, Oct. 2000.
- [32] G. N. Ellison, *Thermal Computations for Electronic Equipment*. New York: Van Nostrand Reinhold, 1983.
- [33] G. E. Ellison, "Thermal analysis of circuit boards and microelectronic components using an analytical solution to the heat conduction equation," in *Proc. 12th IEEE SEMI-THERM Symp.*, 1996, pp. 144–150.
- [34] T. D. Yuan, "Thermal management in PowerPC microprocessor multichip modules applications," in *Proc. 13th IEEE SEMI-THERM Symp.*, 1997, pp. 247–256.