



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ  
ΤΟΜΕΑΣ ΦΥΣΙΚΗΣ

---

Ανάπτυξη προγράμματος σε Java για τον γρήγορο  
υπολογισμό χωρητικότητας σε ανιχνευτές πυριτίου  
για την αναβάθμιση του ανιχνευτή τροχιών του  
πειράματος C.M.S. στο Super L.H.C.

---

## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ασσιούρα Παναγιώτη

Επιβλέποντες : Δημήτρης Λουκάς  
Διευθυντής Ερευνών Ε.Κ.Ε.Φ.Ε Δημόκριτος

Θεόδωρος Αλεξόπουλος  
Καθηγητής Ε.Μ.Π

Αθήνα, 12 Μαρτίου 2015





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών  
Τομέας Φυσικής

**Ανάπτυξη προγράμματος σε Java για τον γρήγορο υπολογισμό χωρητικότητας σε ανιχνευτές πυριτίου για την αναβάθμιση του ανιχνευτή τροχιών του πειράματος C.M.S. στο Super L.H.C.**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**Ασσιούρα Παναγιώτη**

**Επιβλέποντες:** Δημήτρης Λουκάς  
Διευθυντής Ερευνών Ε.Κ.Ε.Φ.Ε. Δημόκριτος

Θεόδωρος Αλεξόπουλος  
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Δημήτρης Λουκάς    Θεόδωρος Αλεξόπουλος    Σταύρος Μαλτέζος

Διευθυντής Ερευνών ,    Καθηγητής Ε.Μ.Π.    Αναπληρωτής  
Ε.Κ.Ε.Φ.Ε. Δημόκριτος    Καθηγητής Ε.Μ.Π.

Αθήνα, 12 Μαρτίου 2015

(Υπογραφή)

.....

**Ασσιούρας Παναγιώτης**

Διπλωματούχος Σχολής Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών Ε.Μ.Π.

© 2014 – All rights reserved





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών  
Τομέας Φυσικής

Copyright ©–All rights reserved Ασιούρας Παναγιώτης,  
Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.



# Ευχαριστίες

Για την διεκπεραίωση της διπλωματικής μου εργασίας θα ήθελα να ευχαριστήσω τον κ.Δημήτρη Λουκά για την εμπιστοσύνη που μου έδειξε και την πολύτιμη καθοδήγηση. Επίσης θα ήθελα να ευχαριστήσω την ομάδα του κ.Λουκά στο Ε.Κ.Ε.Φ.Ε Δημόκριτο και κυρίως τον Παναγιώτη Ρούσο για τη συνεργασία και τις πολύτιμες συμβουλές.

Επίσης θα ήθελα να ευχαριστήσω τον κ.Θεόδωρο Αλεξόπουλο για τις γνώσεις που μας μετέδωσε κατά την διάρκεια των σπουδών μας και για την στήριξη του στη διεκπεραίωση της διπλωματικής μου εργασίας.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη τους όλα τα χρόνια των σπουδών.



# Περίληψη

Οι ανιχνευτές πυριτίου χρησιμοποιούνται στη Φυσική υψηλών ενεργειών για ανίχνευση σωματιδίων με υψηλή διακριτική ικανότητα όπως ο ανιχνευτής τροχιών του CMS ( Compact Muon Solenoid) στο CERN ο οποίος αποτελείται αποκλειστικά από ανιχνευτές πυριτίου. Οι ανιχνευτές αυτοί σχεδιάζονται σε διάφορους τύπους ανάλογα με την λειτουργία που προορίζονται. Δύο βασικοί τύπου ανιχνευτών πυριτίου είναι οι ανιχνευτές μικρο-πλακιδίων(pixel) και μικρο-λωρίδων(strip).

Στη παρούσα διπλωματική αναλύουμε μια αριθμητική μέθοδο για την επίλυση της εξίσωσης Laplace για το δυναμικό σε ανιχνευτές με επίπεδη γεωμετρία όπως έναν ανιχνευτή πυριτίου τύπου μικρο-πλακιδίων (pixel) ή μικρο-λωρίδων (strip). Η αριθμητική μέθοδος αναπτύχθηκε στον Δημόκριτο και περιγράφεται αναλυτικά στις αναφορές [1], [2]. Με τη μέθοδο αυτή γίνεται αριθμητική εκτίμηση του ηλεκτρικού πεδίου ανάλογα με την γεωμετρία μέσα στον όγκο ενός πλήρους απογυμνωμένου ανιχνευτή. Επίσης μέσω της μεθόδου αυτής υπολογίζονται οι χωρητικότητες που αναπτύσσονται μεταξύ των διόδων και μεταξύ των διόδων και του πίσω επιπέδου. Οι χωρητικότητες αυτές είναι πολύ σημαντικές στον σχεδιασμό των ανιχνευτών καθώς σχετίζονται με το θόρυβο και τα φαινόμενα crosstalk ανάμεσα στα pixel ή τα strip. Η γνώση των χωρητικοτήτων ανάλογα με τη γεωμετρία είναι πολύ χρήσιμη για το βέλτιστο σχεδιασμό του ανιχνευτή καθώς και των εξωτερικών ηλεκτρονικών.

Βασιζόμενοι στην παραπάνω αριθμητική μέθοδο υλοποιήσαμε ένα πρόγραμμα σε Java για τον γρήγορο υπολογισμό χωρητικοτήτων σε ανιχνευτές πυριτίου. Το πρόγραμμα αυτό υλοποιήθηκε στο Ε.Κ.Ε.Φ.Ε Δημόκριτος (Ινστιτούτο Πυρηνικής και Σωματιδιακής Φυσικής) για την αναβάθμιση του ανιχνευτή τροχιών του πειράματος CMS στο Super L.H.C.

Στο πρώτο κεφάλαιο εξετάζουμε τους τρόπους αλληλεπίδρασης της ακτινοβολίας και ύλης καθώς και μερικά βασικά στοιχεία της θεωρίας ανιχνευτών.

Στο δεύτερο κεφάλαιο γίνεται εκτενέστερη αναφορά στους ανιχνευτές πυριτίου οι οποίοι αποτελούν και αντικείμενο της παρούσας διπλωματικής εργασίας. Εξετάζουμε τον τρόπο λειτουργίας τους και κατασκευής τους καθώς και τους τύπους των θορύβων που υπεισέρχονται στις μετρήσεις με το συγκεκριμένο τύπου ανιχνευτή. Επίσης εξετάζουμε τον τρόπο με τον οποίο λειτουργούν οι ανιχνευτές αυτοί σε πειράματα της φυσικής υψηλών ενεργειών όπως στον ανιχνευτή τροχιών του C.M.S. και αναφέρουμε τις αναβαθμίσεις που θα λάβουν χώρα τα επόμενα χρόνια, ώστε να είναι δυνατή η ικανοποιητική λειτουργία του ανιχνευτή τροχιών του C.M.S στις αυξημένες απαιτήσεις του Super L.H.C.

Στο τρίτο κεφάλαιο εξετάζουμε τον αλγόριθμο του γρήγορου μετασχηματισμού Fourier καθώς αποτελεί ένα σημαντικό κομμάτι του προγράμματος και της μεθόδου με τα οποία γίνονται οι υπολογισμοί των χωρητικοτήτων.

Στο τέταρτο κεφάλαιο εξετάζουμε την αριθμητική μέθοδο για την επίλυση της εξίσωσης Laplace για την εκτίμηση του ηλεκτρικού πεδίου μέσα στον όγκο του ανιχνευτή

και τον τρόπο με τον οποίο γίνεται ο υπολογισμός της συνολικής χωρητικότητας , της χωρητικότητας σε σχέση με τα γειτονικά pixel ή strip και της χωρητικότητας σε σχέση με το πίσω επίπεδο του ανιχνευτή.

Στο πέμπτο κεφάλαιο γίνεται παρουσίαση του προγράμματος που δημιουργήσαμε καθώς και τού τρόπου λειτουργίας του.

Στο έκτο κεφάλαιο παρουσιάζουμε αναλυτικά αποτελέσματα τα οποία προέκυψαν από την αριθμητική μέθοδο για διάφορες γεωμετρίες pixel και strip. Οι γεωμετρίες αυτές αντιστοιχούν στις αναλογίες διαστάσεων που έχουν επιλεγεί ως οι πιο πιθανές για την αναβάθμιση στη φάση 2 του ανιχνευτή τροχιών του CMS η οποία αναμένεται να ολοκληρωθεί στα τέλη 2024.

Τέλος στο παράρτημα παραθέτουμε αναλυτικά τον κώδικα του προγράμματος σε Java.

# Abstract

The silicon detectors are used in high energy physics for detecting particles with high resolution. For example the tracker of the CMS (Compact Muon Solenoid) experiment at CERN, consists exclusively of silicon detectors. These detectors are designed into various types depending on the operation intended. Two common types of silicon detectors that have been used are micro-pixel detectors (pixel) and micro-strip (strip).

In this thesis we analyze a numerical method for solving the Laplace equation for the potential in detectors with planar geometry as a pixel or strip silicon detector. A more analytical description on this method can be found in the bibliography [1] [2]. This method is based on a numerical estimation of the electric field in a fully depleted detector which depends on the geometry of the detector. Also by this method the capacitances between the diodes as well as the capacitances between the diodes and the backplane are calculated. The capacitances are very important for the design of the detectors as they are related to noise effects as well as crosstalk phenomena between the neighboring pixel or strip. The knowledge of the capacitances depending on the geometry can be very useful for the optimal design of the detector and of the readout electronics. Based on the above numerical method we have implemented a program in Java for fast calculation of capacitances in silicon detectors. This program was implemented in NCSR Demokritos (Nuclear and Particle Physics Institute) for the upgrade of the CMS tracker for the Super LHC.

In the first chapter we examine how radiation interacts with matter and some basic principles of detectors' theory.

The second chapter elaborates on silicon detectors which are the subject of this thesis. We analyze the way they operate and their construction as well as the types of noise involved in measurements with the probe type specific. Also we analyze the operation of these detectors in experiments of high energy physics such as the CMS experiment. Also a brief analysis of the basic aspects of the CMS tracker upgrade is made.

In the third chapter we examine the algorithm of fast Fourier transform. This algorithm is important for the numerical estimation by which the calculations of capacitances are made as well as the implementation of the program.

In the fourth chapter we examine the numerical method for solving the Laplace equation so we can estimate the electric field within the volume of the detector and the method in which the calculation of the total capacity, the capacity in respect to the neighboring pixel or strip and the capacity with respect to the backplane is made.

The fifth chapter is a presentation of the program that has been implemented for the fast calculation of the capacitances.

The sixth chapter presents analytic results obtained from the numerical method for different geometries both for pixel and strip type. The geometries from which the estimations have been made correspond to the aspect ratio that has been considered as the most possible for the phase 2 CMS tracker upgrade that is expected to end in 2024.

Finally in the appendix there is an analytical description of the java code that implements the program



# Περιεχόμενα

Περίληψη	3
Abstract	5
Περιεχόμενα	7
<b>1 Θεωρία ανιχνευτών</b>	<b>9</b>
1.1 Εισαγωγή	9
1.2 Αλληλεπίδραση ακτινοβολίας και ύλης	9
1.2.1 Αλληλεπίδραση φορτισμένων σωματιδίων	9
1.2.2 Αλληλεπίδραση βαρέων φορτισμένων σωματιδίων	10
1.2.3 Αλληλεπίδραση ηλεκτρονίων-ποζιτρονίων με την ύλη	10
1.2.4 Αλληλεπίδραση φωτονίων και ύλης	12
<b>2 Ανιχνευτές πυριτίου στη φυσική υψηλών ενεργειών</b>	<b>17</b>
2.1 Εισαγωγή	17
2.2 Βασική λειτουργία ανιχνευτών πυριτίου	19
2.3 Σχηματισμός Σήματος	23
2.3.1 Διακυμάνσεις του σήματος	26
2.4 Δίοδοι	27
2.4.1 Δημιουργία περιοχών με περίσσεια ηλεκτρονίων ή οπών	27
2.5 Κίνηση φορέων	30
2.6 Κατασκευή ανιχνευτών πυριτίου	31
2.7 Ενίσχυση σήματος	35
2.8 Ηλεκτρονικός θόρυβος	38
2.8.1 Ενίσχυση θορύβου	39
2.8.2 Χωρητικότητες ως πηγές θορύβου	41
2.8.3 Πολυπλοκότερες διατάξεις	42
2.9 Ανιχνευτής CMS	44
2.9.1 Ο ανιχνευτής τροχιών του πειράματος CMS πριν την αναβάθμιση	45
2.9.2 Περιορισμοί του ανιχνευτή τροχιών	46
2.9.3 Αναβάθμιση του ανιχνευτή τροχιών του CMS	48
2.9.4 Αναβάθμιση του ανιχνευτή τροχιών CMS (Φάση I)	50
2.9.5 Αναβάθμιση του ανιχνευτή τροχιών CMS (Φάση II)	52
<b>3 Γρήγορος Μετασχηματισμός Fourier (FFT)</b>	<b>57</b>
3.1 Εισαγωγή	57
3.2 Μετασχηματισμός Fourier	57

3.3	Βασικές ιδιότητες Μετασχηματισμού Fourier . . . . .	58
3.4	Μετασχηματισμός Fourier διακριτής συνάρτησης . . . . .	58
3.5	Διακριτός μετασχηματισμός Fourier (DFT) . . . . .	59
3.6	Γρήγορος μετασχηματισμός Fourier (FFT) . . . . .	60
3.6.1	Λήμμα Danielson-Lanczos . . . . .	62
3.6.2	Διαγράμματα ροής δεδομένων . . . . .	67
3.7	Υπορουτίνα FFT . . . . .	69
3.8	Παραλλαγές του FFT . . . . .	72
3.8.1	FFT για 2 και για περισσότερες διαστάσεις . . . . .	72
<b>4</b>	<b>Εφαρμογή του FFT στην επίλυση της εξίσωσης Laplace</b>	<b>75</b>
4.1	Εισαγωγή . . . . .	75
4.2	Αριθμητική επίλυση της εξίσωσης Laplace . . . . .	75
4.2.1	Ανιχνευτές με επίπεδη γεωμετρία τύπου pixel . . . . .	75
4.2.2	Ανιχνευτές με επίπεδη γεωμετρία τύπου strip . . . . .	81
4.3	Αλγόριθμος επίλυσης εξίσωσης Laplace . . . . .	83
<b>5</b>	<b>Πρόγραμμα γρήγορου υπολογισμού χωρητικότητας σε Java</b>	<b>89</b>
5.1	Εισαγωγή . . . . .	89
5.2	Πρόγραμμα υπολογισμού χωρητικότητας σε Java . . . . .	89
<b>6</b>	<b>Αποτελέσματα της μεθόδου για διάφορες γεωμετρίες</b>	<b>93</b>
6.1	Εισαγωγή . . . . .	93
6.2	Διάταξη τύπου pixel . . . . .	93
6.3	Διάταξη τύπου strip . . . . .	98
6.4	Συμπεράσματα . . . . .	100
	<b>Βιβλιογραφία</b>	<b>101</b>
<b>A'</b>	<b>Αναλυτικά ο κώδικας του προγράμματος σε Java</b>	<b>105</b>
A'.0.1	Κλάση pixel.java . . . . .	105
A'.0.2	Κλάση strip.java . . . . .	122
A'.0.3	Κλάση cap.java . . . . .	133

# Κεφάλαιο 1

## Θεωρία ανιχνευτών

### 1.1 Εισαγωγή

Στο κεφάλαιο αυτό εξετάζουμε τον τρόπο αλληλεπίδρασης της ακτινοβολίας με την ύλη. Η γνώση του τρόπου με τον οποίο τα διάφορα είδη ακτινοβολίας αλληλεπιδρούν με την ύλη είναι πολύ σημαντική στην ανίχνευση σωματιδίων και φωτονίων καθώς και στην παρατήρηση των διάφορων ιδιοτήτων τους.

Στην παρούσα διπλωματική θα μελετήσουμε εκτενέστερα τους ανιχνευτές πυριτίου. Οι ανιχνευτές πυριτίου χρησιμοποιούνται στην φυσική υψηλών ενεργειών κυρίως ως ανιχνευτές φορτισμένων σωματιδίων. Επίσης χρησιμοποιούνται σε πολλές άλλες εφαρμογές, όπως στην ιατρική ως ανιχνευτές  $\gamma$ -ακτινοβολίας και φασματόμετρα ακτίνων-Χ. Επομένως αρχικά θα μελετήσουμε τον τρόπο με τον οποίο φορτισμένα σωματίδια και φωτόνια αλληλεπιδρούν με την ύλη.

### 1.2 Αλληλεπίδραση ακτινοβολίας και ύλης

Η λειτουργία κάθε ανιχνευτή ακτινοβολίας βασίζεται στο τρόπο με τον οποίο η ακτινοβολία αλληλεπιδρά με το υλικό του ανιχνευτή. Ο μηχανισμός που κάθε σωματίδιο αλληλεπιδρά με την ύλη ποικίλει όχι μόνο μεταξύ διαφορετικών σωματιδίων αλλά και για ένα σωματίδιο σε διαφορετικές ενέργειες. Για το λόγο αυτό πολύ διαφορετικοί ανιχνευτές έχουν κατασκευαστεί όπου κάθε ένας εκμεταλλεύεται ορισμένους χαρακτηριστικούς μηχανισμούς με τους οποίους ένα σωματίδιο αλληλεπιδρά με την ύλη ώστε να μπορέσει να το ανιχνεύσει.

#### 1.2.1 Αλληλεπίδραση φορτισμένων σωματιδίων

Η αλληλεπίδραση φορτισμένων σωματιδίων όταν διέρχονται από την ύλη οφείλεται κυρίως σε δυνάμεις Coulomb. Τα φορτισμένα σωματίδια χάνουν κινητική ενέργεια κυρίως από μη ελαστικές σκεδάσεις με τα ατομικά ηλεκτρόνια και δευτερευόντως με τους πυρήνες του υλικού μέσου. Κατά την μη ελαστική σκέδαση με τα περιφερειακά ηλεκτρόνια μπορεί να προκληθεί ιονισμός ή διέγερση του ατόμου και απόκλιση του φορτισμένου σωματιδίου από την αρχική τροχιά. Ενώ κατά την μη ελαστική σκέδαση των σωματιδίων με το πυρήνα του μέσου εκπέμπεται ακτινοβολία πεδήσεως και πολύ σπάνια προκαλείται διέγερση του πυρήνα.

### 1.2.2 Αλληλεπίδραση βαρέων φορτισμένων σωματιδίων

Στην κατηγορία αυτή εντάσσονται φορτισμένα σωματίδια με μάζα πολύ μεγαλύτερη από αυτή των ηλεκτρονίων όπως πρωτόνια, μόνια, σωματίδια  $\alpha$ . Ο κύριος μηχανισμός απώλειας ενέργειας είναι μέσω μη ελαστικών συγκρούσεων με τα περιφερειακά ηλεκτρόνια των ατόμων του υλικού μέσου. Κατά την διαδικασία αυτή γίνεται ιονισμός και διέγερση του υλικού μέσου. Τα βαρέα σωματίδια λόγω της πολύ μεγαλύτερης μάζας τους σε σχέση με τα περιφερειακά ηλεκτρόνια των ατόμων δεν αποκλίνουν από την αρχική τους πορεία. Η τροχιά τους μέσα στην ύλη μπορεί να θεωρηθεί ευθύγραμμη.

Κατά το μηχανισμό αυτό τα φορτισμένα σωματίδια αποδίδουν την ενέργεια τους σταδιακά στο υλικό μέσο από το οποίο διέρχονται και η κινητική τους ενέργεια μετατρέπεται σε θερμότητα στο μέσο αυτό.

Η μείωση της κινητικής ενέργειας ανά μονάδα μήκους διαδρομής ενός σωματιδίου με μάζα πολύ μεγαλύτερη του ηλεκτρονίου  $M \gg m_e$ , φορτίο  $ze$ . Δίνεται από την σχέση Bethe-Bloch:

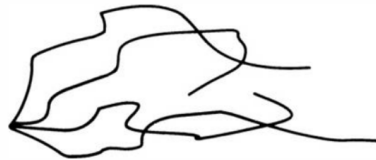
$$-\frac{dE}{dx} = \frac{2\pi ZN_A}{A} \frac{2r_e^2 m_e c^2}{\beta^2} z^2 \left[ \ln \frac{2\gamma^2 \beta^2 m_e c^2}{I} - \beta^2 - \frac{\delta}{2} \right] \quad (1.1)$$

Όπου:

- $N_A$  αριθμός Avogadro
- $r_e^2$  ακτίνα του ηλεκτρονίου
- $m_e c^2$  η ενέργεια που ισοδυναμεί προς την μάζα ηρεμίας του ηλεκτρονίου
- $Z, A$  ατομικός και μαζικός αριθμός του υλικού.
- $\beta = \frac{v}{c}$  σχετική ταχύτητα
- $\gamma = \frac{1}{\sqrt{1-\beta^2}}$  παράγοντας Lorentz.
- $z$  το φορτίο του εισερχόμενου σωματιδίου
- $\delta$  παράγοντας διόρθωσης πυκνότητας.

Περισσότερες πληροφορίες στις βιβλιογραφικές αναφορές [4], [5].

### 1.2.3 Αλληλεπίδραση ηλεκτρονίων-ποζιτρονίων με την ύλη



Σχήμα 1.1: Τροχιά ηλεκτρονίων μέσα στην ύλη. Τα ηλεκτρόνια σκεδάζονται σε μεγάλο εύρος γωνιών κάτι που αντιστοιχεί σε μια ακανόνιστη τροχιά

Τα ηλεκτρόνια λόγω της μικρής τους μάζας σκεδάζεται σε μεγάλες γωνίες κάτι που οδηγεί σε μια ακανόνιστη τροχιά μέσα στον όγκο του υλικού όπως φαίνεται στην εικόνα 1.1.

Κατά τις μη ελαστικές σκεδάσεις των ηλεκτρονίων με τα ατομικά ηλεκτρόνια παρατηρούνται φαινόμενα διεγέρσεων και ιονισμών, κατά τα οποία παρατηρείται αντίστοιχη απώλεια της ενέργειας τους και συγχρόνως απόκλιση από την αρχική τους τροχιά.

Η μείωση της κινητικής ενέργειας των ηλεκτρονίων ανα μονάδα μήκους διαδρομής λόγω διεγέρσεων και ιονισμών για διάφορες τιμές ενεργειών δίνεται από την σχέση των Bethe-Bloch. Περισσότερες πληροφορίες στις βιβλιογραφικές αναφορές[4] , [5].

$$-\frac{dE}{dx} = 4\pi N_A r_e^2 m_e c^2 \frac{Z}{A} \frac{1}{\beta^2} \left[ \ln \frac{\gamma m_e c^2 \beta \sqrt{\gamma-1}}{\sqrt{2}I} + \frac{1}{2}(1-\beta^2) - \frac{2\gamma-1}{2\gamma^2} \ln 2 + \frac{1}{16} \left( \frac{\gamma-1}{\gamma} \right)^2 \right] \quad (1.2)$$

Η απώλεια ενέργειας για τα ποζιτρόνια είναι παρόμοια με τα ηλεκτρόνια αφού είναι σωματίδια με ίδια μάζα. Η συνάρτηση απώλειας ενέργειας ανα μονάδα μήκους για τα ποζιτρόνια είναι:

$$-\frac{dE}{dx} = 4\pi N_A r_e^2 m_e c^2 \frac{Z}{A} \frac{1}{\beta^2} \left[ \ln \frac{\gamma m_e c^2 \beta \sqrt{\gamma-1}}{\sqrt{2}I} - \frac{\beta^2}{24} \left[ 23 + \frac{14}{\gamma+1} + \frac{10}{(\gamma+1)^2} + \frac{4}{(\gamma+1)^3} \right] \right] \quad (1.3)$$

Τα ηλεκτρόνια όταν χάσουν το μεγαλύτερο ποσό της κινητικής τους ενέργειας γίνονται μέρος του πληθυσμού των ηλεκτρονίων της ύλης είτε δεσμεύονται από κενές θέσεις των ηλεκτρονίων είτε προστίθενται στα ελεύθερα ηλεκτρόνια του υλικού.

Αντίστοιχα τα ποζιτρόνια μέσα στην ύλη χάνουν συνεχώς ενέργεια. Όταν έχουν χάσει το μεγαλύτερο ποσό της κινητικής τους ενέργειας, βρίσκονται δηλαδή πρακτικά σε ηρεμία, τα ποζιτρόνια αλληλεπιδρούν με κάποια από τα ηλεκτρόνια που βρίσκονται μέσα στο υλικό. Η αλληλεπίδραση ηλεκτρονίου ποζιτρονίου ονομάζεται εξαύλωση. Κατά την διαδικασία αυτή το ηλεκτρόνιο και το ποζιτρόνιο εξαυλώνονται και εκπέμπονται δύο φωτόνια με ενέργειες 511 keV με αντιδιαμετρικές διευθύνσεις.

Κατά τις μη ελαστικές σκεδάσεις των ηλεκτρονίων με τους πυρήνες των ατόμων η ενέργεια χάνεται κυρίως λόγω ακτινοβολίας πεδήσεως. Όταν ένα ηλεκτρικό φορτίο επιταχύνεται εκπέμπει ηλεκτρομαγνητική ακτινοβολία. Στην περίπτωση του ηλεκτρονίου που διέρχεται μέσα από την ύλη, το ηλεκτρικό πεδίο του πυρήνα επιβραδύνει το ηλεκτρόνιο και το εξαναγκάζει να αλλάξει διεύθυνση με αποτέλεσμα να εκπέμπεται ηλεκτρομαγνητική ακτινοβολία εις βάρος της κινητικής ενέργειας του σωματιδίου. Η ακτινοβολία αυτή ονομάζεται ακτινοβολία πεδήσεως. Το φαινόμενο ακτινοβολίας πεδήσεως είναι ιδιαίτερα έντονο στην περίπτωση ηλεκτρονίων γιατί τα ηλεκτρόνια αποκλίνουν έντονα από την πορεία τους λόγω της μικρής τους μάζας. Στην περίπτωση βαρύτερων σωματιδίων όπως (σωματίδια  $\alpha$ ) το φαινόμενο εκπομπής ακτινοβολίας πεδήσεως μπορεί να θεωρηθεί αμελητέο.

Η ανά μονάδα μήκους μείωση της ακτινοβολίας πεδήσεως λόγω εκπομπής ακτινοβολίας πεδήσεως εκφράζεται θεωρητικά από την σχέση :

$$-\left(\frac{dE}{dx}\right)_r = N\sigma_0 Z^2 \int_0^1 B\left(Z, \frac{hv}{E}\right) d\left(\frac{hv}{E}\right) \quad (1.4)$$

Όπου:

- $Z, N$  ο ατομικός αριθμός και η πυκνότητα ατόμων του μέσου αντίστοιχα
- $E$  η κινητική ενέργεια του ηλεκτρονίου σε MeV
- $m_0 c_0^2$  η ενέργεια που ισοδυναμεί προς την μάζα ηρεμίας του ηλεκτρονίου
- $\sigma_0$  ενεργός διατομή
- $h\nu$  η ενέργεια των εκπεμπόμενων φωτονίων κυμαινόμενη από 0 έως E MeV
- $B(Z, \frac{h\nu}{E})$  συντελεστής, που εξαρτάται από τον ατομικό αριθμό Z και την κινητική ενέργεια των ηλεκτρονίων.

Η συνολική γραμμική μεταβολή της ενέργειας του σωματιδίου σε ένα μέσο θα ισούται με το άθροισμα της ενέργειας που χάνεται λόγω ακτινοβολίας και της ενέργειας που χάνεται λόγω συγκρούσεων. Ισχύει δηλαδή η σχέση

$$\frac{dE}{dx} = \left(\frac{dE}{dx}\right)_c + \left(\frac{dE}{dx}\right)_r \quad (1.5)$$

Ο λόγος που συνδέει τους δύο μηχανισμούς απώλειας ενέργειας δίνεται από τη παρακάτω σχέση

$$\frac{(dE/dx)_r}{(dE/dx)_c} \approx \frac{EZ}{700} \quad (1.6)$$

#### 1.2.4 Αλληλεπίδραση φωτονίων και ύλης

Τα φωτόνια ανιχνεύονται έμμεσα μέσω των αλληλεπιδράσεων τους με την ύλη. Από τις αλληλεπιδράσεις αυτές συνήθως προκύπτουν φορτισμένοι φορείς οι οποίοι μπορούν να συλληχθούν δίνοντας σήμα το οποίο μπορεί να μετρηθεί. Οι κύριοι μηχανισμοί αλληλεπίδρασης των φωτονίων είναι οι εξής:

1. Φωτοηλεκτρικό φαινόμενο
2. Σκέδαση Compton
3. Δίδυμη Γέννηση

Όλοι οι μηχανισμοί οδηγούν σε μερική ή ολική μεταφορά ενέργειας στα ηλεκτρόνια του στόχου. Περισσότερες πληροφορίες στις βιβλιογραφικές αναφορές [4] , [5].

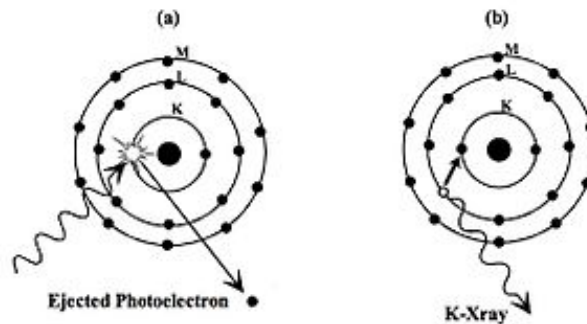
#### Φωτοηλεκτρικό Φαινόμενο

Το φωτόνιο αλληλεπιδρά με ισχυρά συνδεδεμένο ηλεκτρόνιο του ατόμου (π.χ στοιβάδα K). Το φωτόνιο χάνει την συνολική του ενέργεια και εκπέμπεται ένα ηλεκτρόνιο με κινητική ενέργεια

$$E_{e^-} = h\nu - E_b \quad (1.7)$$

Όπου  $E_b$  ενέργεια σύνδεσης του ηλεκτρονίου με την στοιβάδα και  $h\nu$  η ενέργεια του φωτονίου. Το κενό που προκύπτει στις εσωτερικές στοιβάδες όπου έγινε η αλληλεπίδραση καλύπτεται από ελεύθερα ηλεκτρόνια του μέσου ή από αναδιάταξη των ηλεκτρονίων των εξωτερικών στοιβάδων. Έτσι προκύπτουν χαρακτηριστικές X-ακτινοβολίες.

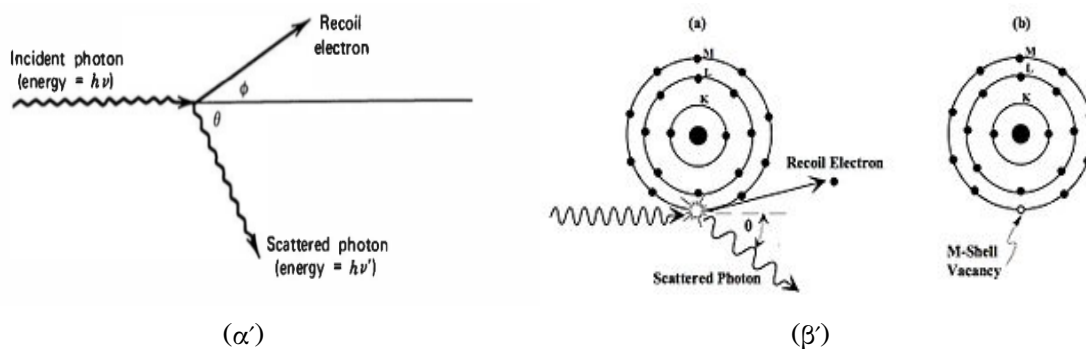
Το φωτοηλεκτρικό φαινόμενο είναι ο κυρίαρχος μηχανισμός αλληλεπίδρασης  $\gamma$  και X ακτινοβολίας για μικρές ενέργειες.



Σχήμα 1.2: Φωτοηλεκτρικό φαινόμενο.

### Σκέδαση Compton

Στη σκέδαση Compton η ακτινοβολία  $\gamma$  προσπίπτει σε ένα ηλεκτρόνιο του υλικού και σκεδάζεται σε γωνία  $\theta$  ανάλογα με την αρχική διεύθυνση. Τα ηλεκτρόνια σε αντίθεση με το φωτοηλεκτρικό φαινόμενο είναι ασθενώς συνδεδεμένα με τον πυρήνα, έχουν δηλαδή μικρή ενέργεια σύνδεσης. Το φωτόνιο χάνει ένα μέρος της αρχικής του ενέργειας. Από τους νόμους διατήρησης ενέργειας και ορμής προκύπτει η παρακάτω σχέση που συνδέει την ενέργεια που μεταφέρθηκε με τη γωνία σκέδασης.



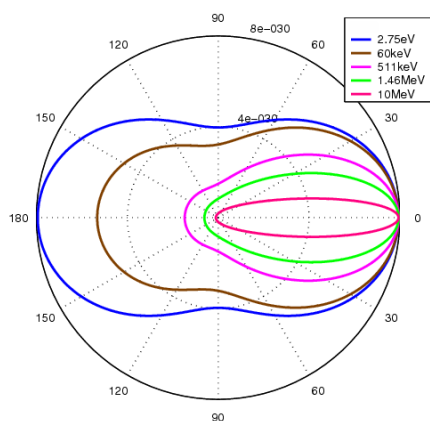
Σχήμα 1.3: Φαινόμενο Compton.

$$h\nu' = \frac{h\nu}{1 + \frac{h\nu}{m_0c^2}(1 - \cos\theta)} \quad (1.8)$$

Επειδή όλες οι γωνίες είναι πιθανές η ενέργεια που μεταφέρεται μπορεί να κυμαίνεται από 0 έως πολύ μεγάλο μέρος της αρχικής ενέργειας. Η ενεργός διατομή της σκέδασης Compton αυξάνεται με το μαζικό αριθμό του υλικού που ακτινοβολείται. Μεγάλο  $Z$  περισσότερα ηλεκτρόνια στόχοι.

Η διαφορική ενεργός διατομή (ενεργός διατομή ανά στερεά γωνία) δίνεται από τη σχέση Klein-Nishina. Στο σχήμα 1.4 μπορούμε να παρατηρήσουμε την κατανομή των γωνιών σκέδασης των φωτονίων λόγω του φαινομένου Compton για διάφορες ενέργειες.

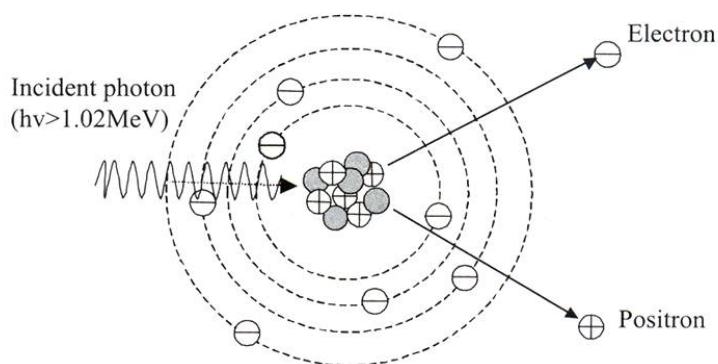
$$\frac{d\sigma}{d\Omega} = Zr_0^2 \left( \frac{1}{1 + \alpha(1 - \cos\theta)} \right)^2 \left( \frac{1 + \cos^2\theta}{2} \right) \left( 1 + \frac{\alpha^2(1 - \cos^2\theta)^2}{(1 + \cos^2\theta)(1 + \alpha(1 - \cos\theta))} \right) \quad (1.9)$$



Σχήμα 1.4: Διάγραμμα Klein-Nishina. Κατανομή των γωνιών σκέδασης για διάφορες ενέργειες Πηγή: [http://en.wikipedia.org/wiki/File:Klein-Nishina\\_distribution.png](http://en.wikipedia.org/wiki/File:Klein-Nishina_distribution.png)

### Δίδυμη γέννηση

Κατά τη δίδυμη γέννηση το φωτόνιο διασπάται σε ηλεκτρόνιο και ποζιτρόνιο. Η δίδυμη γέννηση συμβαίνει όταν η ακτινοβολία  $\gamma$  έχει ενέργεια μεγαλύτερη 1,02 MeV. Όπου 1,02 MeV είναι δύο φορές η ενέργεια ηρεμίας του ηλεκτρονίου ώστε να μην παραβιάζεται ο νόμος διατήρησης της ενέργειας. Η επιπλέον ενέργεια του φωτονίου γίνεται κινητική ενέργεια του ηλεκτρονίου και του ποζιτρονίου. Το ποζιτρόνιο άμεσα θα εξαϋλωθεί μέσα στο όγκο του υλικού και έτσι θα προκύψουν δύο φωτόνια με αντιδιαμετρικές διευθύνσεις.



Σχήμα 1.5: Δίδυμη γέννηση. Πηγή:



### Συντελεστές Εξασθένησης

Εάν θεωρήσουμε μια μονοενεργειακή δέσμη ακτινοβολίας  $\gamma$  που διέρχεται από ένα υλικό όλοι οι παραπάνω μηχανισμοί συμβάλουν στην εξασθένηση της δέσμης. Όλοι οι μηχανισμοί έχουν μια πιθανότητα ανά μονάδα μήκους να συμβούν. Η ολική πιθανότητα εξασθένησης της δέσμης προκύπτει από το άθροισμα των πιθανοτήτων ανά μονάδα μήκους του κάθε μηχανισμού. Ισχύει δηλαδή

$$\mu = \tau(\text{φωτοηλεκτρικό}) + \sigma(\text{Compton}) + \kappa(\text{δίδυμη γέννηση}) \quad (1.10)$$

Προκύπτει ότι δέσμη θα μειώνεται εκθετικά με το χρόνο δηλαδή

$$I = I_0 e^{-\mu t} \quad (1.11)$$

Επειδή η πυκνότητα κάθε υλικού διαφέρει η χρήση του γραμμικού συντελεστή εξασθένησης είναι περιορισμένη. Συνήθως χρησιμοποιείται ο μαζικός συντελεστής εξασθένησης ο οποίος ορίζεται ως

$$\mu_0 = \frac{\mu}{\rho} \quad (1.12)$$

Ανάλογα με την ενέργεια κάποιος μηχανισμός υπερτερεί των άλλων και έχει μεγαλύτερη πιθανότητα να συμβεί. Στο παρακάτω διάγραμμα παρατηρούμε τον τρόπο με τον οποίο η κάθε αλληλεπίδραση συνεισφέρει στον συνολικό μαζικό συντελεστή εξασθένησης. Αρχικά για μικρές ενέργειες υπερισχύει το φωτοηλεκτρικό φαινόμενο. Όσο η ενέργεια αυξάνεται η σκέδαση Compton υπερισχύει. Ενώ για πολύ υψηλές ενέργειες η δίδυμη γέννηση γίνεται ο κύριος μηχανισμός αλληλεπίδρασης.



## Κεφάλαιο 2

# Ανιχνευτές πυριτίου στη φυσική υψηλών ενεργειών

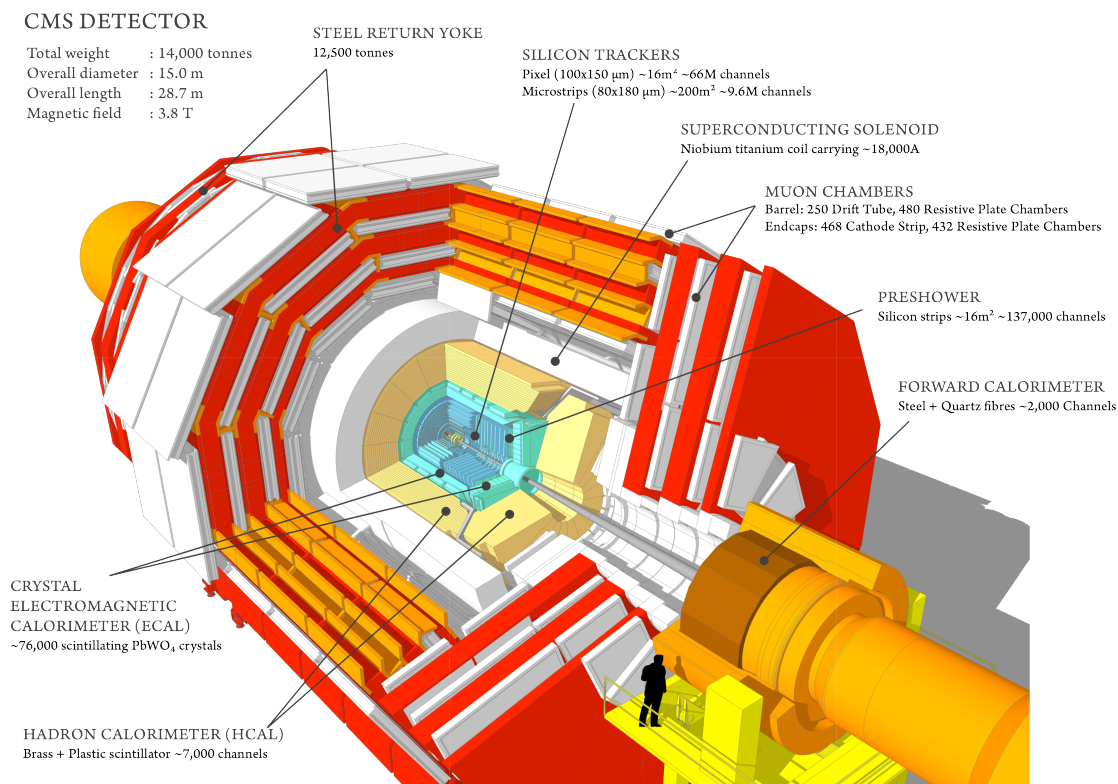
### 2.1 Εισαγωγή

Σε πειράματα της φυσικής υψηλών ενεργειών σωματίδια όπως ηλεκτρόνια, πρωτόνια και ποζιτρόνια επιταχύνονται και συγκρούονται μεταξύ τους με ενέργειες GeV έως και TeV. Από την αρχική αλληλεπίδραση των δύο σωματιδίων που συγκρούονται, δημιουργούνται δευτερεύοντα σωματίδια τα οποία με την σειρά τους μπορούν να αλληλεπιδράσουν και να διασπαστούν. Μελετώντας τις ιδιότητες των δευτερευόντων σωματιδίων όπως το είδος του σωματιδίου, το φορτίο, τη ενέργεια και τη γωνία της διάσπαση μπορεί να γίνει πλήρης ανακατασκευή της αλληλεπίδρασης κάτι που οδηγεί σε συμπεράσματα για την φύση των σωματιδίων. Προφανώς όλες αυτές οι ιδιότητες δεν μπορούν να μετρηθούν από ένα μόνο ανιχνευτικό σύστημα. Για το λόγω αυτό σε ένα πείραμα σύγκρουσης τοποθετούνται πολλοί ανιχνευτές διαφορετικού τύπου σε στρώματα με σχήμα κυλίνδρου γύρω από το σημείο που γίνεται οι αρχική σύγκρουση, καλύπτοντας μια στερεά γωνία  $4\pi$ . Το κάθε στρώμα προορίζεται για έναν συγκεκριμένο σκοπό αποτελώντας έναν υπό-ανιχνευτή. Στην σχήμα 2.1 απεικονίζεται ο ανιχνευτής CMS με τα διάφορα επίπεδα του. Στο παρόν κεφάλαιο θα επικεντρωθούμε περισσότερο στους ανιχνευτές κορυφών και τροχιάς οι οποίοι συγκροτούνται από ανιχνευτές πυριτίου.

Οι ανιχνευτές πυριτίου χρησιμοποιήθηκαν πρώτη φορά στη φυσική υψηλών ενεργειών τη δεκαετία του 80 όταν χρειάστηκαν υψηλής ακρίβειας ανιχνευτές κορυφών για τη μέτρηση του χρόνου ημιζωής του charm σωματιδίου όπου χρειαζόταν χωρική διακριτική ικανότητα μικρότερη από  $10 \mu\text{m}$ . Αρχικά οι ανιχνευτές πυριτίου περιορίζονταν σε πειράματα σταθερού στόχου λόγω του ότι τα εξωτερικά ηλεκτρονικά ήταν ακόμα αρκετά ογκώδη. Η ανάπτυξη συστημάτων VLSI έκανε δυνατή τη κατασκευή ανιχνευτών πυριτίου σε πιο συμπαγή μορφή επιτρέποντας την χρησιμοποίησή τους σε πειράματα συγκρούσεων. Οι ανιχνευτές κορυφών αποδείχθηκαν αναπόσπαστο κομμάτι στους ανιχνευτές του επιταχυντή LEP επιτρέποντας τη μέτρηση των χρόνων ημιζωής του B-μεσονίου και της ακριβής μέτρησης του λόγου διάσπασης  $Z \rightarrow b\bar{b}$ .

Ο πιο απλός τύπος ανιχνευτών πυριτίου είναι ο ανιχνευτής παράλληλων μικρολωρίδων (strip). Οι ανιχνευτές αυτοί είναι ικανοί να ανιχνεύουν την προβολή της τροχιάς ενός σωματιδίου. Μέτρηση δύο διαστάσεων μπορεί να γίνει με ανιχνευτές μικρο-λωρίδων, όταν υπάρχουν δύο παράλληλα στρώματα μικρο-λωρίδων τα οποία σχηματίζουν γωνία μεταξύ τους. Άλλος τύπος ανιχνευτών πυριτίου όπου παρέχονται

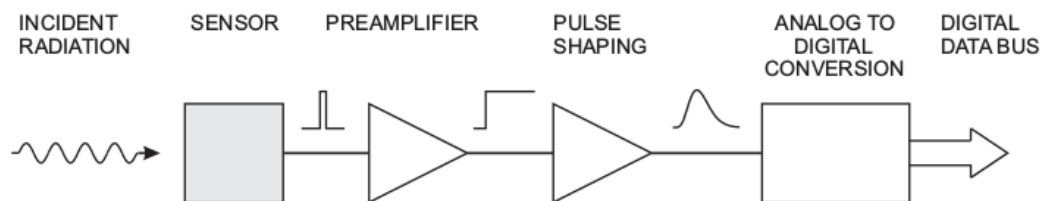
ικανοποιητικές πληροφορίες στις δύο διαστάσεις είναι οι ανιχνευτές μικρο-πλακιδίων (pixel).



Σχήμα 2.1: Ανιχνευτής CMS . Τα διαφορετικά επίπεδα και οι βασικοί υπό-ανιχνευτές οι οποίοι συγκροτούν τον κύριο ανιχνευτή. Το επίπεδο το οποίο είναι κοντά στη αλληλεπίδραση αποτελεί το ανιχνευτή τροχιών ο οποίος αποτελείται από ανιχνευτές πυριτίου. Το επίπεδο αυτό περικλείεται από άλλους υπό-ανιχνευτές όπως το αδρονικό καλορίμετρο , το ηλεκτρομαγνητικό καλορίμετρο και τους σπινθηριστές μιονίων Πηγή: [https://cms-docdb.cern.ch/cgi-bin/PublicDocDB/RetrieveFile?docid=11514&version=1&filename=cms\\_120918\\_03.png](https://cms-docdb.cern.ch/cgi-bin/PublicDocDB/RetrieveFile?docid=11514&version=1&filename=cms_120918_03.png)

## 2.2 Βασική λειτουργία ανιχνευτών πυριτίου

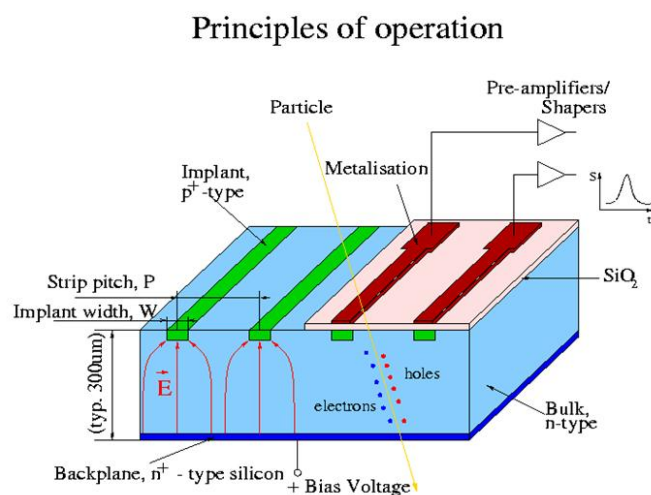
Οι βασικές λειτουργίες του ανιχνευτή περιγράφονται στην εικόνα 2.2



Σχήμα 2.2: Βασικές λειτουργίες ανιχνευτών. Ενέργεια απορροφάται στον όγκο του ανιχνευτή και μετατρέπεται σε ηλεκτρικό σήμα (τετραγωνικός παλμός). Το σήμα ενισχύεται από έναν προενισχυτή (preamplifier) ύστερα τροφοδοτείται σε ένα pulse-shaper και από αυτό σε έναν μετατροπέα αναλογικού σε ψηφιακό σήμα για αποθήκευση και περαιτέρω επεξεργασία. Πηγή: [3]

### 1. Ανιχνευτής

Η ενέργεια που εναποτίθεται σε έναν ανιχνευτή μετατρέπεται σε σήμα. Η βασική δομή του ανιχνευτή αποτελείται από το ενεργό μέσο το οποίο βρίσκεται ανάμεσα από δύο ηλεκτρόδια. Όταν ένα ηλεκτρόνιο ή φωτόνιο διέρχεται μέσα από τον όγκο του ανιχνευτή δημιουργούνται ζευγάρια ηλεκτρονίων-οπών τα οποία κινούνται υπό την επίδραση ενός ηλεκτρικού πεδίου προς τα ηλεκτρόδια. Η κίνηση των φορέων δημιουργεί ένα επαγόμενο φορτίο στα ηλεκτρόδια. Ο αριθμός των ηλεκτρονίων-οπών είναι ανάλογος της απορροφούμενης ενέργειας.



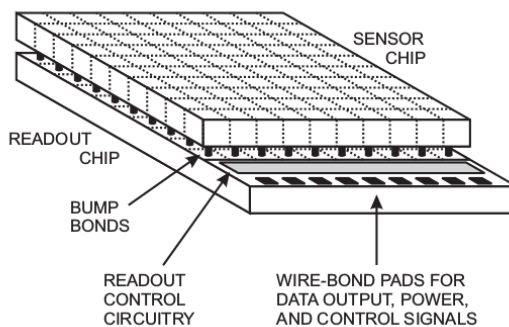
Σχήμα 2.3: Βασική λειτουργία ανιχνευτή μικρο-λωρίδων Πηγή: [http://www.pd.infn.it/\\$%\\$7Edorigo/silicondet\\_working.jpg](http://www.pd.infn.it/$%$7Edorigo/silicondet_working.jpg)

Τα ηλεκτρόδια του ανιχνευτή μπορούν να χωριστούν κατάλληλα παρέχοντας έτσι πληροφορίες για την θέση όπου πέρασε το σωματίδιο. Δύο βασικοί τύποι με τους οποίους μπορούν να χωρισθούν τα ηλεκτρόδια είναι σε μικρο-λωρίδες (strip) και σε μικρο-πλακίδια (pixel).

Στην εικόνα 2.3 παρουσιάζεται σχηματικά η λειτουργία μιας διάταξης ανιχνευτή πυριτίου μικρο-λωρίδων. Αποτελείται από έναν όγκο πολύ-κρυσταλλικού πυριτίου με περίσσεια ηλεκτρονίων ( $n$ ) όπου στην πάνω επιφάνεια έχουν σχηματιστεί περιοχές με ενισχυμένη περίσσεια οπών ( $p^+$ ) σε σχήμα λωρίδων. Δημιουργούνται δηλαδή στον όγκο του ανιχνευτή πολλές διαφορετικές περιοχές όπου η κάθε μια αποτελεί μια δίοδο ( $p^+n$ ). Το πλάτος όπως και το πάχος των λωρίδων είναι πολύ μικρότερο από το μήκος του ανιχνευτή. Οι περιοχές αυτές καλύπτουν όλο το μήκος του ανιχνευτή και συνήθως σχηματίζονται με σταθερό πλάτος και σταθερή απόσταση μεταξύ τους. Το πλάτος καθώς και η απόσταση μεταξύ των λωρίδων παίζει σημαντικό ρόλο στον σχηματισμό του σήματος. Ο όγκος του κρυσταλλικού πυριτίου θα πρέπει να προστατευθεί από ένα στρώμα το οποίο δεν θα επιτρέπει σε εξωτερικούς παράγοντες να διεισδύσουν στο εσωτερικό του ανιχνευτή όπως άτομα από το εξωτερικό περιβάλλοντα χώρο. Για τον λόγο αυτό δημιουργείται με την μέθοδο της χημικής εναπόθεσης ατμών ένα στρώμα οξειδίου του πυριτίου  $SiO_2$ . Το στρώμα  $SiO_2$  προστατεύει τον όγκο από εξωτερικούς παράγοντες καθώς επίσης δεν επιτρέπει την δημιουργία στατικών φορτίων στην επιφάνεια (παθητικοποίηση). Στην κάτω επιφάνεια σχηματίζεται με ιοντική εμφύτευση μια ενιαία επιφάνεια με ενισχυμένη περίσσεια ηλεκτρονίων ( $n^+$ ). Στις περιοχές όπου έχουν δημιουργηθεί οι δίοδοι των λωρίδων αλλά και στο πίσω επίπεδο τοποθετείται ένα στρώμα μετάλλου για την δημιουργία ηλεκτροδίων τα οποία επιτρέπουν την σύνδεση του ανιχνευτή με εξωτερικά ηλεκτρονικά. Στα ηλεκτρόδια αυτά ασκείται μια υψηλή τάση η οποία δημιουργεί ένα υψηλό ηλεκτρικό πεδίο στον ανιχνευτή έτσι ώστε οι φορείς ( $e^-$ ,  $h^+$ ) που θα δημιουργηθούν στο εσωτερικό του ανιχνευτή να μετακινηθούν γρήγορα προς τα ηλεκτρόδια. Οι ανιχνευτές μικρο-λωρίδων μπορούν να μετρήσουν την προβολή της τροχιάς του διερχόμενου σωματιδίου. Τα εξωτερικά ηλεκτρονικά ανάγνωσης μπορούν να τοποθετηθούν έξω από την επιφάνεια αποδοχής του ανιχνευτή. Οι ανιχνευτές πυριτίου μικρο-λωρίδων μπορούν να καλύψουν μια μεγάλη επιφάνεια με λιγότερα εξωτερικά ηλεκτρονικά.

Η πίσω επιφάνεια μπορεί να χωρισθεί και αυτή σε λωρίδες έτσι ώστε οι λωρίδες του κάτω επιπέδου να σχηματίζουν γωνία με το πάνω επίπεδο με αυτό τον τρόπο είναι δυνατός ο καθορισμός της θέσης από την οποία πέρασε το σωματίδιο στις δύο διαστάσεις.

Προκειμένου να υπάρξει σαφής διαχωρισμός της θέσης στις δύο διαστάσεις απαιτείται καλή τμηματοποίηση της επιφάνειας των διόδων. Όσο περισσότερες και μικρότερες δίοδοι υπάρχουν τόσο πιο σαφή πληροφορία λαμβάνουμε σε σχέση με την τροχιά που διέγραψε ένα σωματίδιο μέσα στον όγκο του ανιχνευτή. Αυτό μπορεί να επιτευχθεί σχηματίζοντας περιοχές με ενισχυμένη περίσσεια οπών ( $p^+$ ) σε σχήμα μικρών πλακιδίων pixel. Η βασική δομή ενός τέτοιου ανιχνευτή φαίνεται στην εικόνα 2.4. Τα ηλεκτρόδια του ανιχνευτή σχηματίζουν ένα μοτίβο σαν "σκακιέρα". Τα ηλεκτρόδια του ανιχνευτή συνδέονται με μια εξωτερική διάταξη ενισχυτών μέσω ενός δύο διαστάσεων πλέγμα από επαφές ("solder bumps"). Το μέγεθος κάθε pixel περιορίζεται από την επιφάνεια που καλύπτει κάθε κανάλι τον εξωτερικών ηλεκτρονικών ανάγνωσης. Τα εξωτερικά ηλεκτρονικά ανάγνωσης τοποθετούνται ανάμεσα στην ευαίσθητη περιοχή του ανιχνευτή προσθέτοντας υλικό στον ανιχνευτή. Επίσης η μεγάλη πυκνότητα ηλεκτρονικών μπορεί να επιφέρει σημαντική κατανάλωση ισχύος.



Σχήμα 2.4: Βασική λειτουργία ανιχνευτή μικρο-πλακιδίων Πηγή: [3]

## 2. Προενισχυτής

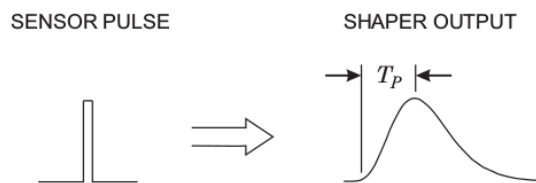
Το σήμα αρχικά μπορεί να είναι αρκετά μικρό, περίπου ( $5 \times 10^{-17}$  C) για ακτίνες  $x$  ενέργειας ( $1keV$ ). Επομένως το σήμα θα πρέπει να ενισχυθεί. Στο σήμα υπεισέρχονται και άλλοι παράγοντες όπως στατιστικές διακυμάνσεις σήματος και ηλεκτρονικός θόρυβος. Άρα ο ανιχνευτής καθώς και ο προενισχυτής θα πρέπει να σχεδιάζονται με το βέλτιστο τρόπο ο οποίος οδηγεί στο μέγιστο λόγο σήματος προς θόρυβο. Μια κρίσιμη παράμετρος που επηρεάζει τον θόρυβο είναι η συνολική χωρητικότητα η οποία είναι το άθροισμα της χωρητικότητας του ανιχνευτή και η χωρητικότητα στην είσοδο του ενισχυτή. Ο λόγος σήματος προς θόρυβο αυξάνεται με την μείωση της χωρητικότητας.

## 3. Διαμορφωτής παλμού

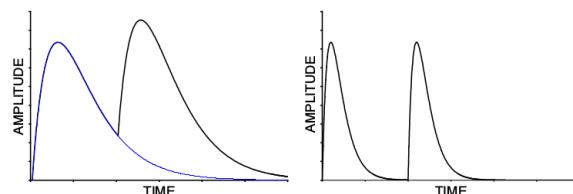
Η βασική λειτουργία ενός διαμορφωτή παλμού (pulse shaper) είναι η βελτίωση του λόγου σήματος προς θόρυβο. Το φάσμα συχνοτήτων του σήματος διαφέρει από το φάσμα του θορύβου το οποίο λόγω του ότι βασίζεται σε στατιστικά ασυσχέτιστα φαινόμενα θα επεκτείνεται σε ένα πολύ μεγαλύτερο εύρος ζώνης συχνοτήτων σε σχέση με το σήμα. Μπορούμε επομένως χρησιμοποιώντας κατάλληλα φίλτρα να περιορίσουμε το θόρυβο προς όφελος του σήματος. Όμως αλλάζοντας την απόκριση συχνότητας ενός παλμού αλλάζει και η χρονική απόκριση και άρα το σχήμα του παλμού, η διαδικασία αυτή καλείται διαμόρφωση σήματος. Η μείωση του εύρους ζώνης του σήματος βελτιώνει τον λόγο σήματος προς θόρυβο αυξάνει όμως και τη διάρκεια του παλμού. Έτσι ένας τετραγωνικός παλμός που σχηματίζεται από τον ανιχνευτή μετατρέπεται σε έναν ευρύτερο παλμό με έναν χαρακτηριστικό χρόνο ανόδου ( $T_p$ ).

Συνήθως μας ενδιαφέρει να μετρήσουμε πολλούς παλμούς διαδοχικά. Επομένως το μεγάλο πλάτος ενός παλμού οδηγεί στο φαινόμενο pile-up στο οποίο κομμάτι ενός παλμού υπεισέρχεται στον επόμενο παλμό αυξάνοντας το μέγεθος του. Αν ο ρυθμός των παλμών είναι υψηλός αυτό οδηγεί σε λανθασμένη καταγραφή του ύψους των παλμών. Το φαινόμενο αυτό μπορεί να περιοριστεί μειώνοντας το πλάτος των παλμών 2.6.

Τα βασικά ηλεκτρονικά στοιχεία διαμόρφωσης παλμών φαίνονται στην εικόνα 2.7. Ο ανιχνευτής συνήθως απεικονίζεται ως ένας πυκνωτής, ενώ ο προενισχυ-



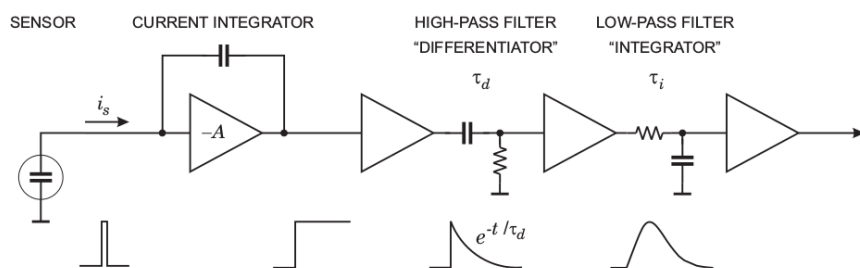
Σχήμα 2.5: Το σήμα που προέρχεται από τον ανιχνευτή μετασχηματίζεται σε έναν ευρύτερο παλμό με έναν χαρακτηριστικό χρόνο ανόδου ( $T_p$ ) Πηγή: [3]



Σχήμα 2.6: Το φαινόμενο pile-up συμβαίνει όταν δύο παλμοί επικαλύπτονται. Μειώνοντας τον χρόνο πτώσης του σήματος ο πρώτος παλμός φτάνει στην αρχική τιμή πριν τον δεύτερο παλμό. Πηγή: [3]

τής απεικονίζεται από ένα κύκλωμα ολοκληρωτή. Το ολοκλήρωμα του στενού παλμού θα είναι ένας βηματικός παλμός με πολύ μεγάλο χρόνο πτώσης.

Ο διαμορφωτής παλμών (pulse shaper) αποτελείται στην ουσία από δύο παθητικά φίλτρα ένα ύψι-περατό (CR) και ένα χαμηλό-περατό (RC). Το ύψι-περατό φίλτρο εισάγει τον επιθυμητό χρόνο πτώσης του παλμού ενώ το χαμηλό-περατό ρυθμίζει τον χρόνο ανόδου του παλμού.



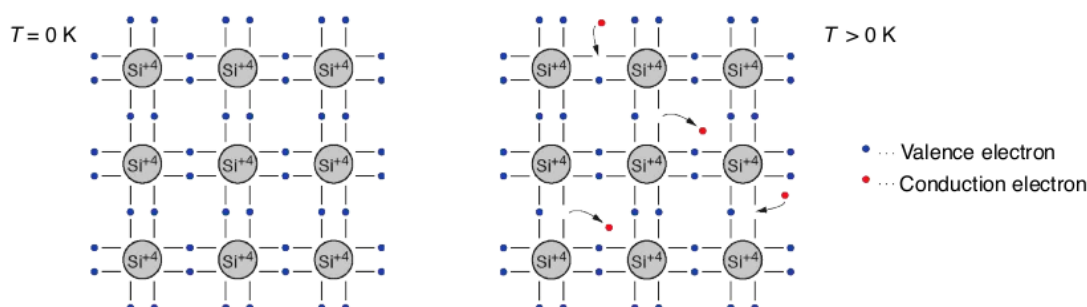
Σχήμα 2.7: Τα βασικά στοιχεία διαμόρφωσης παλμού. Πηγή: [3]



## 2.3 Σχηματισμός Σήματος

Οι ανιχνευτές ημιαγωγών λειτουργούν στην ουσία ως θάλαμοι ιονισμού. Το ηλεκτρικό σήμα σχηματίζεται απευθείας από ιονισμό. Η ακτινοβολία που προσπίπτει στο ενεργό μέσο του ανιχνευτή σχηματίζει ζευγάρια ηλεκτρονίων οπών. Ο μηχανισμός αυτός ανταγωνίζεται τους άλλους μηχανισμούς αλληλεπίδρασης ακτινοβολίας και ύλης.

Το πυρίτιο(Si) όπως και το γερμάνιο (Ge) είναι στοιχεία με 4 ηλεκτρόνια στην εξωτερική τους στοιβάδα. Τα ηλεκτρόνια αυτά σχηματίζουν δεσμούς με τα γειτονικά άτομα. Για μικρές θερμοκρασίες όλα τα ηλεκτρόνια σχηματίζουν δεσμούς. Όσο η θερμοκρασία αυξάνεται οι δεσμοί σπάνε απελευθερώνοντας ένα ηλεκτρόνιο το οποίο συμβάλει την αγωγιμότητα του υλικού. Ο ανοιχτός δεσμός μπορεί να προσελκύσει γειτονικά ηλεκτρόνια. Έτσι σχηματίζεται μια οπή η οποία μπορεί να κινείται έμμεσα ανάλογα με την κίνηση των ηλεκτρονίων.

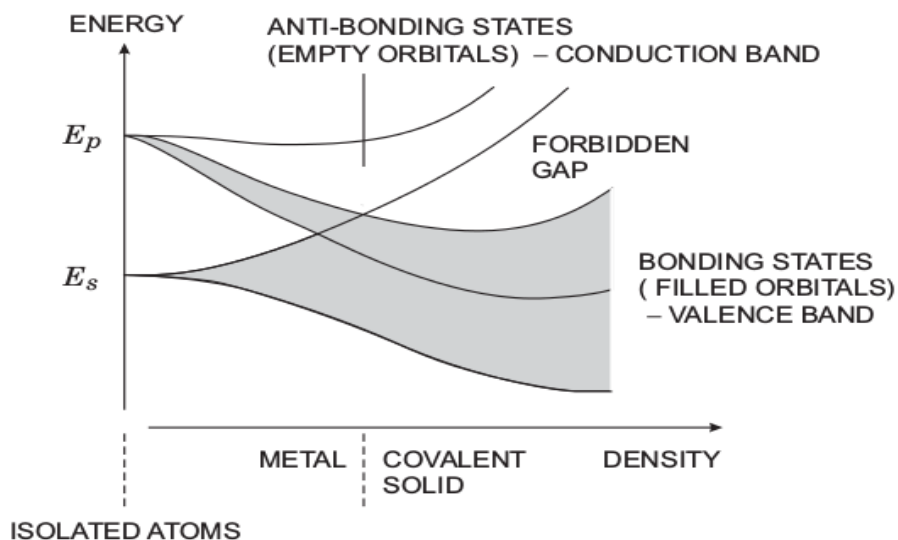


Σχήμα 2.8: Για  $T=0$  όλα τα ηλεκτρόνια σχηματίζουν δεσμούς με γειτονικά άτομα. Για  $T>0$  οι δεσμοί σπάνε Πηγή: [3]

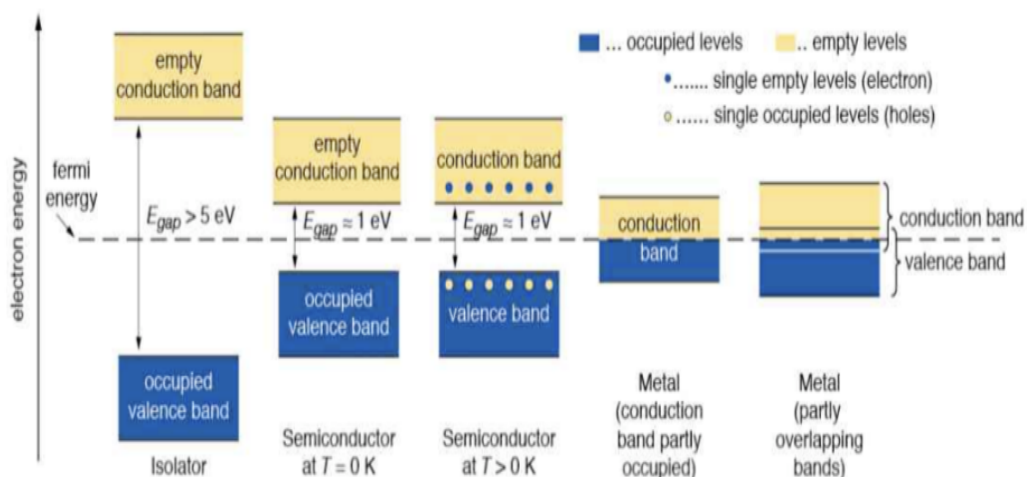
Ένα απομονωμένο άτομο έχει διακριτές στάθμες ενέργειας. Δηλαδή στην περίπτωση του πυρίτιου σε μεγάλες αποστάσεις θα υπάρχουν 2 ηλεκτρόνια στην s στοιβάδα και 2 στην p στοιβάδα. Όταν όμως η διατομική απόσταση μειώνεται έτσι ώστε να σχηματιστεί κρυσταλλικό πλέγμα οι ενεργειακές καταστάσεις συγχωνεύονται ώστε να σχηματίσουν καταστάσεις δεσμού και καταστάσεις με μηδενική πιθανότητα κατάληψης. Οι καταστάσεις αυτές ονομάζονται ενεργειακές ζώνες. Οι ενεργειακές ζώνες χωρίζονται στη ζώνη σθένους, απαγορευμένη ζώνη και ζώνη αγωγιμότητας εικόνα 2.9.

Στα μέταλλα οι ενεργειακές ζώνες αγωγιμότητας και σθένους επικαλύπτονται. Ενώ στους μονωτές και στους ημιαγωγούς οι ζώνες αυτές διαχωρίζονται από ένα ενεργειακό χάσμα. Στους μονωτές το χάσμα είναι πολύ μεγαλύτερο. Ο αριθμός των καταστάσεων δεσμού είναι ίσος με τον αριθμό των ηλεκτρονίων, Έτσι καμία υψηλότερη ενέργεια δεν έχει καταληφθεί. Επίσης δεν υπάρχει καμία κενή κατάσταση στη ζώνη σθένους. Για να μεταφερθεί ένα ηλεκτρόνιο από την ζώνη σθένους στη ζώνη αγωγιμότητας πρέπει να του δοθεί ενέργεια (π.χ. υπό μορφή θερμότητας) μεγαλύτερη από το ενεργειακό χάσμα. Στους ημιαγωγούς σε θερμοκρασία δωματίου υπάρχουν καταστάσεις στη ζώνη αγωγιμότητας που έχουν καταληφθεί λόγω του μικρού ενεργειακού χάσματος. Εικόνα 2.10.

Ο δεσμός μπορεί να σπάσει με οποιοδήποτε τρόπο του δοθεί αρκετή ενέργεια όπως για παράδειγμα η ενέργεια που μπορεί να εναποτεθεί στον δεσμό από μία διερχόμενη ακτινοβολία. Με αυτό τον τρόπο το ηλεκτρόνιο μεταβαίνει από την ζώνη σθένους στη ζώνη αγωγιμότητας αφήνοντας στη ζώνη σθένους μια κενή θέση "οπή". Οι οπές μπορούν να θεωρηθούν ως φορείς θετικού φορτίου και μπορούν φαινομενικά να κινούνται παρόλο που βρίσκονται στη ζώνη σθένους όπως και το ηλεκτρόνιο κινείται ελεύθερα

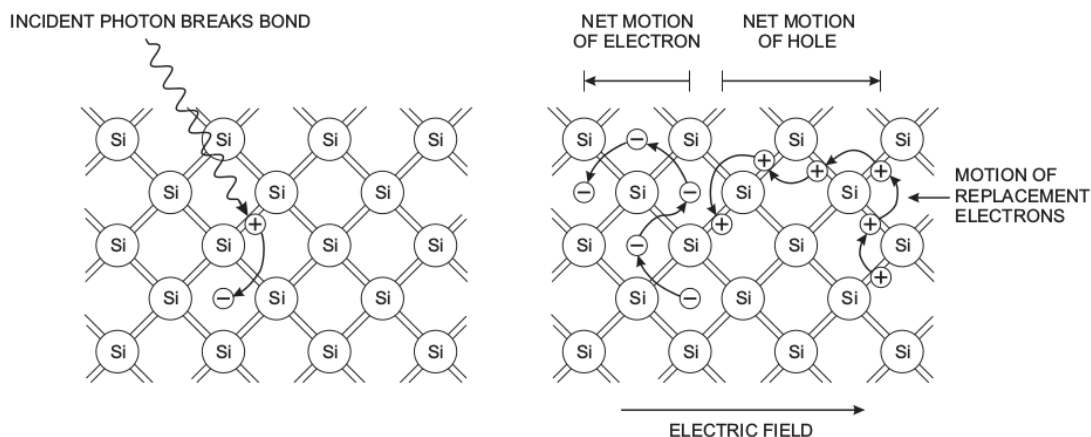


Σχήμα 2.9: Όσο η πυκνότητα αυξάνεται οι διακριτές στάθμες συγχωνεύονται ώστε να δημιουργήσουν τις ζώνες σθένους, τη απαγορευμένη ζώνη και τη ζώνη αγωγιμότητας. Πηγή: [3]



Σχήμα 2.10: Ενεργειακές ζώνες για ημιαγωγούς, μονωτές, μέταλλα. Το ενεργειακό χάσμα στους μονωτές είναι αρκετά μεγάλο. Στους ημιαγωγούς η ζώνη αγωγιμότητας είναι άδεια για  $T = 0k$ . Τα ηλεκτρόνια στους ημιαγωγούς αποκτούν αρκετή ενέργεια ώστε να μεταβούν στην ζώνη αγωγιμότητας για  $T > 0k$ . Η ζώνη αγωγιμότητας και η ζώνη σθένους επικαλύπτονται στα μέταλλα Πηγή: [3]

στη ζώνη αγωγιμότητας, εικόνα 2.11 Μπορούμε να κατευθύνουμε την κίνηση του ηλεκτρονίου και της οπής εφαρμόζοντας ένα ηλεκτρικό πεδίο και έπειτα να συλλεχθεί από τα ηλεκτρόδια. Με αυτό τον τρόπο ο όγκος του υλικού λειτουργεί ως θάλαμος ιονισμού. Παρατηρούμε ότι το ελάχιστο κβάντο ενέργειας που μπορεί να ανιχνευθεί είναι το ενεργειακό χάσμα. Μεγαλύτερες ενέργειες θα παράγουν περισσότερους φορείς σχηματίζοντας έτσι ένα μεγαλύτερο σήμα.



**Σχήμα 2.11:** Η ακτινοβολία που προσπίπτει στο υλικό μπορεί να δώσει σε ένα ηλεκτρόνιο σθένους οπύ σχηματίζει δεσμό με άλλα ηλεκτρόνια (βρίσκεται σε ζώνη σθένους) αρκετή ενέργεια έτσι ώστε ο δεσμός να σπάσει και να μεταβεί το ηλεκτρόνιο στην ζώνη αγωγιμότητας. Δημιουργείται μια κενή θέση οπή. Το ηλεκτρικό πεδίο εξαναγκάζει τα ηλεκτρόνια και έμμεσα τις οπές να κινηθούν σε αντίθετες διευθύνσεις. Πηγή: [3]

Εφόσον τα ηλεκτρόνια είναι φερμιόνια ακολουθούν κατανομή Fermi-Dirac οπότε η πιθανότητα κατάληψης μιας ενεργειακής κατάστασης είναι.

$$f_e(E) = \frac{1}{e^{(E-E_f)/kT} + 1} \quad (2.1)$$

Όπου  $E_f$  είναι η ενέργεια Fermi. Αντίστοιχα η πιθανότητα μια κατάσταση να μην έχει καταληφθεί από ηλεκτρόνιο ή αλλιώς να έχει καταληφθεί από μια οπή δίνεται από την σχέση

$$f_e(E) = 1 - f_e(E) = \frac{1}{e^{(E_f-E)/kT} + 1} \quad (2.2)$$

Σε θερμική ισορροπία ο αριθμός των ηλεκτρονίων και οπών είναι ίσος ( $n_e = n_h = n_i$ ) και η ενέργεια Fermi τοποθετείται στο μέσο του ενεργειακού χάσματος. Στη περίπτωση του Si το ενεργειακό χάσμα είναι  $E_g = 1.12\text{eV}$  και για θερμικές ενέργειες  $26\text{meV}$  ( $\approx 29^\circ\text{C}$ ) η πιθανότητα ενός ηλεκτρονίου να καταλάβει μια ενεργειακή κατάσταση στη ζώνη σθένους είναι  $4,4 \times 10^{-10}$ . Ωστόσο λόγω της μεγάλης πυκνότητας των ενεργειών καταστάσεων προκύπτει ότι οι φορείς θα έχουν συγκέντρωση  $n_i = 1.45 \times 10^{10}\text{cm}^{-3}$

Σε έναν ανιχνευτή ακτινοβολίας ο θερμικός ιονισμός οδηγεί σε μια συνεχή ροή ρεύματος από την οποία η παρουσία οποιαδήποτε σήματος λόγω ακτινοβολίας θα πρέπει να διαχωριστεί. Το υπόβαθρο αυτό είναι δύσκολο να προσδιοριστεί με ακρίβεια λόγω

των τυχαίων διακυμάνσεων πρέπει έτσι να περιοριστεί θέτοντας ένα κατώφλι. Η επιλογή του κατάλληλου υλικού ως ενεργό μέσο πρέπει να αντισταθμίζει δύο αντικρουόμενα γεγονότα. Ένα μεγάλο ενεργειακό χάσμα μειώνει το ρεύμα που παράγεται από θερμικό ιονισμό μειώνει όμως και το επιθυμητό σήμα αντίθετα ένα μικρό ενεργειακό χάσμα θα δώσει μεγάλο σήμα αλλά και μεγάλο θόρυβο. Αυτό περιορίζει αρκετά τον αριθμό των υλικών που μπορούν να χρησιμοποιηθούν για ανίχνευση ακτινοβολίας.

### 2.3.1 Διακυμάνσεις του σήματος

Η ενέργεια μπορεί να απορροφηθεί από το πλέγμα δημιουργώντας φωνόνια χωρίς δηλαδή την δημιουργία φορέων ή να προκαλέσει ιονισμό δημιουργώντας φορείς (ζευγάρια ηλεκτρονίου-οπής). Έστω ότι λόγω μιας ενέργειας  $E_0$  γίνονται  $N_x$  διεγέρσεις πλέγματος παράγοντας φωνόνια και  $N_{ion}$  αλληλεπιδράσεις ιονισμού και ότι ισχύει η σχέση

$$E_0 = E_{ion}N_{ion} + E_x N_x \quad (2.3)$$

Όπου  $E_{ion}$  η ενέργεια ιονισμού και  $E_x$  η ενέργεια διέγερσης πλέγματος. Αν υποθέσουμε ότι δεν υπάρχουν διακυμάνσεις στην ενέργεια  $E_0$  προκύπτει ότι

$$E_{ion}\Delta N_{ion} + E_x\Delta N_x = 0 \quad (2.4)$$

Παρατηρούμε ότι για ένα γεγονός αν περισσότερη ενέργεια συμβάλει στην δημιουργία φορέων λιγότερη θα συμβάλει στη διέγερση πλέγματος. Από το μέσο όρο για πολλά γεγονότα θα προκύψει ότι οι διακυμάνσεις για τους δύο μηχανισμούς θα πρέπει να είναι ίσες, Οπότε θα προκύψει η σχέση

$$\sigma_{ion} = \frac{E_x}{E_{ion}} \sqrt{N_x} \quad (2.5)$$

Από τη σχέση για την ολική ενέργεια προκύπτει

$$N_x = \frac{E_0 - E_{ion}N_{ion}}{E_x} \quad (2.6)$$

Επίσης υποθέτουμε ότι το πλήθος  $N_Q$  των φορτισμένων φορέων προκύπτει από την ολική ενέργεια προς τη ενέργεια που χρειάζεται για τη δημιουργία ενός ζευγαριού. Δηλαδή  $N_Q = N_{ion} = E_0/E_i$ . Τότε η 2.5 μπορεί να γραφεί στη παρακάτω μορφή.

$$\sigma_{ion} = \sqrt{\frac{E_0}{E_{ion}}} \sqrt{\frac{E_x}{E_{ion}} \left( \frac{E_i}{E_{ion}} - 1 \right)} \quad (2.7)$$

Όπου ο δεύτερος όρος στο δεξί μέλος της εξίσωσης 2.8 είναι ο παράγοντας Fano και το συμβολίζουμε με  $F$  οπότε τελικά η διακύμανση του αριθμού των φορτισμένων φορέων δίνεται από την σχέση.

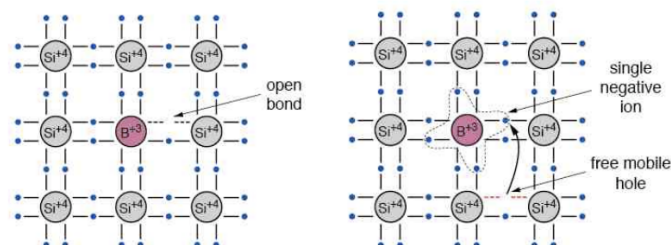
$$\sigma_{ion} = \sqrt{FN_Q} \quad (2.8)$$

Για το πυρίτιο  $E_x = 0.037eV$ ,  $E_{ion} = 1.1eV$ ,  $E_i = 3.6eV$  προκύπτει ότι η διακύμανση είναι  $\sigma_Q \approx 0.3\sqrt{N_Q}$ .

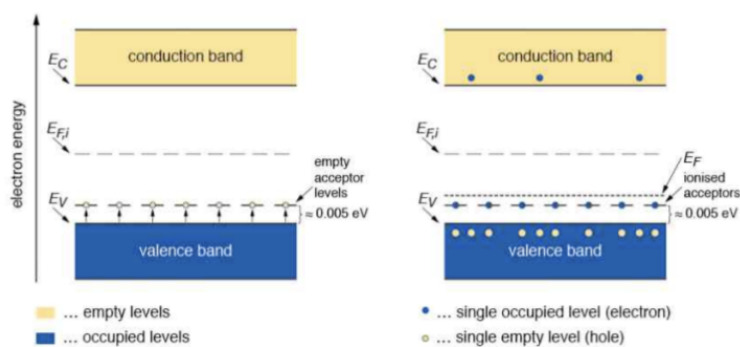
Οι φορείς που δημιουργούνται έχουν μια πιθανότητα να επανασυνδεθούν. Επίσης τα κρυσταλλικά πλέγματα περιέχουν ατέλειες οι οποίες μάλιστα αυξάνονται με τη θερμοκρασία. Οι ατέλειες αυτές μπορούν να παγιδέψουν τους φορείς. Η πιθανότητα



αντικατέστησε το πυρίτιο θα δανειστεί από το πλέγμα ένα ηλεκτρόνιο ώστε να καλυφθεί ο ασύζευκτος δεσμός. Το στοιχείο που εισάγεται στην περίπτωση αυτή ονομάζεται δέκτης. Η ενεργειακή κατάσταση του ασύζευκτου δεσμού θα βρίσκεται ελάχιστα πάνω από τη ζώνη σθένους. Λόγω της μικρής ενεργειακής διαφορά ακόμα και σε θερμοκρασίες δωματίου ηλεκτρόνια από τη ζώνη σθένους θα μεταβούν στη ενεργειακή ζώνη του δότη σχηματίζοντας μια οπή στη ζώνη σθένους.



(α')



(β')

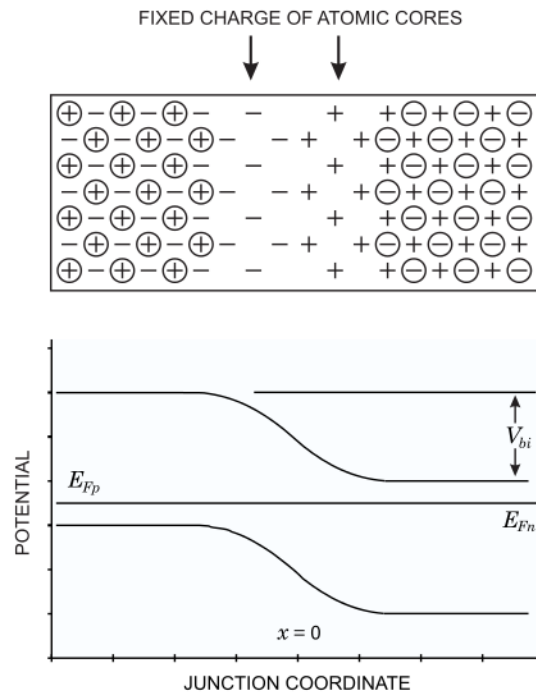
Σχήμα 2.13: Εμφύτευση ατόμου με 3 ηλεκτρόνια σθένους σε πυρίτιο. Η ενεργειακές στάθμες των ηλεκτρονίων του αποδέκτη βρίσκονται κοντά στην ζώνη σθένους. Πηγή: [3]

### Δίοδοι p-n

Στην περίπτωση που ένα υλικό ντοπαριστεί με τέτοιο τρόπο ώστε οι n και p περιοχές έρθουν σε επαφή. Αρχικά οι δύο περιοχές θα είναι ηλεκτρικά ουδέτερες. Όμως λόγω διάχυσης οι φορείς πλειονότητας στις δύο περιοχές, ηλεκτρόνια και οπές αντίστοιχα θα μεταφερθούν από την μια περιοχή στην άλλη μέχρι να υπάρξει θερμοκή ισορροπία. Με αποτέλεσμα τα εναπομείναντα ιόντα στις δύο περιοχές να δημιουργούν ένα ηλεκτρικό πεδίο το οποίο σταματάει τη περαιτέρω διάχυση. Η διάχυση ηλεκτρονίων οπών σχηματίζει μια περιοχή ελεύθερη από φορείς η οποία ονομάζεται περιοχή απογύμνωσης.

Στη θερμοκή ισορροπία η στάθμη Fermi πρέπει να είναι σταθερή σε όλο τον όγκο. Έτσι οι ενεργειακές ζώνες κάμπτονται δημιουργώντας μια διαφορά δυναμικού η οποία είναι ίση με την διαφορά μεταξύ των δύο επιπέδων fermi.  $V_{bi} = E_{Fn} - E_{Fp}$  και ονομάζεται εσωτερικό δυναμικό. Εάν εφαρμοστεί ένα εξωτερικό δυναμικό δεν υπάρχει πια θερμοκή ισορροπία. Αν η θετική τάση εφαρμοστεί στην περιοχή p και η αρνητική στην n τότε η κινητικότητα των ηλεκτρονίων και των οπών μέσα στη διόδο





Σχήμα 2.14: Δίοδος p-n. Στη δίοδο p-n οι ενεργειακές ζώνες κάμπτονται δημιουργώντας μια διαφορά δυναμικού η οποία ονομάζεται εσωτερικό δυναμικό  $V_{bi}$ . Πηγή: [3]

αυξάνεται και το  $V_{bi}$  μειώνεται. Αν εφαρμοστεί ανάστροφη πόλωση το  $V_{bi}$  αυξάνεται και το πλάτος της περιοχής απογύμνωσης αυξάνεται.

Εφόσον η περιοχή απογύμνωσης είναι ένας όγκος χωρίς φορείς θα σχηματιστεί ένας πυκνωτής όπου η περιοχή απογύμνωσης αποτελεί το διηλεκτρικό ενώ οι περιοχές p και n τα ηλεκτρόδια. Το εξωτερικό ηλεκτρικό πεδίο θα εξαναγκάζει τους φορείς να κινηθούν προς τα ηλεκτρόδια με αποτέλεσμα η δίοδος να σχηματίζει έναν θάλαμο ιονισμού. Αυτή η ιδιότητα της ανάστροφα πολωμένης διόδου τη καθιστά ιδιαίτερα χρήσιμη στην ανίχνευση ακτινοβολίας

Το συνολικό πλάτος απογύμνωσης δίνεται από τη σχέση

$$w = x_n + x_p = \sqrt{\frac{2\varepsilon V_b (N_a + N_b)}{e N_a N_b}} \quad (2.9)$$

Όπου:

- $V_b$  είναι το εφαρμοζόμενο εξωτερικό δυναμικό
- $N_a$ ,  $N_b$  οι συγκεντρώσεις δότη και αποδέκτη αντίστοιχα
- $\varepsilon$  διηλεκτρική σταθερά του υλικού.
- $x_n$ ,  $x_p$  τα πλάτη απογύμνωσης στη περιοχή p και n αντίστοιχα.

Το πλάτος απογύμνωσης αυξάνεται με την τετραγωνική ρίζα του ανάστροφα πολωμένου δυναμικού. Οπότε αυξάνοντας την τάση πόλωσης ο ενεργός όγκος αυξάνεται και η χωρητικότητα μειώνεται. Αυτό είναι ιδιαίτερα χρήσιμο αφού αυξάνεται το σήμα και μειώνεται ο θόρυβος. Παρόλα αυτά αν η τάση αυξηθεί πάρα πολύ τα ηλεκτρόνια

αποκτούν αρκετή ενέργεια ώστε να προκαλέσουν δευτερεύοντα ζευγάρια ηλεκτρονίων-οπών. Το πλάτος απογύμνωσης μπορεί να αυξηθεί μειώνοντας την συγκέντρωση του δότη ή του αποδέκτη όπως μπορούμε να παρατηρήσουμε από την εξίσωση 2.9. Αν  $N_a \gg N_d$  τότε η περιοχή απογύμνωσης εκτείνεται κυρίως στην περιοχή  $n$  και η εξίσωση 2.9 γίνεται:

$$w \approx x_n \approx \sqrt{\frac{2\varepsilon V_b}{eN_b}} \quad (2.10)$$

Αν λάβουμε υπόψιν και το εσωτερικό δυναμικό ( $V_{bi}$ ) τότε προκύπτει η σχέση

$$w \approx x_n \approx \sqrt{\frac{2\varepsilon (V_b - V_{bi})}{eN_b}} \quad (2.11)$$

Εφόσον ο όγκος του ανιχνευτή είναι ελεύθερος από φορείς σχηματίζεται ένας πυκνωτής του οποίου η χωρητικότητα δίνεται από τη σχέση.

$$C = \varepsilon \frac{A}{w} = A \sqrt{\frac{\varepsilon e N}{2(V_b + V_{bi})}} \quad (2.12)$$

Η χωρητικότητα είναι σημαντική για το λόγο σήματος προς θόρυβο του ανιχνευτή. Σε ανιχνευτές που έχουν χωριστεί κατά τμήματα η χωρητικότητα μεταξύ των τμημάτων γίνεται σημαντική.

Σε μια ανάστροφα πολωμένη δίοδο μόνο ένα πολύ μικρό ρεύμα μπορεί να ρέει από τον όγκο της, ενώ πολύ μεγάλη τάση και άρα ηλεκτρικό πεδίο μπορεί να εφαρμοστεί στη ζώνη απογύμνωσης. Το ρεύμα που δημιουργείται σε μια ανάστροφα πολωμένη δίοδο οφείλεται κυρίως στις ατέλειες που υπάρχουν στον όγκο του υλικού και εξαρτάται από τη θερμοκρασία. Το ρεύμα αυτό ονομάζεται ρεύμα Shockley ή ρεύμα διαρροής.

$$I(T) = I_0 \left( \frac{T}{T_0} \right) \exp \left( \frac{E_g}{2kT_0} - \frac{E_g}{2kT} \right) \quad (2.13)$$

Όπου

- $E_g$  η ενέργεια της ζώνης χάσματος
- $I_0, T_0$  αρχικό ρεύμα και θερμοκρασία αντίστοιχα
- $k$  σταθερά Boltzman

## 2.5 Κίνηση φορέων

Οι φορτισμένοι φορείς μπορούν να κινούνται μέσα στο υλικό μέσω τον μηχανισμών της διάχυσης ή της ολίσθησης. Η διάχυση οφείλεται στη διαφορά συγκέντρωσής ανάμεσα στις περιοχές. Οι φορείς κινούνται από τις περιοχές με υψηλή συγκέντρωση στις χαμηλές. Μετά από χρόνο  $t$  η συγκέντρωση έχει κατανεμηθεί στον χώρο με τυπική απόκλιση

$$\sigma = \sqrt{Dt} \quad (2.14)$$

Όπου  $D$  είναι η σταθερά ολίσθησης



Η ολίσθηση οφείλεται στη παρουσία ηλεκτρικού πεδίου. Τα φορτία τείνουν να κινηθούν παράλληλα με το ηλεκτρικό πεδίο με ταχύτητα που δίνεται από τη σχέση:

$$\vec{u} = \mu \vec{E} \quad (2.15)$$

Όπου  $\mu$  είναι η κινητικότητα των φορέων. Η κινητικότητα των οπών είναι πολύ μικρότερη της κινητικότητας των ηλεκτρονίων. Τα ηλεκτρόνια έχουν τρεις φορές μεγαλύτερη κινητικότητα από τις οπές. Οι κινητικότητες των ηλεκτρονίων και των οπών καθορίζουν το χρόνο συλλογής του φορτίου. Επίσης τα γινόμενα της κινητικότητας με τον χρόνο ημιζωής για τα ηλεκτρόνια και τις οπές  $(\mu\tau)_e$  και  $(\mu\tau)_h$  καθορίζουν το μέγιστο χρήσιμο πάχος του ανιχνευτή. Η σταθερά ολίσθησης και η κινητικότητα συνδέονται μέσω της σχέσης του Einstein

$$D = \mu \frac{kT}{e} \quad (2.16)$$

Το σήμα εκ πρώτης όψεως δημιουργείται από τη στιγμή που τα φορτία φθάνουν στα ηλεκτρόδια του ανιχνευτή όμως αυτό δεν είναι ακριβώς σωστό αφού η ροή ρεύματος στα ηλεκτρόδια ξεκινάει ακαριαία με την κίνηση των φορτίων στον όγκο του ανιχνευτή.

## 2.6 Κατασκευή ανιχνευτών πυριτίου

Ένα βασικό ερώτημα είναι γιατί η επιλογή του πυριτίου ως ενεργό μέσο αφού άλλοι τύποι ημιαγωγών π.χ GaAs έχουν καλύτερες ιδιότητες, όπως μεγαλύτερη ευκινησία φορέων (κινητικότητα ηλεκτρονίου 8800,  $\frac{cm^2}{V}$  ενώ 1450,  $\frac{cm^2}{V}$  για το Si ) ή μεγαλύτερο ενεργειακό χάσμα (1,32 eV για το GaAs , 1.1 eV για το Si). Η απάντηση είναι ο ίδιος λόγος που το πυρίτιο έχει γίνει ο πιο σημαντικός ημιαγωγός στη βιομηχανία των ηλεκτρονικών. Το πυρίτιο έχει ικανοποιητικές ιδιότητες και παράλληλα είναι φθηνό και εύκολο στη χρήση. Το πυρίτιο έχει ένα πολύ ανθεκτικό φυσικό οξειδίο το οποίο αν αναπτυχθεί στις επιφάνειες παρέχει μια καλά ελεγχόμενη διεπαφή, ενώ προστατεύει τον εσωτερικό όγκο από εξωτερικούς παράγοντες.

### Ιοντική Εμφύτευση

Η ιοντική εμφύτευση είναι μια βασική τεχνική για τη δημιουργία υλικών με περίσσια ηλεκτρονίων ή οπών καθώς επιτρέπει τον ακριβή έλεγχο της συγκέντρωσης και του βάθους. Η ιοντική εμφύτευση γίνεται με επιταχυντές οι οποίοι βομβαρδίζουν το στόχο με άτομα δότες ή με άτομα αποδέκτες αντίστοιχα. Τα άτομα αυτά τοποθετούνται σε ενδιάμεσες ενδοπλεγματικές θέσεις και είναι ηλεκτρικά ανενεργά. Μετά την ιοντική εμφύτευση του υλικού γίνεται θερμική ανώπτηση του υλικού. Κατά τη διαδικασία αυτή το υλικό θερμαίνεται σε πολύ υψηλές θερμοκρασίες για πολύ μικρό χρονικό διάστημα και έπειτα ψύχεται. Τα άτομα που βρίσκονταν αρχικά σε ενδιάμεσες ενδοπλεγματικές θέσεις διαχέονται στο υλικό ενώ παράλληλα λόγω της αύξησης της θερμοκρασίας δημιουργούνται ατέλειες κενών θέσεων (κενές θέσεις πλέγματος ) οι οποίες μπορούν να καταληφθούν από τα άτομα που εμφυτεύτηκαν σχηματίζοντας δεσμούς. Η διαδικασία αυτή επιτρέπει τη ηλεκτρική ενεργοποίηση του υλικού και θα πρέπει να είναι αρκετά γρήγορη ώστε να μην δημιουργηθούν περαιτέρω ατέλειες.

### Θερμική οξείδωση

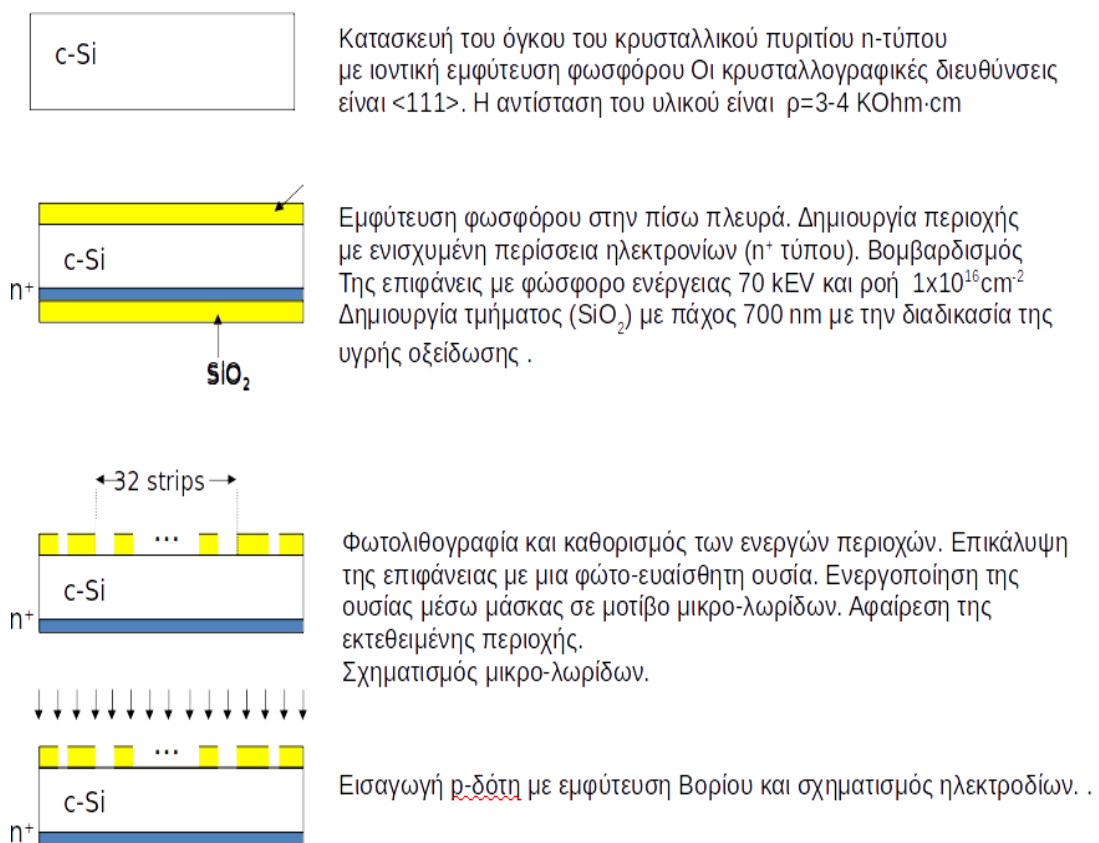
Οι επιφάνειες του πυριτίου θα πρέπει να προστατευτούν από ένα στρώμα οξειδίου του πυριτίου το οποίο θα απομονώνει το υλικό ενώ παράλληλα θα λειτουργεί ως διηλεκτρικό για πυκνωτές και πύλες τρανζίστορ. Τα οξείδια δημιουργούνται θερμικά ή μέσω μιας διαδικασίας που ονομάζεται χημική εναπόθεση ατμών.

### Επιμετάλλωση

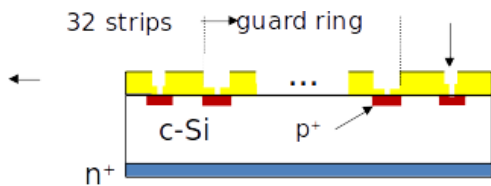
Για να υπάρξει ηλεκτρική επαφή μεταξύ των ντοπαρισμένων περιοχών του πυριτίου και των εξωτερικών ηλεκτρονικών απαιτείται επιμετάλλωση. Σε έναν ανιχνευτή μικρο-λωρίδων η επιμετάλλωση θα πρέπει να γίνει σε όλη την έκταση της. Συνήθως κατά την επιμετάλλωση χρησιμοποιείται αλουμίνιο (Al). Τα επίπεδα Fermi των δύο υλικών που θα έρθουν σε επαφή κατά την επιμετάλλωση θα πρέπει να έχουν συμβατά επίπεδα Fermi έτσι ώστε να επιτευχθεί ωμική επαφή. Η επιμετάλλωση γίνεται είτε με θερμική εξάτμιση είτε με την διαδικασία sputtering κατά την οποία μια κάθοδος από αλουμίνιο βομβαρδίζεται από άλλα ιόντα όπως Αργό εκπέμποντας άτομα Al.

### Φωτολιθογραφία

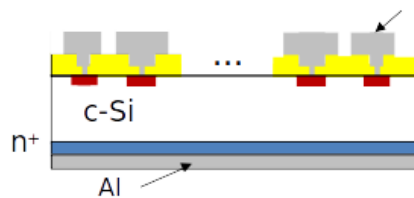
Το σχήμα και το μέγεθος της συσκευής καθορίζεται με την μέθοδο της φωτολιθογραφίας. Μια φωτοευαίσθητη ουσία επικαλύπτει την επιφάνεια. Η ουσία αυτή ενεργοποιείται με έκθεση σε φωτεινή ακτινοβολία αφού πρώτα έχει τοποθετηθεί πάνω από αυτή μια μάσκα η οποία θα δώσει το κατάλληλο σχήμα. Η ενεργοποιημένες περιοχές διαλύονται με την χρήση κατάλληλων διαλυμάτων. Η ουσία αυτή ονομάζεται θετικά φωτοευαίσθητη. Μια αρνητικά φωτοευαίσθητη ουσία δρα με το αντίθετο ακριβώς τρόπο όταν εκτεθούν σε φωτεινή ακτινοβολία δεν μπορούν να διαλυθούν. Αυτό είναι χρήσιμο κατά την κατασκευή επαφών στην επιμετάλλωση.



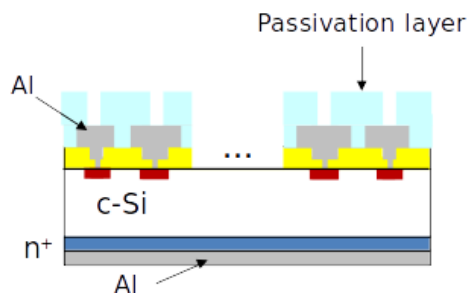
Σχήμα 2.15: Διαδικασία κατασκευής ανιχνευτών μικρο-λωρίδων.



Οξειδωση στην περιοχή των ηλεκτροδίων.  
Φωτολιθογραφία και σχηματισμός ηλεκτρικών επαφών.



Εναπόθεση αλουμινίου (Al) και δημιουργία ηλεκτρικών επαφών  
Φωτολιθογραφία καθορισμός των ηλεκτρικών επαφών κάθε Ηλεκτροδίου.



Επίστρωση  $\text{SiO}_2$  παθητικοποίηση του ανιχνευτή.  
Φωτολιθογραφία για την δημιουργία περιοχών συγκόλλησης με τα εξωτερικά ηλεκτρονικά.

Σχήμα 2.16

Σχήμα 2.17: Διαδικασία κατασκευής ανιχνευτών μικρο-λωρίδων.

## 2.7 Ενίσχυση σήματος

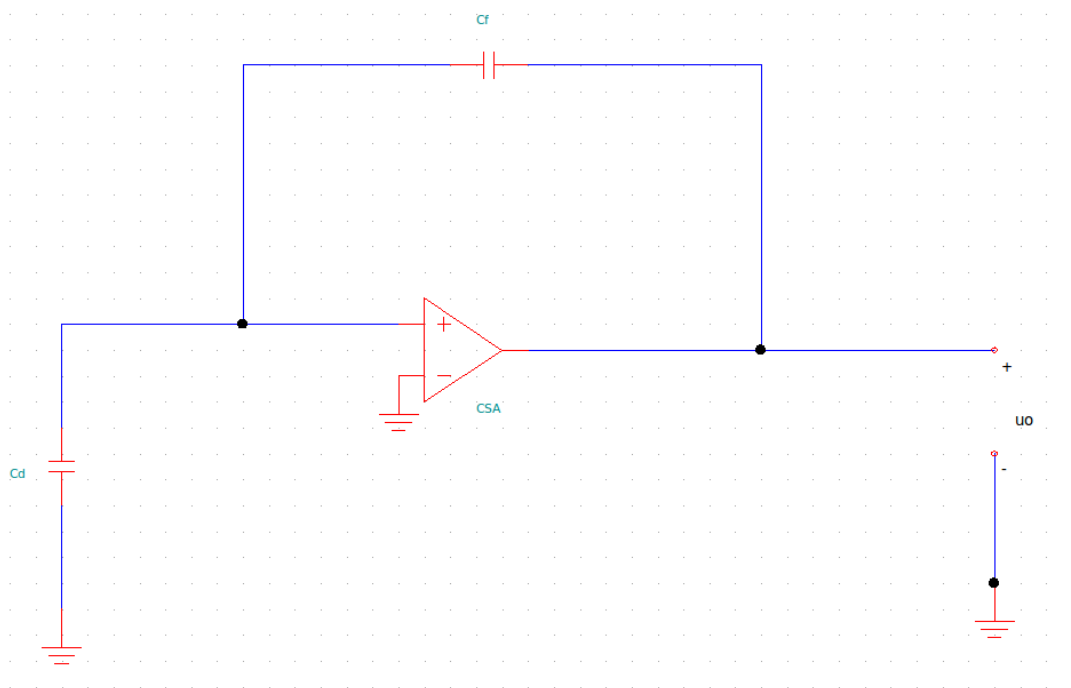
Εφόσον το σήμα σε έναν ανιχνευτή πυριτίου δημιουργείται από το φορτίο ο ενισχυτής σήματος που συνήθως χρησιμοποιείται είναι ο ενισχυτής φορτίου (charge sensitive amplifier CSA). ένα φορτίο  $Q_{in}$  το οποίο δημιουργείται στον όγκο του ανιχνευτή επάγει φορτίο στους οπλισμούς του πυκνωτή το οποίο προκαλεί ένα βηματικό παλμό τάσης στην είσοδο του ενισχυτή.

Το βασικό στοιχείο ενίσχυσης είναι ένας αναστρέφων ενισχυτής τάσης με πολύ μεγάλη αντίσταση εισόδου. Όπου ένας πυκνωτής ανατροφοδότησης ( $C_f$ ) συνδέει την έξοδο με την είσοδο. Για απλοποίηση του προβλήματος μπορούμε να θεωρήσουμε ένα ιδανικό ενισχυτή με άπειρη αντίσταση εισόδου έτσι ώστε να μην υπάρχει ροή ρεύματος στον ενισχυτή.

Το κέρδος τάσης θα δίνεται από τη σχέση:

$$\frac{du_0}{du_i} = -A \quad \text{άρα} \quad u_0 = -Au_i \quad (2.17)$$

Η διαφορά τάσης στους οπλισμούς του πυκνωτή ( $C_f$ ) θα είναι  $u_f = (A + 1)u_i$  και άρα το φορτίο στον πυκνωτή ( $C_f$ ) θα δίνεται από τη σχέση  $Q_f = C_f(A + 1)u_i$ . Εφόσον δεν μπορεί να διαρεύσει ρεύμα από τον ενισχυτή όλο το ρεύμα θα φορτίσει τον αναστρέφων πυκνωτή. Το φορτίο στον αναστρέφων πυκνωτή στην ιδανική περίπτωση θα είναι ίσο σε το φορτίο στη είσοδο του πυκνωτή ( $Q_i$ ).



Σχήμα 2.18: Βασική μορφή ενισυτή φορτίου

Η είσοδος του ενισχυτή μπορεί να αναπαρασταθεί από μια δυναμική χωρητικότητα εισόδου ( $C_i$ ). όπου :

$$C_i = \frac{Q_i}{u_i} = C_f(A + 1) \quad (2.18)$$

Η τάση εξόδου ανα μονάδα φορτίου στην είσοδο του ενισχυτή θα δίνεται από τη σχέση :

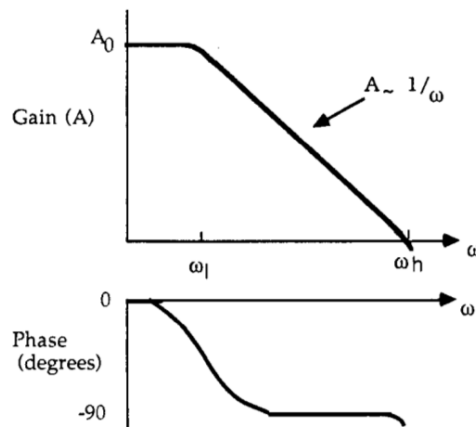
$$A_Q = \frac{u_0}{Q_i} = \frac{A u_i}{C_i u_i} = \frac{A}{C_i} = \frac{A}{A+1} \frac{1}{C_f} \approx \frac{1}{C_f} \quad (A \gg 1) \quad (2.19)$$

Το φορτίο σήματος ( $Q_s$ ) θα καταναμηθεί μεταξύ της χωρητικότητας του πυκνωτή ( $C_d$ ) και της χωρητικότητας του ενισχυτή ( $C_i$ ). Δηλαδή ( $Q_s = Q_d + Q_i$ ). Ο λόγος του φορτίου που αποθηκεύεται στον ενισχυτή προς το συνολικό φορτίο του σήματος είναι:

$$\frac{Q_i}{Q_s} = \frac{Q_i}{Q_d + Q_i} = \frac{C_i}{C_d + C_i} = \frac{1}{1 + \frac{C_d}{C_i}} \quad (2.20)$$

Το ( $Q_i$ ) που αποθηκεύεται στον ενισχυτή θα είναι και το μετρούμενο φορτίο (σήμα). Θέλουμε επομένως ( $Q_i \approx Q_s$ ) δηλαδή η μεγαλύτερη ποσότητα του φορτίου (σήματος) να αποθηκευτεί στον ενισχυτή. Άρα θα πρέπει ( $C_i > C_d$ )

Στην πραγματικότητα το κέρδος  $A$  του ενισχυτή είναι συνάρτηση της συχνότητας. Το κέρδος του ενισχυτή είναι αντιστρόφως ανάλογο της συχνότητας για συχνότητες μεγαλύτερες από την συχνότητα αποκοπής του ενισχυτή. Επίσης το ενισχυμένο σήμα παρουσιάζει διαφορά φάσης  $90^\circ$  για μεγάλες συχνότητες.



Σχήμα 2.19: Ο ενισχυτής μπορεί να αναπαρασταθεί από μια τάση και έναν πυκνωτή σε σειρά

Ο ενισχυτής εισόδου θα έχει μια εμπέδηση εισόδου η οποία εξαρτάται από την συχνότητα ( $Z_i = \frac{1}{\omega_0 C_f}$ ). Οπότε προκύπτει ότι

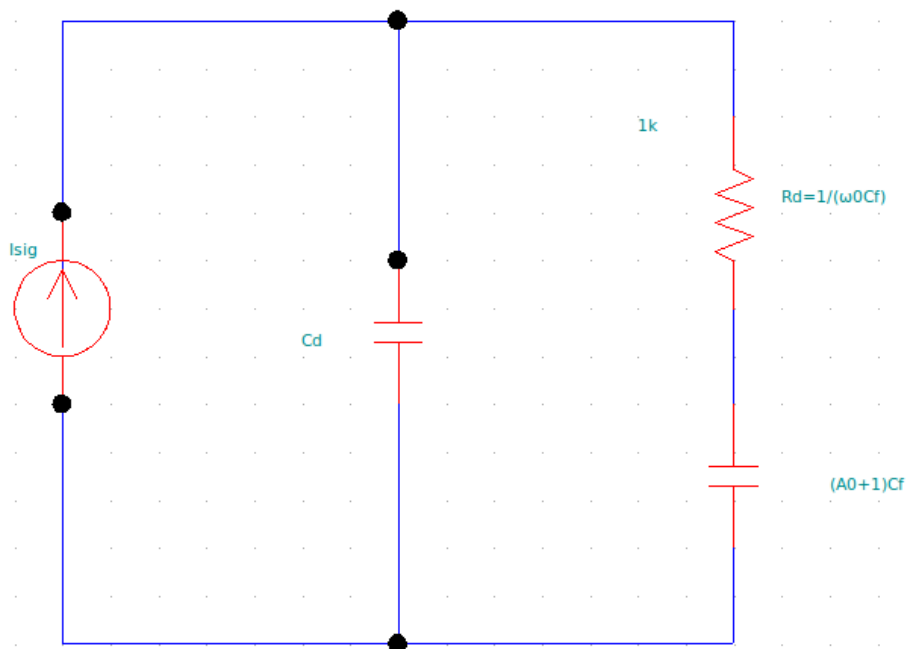
$$v_i = \frac{i}{(A+1)\omega_0 C_f} \quad (2.21)$$

Όπου  $\omega_0$  συχνότητα αποκοπής.

Ο ενισχυτής φορτίου μπορεί να αναπαρασταθεί σαν ένας πυκνωτής σε σειρά με μια αντίσταση κάτι που προκύπτει από την σχέση 2.21 και αφού σε πολύ χαμηλές συχνότητες θέλουμε να συμπεριφέρεται σαν ανοιχτό κύκλωμα "άπειρη αντίσταση".

Σε έναν μη ιδανικό ενισχυτή η απόκριση του σήματος δεν είναι ακαριαία αλλά υπάρχει μια καθυστέρηση η οποία εξαρτάται από τον χαρακτηριστικό χρόνο ανόδου του ενισχυτή

$$\tau_r = C_d R_d = \frac{C_d}{\omega_h C_f} \quad (2.22)$$



Σχήμα 2.20: Ο ενισχυτής μπορεί να αναπαρασταθεί από μια τάση και έναν πυκνωτή σε σειρά

Όπου  $\omega_h$  συχνότητα μοναδιαίου εύρους. Για ένα βηματικό παλμό το σήμα στην έξοδο του ενισχυτή θα δίνεται από τη σχέση:

$$v_0(t) = -(Q/C_f)(1 - \exp(-t/\tau_r)) \quad (2.23)$$

## 2.8 Ηλεκτρονικός θόρυβος

Το ελάχιστο όριο ανίχνευσης εκτός από τις διακυμάνσεις του σήματος που προέρχονται από τον ανιχνευτή και από την στατιστική φύση της διαδικασίας της ανίχνευσης περιορίζεται και από διακυμάνσεις λόγω των ηλεκτρονικών. Ο ηλεκτρονικός θόρυβος προέρχεται από διακυμάνσεις στην ταχύτητα και τον αριθμό των φορέων.

Όπως αναφέραμε στην προηγούμενη ενότητα το φορτίο που δημιουργείται μέσα στον όγκο ενός ανιχνευτή επάγει φορτίο στα ηλεκτρόδια καθώς αυτό κινείται μέσα στον όγκο του ανιχνευτή πριν αυτό συλλεχθεί. Αν υποθέσουμε ότι ( $n$ ) ηλεκτρόνια κινούνται μέσα στον ανιχνευτή με ταχύτητα ( $u$ ) τότε το επαγόμενο ρεύμα θα εξαρτάται από την απόσταση μεταξύ των ηλεκτροδίων ( $s$ ). Θα ισχύει δηλαδή η σχέση: (Θεώρημα Ramo [3]).

$$i = \frac{neu}{s} \quad (2.24)$$

Η διακύμανση του ρεύματος δίνεται από την σχέση

$$\langle di \rangle^2 = \left( \frac{ne}{s} \langle du \rangle \right)^2 + \left( \frac{eu}{s} \langle dn \rangle \right)^2 \quad (2.25)$$

Παρατηρούμε ότι δύο παράγοντες συμβάλλουν στον συνολικό θόρυβο, οι διακυμάνσεις στην ταχύτητα και οι διακυμάνσεις στον αριθμό των φορέων.

Οι διακυμάνσεις στην ταχύτητα οφείλονται κυρίως στη θερμική κίνηση των ηλεκτρονίων. Στη διακύμανση του αριθμού των φορέων συμβάλλουν πολλοί παράγοντες. Ένας παράγοντας είναι το ότι η πιθανότητα να μπορέσει ένα ηλεκτρόνιο να περάσει το φράγμα δυναμικού είναι ανεξάρτητη από την πιθανότητα ένα άλλο ηλεκτρόνιο να περάσει να το φράγμα δυναμικού (θόρυβος "Shot"). Είναι δηλαδή δυο φαινόμενα τυχαία και δεν σχετίζονται μεταξύ τους. Επίσης ατέλειες του κρυσταλλικού πλέγματος μπορούν να παγιδεύσουν τους φορείς με μία πιθανότητα να τους αποδεσμεύσουν πάλι μετά από ένα χαρακτηριστικό χρόνο ζωής.

Μερικοί κοινοί τύποι θορύβου που συναντάμε και στους ανιχνευτές πυριτίου είναι:

### 1. Θόρυβος "Shot"

Οι φορείς στον όγκο του υλικού δημιουργούνται ανεξάρτητα μεταξύ τους. Η φασματική πυκνότητα ρεύματος δίνεται από τη σχέση

$$i_n^2 = 2eI \quad (2.26)$$

### 2. Θερμικός θόρυβος (Johnson)

Το πιο κοινό παράδειγμα θορύβου λόγω διακυμάνσεις στην ταχύτητα των φορέων είναι ο θερμικός θόρυβος (Johnson).

Η φασματική πυκνότητα θορύβου ανά συχνότητα δίνεται από τη σχέση

$$\frac{dP_n}{df} = 4kT \quad (2.27)$$



όπου  $k$  είναι η σταθερά Boltzmann και  $T$  η απόλυτη θερμοκρασία. Αντίστοιχα η φασματική πυκνότητα τάσης του θορύβου καθώς και η φασματική πυκνότητα ρεύματος δίνονται από τις σχέσεις:

$$\frac{di_n^2}{df} = \frac{4kT}{R}$$

$$\frac{du_n^2}{df} = e_n = 4kTR$$

### 3. Θόρυβος χαμηλής συχνότητας (1/f)

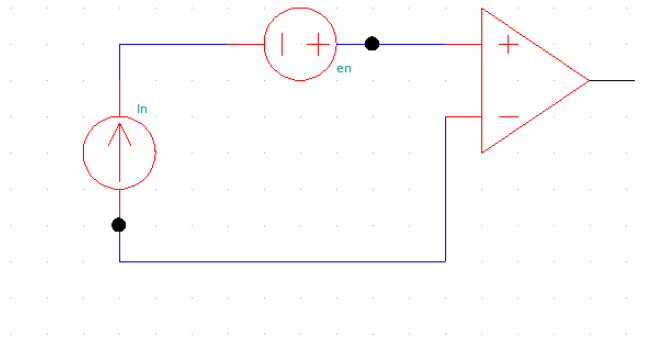
Το φάσμα του θορύβου δεν είναι ομοιόμορφο σε όλες τις συχνότητες στην περίπτωση που οι διακυμάνσεις δεν είναι εντελώς τυχαίες για παράδειγμα όταν φορείς παγιδεύονται στο κρυσταλλικό πλέγμα και απελευθερώνονται μετά από ένα συγκεκριμένη χρονική σταθερά  $\tau$ .

Εφόσον οι ανιχνευτές συνήθως χρησιμοποιούνται για να μετρούν φορτίο για να εκφράσουμε τον θόρυβο σε απολυτές μονάδες συνήθως ο θόρυβος χαρακτηρίζεται από μία ποσότητα που ονομάζεται θόρυβος ισοδύναμου φορτίου (ENC) και δίνεται από τη σχέση 2.28 και ισοδυναμεί με την ποσότητα εκείνη του φορτίου που θα συμβάλει σε λόγο σήματος προς θόρυβο ( $S/N = 1$ ).

$$ENC = \frac{Q_s}{S/N} \quad (2.28)$$

#### 2.8.1 Ενίσχυση θορύβου

Θεωρούμε ότι ο θόρυβος περιγράφεται πλήρως από μια πηγή θορύβου τάσης σε σειρά και μια πηγή θορύβου ρεύματος παράλληλα στον ενισχυτή.



Σχήμα 2.21: Κύκλωμα ανάλυσης θορύβου για ενισχυτή φορτίου.

Η πηγή θορύβου που είναι συνδεδεμένη παράλληλα προέρχεται από το θόρυβο shot και το ρεύμα διαρροής  $I$  καθώς και στο θερμικό θόρυβο λόγω των αντιστάσεων. Προκύπτει

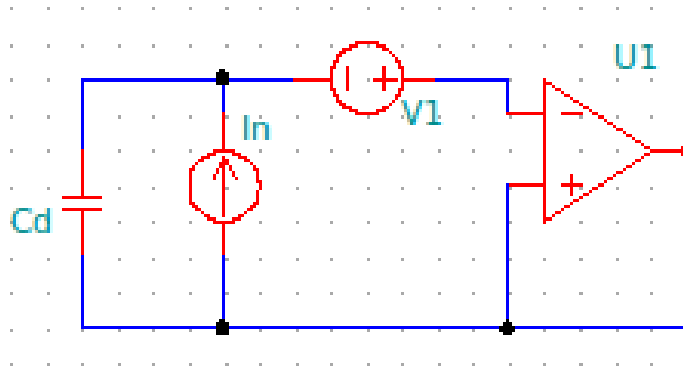
$$i_n^2 = (2eI + 4kT/R) \quad (2.29)$$

Συνήθως επιλέγεται η αντίσταση R να είναι αρκετά μεγάλη με αποτέλεσμα να κυριαρχεί το ρεύμα διαρροής.

Η πηγή θορύβου που συνδέεται σε σειρά προέρχεται από της διακυμάνσεις του ρεύματος η οποία αντιστοιχεί σε πηγή τάσης στην είσοδο του ενισχυτή.

$$e_n^2 = 4kT(2/3g_m) \quad (2.30)$$

Όπου  $g_m$  η διαγωγιμότητα του ενισχυτή. Το μέγεθος του θορύβου είναι ανεξάρτητο της συχνότητας. Θεωρείται ίδιο σε όλο το φάσμα συχνοτήτων λόγω της τυχαιότητας της φύσης των συγκεκριμένων θορύβων. Όταν συνδεθεί όμως ο ανιχνευτής, προκύπτει μια ποσότητα που εξαρτάται από την συχνότητα λόγω της εμπέδησης του ανιχνευτή. Ο συνολικός θόρυβος στην είσοδο του ενισχυτή εκφρασμένος ως ρεύμα θορύβου είναι:



Σχήμα 2.22: Κύκλωμα ανάλυσης θορύβου για ενισχυτή φορτίου με συνδεδεμένο τον ανιχνευτή.

$$I_n^2 = i_n^2 + e_n^2 \omega^2 C_d^2 \quad (2.31)$$

Όπου πλέον ο θόρυβος εξαρτάται από την συχνότητα.

Το σήμα όπως αναφέραμε στην αρχή του κεφαλαίου τροφοδοτείται αφού ενισχυθεί σε έναν διαμορφωτή παλμών ο οποίος στην απλή περίπτωση αποτελείται από δύο φίλτρα ένα χαμηλό-περατό και ένα υψηλό-περατό. Το σήμα στην έξοδο από τον διαμορφωτή παλμών θα έχει έναν χαρακτηριστικό χρόνο ανόδου και πτώσης ( $\tau$ ) ο οποίος θα προκύπτει από τα χαρακτηριστικά (R, C) των δύο φίλτρων. Ο θόρυβος μπορεί να εκφραστεί στην ποσότητα ισοδύναμη του φορτίου (ENC) σύμφωνα με τις σχέσεις:

$$\sigma_{noise}^2 = ENC^2 = ai_n^2\tau + be_n^2C_d^2/\tau + c \quad (2.32)$$

$$\sigma_{noise}^2 = a2eI + b4kT(2/3g_m)C_d^2/\tau + c \quad (2.33)$$

Όπου οι σταθερές a,b,c σχετίζονται με τις ιδιότητες του διαμορφωτή παλμών. Προκύπτει ότι ο θόρυβος εξαρτάται από το ρεύμα διαρροής τη χωρητικότητα και τον χαρακτηριστικό χρόνο από τα δύο φίλτρα του διαμορφωτή παλμών [6].

### 2.8.2 Χωρητικότητες ως πηγές θορύβου

Ο θόρυβος που εξέρχεται από τον ανιχνευτή ανατροφοδοτείται στην είσοδο με αντίθετη φάση μειώνοντας το θόρυβο στην έξοδο μέσω της σχέσης  $e_{n0} = e_{ni}A_u$  όπου  $A_u$  είναι το κέρδος ανοιχτού βρόγχου. Το μέγεθος του σήματος που ανατροφοδοτείται εξαρτάται από την συνολική εμπέδηση εισόδου και άρα από την χωρητικότητα του ανιχνευτή. Επομένως ο λόγος σήματος προς θόρυβο στην έξοδο του ανιχνευτή εξαρτάται από την ποσότητα του σήματος που ανατροφοδοτείται, το οποίο μεταβάλλεται με την χωρητικότητα του ανιχνευτή.

Ένα απλό σχήμα για να εξετάσουμε τον θόρυβο παρουσιάζεται στην εικόνα 2.23. Η εξερχόμενη τάση θορύβου ανατροφοδοτείται στην είσοδο μέσω του διαιρέτη τάσης  $C_f$  και  $C_d$ . Προκύπτει η σχέση που συνδέει την τάση θορύβου που ανατροφοδοτείται με την τάση θορύβου εξόδου.

$$e_{n0} = e_{nf} \frac{X_{C_f} + X_{C_d}}{X_{C_d}} = e_{nf} \frac{1/\omega C_f + 1/\omega C_d}{1/\omega C_d} = e_{nf} \left(1 + \frac{C_d}{C_f}\right) \quad (2.34)$$

Ο ισοδύναμος θόρυβος φορτίου σύμφωνα με την σχέση 2.19 ισούται με :

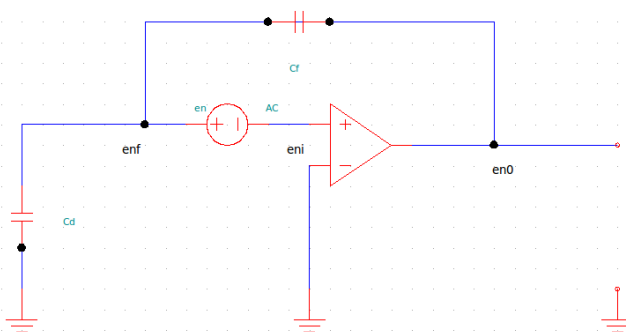
$$Q_{ni} = \frac{e_{n0}}{A_Q} = e_{n0} C_f \quad (2.35)$$

$$Q_{ni} = e_{ni} (C_D + C_f) \quad (2.36)$$

Προκύπτει ότι ο λόγος σήματος προς θόρυβο είναι:

$$\frac{Q_s}{Q_{ni}} = \frac{Q_s}{e_{ni} (C_d + C_f)} \approx \frac{1}{C_d} \frac{Q_s}{e_{ni}} \quad (C_d \gg C_f) \quad (2.37)$$

Παρατηρούμε ότι το σήμα παραμένει σταθερό ενώ ο θόρυβος αυξάνεται με την χωρητικότητα του ανιχνευτή  $C_d$ . Επίσης η χωρητικότητα ανατροφοδότησης προστίθεται στην χωρητικότητα συμβάλλοντας έτσι στον συνολικό θόρυβο. Όμως αυτή η συνεισφορά μπορεί να θεωρηθεί αμελητέα εφόσον ο σχεδιαστικός στόχος είναι  $C_f \ll C_d$ . Επιπλέον στοιχεία αντίστασης στο κύκλωμα ανατροφοδότησης θα συνεισφέρουν στον θερμικό θόρυβο.



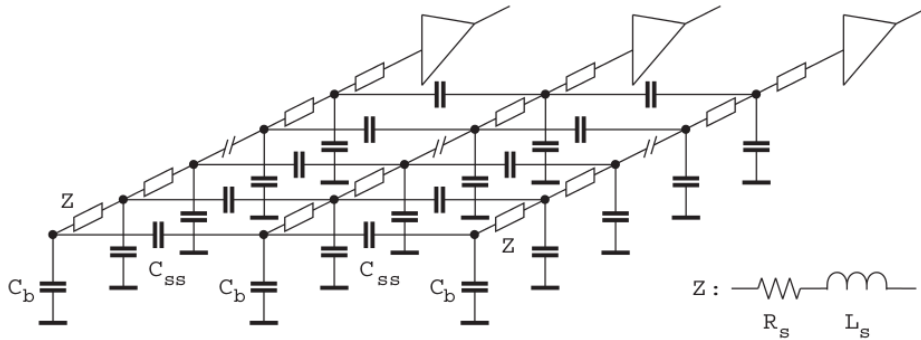
Σχήμα 2.23: Κύκλωμα ανάλυσης θορύβου για ενισχυτή φορτίου.

### 2.8.3 Πολυπλοκότερες διατάξεις

Μέχρι στιγμής εξετάσαμε την χωρητικότητα στην απλή περίπτωση όπου τα ηλεκτρόδια του ανιχνευτή θεωρούνται ενιαία. Όμως το συγκεκριμένο μοντέλο δεν μπορεί να θεωρηθεί επαρκές στην περίπτωση που στον ανιχνευτή έχουμε περισσότερα από ένα κανάλια. Ένα πιο πολύπλοκο σύστημα όπως στην περίπτωση ενός ανιχνευτή μικρο-λωρίδων χρειάζεται διαφορετική αντιμετώπιση. Ο θόρυβος εκτιμάται ως προς το κεντρικό κανάλι και συμπεριλαμβάνεται και ο θόρυβος σε συνδυασμό με τα γειτονικά κανάλια.

Στην εικόνα 2.24 παρουσιάζεται ένα μοντέλο για την εκτίμηση του θορύβου σε ένα σύστημα που αποτελείται από μικρο-λωρίδες. Κάθε μικρο-λωρίδα θεωρείται ένα κύκλωμα RLC όπου  $R$ ,  $L$  είναι η αντίσταση και η αυτεπαγωγή κάθε μικρο-λωρίδας ανα μονάδα μήκους και  $C_{ss}$ ,  $C_b$  οι χωρητικότητες μεταξύ των μικρο-λωρίδων και μεταξύ των μικρο-λωρίδων και του κάτω επιπέδου.

Στο μοντέλο αυτό κάθε μικρο-λωρίδα συμπεριφέρεται σαν ένα χαμηλό-περατό φίλτρο με συχνότητα αποκοπής  $f_u = (2\pi R_{strip} C_{strip})$ , οπότε ο θερμικός θόρυβος σε υψηλές συχνότητες θα είναι εξασθενημένος.



Σχήμα 2.24: Κύκλωμα ανάλυσης θορύβου για ανιχνευτές μικρο-λωρίδων. Τα ηλεκτρόδια των μικρο-λωρίδων θεωρούνται κυκλώματα RLC. Η εμπέδηση  $Z$  ισούται με την αντίσταση και την αυτεπαγωγή της μικρο-λωρίδας ανα μονάδα μήκους.  $C_{ss}$  είναι η χωρητικότητα ως προς της γειτονικές μικρο-λωρίδες και  $C_b$  η χωρητικότητα ως προς το κάτω επίπεδο Πηγή: [3]

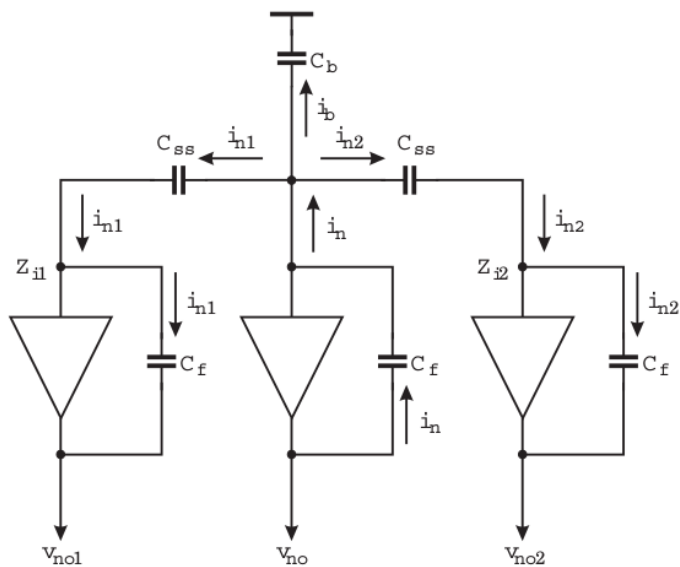
Το ρεύμα που προέρχεται από τον κεντρικό ενισχυτή διαχωρίζεται μεταξύ των γειτονικών χωρητικότητων και της χωρητικότητας σε σχέση με το πίσω επίπεδο. Σύμφωνα με την εικόνα 2.25 το ρεύμα που διαρρέει τον κεντρικό κλάδο θα δίνεται από την σχέση:

$$i_n = \frac{u_{n0}}{\frac{1}{\omega C_f} + \frac{1}{\omega(2C_{ss} + C_b)}} \quad (2.38)$$

Το ρεύμα αυτό διαχωρίζεται σε δύο ρεύματα  $i_{n1}$ ,  $i_{n2}$

$$i_{n1} = i_{n2} = \frac{i_n}{2} \left( 1 - \frac{1}{1 + 2C_b/C_{ss}} \right) \quad (2.39)$$

Οπότε το επιπλέον τάση θορύβου στην έξοδο των γειτονικών ενισχυτών λόγω του ρεύματος που ρέει το κεντρικό θα είναι :



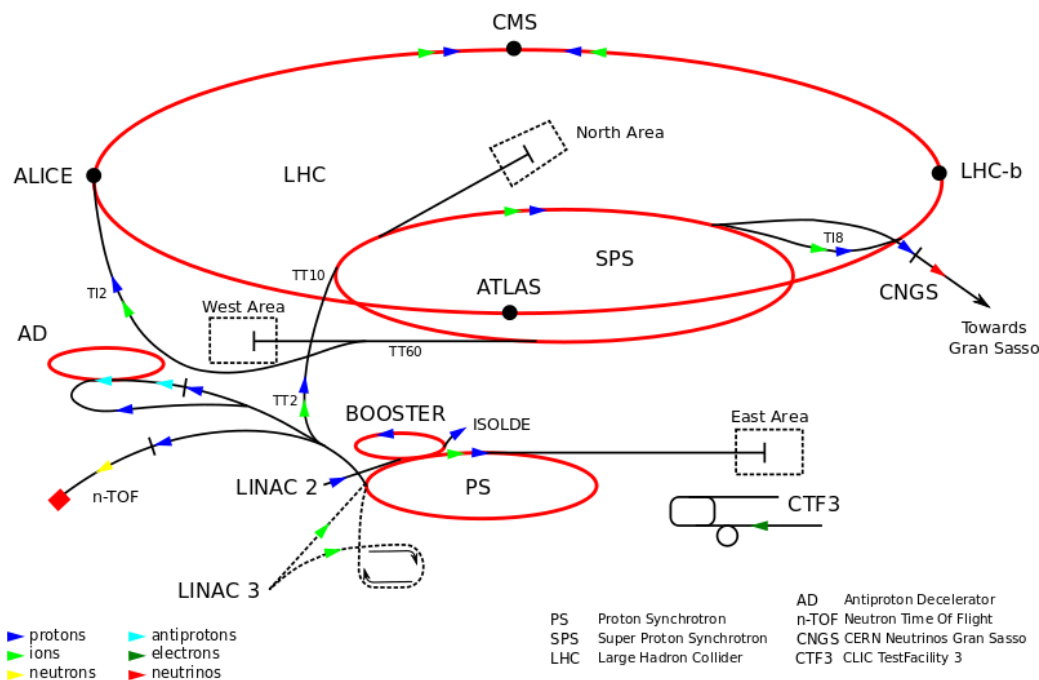
Σχήμα 2.25: Κύκλωμα για την μελέτη του τρόπου συνεισφοράς των γειτονικών μικρο-λωρίδων στο θόρυβο Πηγή: [3]

$$u_{n01} = u_{n02} = \frac{i_{n1}}{\omega C_f} = \frac{u_{n0}}{2} \frac{2C_{ss} + C_b}{2C_{ss} + C_b + C_f} \left( \frac{C_{ss}}{2C_b + C_{ss}} \right) \quad (2.40)$$

$$\approx \frac{u_{n0}}{2} \frac{1}{1 + 2C_b/C_{ss}} \quad (C_f \ll 2C_{ss} + C_b) \quad (2.41)$$

## 2.9 Ανιχνευτής CMS

Ο ανιχνευτής Compact Muon Solenoid (CMS) είναι ένας από τους μεγαλύτερους ανιχνευτές στοιχειωδών σωματιδίων ο οποίος έχει κατασκευαστεί στον μεγάλο αδρονικό επιταχυντή (LHC) στο CERN. Ο ανιχνευτής έχει συνεισφέρει σε πολλούς τομείς της σύγχρονης φυσικής όπως έρευνες για την ύπαρξη και τις ιδιότητες του μποζονίου Higgs, έρευνες για την ύπαρξη της σκοτεινής ύλης καθώς και μέτρηση των ιδιοτήτων των είδη γνωστών σωματιδίων με μεγάλη ακρίβεια.

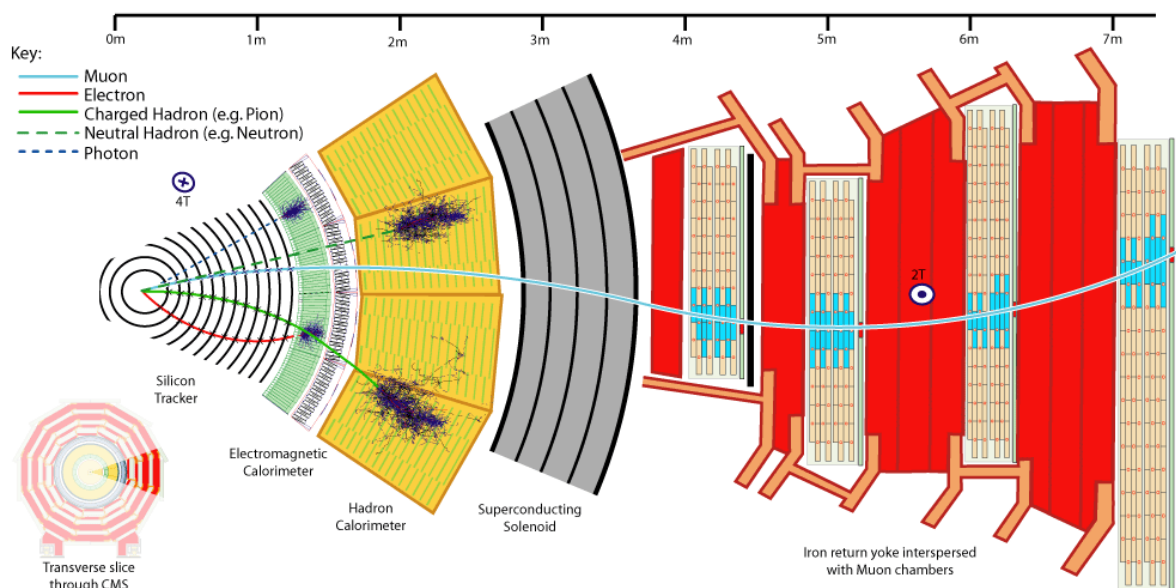


Σχήμα 2.26: Η βασική δομή του συστήματος ανιχνευτών και επιταχυντών του CERN. Πηγή: <http://upload.wikimedia.org/wikipedia/commons/thumb/5/57/Cern-accelerator-complex-fr.svg/220px-Cern-accelerator-complex-fr.svg.png>

Ο ανιχνευτής αποτελείται από διάφορα μέλη κατασκευασμένα γύρω από ένα υπέρ-μαγνήτη  $s$  σε σχήμα σωληνοειδούς. Το μαγνητικό πεδίο που παρέχει ο υπέρ-μαγνήτης είναι περίπου 4 tesla 100.000 φορές ισχυρότερο από το μαγνητικό πεδίο της Γης.

Τα υποσυστήματα του ανιχνευτή έχουν σχεδιαστεί ώστε να μετρούν την ενέργεια, την ορμή φωτονίων, ηλεκτρονίων, μιονίων και άλλων παραγόμενων από τις συγκρούσεις σωματιδίων. Το εσωτερικό τμήμα το οποίο περικλείει το σημείο της σύγκρουσης είναι ο ανιχνευτής τροχιών πυριτίου. Ο ανιχνευτής αυτός περικλείεται από κρυστάλλους σπινθηρισμών. Τον ανιχνευτή τροχιών περικλείει το ηλεκτρομαγνητικό καλορίμετρο το οποίο με την σειρά του περικλείεται από το αδρονικό καλορίμετρο.

Ο ανιχνευτής τροχιών καθώς και τα καλορίμετρα είναι τοποθετημένα στο εσωτερικό του σωληνοειδούς μαγνήτη. Εξωτερικά από τον μαγνήτη είναι τοποθετημένοι ανιχνευτές μιονίων. Στην εικόνα 2.27 παρουσιάζονται τα επιμέρους τμήματα του ανιχνευτή.



Σχήμα 2.27: Τα επιμέρους τμήματα του ανιχνευτή CMS. Στο εσωτερικό τμήμα είναι τοποθετημένος ο ανιχνευτής τροχιών πυριτίου. Εξωτερικά από τον ανιχνευτή τροχιών και πριν από τον υπέρ-μαγνήτη τοποθετούνται το ηλεκτρομαγνητικό και το αδρονικό καλορίμετρο. Εξωτερικά του υπέρ-μαγνήτη είναι τοποθετημένοι σπινθηριστές μιονίων. Πηγή: [http://upload.wikimedia.org/wikipedia/commons/4/48/CMS\\_Slice.png](http://upload.wikimedia.org/wikipedia/commons/4/48/CMS_Slice.png)

### 2.9.1 Ο ανιχνευτής τροχιών του πειράματος CMS πριν την αναβάθμιση

Ο ανιχνευτής τροχιών του CMS αποτελείται συνολικά από ανιχνευτές πυριτίου. Οι ανιχνευτές τοποθετούνται σε στρώματα σε ομόκεντρους κυλίνδρους οι οποίοι περικλείουν τη περιοχή της σύγκρουσης από της δέσμες του LHC. Ο σκοπός του ανιχνευτή είναι να παρέχει υψηλής ακρίβειας μετρήσεις στις τρεις διαστάσεις των σημείων κατά μήκος της καμπύλης που σχηματίζει η τροχιά φορτισμένων σωματιδίων. Η τροχιά των φορτισμένων σωματιδίων επιτρέπει την ανακατασκευή της θέσης της αρχικής αλληλεπίδρασης και των δευτερευόντων διασπάσεων.

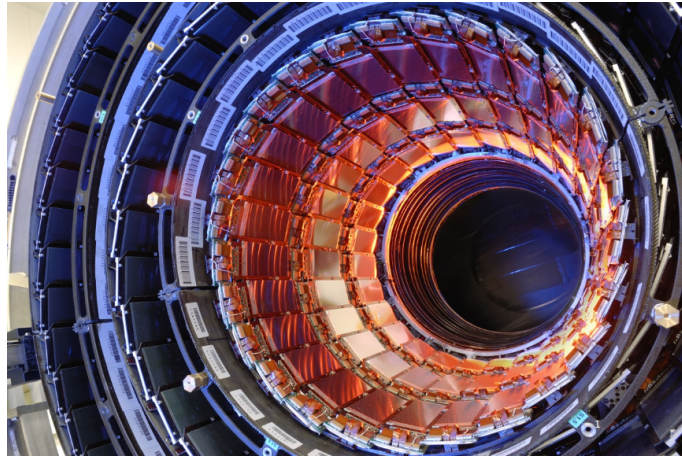
Στην εικόνα 2.31β' φαίνεται τα διάφορα τμήματα του ανιχνευτή τροχιών. Ο κάθετος άξονας παριστάνει τις ακτίνες των διάφορων κυλινδρικών τμημάτων και ο οριζόντιος την απόσταση από το κεντρικό σημείο της σύγκρουσης. Η αρίθμηση που βρίσκεται πάνω και αριστερά της εικόνας είναι η ψευδο-ταχύτητα ( $\eta$ ) η οποία εξαρτάται από τη γωνία που σχηματίζεται από το σημείο της σύγκρουσης.

Ο ανιχνευτής τροχιών αποτελείται από δύο υπό-ανιχνευτές με ανεξάρτητα συστήματα φύξης, τροφοδότησης και λήψης του σήματος. Ο εσωτερικός ανιχνευτής αποτελείται από ανιχνευτές πυριτίου μικρο-πλακιδίων (Pixel). Συνολικά ο ανιχνευτής αποτελείται από 66 εκατομμύρια pixel τύπου ( $n^+$ ) μεγέθους ( $100\mu\text{m} \times 150\mu\text{m}$ ) τα οποία είναι εμφυτευμένα σε όγκο πυριτίου ( $n$ ) ενώ η πίσω πλευρά είναι  $p$ -τύπου.

Καλύπτει μια ενεργή επιφάνεια ( $1,1\text{m}^2$ ). Αποτελείται από τρία ομόκεντρα κυλινδρικά στρώματα με ακτίνες 4,3, 7,2, 11 cm αντίστοιχα. Στις δύο ακριανές πλευρές τοποθετούνται οι δίσκοι (καπάκια) σε απόσταση 36,5 cm και 46,5 cm από το σημείο της σύγκρουσης.

Ο υπό-ανιχνευτής των μικρο-πλακιδίων περικλείεται από τον δεύτερο υπό-ανιχνευτή ο οποίος αποτελείται από ανιχνευτές πυριτίου λωρίδων (strip). Συνολικά αποτελείται





Σχήμα 2.28: Εσωτερική εικόνα από τον εξωτερικό ανιχνευτή τροχιών. Μέσα στην κοιλότητα του εξωτερικού ανιχνευτή τοποθετείται ο εσωτερικός ανιχνευτής τροχιών Pixel. Κάθε ανιχνευτική μονάδα τοποθετείται πολύ κοντά η μια στην άλλη σχηματίζοντας ένα συμπαγές στρώμα. Πηγή: [12]

από 9,6 εκατομμύρια λωρίδες  $p^+$ -τύπου εμφυτευμένες σε όγκο n-τύπου πυριτίου. Το κυλινδρικό κομμάτι του υπό-ανιχνευτή αποτελείται από δύο περιοχές την εσωτερική και την εξωτερική. Συνολικά αποτελείται από 10 ομόκεντρους κυλίνδρους, 4 στο εσωτερικό (TIB) και 6 στο εξωτερικό (TOB).

Επίσης στις άκρες υπάρχουν 12 δίσκοι, 3 στο εσωτερικό (TID) και 9 στο εξωτερικό (TOB).

Στο κυλινδρικό κομμάτι όπως και στο κομμάτι των δίσκων στα άκρα υπάρχουν τμήματα τα οποία αποτελούνται από ανιχνευτές λωρίδων οι οποίοι έχουν λωρίδες και στις δύο τις πλευρές, με τις λωρίδες και στις δύο πλευρές να σχηματίζουν γωνία μεταξύ τους ( $100 \text{ mrad}$ ). Οι ανιχνευτές αυτοί παρέχουν ακριβέστερες μετρήσεις στις 2 διαστάσεις.

Το πρώτο στάδιο λειτουργίας του LHC Run1 τελείωσε το Φεβρουάριο του 2013. Ο ανιχνευτής τροχιών του CMS λειτούργησε με μεγάλη επιτυχία για τρία χρόνια. Στο διάστημα αυτό λιγότερο από το 3% του ανιχνευτή κατέστη ανενεργό και λιγότερο από το 5% της συνολικής φωτεινότητας (luminosity) που δέχτηκε, χάθηκε λόγω του ανιχνευτή. Επίσης λειτούργησε με μεγάλη αποδοτικότητα παρόλο το μεγάλο (pile-up).

Πριν την επανεκκίνηση του ανιχνευτή το 2014 αναμένεται να υπάρξουν διάφορες βελτιώσεις. Μερικά κομμάτια του ανιχνευτή θα αντικατασταθούν έτσι ώστε να υπάρξουν λιγότερο από (1%) ανενεργές ανιχνευτικές μονάδες.

Προκειμένου να αντισταθμιστεί η αναμενόμενη συσσώρευση ζημίας λόγω ακτινοβολίας, η θερμοκρασία των συστημάτων ψύξης θα ελαττωθεί και στους δύο υπό-ανιχνευτές (pixel) και (strip).

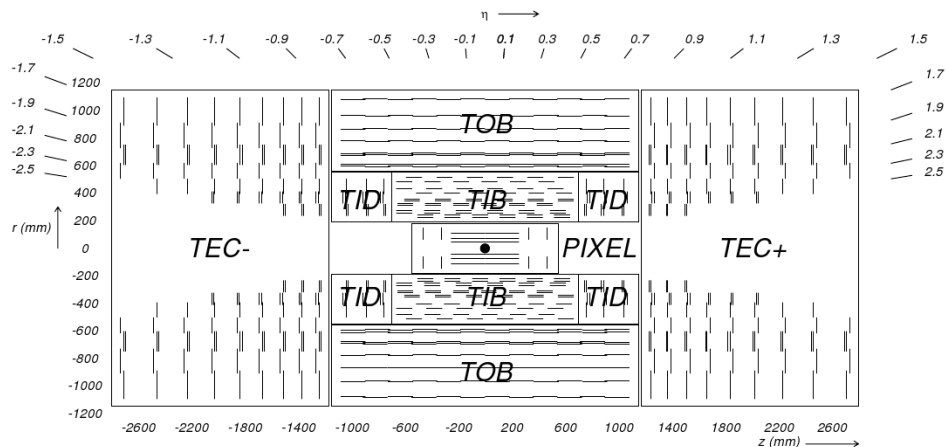
Στο δεύτερο γύρω λειτουργίας του LHC τόσο η ενέργεια της σύγκρουσης όσο και η στιγμιαία φωτεινότητα (instantaneous luminosity) θα διπλασιαστούν κάτι που οδηγεί σε μεγαλύτερο όγκο δεδομένων καθώς και μεγαλύτερη ζημία λόγω ακτινοβολίας.

### 2.9.2 Περιορισμοί του ανιχνευτή τροχιών

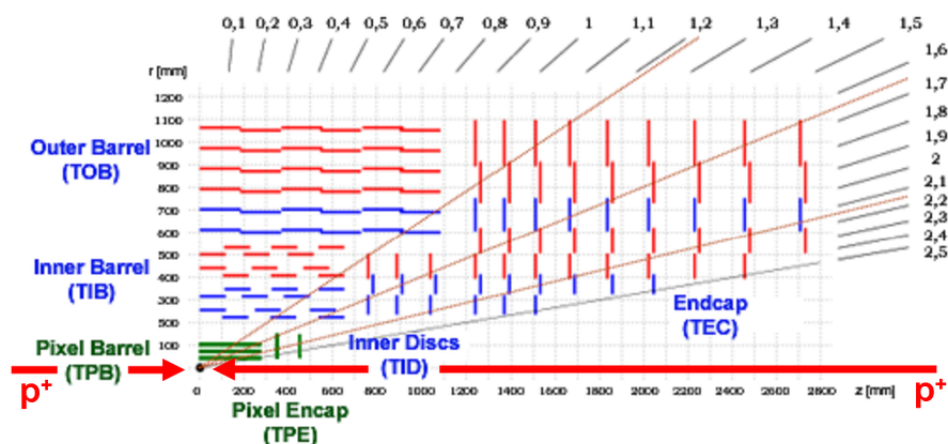
Ο ανιχνευτής τροχιών του CMS είχε σχεδιαστεί να λειτουργεί με μεγάλη ακρίβεια για φωτεινότητα  $500 \text{ fb}^{-1}$  και για λιγότερο από 50 συγκρούσεις ανα δέσμη.

Η συσσωρευμένη ζημία λόγω της ακτινοβολίας μειώνει την συλλογή φορτίου και τον





(α) Πηγή: [http://inspirehep.net/record/850264/files/fig\\_cmstracker.png](http://inspirehep.net/record/850264/files/fig_cmstracker.png)



(β) Πηγή: <http://www.hephy.at/typo3temp/pics/34a0a6f56b.png>

Σχήμα 2.29: Ανιχνευτής τροχιών του πειράματος CMS

Τα διάφορα τμήματα του ανιχνευτή τροχιών. Το εσωτερικό τμήμα που περικλείει την περιοχή της σύγκρουσης αποτελεί τον υπο-ανιχνευτή (pixel) (TPB). Τον (TPB) περικλύει ο υπο-ανιχνευτής λωρίδων με τα διάφορα τμηματά του, το εσωτερικό κυλινδρικό τμήμα (TIB), το εξωτερικό κυλινδρικό τμήμα (TOB) και τα τμήματα των δίσκων. Με μπλέ χρώμα επισημάνονται οι ανιχνευτές με δύο στρώματα λωρίδων

τρόπο με τον οποίο διαμοιράζεται το φορτίο μεταξύ των γειτονικών μικρο-λωρίδων και μικρο-πλακιδίων κάτι που περιορίζει την χωρική διακριτική ικανότητα.

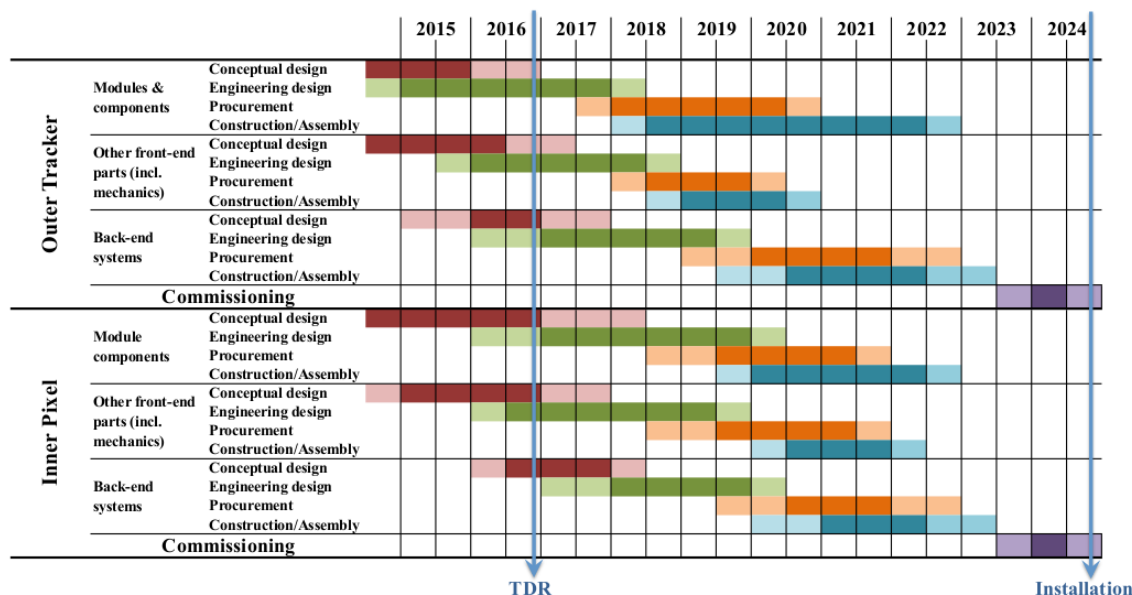
Στον εξωτερικό ανιχνευτή τροχιών η πιο σημαντική αλλαγή στα χαρακτηριστικά του ανιχνευτή λόγω ακτινοβολίας είναι η αύξηση του ρεύματος διαρροής το οποίο μπορεί να περιοριστεί σε ένα βαθμό μειώνοντας την θερμοκρασία ψύξης. Το ρεύμα διαρροής ελέγχεται από το σύστημα ελέγχου του ανιχνευτή για κάθε μια ανιχνευτική μονάδα. Ο τρόπος με τον οποίο μεταβάλλεται το ρεύμα διαρροής σε σχέση με την προβλεπόμενη φωτεινότητα, την θέση και το μέγεθος κάθε ανιχνευτικής μονάδας μπορεί να προβλεφθεί από ακριβέστατα συστήματα προσομοίωσης συμπεριλαμβάνοντας διάφορα σενάρια θερμοκρασίας. Με αυτό τον τρόπο μπορεί να προβλεφθεί η μείωση της απόδοσης του ανιχνευτή λόγω του ρεύματος διαρροής.

Οι ανιχνευτικές μονάδες που μπορεί να μην δώσουν σήμα κατά την διέλευση ενός σωματιδίου οδηγεί σε υποβαθμισμένη ικανότητα ανακατασκευής της τροχιάς. Για τιμές  $|n| < 2.4$  η ανίχνευση τροχιών στον ανιχνευτή strip για μόνιο με ορμή  $p_T = 10\text{GeV}$  μειώνεται από 100% σε 75 – 95% μετά από  $1000\text{fb}^{-1}$  στις ίδιες συνθήκες.

### 2.9.3 Αναβάθμιση του ανιχνευτή τροχιών του CMS

Η αναβάθμιση του L.H.C. θα πραγματοποιηθεί κατά τη διάρκεια τριών μεγάλων παύσεων λειτουργίας LS1 την περίοδο 2013-2014, LS2 την περίοδο 2017-2018 και LS3 την περίοδο 2022. Η αναβάθμιση του ανιχνευτή τροχιών θα γίνει σε δύο διαφορετικές φάσεις. Η φάση 1 θα έχει ολοκληρωθεί ως το τέλος του 2016 κατά την διάρκεια μιας εκτεταμένης παύσης λειτουργίας. Πριν την επαναλειτουργία του ανιχνευτή τόσο ο εξωτερικός ανιχνευτής όσο και ο αναβαθμισμένος ανιχνευτής στη φάση 1 θα πρέπει να αντικατασταθούν λόγω της αυξημένης ζημιάς λόγω ακτινοβολίας που θα δεχτούν.

Σύμφωνα με το χρονοδιάγραμμα του LHC η εγκατάσταση του αναβαθμισμένου στη φάση 2 ανιχνευτή τροχιών αναμένεται να γίνει στο τέλος του 2024. Στην εικόνα 2.30 παρουσιάζεται το χρονοδιάγραμμα των διεργασιών που θα γίνουν για την αναβάθμιση του ανιχνευτή τροχιών του CMS.



Σχήμα 2.30: Χρονοδιάγραμμα της αναβάθμισης του ανιχνευτή τροχιών στο C.M.S. μέχρι το 2024. Πηγή: [12]

### Προδιαγραφές της αναβάθμισης

Οι βασικές προδιαγραφές για την αναβάθμιση του ανιχνευτή τροχιών του CMS συνοψίζονται ως εξής :

- Αυξημένη αντοχή στην ακτινοβολία.

Ο αναβαθμισμένος ανιχνευτής τροχιών θα πρέπει να λειτουργεί αποτελεσματικά στις αυξημένες απαιτήσεις του HL-LHC. Αυτές οι απαιτήσεις θα πρέπει να εκπληρωθούν χωρίς παρεμβάσεις συντήρησης για τον εξωτερικό ανιχνευτή τροχιών, ενώ ο εσωτερικός ανιχνευτής Pixel προβλέπεται να έχει ευκολότερη πρόσβαση προσφέροντας την δυνατότητα εργασιών συντήρησης, καθώς θα μπορούν να αντικατασταθούν ανιχνευτικές μονάδες στο εσωτερικότερο τμήμα του ανιχνευτή όπου η ροή σωματιδίων ανα επιφάνεια θα είναι και μεγαλύτερη.

- Μειωμένη ποσότητα υλικού.

Η απόδοση του ανιχνευτή σχετίζεται άμεσα με την ποσότητα του υλικού. Από το υλικό του ανιχνευτή τροχιών επηρεάζεται επίσης και η απόδοση των εξωτερικών τμημάτων του ανιχνευτή όπως τα καλορίμετρα.

- Βελτιωμένο διαχωρισμό μεταξύ δύο τροχιών.

Ο ανιχνευτής τροχιών πριν την αναβάθμιση είχε υποβαθμισμένη απόδοση ως προς τον διαχωρισμό δύο διαφορετικών γεγονότων με πολύ κοντινή τροχιά.

- Αυξημένος βαθμός ανάλυσης.

Με σκοπό να εξασφαλιστεί επαρκής απόδοση στην ανίχνευση τροχιών σε καταστάσεις λειτουργίας υψηλής επιστοιβάσης γεγονότων (pile-up). Η απασχόληση κάθε καναλιού από την συλλογή του φορτίου μέχρι την τελική ψηφιοποίηση του

θα πρέπει να είναι αρκετά μικρή σε όλες τις περιοχές του ανιχνευτή. Αυτό απαιτεί υψηλή πυκνότητα καναλιών. Ως δείκτης αναφοράς έχει οριστεί μια μέση τιμή των 140 συγκρούσεων ανα διασταύρωση δέσμης.

- Ισχυρότερη διαδικασία αναγνώρισης προτύπων.

Ο αναβαθμισμένος ανιχνευτής τροχιών θα πρέπει να επιτρέπει γρήγορη και αποδοτική ανίχνευση τροχιών. Κάθε σήμα από διαφορετικές ανιχνευτικές μονάδες θα πρέπει να κατανέμεται αποδοτικά και γρήγορα στην τροχιά που ανήκει επιτρέποντας την καλύτερη ανακατασκευή της τροχιάς του σωματιδίου .

- Μονάδες με λειτουργία triggering

Η επιλογή των γεγονότων που είναι ενδιαφέροντα από πλευρά φυσικής γίνεται ιδιαίτερα απαιτητική σε υψηλή φωτεινότητα όχι μόνο λόγω της αυξημένης ροής αλλά και επειδή οι αλγόριθμοι που είναι υπεύθυνοι για την διαδικασία της επιλογής γίνονται λιγότερο αποδοτικοί σε καταστάσεις υψηλής επιστοίβασης (pile-up). Προκειμένου να διατηρηθεί η απόδοση του CMS καλύπτοντας το φάσμα των γεγονότων που έχουν ενδιαφέρον από πλευράς της φυσικής θα πρέπει να γίνεται κατάλληλη επιλογή των γεγονότων στον ανιχνευτή τροχιών.

#### 2.9.4 Αναβάθμιση του ανιχνευτή τροχιών CMS (Φάση I)

Σε αυτή τη φάση ο ανιχνευτής τροχιών Pixel θα αντικατασταθεί για πρώτη φορά. Αναμένονται ορισμένες τεχνολογικές βελτιώσεις όπως μία νέα ελαφρύτερη κατασκευή με λιγότερο υλικό , την αλλαγή του συστήματος ψύξης θα χρησιμοποιείται ένα σύστημα  $CO_2$  για ψύξη αντί του συστήματος  $C_6F_{14}$  που χρησιμοποιούταν στο προηγούμενο σύστημα και αλλαγές στα περιφερικά συστήματα των εξωτερικών ηλεκτρονικών ανάγνωσης.

Επίσης θα τοποθετηθούν 4 επίπεδα αντί για τρία στο κυλινδρικό κομμάτι και 3 αντί για 2 δίσκους στα καπάκια κάτι που επιτρέπει καλύτερη διακριτική ικανότητα για  $|n| < 2.5$ . Η εσωτερική ακτίνα στο κυλινδρικό κομμάτι θα μειωθεί έτσι ώστε το πιο εσωτερικό επίπεδο να βρεθεί πολύ κοντά στην σωλήνα όπου θα γίνονται οι συγκρούσεις.Ενώ θα αυξηθεί η απόσταση μεταξύ των επιπέδων.

Οι ανιχνευτικές μονάδες που θα χρησιμοποιηθούν σε αυτή τη φάση θα είναι ενός τύπου με αναλογία διαστάσεων pixel  $100 \times 150 \mu m^2$  με ενεργή περιοχή  $16, 2 \times 64.8 mm^2$  η οποία χρησιμοποιήθηκε και πριν την αναβάθμιση. Οι ανιχνευτές θα κατασκευαστούν με την ίδια  $n^+ - n$  τεχνολογία.Ο αριθμός των ανιχνευτικών μονάδων θα αυξηθεί από 768 σε 1184 στο κυλινδρικό κομμάτι ενώ στους δίσκους θα παραμείνει ο ίδιος αριθμός 672 αλλά με μεγαλύτερο μέγεθος. Αναμένεται να τοποθετηθούν συνολικά 79 εκατομμύρια pixel στους δίσκους και 45 εκατομμύρια στους δίσκους.Η αύξηση των pixel θα επιτρέψει στον ανιχνευτή να μπορεί να δουλέψει με την ίδια και καλύτερη απόδοση στις νέες αυξημένες απαιτήσεις.

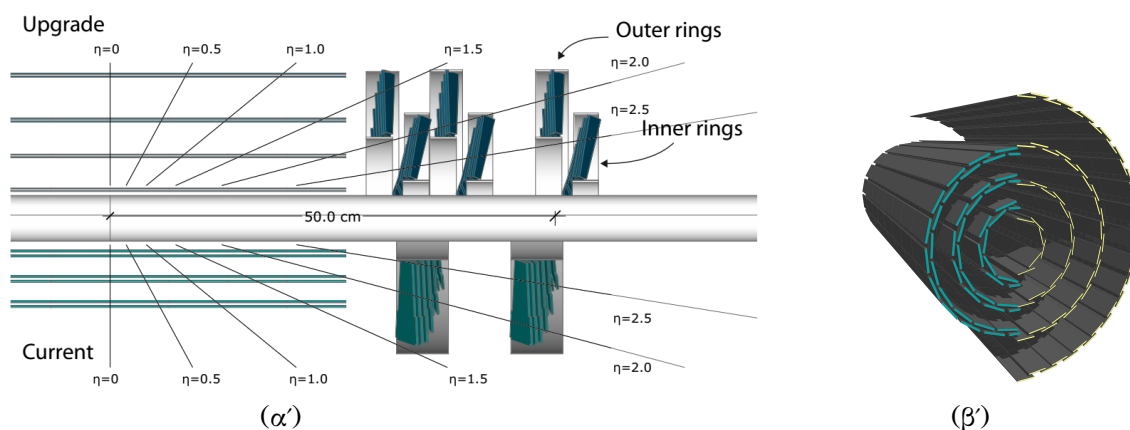
Τα εξωτερικά ηλεκτρικά ανάγνωσης θα είναι πιο αποδοτικά μειώνοντας σημαντικά τον όγκο δεδομένων που χάνεται λόγω των ηλεκτρονικών. Πιο εξελιγμένοι τύποι εξωτερικών ηλεκτρονικών έχουν σχεδιαστεί και κατασκευαστεί λαμβάνοντας υπόψιν την πιθανότητα να δημιουργηθούν ανιχνευτές με μικρότερη αναλογία Pixel.Η διαδικασία σχεδιασμού και κατασκευής των εξωτερικών ηλεκτρονικών στις δύο φάσεις 1,2 συμβαδίζουν σε αυτό το σημείο.

Έρευνες για την καλύτερη αντοχή των ανιχνευτών και των ηλεκτρονικών στην ακτινοβολία έχουν γίνει και χρησιμοποιούνται για το σχεδιασμό των αναβαθμισμένων ανιχνευτικών μονάδων.

Η επανεκκίνηση της λειτουργίας μετά την αναβάθμιση του ανιχνευτή στη φάση 1 αναμένεται μετά το 2017 και αναμένεται να λειτουργήσει για 3 χρόνια. Κατά την διάρκεια αυτή η απόδοση του ανιχνευτή θα μειωθεί εξαιτίας της ζημιάς που θα προκληθεί εξαιτίας της ακτινοβολίας. Αναμένεται κατά την διάρκεια λειτουργίας μετά την αναβάθμιση αντικατασταθούν κάποια τμήματα του ανιχνευτή ώστε να διατηρηθεί η απόδοση.

Οι βασικοί στόχοι της αναβάθμισης στη φάση 1 συνοψίζονται ως εξής:

- Ο ανιχνευτής θα πρέπει να λειτουργεί με ικανοποιητική απόδοση στην υψηλότερη φωτεινότητα μεγαλύτερη από  $500 \text{ fb}^{-1}$
- Βελτιωμένη κατασκευή έτσι ώστε να επιτυγχάνεται καλύτερη χωρική διακριτική ικανότητα (4-pixel-hit) για  $|n| < 2.5$ . Εικόνα 2.31
- Μείωση της ποσότητας του υλικού. Εικόνα 2.32
- Ύψηλότερη διακριτική ικανότητα. Μικρότερο ρυθμό ψευδών γεγονότων σε μεγαλύτερη επιστοίβαση σήματος (pile-up). Εικόνα 2.33
- Ελαχιστοποίηση της υποβάθμισης της απόδοσης του ανιχνευτή λόγω ζημιάς εξαιτίας της ακτινοβολίας.
- Ελαχιστοποίηση του όγκου δεδομένων που χάνονται με την κατασκευή νέων εξωτερικών ηλεκτρονικών ανάγνωσης (ROC).

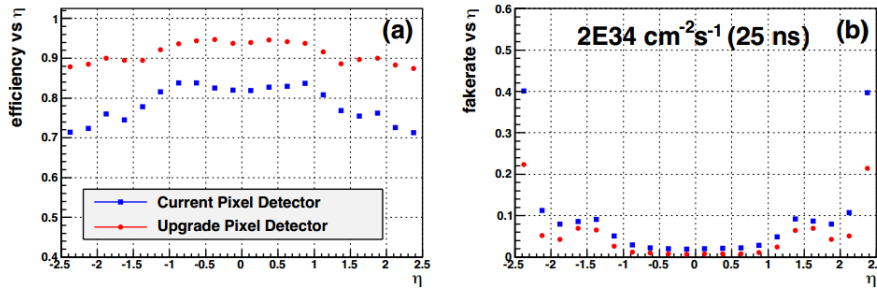


Σχήμα 2.31: Τα διάφορα στρώματα του ανιχνευτή τροχιών πριν και μετά την αναβάθμιση

Στην αναβάθμιση θα τοποθετηθούν 4 στρώματα αντί για τρία με μεγαλύτερο κενό ανάμεσα σε κάθε στρώμα. Στο κομμάτι των δίσκων θα τοποθετηθούν 3 δίσκοι για τον εσωτερικό ανιχνευτή τροχιών Pixel και αντίστοιχα 3 δίσκοι για των εξωτερικό ανιχνευτή τροχιών αντί για 2. Με αυτό τον τρόπο παρέχεται καλύτερη διακριτική ικανότητα για  $|n| > 2.5$ . Πηγή: [11]

Volume	Mass (g)	
	Present Detector	Phase 1 Detector
BPIX $ \eta  < 2.16$	16801	6686
FPIX $ \eta  < 2.50$	8582	7040

Σχήμα 2.32: Μάζα σε γραμμάρια του ανιχνευτή τροχιών πριν και μετά την αναβάθμιση στη φάση 1 Πηγή: [11]



Σχήμα 2.33: Απόδοση του ανιχνευτή και ρυθμός ψευδών γεγονότων πριν και μετά την αναβάθμιση όπως προέκυψαν από προσομοιώσεις Πηγή: [11]

### 2.9.5 Αναβάθμιση του ανιχνευτή τροχιών CMS (Φάση II)

Σκοπός της αναβάθμισης είναι να μπορεί ο ανιχνευτής τροχιών να ανταποκριθεί στις αυξημένες απαιτήσεις του HL-LHC όπου η φωτεινότητα αναμένεται να φτάσει  $3000 \text{ fb}^{-1}$ . Η διαδικασία έρευνας και κατασκευής για την φάση 2 της αναβάθμισης βασίζεται στο σχεδιασμό και την κατασκευή στην φάση 1. Μερικά στοιχεία της αναβάθμισης στη φάση 1 έχουν σχεδιαστεί ώστε να μπορέσουν να χρησιμοποιηθούν και στη φάση 2

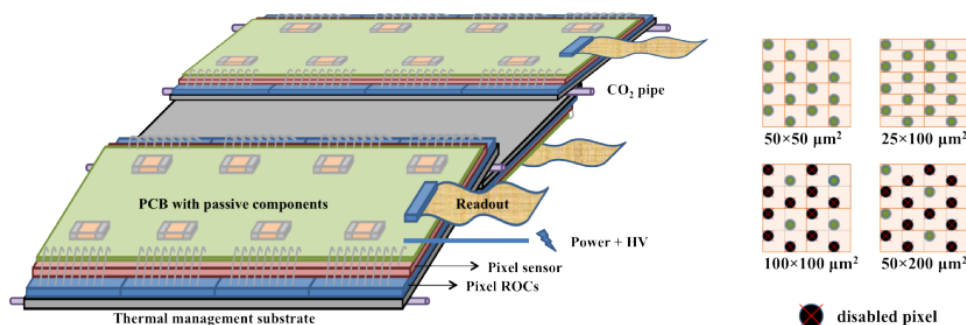
#### Εσωτερικός ανιχνευτής τροχιών Pixel

Ο εσωτερικός ανιχνευτής pixel θα πρέπει να είναι αρκετά ανθεκτικός στην ακτινοβολία προκειμένου να ανταποκρίνεται στις νέες αυξημένες απαιτήσεις του HL-LHC. Αυτό μπορεί να επιτευχθεί με ανιχνευτές πυριτίου επίπεδης γεωμετρίας με την μια επιφάνεια χωρισμένη σε πολύ μικρά μικρο-πλακίδια(pixel). Τα μεγέθη των pixel που μελετώνται είναι  $25 \times 100 \mu\text{m}^2$  και  $50 \times 50 \mu\text{m}^2$  τα οποία αντιστοιχούν σε μια μείωση της επιφάνειας των pixel κατά ένα παράγοντα 6 σε σχέση με τα pixel που χρησιμοποιήθηκαν ως τώρα. Η ίδια τεχνολογία  $n^+ - n$  θα χρησιμοποιηθεί και στην αναβάθμιση στη φάση 2.

Μια εναλλακτική επιλογή είναι να χρησιμοποιηθούν 3D-ανιχνευτές πυριτίου οι οποίοι είναι πολύ πιο ανθεκτικοί στην ακτινοβολία λόγω του μικρού χρόνου συλλογής φορτίου. Όμως αυτή η επιλογή θα περιοριστεί σε ένα μικρό τμήμα του ανιχνευτή όπου η ροή των σωματιδίων είναι μεγαλύτερη λόγω του υψηλού κόστους.

Η γενική ιδέα για την σχεδίαση και την δημιουργία του συστήματος ηλεκτρονικών για τον ανιχνευτή μικρο-πλακιδίων (pixel) περιγράφεται στην εικόνα 2.35. Η κάθε μονάδα αποτελείται από έναν ανιχνευτή πυριτίου pixel συνδεδεμένο με πολλαπλά Read-Out Chips (ROC). Η θερμοκρασία που δημιουργείται στο τσιπάκι του ανιχνευτή

απάγεται μέσω των δύο λεπτών σωλήνων  $CO_2$ .



Σχήμα 2.34: Αριστερά Βασική δομή συστήματος ηλεκτρονικών του ανιχνευτή τροχιών pixel Δεξιά: Πλέγμα συγκόλλησης εξωτερικών ηλεκτρονικών και ανιχνευτή για διαφορετικές αναλογίες pixel και μεγέθη Πηγή: [12]

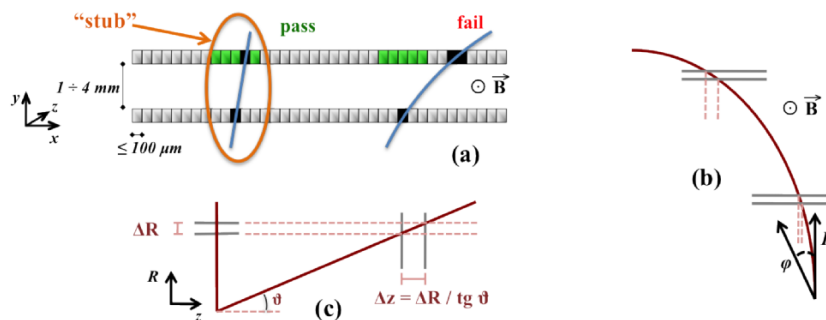
Ως στόχος έχει επιλεχθεί ανιχνευτές με επιφάνεια  $2500 \mu m^2$ , με δύο αναλογίες διαστάσεων pixel  $50 \times 50$ ,  $25 \times 100$ . Στα εξωτερικά τμήματα του εσωτερικού ανιχνευτή τροχιών Pixel ανιχνευτές με μεγαλύτερες αναλογίες διαστάσεων pixel μπορεί να επιφέρουν πλεονεκτήματα μειώνοντας την συνολική κατανάλωση ισχύος. Με την κατάλληλη συγκόλληση μεταξύ του ανιχνευτή και των εξωτερικών ηλεκτρονικών τα ίδια τσιπάκια εξωτερικών ηλεκτρονικών ανάγνωσης μπορούν να χρησιμοποιηθούν για τις δύο διαφορετικές αναλογίες διαστάσεων pixel και για πιθανών μεγαλύτερες συμβατές αναλογίες pixel. Αυτό μπορεί να γίνει απενεργοποιώντας τα κανάλια που δεν θα χρησιμοποιούνται όπως φαίνεται και στην δεξιά πλευρά της εικόνας 2.35. Με αυτό τον τρόπο θα μπορεί να χρησιμοποιηθεί το ίδιο (readout chip) για διαφορετικές αναλογίες διαστάσεων των pixel στις διαφορετικές περιοχές. Επίσης εφόσον οι αναλογίες των pixel θα καθοριστούν σε ένα μεταγενέστερο στάδιο τα readout chip θα σχεδιαστούν ώστε να μπορούν να λειτουργήσουν σε ένα ευρύ φάσμα τιμών χωρητικότητας και συλλογής φορτίου σύμφωνα με τις διατάξεις ανιχνευτών που προβλέπονται.

### Εξωτερικός ανιχνευτής τροχιών

Στο τμήμα αυτό του ανιχνευτή θα γίνεται κατάλληλη επιλογή των γεγονότων, κρατώντας μόνο αυτά που μας ενδιαφέρουν από πλευράς φυσικής. Αυτό μπορεί να συμβεί με ανιχνευτικές μονάδες οι οποίες μπορούν να απορρίψουν το σήμα από ένα σωματίδιο για το οποίο η εγκάρσια ορμή είναι μικρότερη από μια συγκεκριμένη τιμή. Σωματίδια με μικρή εγκάρσια ορμή κάμπτονται περισσότερο από το μαγνητικό πεδίο. Οπότε μπορούμε να απορρίψουμε γεγονότα από σωματίδια τα οποία κάμπτονται περισσότερο από το κατώφλι που έχει οριστεί. Η διαδικασία αυτή θα πρέπει να γίνεται στον ανιχνευτή έτσι ώστε ο συνολικός όγκος δεδομένων προς επεξεργασία να μην περιέχει ποσότητα πληροφορίας που δεν θα είναι χρήσιμη.

Η κάθε ανιχνευτική μονάδα αποτελείται από δύο ανιχνευτές πυριτίου σε πολύ κοντινή απόσταση συνδεδεμένα σε ένα κοινό σύστημα ηλεκτρονικών front-end ASIC (Application-specific integrated circuit). Συλλέγεται το σήμα και από τους δύο ανιχνευτές και επιλέγονται τα ζευγάρια που αντιστοιχούν σε σωματίδιο με εγκάρσια ορμή ( $p_T$ ) μεγαλύτερη από το κατώφλι που έχει οριστεί. Το ισχυρό μαγνητικό πεδίο του CMS επιτρέπει να υπάρχει επαρκής ευαισθησία στην μέτρηση της ορμής.



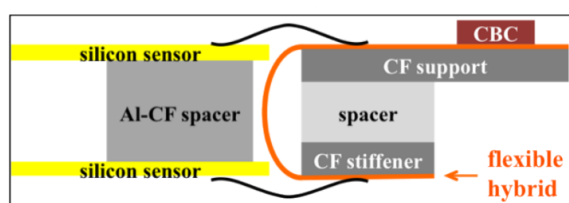


Σχήμα 2.35: Διαδικασία επιλογής γεγονότων. Σωματίδια τα οποία η εγκάρσια ορμή είναι μικρότερη από μία συγκεκριμένη τιμή δεν συλλέγονται έτσι ώστε ο συνολικός όγκος δεδομένων να μην περιέχει πληροφορία που δεν είναι χρήσιμη. Πηγή: [12]

Δύο τύποι ανιχνευτικών μονάδων είναι υπό μελέτη

- Ανιχνευτικές μονάδες 2S οι οποίες αποτελούνται από δύο ανιχνευτές πυριτίου τύπου strip διαστάσεων περίπου  $10 \times 10 \text{ cm}$  με τα strips να είναι παράλληλα μεταξύ τους. Οι μονάδες αυτές τοποθετούνται στις εξωτερικές περιοχές σε ακτίνα περίπου  $R \approx 60 \text{ cm}$  καλύπτοντας μια επιφάνεια  $150 \text{ m}^2$ . Για να είναι δυνατόν ο ανιχνευτής τροχιών να απορρίπτει γεγονότα από σωματίδια με μικρή εγκάρσια ορμή κάθε ανιχνευτική μονάδα θα έχει διαφορετική απόσταση μεταξύ των λωρίδων καλύπτοντας ένα φάσμα από 1,8 έως 4,0 mm.

Στην εικόνα 2.36 φαίνεται σχηματικά η διάταξη ανιχνευτών τύπου 2S. Οι δύο ανιχνευτές με διατάξεις strip συνδέονται μεταξύ τους μέσω ενός υλικού το οποίο αποτελείται από νανοσωλήνες άνθρακα AL-CF (carbon fibre aluminum reinforced) το οποίο λειτουργεί ως στήριξη, διαχωριστής και απαγωγός θερμότητας. Τα ηλεκτρονικά για την παροχή τάσης και την λήψη του σήματος τοποθετούνται εξωτερικά του ανιχνευτή και συνδέονται με τις δύο πλευρές του ανιχνευτή. Αντίστοιχα και στα εξωτερικά ηλεκτρονικά τοποθετείται AL-CF έτσι ώστε να επιτυγχάνεται απομόνωση των ανιχνευτών.



Σχήμα 2.36: Πηγή:[12]

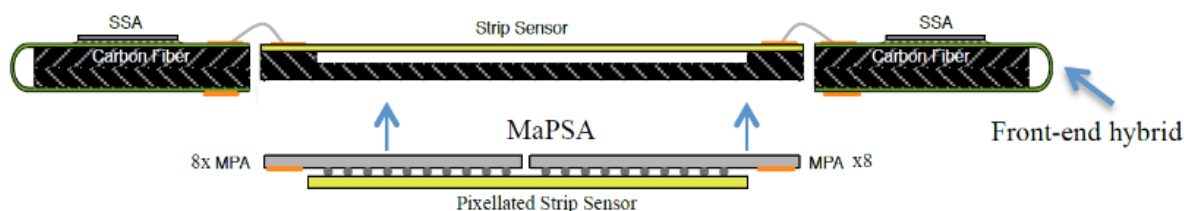
- Ανιχνευτικές μονάδες PS οι οποίες αποτελούνται από δύο ανιχνευτές συνολικής επιφάνειας  $5 \times 10 \text{ cm}^2$  σε πολύ κοντινή απόσταση ο ένας με τον άλλον. Ο ένας ανιχνευτής είναι τύπου strip και ο άλλος macro-pixel με αναλογία διαστάσεων pixel  $100 \mu\text{m} \times 1.5 \text{ mm}$ . Οι μονάδες PS τοποθετούνται σε ακτίνα μεταξύ  $R \approx 20 \text{ cm}$  και  $R \approx 60 \text{ cm}$

Οι ανιχνευτές με διάταξη macro-pixel παρέχουν επαρκής μετρήσεις στην κάθετη  $z$  συντεταγμένη για την ακριβή αποτύπωση της τροχιάς και επιτρέπουν



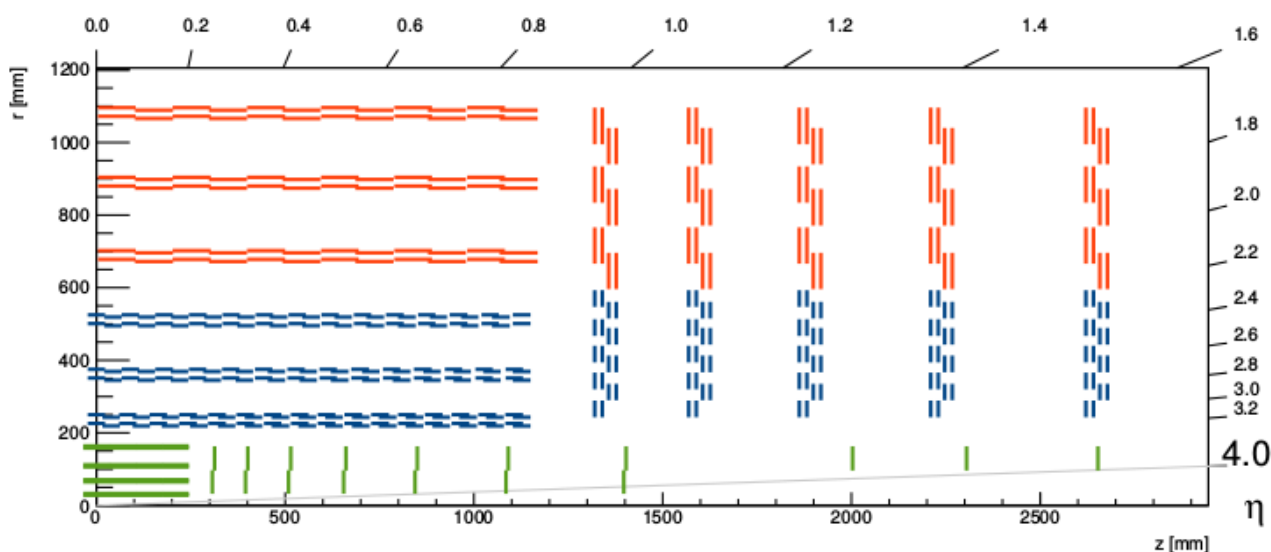
την διάκριση πρωτογενών κορυφών διάσπασης σωματιδίων ενισχύοντας την αναγνώριση προτύπων με πιο αποτελεσματικό τρόπο από μια πιθανή επέκταση του εσωτερικού ανιχνευτή Pixel σε μεγαλύτερες ακτίνες.

Στην εικόνα 2.37 φαίνεται σχηματικά η διάταξη ανιχνευτών τύπου PS. Ο ανιχνευτής με διάταξη macro-pixel συνδέεται μέσω bump-bondings με τις πλακέτες των εξωτερικών ηλεκτρονικών ανάγνωσης MPA με αποτέλεσμα να μπορεί να χρησιμοποιηθεί σαν ενιαία μονάδα. Ο ανιχνευτής strip έπειτα συνδέεται με τα MPAs με την βοήθεια AL-CF. Τα ηλεκτρονικά για την παροχή τάσης και την λήψη του σήματος τοποθετούνται ξεχωριστά εκατέρωθεν του ανιχνευτή και συνδέονται και με τις δύο πλευρές..



Σχήμα 2.37: Πηγή:[12]

Στην εικόνα 2.38 παρουσιάζεται το ένα τέταρτο της προβλεπόμενης αναβάθμισης του ανιχνευτή τροχιών του CMS. Αναλυτικότερη περιγραφή της αναβάθμισης μπορεί να βρεθεί στη βιβλιογραφία [12], [14].



Σχήμα 2.38: Ο εξωτερικός ανιχνευτής τροχιών αποτελείται από PS ανιχνευτικές μονάδες μπλε γραμμές και από SP ανιχνευτικές μονάδες κόκκινες γραμμές. Τα επίπεδα του εσωτερικού ανιχνευτή Pixel σημειώνεται με πράσινη ευθεία γραμμή Πηγή:[12]



## Κεφάλαιο 3

# Γρήγορος Μετασχηματισμός Fourier (FFT)

### 3.1 Εισαγωγή

Ο γρήγορος μετασχηματισμός Fourier είναι ένα απαραίτητο κομμάτι της παρούσας διπλωματικής εργασίας και του προγράμματος που υλοποιήθηκε για τον υπολογισμό των χωρητικοτήτων το οποία θα εξετάσουμε στα επόμενα κεφάλαια. Κρίνεται λοιπόν απαραίτητο να εξετάσουμε πως δουλεύει ο συγκεκριμένος αλγόριθμος.

Ο γρήγορος μετασχηματισμός Fourier είναι ένας αλγόριθμος ο οποίος μας επιτρέπει να υπολογίσουμε τον διακριτό μετασχηματισμό Fourier (DFT) μειώνοντας δραστηκά τον αριθμό και τον χρόνο των υπολογισμών που απαιτούνται. Είναι ένας από τους πιο χρήσιμους αλγορίθμους της σύγχρονης επιστήμης καθώς αποτελεί αναπόσπαστο κομμάτι σε πολλούς κλάδους της φυσικής, των μαθηματικών και της τεχνολογίας.

Υπάρχουν πολλοί διαφορετικοί αλγόριθμοι γρήγορου μετασχηματισμού Fourier. Μερικοί από αυτούς είναι οι : Prime-factor FFT algorithm, Winograd algorithm, Rader's algorithm, με τον πιο γνωστό να είναι ο αλγόριθμος των Cooley–Tukey FFT algorithm. Επίσης έχουν δημιουργηθεί πολλές διαφορετικές βιβλιοθήκες σε διάφορες γλώσσες προγραμματισμού οι οποίες υλοποιούν έναν ή και περισσότερους απο τους παραπάνω αλγορίθμους όπως η βιβλιοθήκη FFTW που είναι γραμμένη σε C και χρησιμοποιείται από διάφορα πακέτα μαθηματικών όπως το Matlab ή την JTransforms που είναι γραμμένη σε Java και την οποία θα χρησιμοποιήσουμε και στην παρούσα διπλωματική. Ο αλγόριθμος του FFT αποτελεί ένα ανοιχτό ακόμα πεδίο το οποίο μελετάται από αρκετούς επιστήμονες των μαθηματικών και της επιστήμης υπολογιστών καθώς δεν έχει βρεθεί ο βέλτιστος αλγόριθμος. Περισσότερες πληροφορίες μπορούν να βρεθούν στην βιβλιογραφία [26], [28], [25], [27].

### 3.2 Μετασχηματισμός Fourier

Ο μετασχηματισμός Fourier είναι ένα από τα πιο χρήσιμα εργαλεία στην σύγχρονη φυσική. Για μερικά προβλήματα ο μετασχηματισμός Fourier είναι απλά ένα υπολογιστικό εργαλείο για την επεξεργασία δεδομένων ενώ σε άλλες περιπτώσεις ο μετασχηματισμός Fourier είναι το κεντρικό θέμα του προβλήματος. Μια φυσική διαδικασία μπορεί να περιγραφεί είτε στο πεδίο του χρόνου είτε στο πεδίο των συχνοτήτων. Μπορούμε να πάμε από το ένα πεδίο στο άλλο μέσω τον μετασχηματισμών :

$$H(f) = \int_{-\infty}^{+\infty} h(t)e^{2\pi i ft} dt \quad (3.1)$$

$$H(t) = \int_{-\infty}^{+\infty} h(f)e^{-2\pi i ft} df \quad (3.2)$$

Αν το  $t$  μετριέται σε δευτερόλεπτα τότε το  $f$  μετριέται σε κύκλους ανά δευτερόλεπτο. Ενώ αν το  $h$  είναι συνάρτηση της θέσης  $x$  τότε το  $H$  μετριέται σε κύκλους ανά μέτρο (δηλαδή το ανάστροφο του μήκους κύματος) Ο μετασχηματισμός Fourier είναι γραμμικός μετασχηματισμός δηλαδή ο μετασχηματισμός του αθροίσματος δυο συναρτήσεων θα είναι ίσος με το άθροισμα των μετασχηματισμών των συναρτήσεων. Η συνάρτηση  $h(t)$  μπορεί να έχει μια ή περισσότερες συμμετρίες. Στο πεδίο των συχνοτήτων οι συμμετρίες αυτές ισοδυναμούν με σχέσεις μεταξύ  $H(f)$  και  $H(-f)$ . Οι συμμετρίες αυτές μπορούν να χρησιμοποιηθούν ώστε να αυξήσουν την υπολογιστική ικανότητα.

### 3.3 Βασικές ιδιότητες Μετασχηματισμού Fourier

Η συνάρτηση  $h(t)$  μπορεί να έχει μια ή και περισσότερες συμμετρίες. Δηλαδή μπορεί να είναι καθαρά πραγματική συνάρτηση ή καθαρά φανταστική ή να είναι άρτια και περιττή. Από τις συμμετρίες αυτές προκύπτουν διάφορες σχέσεις μεταξύ των τιμών  $H(f)$  και  $H(-f)$ .

Μερικές βασικές ιδιότητες του μετασχηματισμού Fourier είναι:

1. *Άρτιες συναρτήσεις έχουν μόνο πραγματικό μέρος* Εάν μια συνάρτηση είναι πραγματική και άρτια δηλαδή ισχύει  $h(t) = h(-t)$  τότε ο μετασχηματισμός που θα προκύψει θα είναι μια πραγματική και άρτια συνάρτηση της συχνότητας. Αφού το πραγματικό μέρος θα είναι μηδέν.
2. *Περιττές συναρτήσεις έχουν μόνο φανταστικό μέρος* Εάν μια συνάρτηση είναι πραγματική και περιττή δηλαδή ισχύει  $h(t) = -h(-t)$ . τότε ο μετασχηματισμός που θα προκύψει είναι μια φανταστική και περιττή συνάρτηση της συχνότητας.
3. *Τυχαίες συναρτήσεις μπορούν να γραφούν ως άθροισμα άρτιων και περιττών συναρτήσεων.* Κάθε συνάρτηση μπορεί να γραφεί ως άθροισμα μιας περιττής και μιας άρτιας συνάρτησης.
4. *Γραμμικότητα* Ο μετασχηματισμός του αθροίσματος συναρτήσεων ισούται με το άθροισμα τον μετασχηματισμών των συναρτήσεων. Ο μετασχηματισμός μιας συνάρτησης επί μία σταθερά ισούται με την σταθερά επί το μετασχηματισμό της συνάρτησης.

### 3.4 Μετασχηματισμός Fourier διακριτής συνάρτησης

Οι κυματοσυναρτήσεις που συναντάμε στη φύση είναι συνήθως αναλογικές δηλαδή συνεχείς με το χρόνο. Παρόλα αυτά στις πιο πολλές περιπτώσεις τα δεδομένα ενός προβλήματος έχουν πεπερασμένο εύρος και είναι διακριτά χωρισμένα συνήθως σε ίσου μήκους διαστήματα. Όπως για παράδειγμα δεδομένα που προκύπτουν από μια πειραματική ή υπολογιστική διαδικασία. Η διαδικασία μετατροπής ενός σήματος σε

διακριτό ή του χωρισμού μιας συνάρτησης σε διακριτά σημεία ονομάζεται δειγματοληψία. Έστω μια συνάρτηση  $h(t)$  στην οποία θέλουμε να γίνει δειγματοληψία. Η δειγματοληψία του σήματος θα γίνει παίρνοντας δείγματα της συνάρτησης ανά διακριτές τιμές για κάθε  $t$  με διάστημα μεταξύ δύο διαδοχικών τιμών  $\Delta$  προκύπτει έτσι η ακολουθία τιμών

$$h_n = h(n\Delta) \quad \text{για} \quad n = \dots -3, -2, -1, 0, 1, 2, 3\dots \quad (3.3)$$

Όπου το  $\Delta$  καλείται ρυθμός δειγματοληψίας ή συχνότητα δειγματοληψίας Η συχνότητα δειγματοληψίας μας δείχνει πόσα δείγματα έχουν ληφθεί σε διάρκεια ενός δευτερολέπτου και προκύπτει από την σχέση  $f = 1/\Delta$ .

### Θεώρημα Δειγματοληψίας

Το θεώρημα δειγματοληψίας εγγυάται ότι σήματα περιορισμένου φάσματος (π.χ. σήματα τα οποία στο πεδίο συχνοτήτων έχουν τιμές μέχρι μια μέγιστη συχνότητα) μπορούν να ανακατασκευαστούν πλήρως από την δειγματοληπτημένη μορφή τους εάν η συχνότητα δειγματοληψίας είναι μεγαλύτερη ή ίση από το διπλάσιο της μέγιστης συχνότητάς τους. Για κάθε ρυθμό δειγματοληψίας  $\Delta$  υπάρχει μια κρίσιμη συχνότητα  $f_c$  η οποία καλείται κρίσιμη συχνότητα Nyquist και δίνεται από την σχέση

$$f_c = \frac{1}{2\Delta} \quad (3.4)$$

Αν ένα ημιτονοειδές σήμα έχει συχνότητα ίση με τη συχνότητα Nyquist και έχει το πρώτο δείγμα στο θετικό μέγιστο τότε το επόμενο δείγμα θα είναι στο αρνητικό μέγιστο. Με άλλα λόγια η κρίσιμη δειγματοληψία ενός ημιτονοειδούς κύματος είναι δύο σημεία ανά κύκλο. Η συχνότητα Nyquist είναι σημαντική για δύο λόγους. Αν μια συνεχής συνάρτηση  $h(t)$  έχει ρυθμό δειγματοληψίας  $\Delta$ , και έχει εύρος ζώνης συχνοτήτων μικρότερο της κρίσιμης συχνότητας  $f_c$ , δηλαδή για κάθε  $|f| \geq f_c$  ισχύει  $H(f) = 0$ , τότε η  $h(t)$  καθορίζεται πλήρως από τα δείγματα της  $h_n$ . Ισχύει δηλαδή η σχέση

$$h(t) = \Delta \sum_{n=-\infty}^{+\infty} h_n \frac{\sin[2\pi f_c(t - n\Delta)]}{\pi(t - n\Delta)} \quad (3.5)$$

Σύμφωνα με το θεώρημα δειγματοληψίας η συνολική πληροφορία μιας συνεχούς συνάρτησης ή σήματος μπορεί να καταγραφεί με ρυθμό δειγματοληψίας  $\Delta$  ίσο με το διπλάσιο της κρίσιμης συχνότητας. Αν μια συνάρτηση όμως δεν έχει εύρος συχνοτήτων μικρότερο της κρίσιμης τότε σε αυτή την περίπτωση αποδεικνύεται ότι η συνολική πυκνότητα φασματικής ισχύος η οποία βρίσκεται εκτός του ορίου  $-f_c \leq f \leq +f_c$  προστίθεται μέσα στο όριο.

### 3.5 Διακριτός μετασχηματισμός Fourier (DFT)

Έστω ότι έχουμε  $N$  διακριτά σημεία μιας συνάρτησης μετά από δειγματοληψία.

$$h_k = h(t_k) \quad \text{όπου} \quad t_k = k\Delta \quad \text{με} \quad k = 0, 1, 2, \dots, N-1 \quad (3.6)$$

Όμοια η συχνότητα χωρίζεται σε διαστήματα που δίνεται από τη σχέση  $f = \frac{1}{N\Delta}$ .

$$f_n = \frac{n}{N\Delta}, \quad \text{όπου} \quad n = -\frac{N}{2} \dots \frac{N}{2} \quad (3.7)$$

Οι τιμές  $-\frac{N}{2}$ ,  $\frac{N}{2}$  αντιστοιχούν στην συχνότητα Nyquist. Έτσι αντί να υπολογίζουμε τον μετασχηματισμό Fourier σε όλες τις τιμές μέσα στο διάστημα από  $-f_c$  μέχρι  $f_c$  υπολογίζουμε μόνο τις διακριτές τιμές. Προκύπτει ότι ο διακριτός μετασχηματισμός Fourier δίνεται από τη παρακάτω σχέση 3.8:

$$h_k = \sum_{k=0}^{N-1} h_k e^{2\pi i k n / N} \quad (3.8)$$

Ενώ αντίστοιχα ο ανάστροφος μετασχηματισμός Fourier από τη σχέση 3.9:

$$H_n = \frac{1}{N} \sum_{k=0}^{N-1} h_k e^{-2\pi i k n / N} \quad (3.9)$$

Όπου και στις δυο εκφράσεις έχουμε:

- N αριθμός δειγμάτων
- Δt Το διάστημα μεταξύ δύο δειγμάτων στο πεδίο του χρόνου
- Δf Το διάστημα μεταξύ δύο δειγμάτων στο πεδίο των συχνοτήτων
- $1/N$  είναι παράγοντας κανονικοποίησης, μπορούμε να το βάλουμε είτε στον ευθύ είτε στον ανάστροφο αλλά όχι και στα δύο μαζί.

Οι παραπάνω εκφράσεις μπορούν να γραφούν από την ταυτότητα του Euler ( $e^{\pm\Theta} = \cos(\Theta) \pm \sin(\Theta)$ ) στην παρακάτω μορφή

$$h_d(k) = \sum_{n=0}^{N-1} h(n) \cos\left(\frac{2\pi k n}{N}\right) - i h(n) \sin\left(\frac{2\pi k n}{N}\right) \quad (3.10)$$

$$h(n) = \frac{1}{N} \sum_{k=0}^{N-1} h_d(k) \cos\left(\frac{2\pi k n}{N}\right) + i h(n) \sin\left(\frac{2\pi k n}{N}\right) \quad (3.11)$$

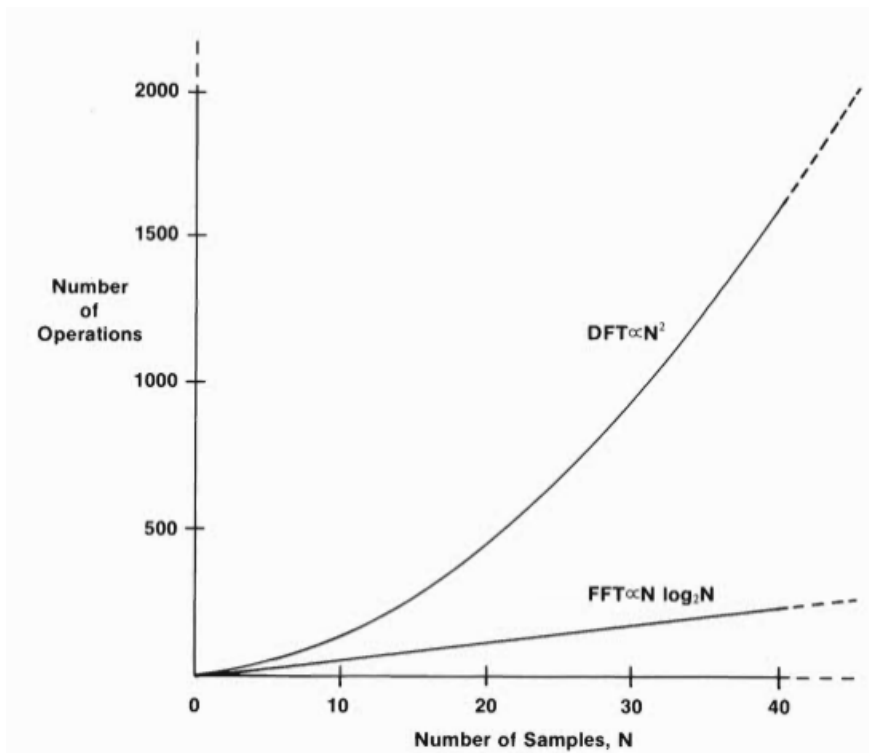
Με την παραπάνω έκφραση μπορούμε με μια απλή σειρά από υπολογισμούς να υπολογίσουμε τον διακριτό μετασχηματισμό Fourier οποιασδήποτε συνάρτησης. Σύμφωνα με την παραπάνω έκφραση κάθε συντελεστής  $h_d(k)$  θα προκύψει αθροίζοντας τα γινόμενα  $h(n) \sin\left(\frac{2\pi k n}{N}\right) - i \cos\left(\frac{2\pi k n}{N}\right)$  για κάθε  $n$  και για κάθε  $k$ . Δηλαδή για  $k = 0$  αθροίζουμε  $h(n) \sin\left(\frac{2\pi k n}{N}\right) - i \cos\left(\frac{2\pi k n}{N}\right)$  για  $n = 0$  μέχρι  $n = N - 1$ . Ομοίως για  $k=1$  μέχρι  $k = N - 1$

Προκύπτει ότι για 32 σημεία πρέπει να αθροιστούν 32 όροι για κάθε ένα από τις 32 τιμές του  $k$ . Δηλαδή  $32 \times 32 = 1024$  υπολογισμούς. Παρατηρούμε ότι για  $N$  σημεία προκύπτουν  $N^2$  υπολογισμοί. Αν σκεφτούμε ότι συνήθως ο αριθμός των σημείων  $N$  που χρειαζόμαστε για να καθορίσουμε μια τυχαία συνάρτηση είναι εκατοντάδες ή και παραπάνω καταλαβαίνουμε γιατί ο διακριτός μετασχηματισμός Fourier αποφευγόταν πριν την ανάπτυξη του σύγχρονου ηλεκτρονικού υπολογιστή.

### 3.6 Γρήγορος μετασχηματισμός Fourier (FFT)

Ο γρήγορος μετασχηματισμός Fourier έγινε ευρύτερα γνωστός από τον αλγόριθμο των Cooley και Tukey στα μέσα του 1960. Παρόλα αυτά αποδοτικές μέθοδοι είχαν ανακαλυφθεί από πολλά διαφορετικά άτομα, ξεκινώντας από τον Gauss το 1805. Το 1903 C. Runge περιέγραψε μια τεχνική που αργότερα θα γίνει γνωστή ως ο γρήγορος

μετασχηματισμός Fourier (FFT). Αργότερα το 1942 μια πιο γενική μέθοδος αναπτύχθηκε από τους Danielson και Lanczos. Αναγνωρίζοντας διάφορες συμμετρίες κατάφεραν να μειώσουν τον αριθμό των υπολογισμών στην περίπτωση όπου έχουμε πλήθος σημείων που είναι δύναμη του 2 (όπου ισχύει δηλαδή η σχέση  $N = 2^k$ ). Ο αριθμός των υπολογισμών μειώθηκε από  $N^2$  σε  $N \log_2 N$ . Η δραστηκή μείωση των υπολογισμών περιγράφεται στο σχήμα 3.1.



Σχήμα 3.1: Σύγκριση FFT, DFT. Αριθμός υπολογισμών Πηγή: [28]

$N$	1000	$10^6$	$10^9$
$N^2$	$10^6$	$10^{12}$	$10^{18}$
$N \log_2 N$	$10^4$	$20 \times 10^6$	$30 \times 10^9$

Πίνακας 3.1: Αριθμός υπολογισμών για διάφορα πλήθη τιμών

Παρατηρούμε ότι για μεγάλο  $N$  οι υπολογισμοί μειώνονται δραστηκά. Όπως φαίνεται και στον παραπάνω πίνακα. Αν υποθέσουμε ότι κάθε υπολογισμός γίνεται σε χρόνο 1 ns τότε για  $N = 10^9$  προκύπτει ότι θα χρειάζονταν  $10^{18}$  ns  $\approx 31,2$  χρόνια για  $N^2$  υπολογισμούς ενώ για  $N \log_2 N$  θα χρειάζονταν 30 sec.

Ορίζουμε ένα μιγαδικό αριθμό  $W$  τέτοιο ώστε

$$W = e^{\frac{2\pi i}{N}} \quad (3.12)$$

Τότε η εξίσωση 3.8 μπορεί να γραφεί στην παρακάτω πιο συμπαγή μορφή

$$H_n = \sum_{k=0}^{N-1} h_k W_N^{nk} \quad (3.13)$$

Παρατηρούμε ότι το διάνυσμα  $h_k$  πολλαπλασιάζεται με έναν πίνακα όπου το στοιχείο στην θέση  $(n,k)$  είναι μια σταθερά υψωμένη στην δύναμη  $n \times k$ . Ο γρήγορος μετασχηματισμός Fourier εκμεταλλεύεται τις συμμετρίες του παράγοντα  $W$ .

1. Περιοδικότητα ως προς  $n$ ,  $k$  με περίοδο  $N$

$$W_N^{nk} = W_N^{(n+N)k} = W_N^{n(k+N)}$$

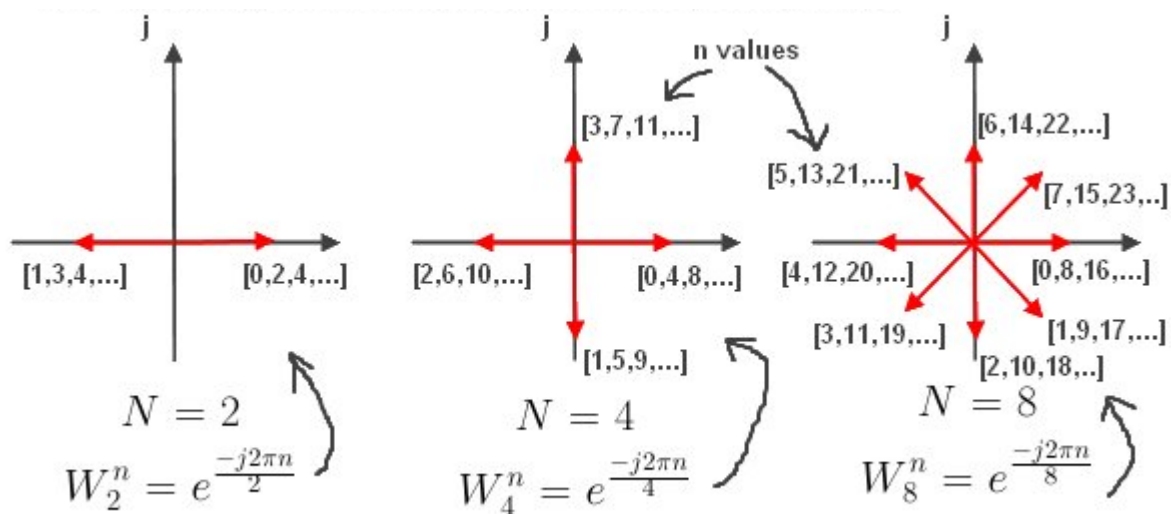
Αφού:

$$W_N^{(n+N)k} = e^{2\pi(n+N)k/N} = e^{2\pi nk/N} e^{2\pi kN/N} = 1e^{2\pi nk/N}$$

2. Συμμετρία μιγαδικού συζυγούς

$$W_N^{(N-n)k} = W_N^{-nk} = (W_N^{nk})^* \quad (3.14)$$

Ο παράγοντας  $W$  περιγράφει ένα διάνυσμα στροφής το οποίο περιστρέφεται ανάλογα με το  $N$ . Στην εικόνα 3.2 παρατηρούμε πώς μεταβάλλεται το  $W$  για  $N=2,4,8$ . Οι τιμές του  $W$  επαναλαμβάνονται ξανά και ξανά για διαφορετικές τιμές του  $n$ . Επίσης οι τιμές που έχουν διαφορά φάσης  $180^\circ$  είναι αντίθετες μεταξύ τους.



Σχήμα 3.2: Διάνυσμα στροφής.  $\Pi$  : <http://www.alwayslearn.com/dft>

### 3.6.1 Λήμμα Danielson-Lanczos

Οι Danielson και Lanczos απέδειξαν ότι ένας διακριτός μετασχηματισμός Fourier με πλήθος στοιχείων  $N$  μπορεί να γραφτεί ως το άθροισμα δύο διακριτών μετασχηματισμών Fourier μεγέθους  $N/2$ . Το ένα άθροισμα σχηματίζεται από το περιττά σημεία (περιττά αριθμημένα) του πλήθους  $N$  ενώ το άλλο από τα άρτια σημεία (άρτια αριθμημένα).



$$\begin{aligned}
F_k &= \sum_{j=0}^{N-1} e^{2\pi i j k / N} f_j \\
&= \sum_{j=0}^{N/2-1} e^{2\pi i (2j) k / N} f_{2j} + \sum_{j=0}^{N/2-1} e^{2\pi i (2j+1) k / N} f_{2j+1} \\
&= \sum_{j=0}^{N/2-1} e^{2\pi i j k / (N/2)} f_{2j} + W^k \sum_{j=0}^{N/2-1} e^{2\pi i j k / (N/2)} f_{2j+1} \\
&= F_k^e + W^k F_k^o
\end{aligned} \tag{3.15}$$

Όπου  $W = e^{2\pi i / N}$  και  $F_k^e$  ο μετασχηματισμός Fourier για πλήθος στοιχείων  $N/2$  του στοιχείου  $k$  ο οποίος σχηματίζεται μόνο για τα άρτια στοιχεία ενώ  $F_k^o$  είναι ο αντίστοιχος μετασχηματισμός με πλήθος  $N/2$  μόνο για το περιττά στοιχεία.

Για να δούμε πως αυτή η διαδικασία μειώνει τον αριθμό των υπολογισμών αρκεί να αθροίσουμε τον αριθμό των πολλαπλασιασμών που γίνονται. Παρατηρούμε ότι έχουν προκύψει 2 διακριτοί μετασχηματισμοί με  $N/2$  στοιχεία ο κάθε ένας. Άρα αφού κάθε διακριτός μετασχηματισμός απαιτεί  $(N)^2$  πολλαπλασιασμούς προκύπτει ότι χρειάζονται  $2(N/2)^2$  πολλαπλασιασμοί από τους δύο νέους διακριτούς μετασχηματισμούς. Επιπλέον προσθέτουμε  $N$  πολλαπλασιασμούς που προκύπτουν από τα διανύσματα στροφής  $W$ . Άρα συνολικά χρειάζονται

$$\frac{(N)^2}{2} + N$$

υπολογισμοί.

Αν επαναλάβουμε την παραπάνω διαδικασία για τα  $F_k^e$  και  $F_k^o$  προκύπτουν δυο μετασχηματισμοί για το καθένα με πλήθος στοιχείων  $N/4$ . Δηλαδή θα προκύψουν οι διακριτοί μετασχηματισμοί  $F_k^{ee}$  και  $F_k^{eo}$  για τα σημεία που είναι περιττά-περιττά και περιττά-άρτια και ομοίως  $F_k^{oe}$  και  $F_k^{oo}$  για τα άρτια-άρτια και άρτια-περιττά. Γίνεται έτσι ένας διαδοχικός διαχωρισμός των δεδομένων με το πλήθος στοιχείων να διαιρείται συνεχώς με το 2 μέχρι να προκύψει πλήθος στοιχείων ίσο με τη μονάδα:

$$\frac{N}{2}, \frac{N}{4}, \frac{N}{8}, \dots, \frac{N}{2^\rho} = 1 \tag{3.16}$$

όπου  $\rho = \log_2(N)$

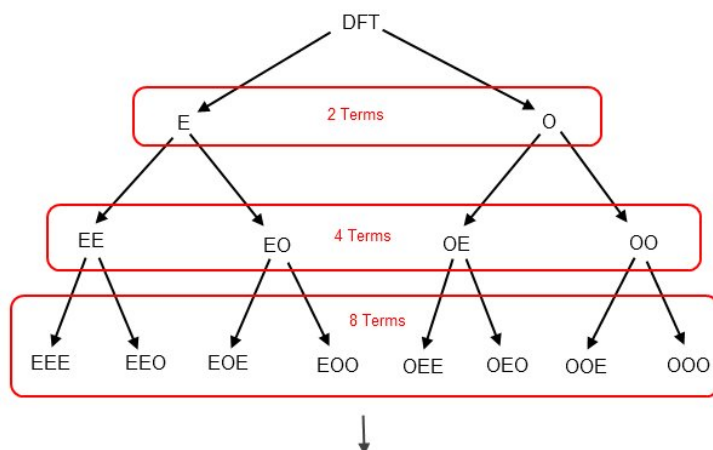
Στην πιο απλή περίπτωση όπου το  $N$  είναι δύναμη του 2. Μπορούμε να εφαρμόζουμε διαδοχικά την παραπάνω διαδικασία μέχρι να φτάσουμε στον μετασχηματισμό ενός στοιχείου. Φυσικά ο μετασχηματισμός αυτός θα μας δώσει το ίδιο το στοιχείο. Αφού θα προκύψουν αθροίσματα της μορφής:

$$\sum_{k=0}^0 \tag{3.17}$$

Για κάθε  $N$  απαιτούνται  $\rho = \log_2 N$  διαχωρισμοί σε άρτια και περιττά. Έτσι προκύπτει ο μετασχηματισμός ενός σημείου ο οποίος θα είναι ένα από τα δεδομένα  $f_n$ . Δηλαδή:

$$F_k^{eooooe\dots oee} = f_n \tag{3.18}$$

Οπότε επαναλαμβάνοντας την παραπάνω διαδικασία προκύπτει :



This shows the binary nature of the Danielson-Lanczos Lemma. Each term can be broken again and again into even and odd terms until samples are exhausted. This is why an FFT requires a power of base 2 samples, e.g. 2, 4, 8, 16, etc.

Σχήμα 3.3: Διαδική φύση του FFT. *Π* :<http://www.alwayslearn.com/dft>

1.  $\frac{N}{2} \rightarrow 2\left(\frac{N^2}{2}\right)^2 + N \rightarrow \frac{N^2}{2} + N$
2.  $\frac{N}{4} \rightarrow 2\left(2\frac{N^2}{4} + \frac{N}{2}\right) + N \rightarrow \frac{N^2}{4} + 2N$
3.  $\frac{N}{8} \rightarrow 2\left[2\left(2\frac{N^2}{8} + \frac{N}{4}\right) + \frac{N}{2}\right] + N \rightarrow \frac{N^2}{8} + 3N$

⋮

$$\rho. \frac{N}{2^p} = 1 \rightarrow \frac{N}{2^p} + \rho N \rightarrow \frac{N^2}{N} + N \log_2 N$$

Για  $N$  μεγάλο προκύπτει ότι ο αριθμός υπολογισμών είναι

$$\approx O(N \log_2 N)$$

Αυτή η διαδικασία είναι ένα παράδειγμα της γενικής τεχνικής του διαίρει και βασίλευε αλγορίθμων.

**Παράδειγμα για N=2**

Εφαρμόζοντας το λήμμα των Danielson-Lanczos για N=2 προκύπτει

$$\begin{aligned}
 F_k &= \sum_{j=0}^{2-1} e^{2\pi i j k / 2} f_j \\
 &= \sum_{j=0}^{2/2-1} e^{2\pi i j k / (2/2)} f_{2j} + W^k \sum_{j=0}^{2/2-1} e^{2\pi i j k / (2/2)} f_{2j+1} \\
 &= F_k^e + W^k F_k^o \\
 &= \sum_{j=0}^0 f_{2j} e^{\frac{-i2\pi j k}{(\frac{2}{2})}} + W^k \sum_{k=0}^0 f_{2k+1} e^{\frac{-i2\pi j k}{(\frac{2}{2})}} \\
 &= f_0 + W^k f_1
 \end{aligned}$$

Ο πρώτος διαχωρισμός είναι αρκετός για να μειωθούν τα αθροίσματα στη μονάδα και αυτό που παραμένει είναι τα δεδομένα  $f_0$ ,  $f_1$  και ο όρος  $W^k$  όπου  $W^k = e^{2\pi i k / 2}$

Όπου για  $k=0,1$

$$\begin{aligned}
 F(0) &= f_0 + W_2^0 f_1 \\
 F(1) &= f_0 + W_2^1 f_1 = f_0 - W_2^0 f_1
 \end{aligned}$$

Αφού όπως φαίνεται και στην εικόνα 3.2

$$W_2^0 = -W_2^1$$

**Παράδειγμα για N=4**

Εφαρμόζοντας το λήμμα των Danielson-Lanczos για N=4 προκύπτει

$$\begin{aligned}
 F_k &= \sum_{j=0}^{N-1} e^{2\pi i j k / N} f_j \\
 &= E + O
 \end{aligned}$$

$$E = \sum_{j=0}^{N/2-1} f_{2j} e^{\frac{-2\pi i (2j)k}{(\frac{N}{2})}} \quad \text{kai} \quad O = \sum_{j=0}^{N/2-1} f_{2j+1} e^{\frac{-2\pi i (2j+1)k}{(\frac{N}{2})}}$$

Ο πρώτος διαχωρισμός δίνει δύο αθροίσματα για τους άρτιους και περιττούς όρους αντίστοιχα. Με τον αριθμό των παραγόντων που αθροίζονται να μειώνεται στο μισό και στα δύο αθροίσματα. Κάνουμε την ίδια διαδικασία στα αθροίσματα E , O. Προκύπτει έτσι :

$$E = EE + EO \quad \text{και} \quad O = OE + OO$$

Όπου EE, EO είναι ο διαχωρισμός σε άρτια-άρτια και άρτια-περιττά αντίστοιχα

$$EE = \sum_{j=0}^{\frac{N}{2}-1} f_{2(2j)} e^{\frac{-2\pi i(2j)k}{\left(\frac{N}{2}\right)}} = \sum_{j=0}^{\frac{N}{4}-1} f_{4j} e^{\frac{2\pi ijk}{\left(\frac{N}{4}\right)}}$$

Και:

$$EO = \sum_{j=0}^{\frac{N}{2}-1} f_{2(2j+1)} e^{\frac{-2\pi i2(2j+1)k}{\left(\frac{N}{2}\right)}} = \sum_{j=0}^{\frac{N}{4}-1} f_{4j+2} e^{\frac{-2\pi ijk}{\left(\frac{N}{4}\right)}} e^{\frac{-2\pi ik}{\left(\frac{N}{2}\right)}} = W_{N/2}^k \sum_{j=0}^{N/4-1} f_{4k+2} e^{\frac{-2\pi ijk}{\left(\frac{N}{4}\right)}}$$

Ενώ OE, OO ο διαχωρισμός για περιττά-άρτια, περιττά-περιττά στοιχεία :

$$OE = W_N^k \sum_{j=0}^{\frac{N}{2}-1} f_{2(2k)+1} e^{\frac{-2\pi i(2j)k}{\left(\frac{N}{2}\right)}} = W_N^k \sum_{j=0}^{\frac{N}{4}-1} f_{4k+1} e^{\frac{-2\pi i(j)k}{\left(\frac{N}{4}\right)}}$$

$$\begin{aligned} OO &= W_N^k \sum_{j=0}^{\frac{N}{2}-1} f_{(2(2j+1)+1)} e^{\frac{-i2\pi(2j+1)k}{\left(\frac{N}{2}\right)}} \\ &= W_N^k \sum_{j=0}^{\frac{N}{4}-1} f_{(4j+3)} e^{\frac{-i2\pi jk}{\left(\frac{N}{4}\right)}} e^{\frac{-2\pi k}{\left(\frac{N}{2}\right)}} \\ &= W_{\frac{N}{2}}^k W_N^k \sum_{j=0}^{\frac{N}{4}-1} f_{(4j+3)} e^{\frac{-i2\pi jk}{\left(\frac{N}{4}\right)}} \end{aligned}$$

Τελικά προκύπτει :

$$\begin{aligned} F_k &= E + O = EE + EO + OO + OE \\ &= \sum_{j=0}^{\frac{N}{4}-1} f_{4j} e^{\frac{2\pi ijk}{\left(\frac{N}{4}\right)}} + W_{N/2}^k \sum_{j=0}^{N/4-1} f_{4k+2} e^{\frac{-2\pi ijk}{\left(\frac{N}{4}\right)}} \\ &\quad + W_N^k \sum_{j=0}^{\frac{N}{4}-1} f_{4k+1} e^{\frac{-2\pi i(j)k}{\left(\frac{N}{4}\right)}} + W_{\frac{N}{2}}^k W_N^k \sum_{j=0}^{\frac{N}{4}-1} f_{(4j+3)} e^{\frac{-i2\pi jk}{\left(\frac{N}{4}\right)}} \end{aligned}$$

Οπότε για N=4

$$F_k = f_0 + W_2^k f_2 + W_4^k f_1 + W_4^k W_2^k f_3$$

## 3.6.2 Διαγράμματα ροής δεδομένων

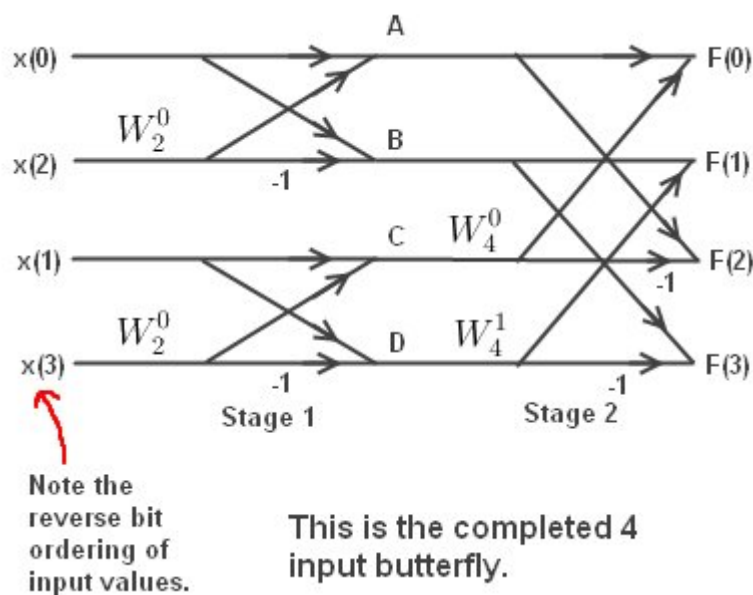
Ένας τρόπος να αναπαραστήσουμε τον γρήγορο μετασχηματισμό Fourier είναι τα διαγράμματα ροής δεδομένων ή διαγράμματα "πεταλούδας". Στην πιο απλή περίπτωση για  $N = 2$  το διάγραμμα ροής είναι όπως στην εικόνα 3.4.



Σχήμα 3.4: Διάγραμμα ροής για 2 σημεία  $\Pi$  : <http://www.alwayslearn.com/dft>

Στο διάγραμμα ροής μπορούμε να ακολουθήσουμε μόνο μια διαδρομή όπου κάθε είσοδος πολλαπλασιάζεται με το "στοιχείο" που βρίσκεται στην διαδρομή. Στους κόμβους γίνεται η άθροιση όπου παίρνουμε και την έξοδο. Δηλαδή το  $x(1)$  στην μια διαδρομή πολλαπλασιάζεται με  $W_2^0$  και τέλος αθροίζεται με  $x(0)$  ώστε να δώσει την έξοδο  $F(0)$  ενώ στην άλλη περίπτωση πολλαπλασιάζεται με  $-1W_2^0$  και αθροίζεται με το  $x(0)$  ώστε να δώσει την έξοδο  $F(1)$ .

Για να σχεδιάσουμε ένα διάγραμμα ροής με  $N=4$ . Αρχικά σχεδιάζουμε δύο διαγράμματα για  $N=2$  και τα ενώνουμε όπως φαίνεται στην εικόνα 3.5



Σχήμα 3.5: Διάγραμμα ροής για 4 σημεία  $\Pi$  : <http://www.alwayslearn.com/dft>

Συνολικά έχουμε 4 διαγράμματα "πεταλούδας" στο πρώτο στάδιο προκύπτουν οι έξοδοι A,B,C,D όπου

$$A = x(0) + W_2^0 x(2)$$

$$B = x(0) - W_2^0 x(2)$$

$$C = x(1) + W_2^0 x(3)$$

$$D = x(1) - W_2^0 x(3)$$

Στο δεύτερο στάδιο οι έξοδοι γίνονται είσοδοι στα δύο επόμενα διαγράμματα πεταλούδας

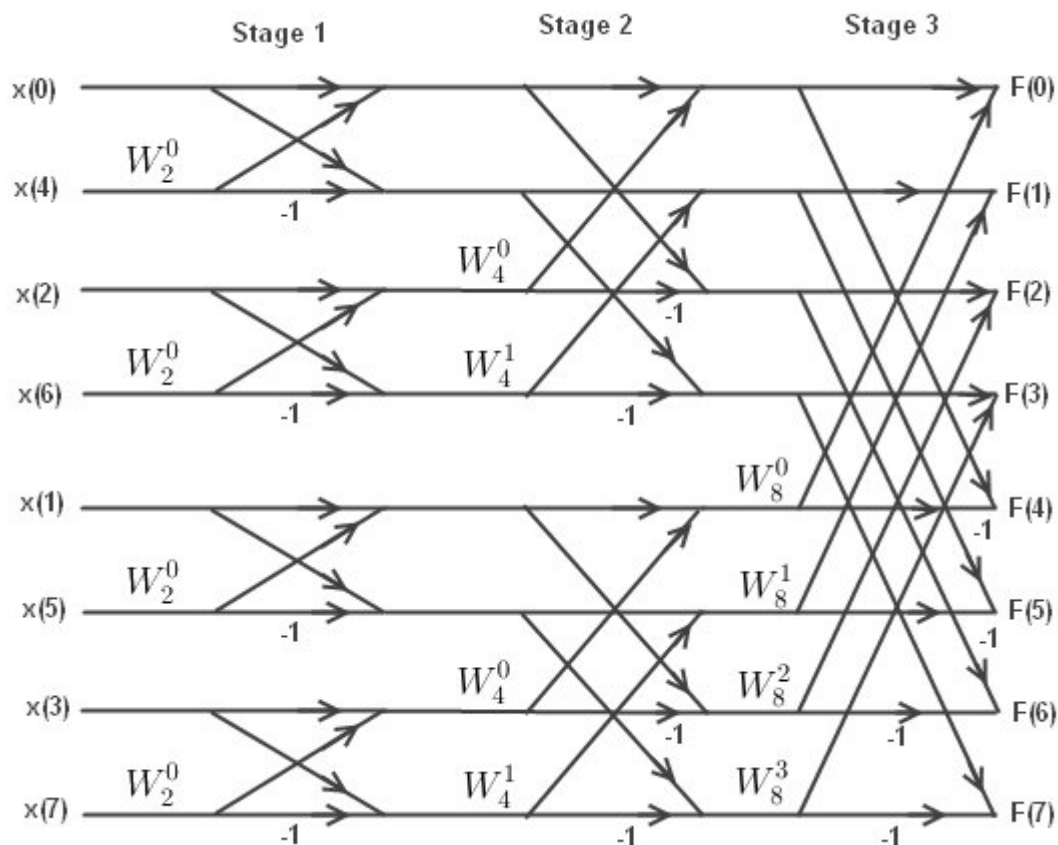
$$F(0) = A + W_4^0 C$$

$$F(1) = B + W_4^1 D$$

$$F(2) = A - W_4^0 C$$

$$F(3) = B - W_4^1 D$$

Με παρόμοιο τρόπο μπορούμε να κατασκευάσουμε τα διαγράμματα για  $N=8$  ή και παραπάνω στοιχεία εικόνα 3.6



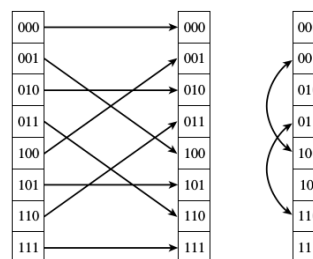
Σχήμα 3.6: Διάγραμμα ροής για 8 σημεία [Π :http://www.alwayslearn.com/dft](http://www.alwayslearn.com/dft)

Μπορούμε να παρατηρήσουμε από τα διαγράμματα καθώς και από τις εξισώσεις από τα παραδείγματα για  $N=2,4,8$  ότι τα στοιχεία που δίνονται ως είσοδοι είναι σε

ανακατεμένη σειρά δηλαδή  $x(0), x(4), x(2), x(6), \dots$ . Αυτή η αναδιάταξη των όρων ήταν αναμενόμενη αφού το λήμμα τον Danielson-Lanczos στηρίζεται σε αυτή. Οπότε για να γνωρίζουμε πια θα πρέπει να είναι η σειρά των πράξεων θα πρέπει να αναδιατάξουμε τους όρους εξ αρχής ή το αντίστροφο. Αυτό μπορεί να γίνει κωδικοποιώντας τον τρόπο με τον οποίο ανακατεύονται οι όροι. Αν μετρήσουμε από το 0 μέχρι 7 στο δυαδικό σύστημα προκύπτει η ακολουθία 000, 001, 010, 011, 100, 101, 110, 111. Αν αντιστρέψουμε τώρα τα ψηφία κάθε δεκαδικού αριθμού προκύπτει 000, 100, 010, 110, 001, 101, 011, 111 η οποία αντιστοιχεί σε δεκαδικό στην ακολουθία 0, 4, 2, 6, 1, 5, 3, and 7. Η διαδικασία αυτή ονομάζεται αναστροφή δυαδικού ψηφίου (reverse binary). Η αναστροφή δυαδικού στοιχείου αποτελεί αναπόσπαστο κομμάτι του γρήγορου μετασχηματισμού Fourier αφού στις περισσότερες περιπτώσεις θέλουμε να γνωρίζουμε πια συχνότητα ανήκει σε ποια σημεία εισόδου και το αντίστροφο.

0	→	000	→	000	→	0
1	→	001	→	100	→	4
2	→	010	→	010	→	2
3	→	011	→	110	→	6
4	→	100	→	001	→	1
5	→	101	→	101	→	5
6	→	110	→	011	→	3
7	→	111	→	111	→	7

(α')



(a)

(b)

(β')

Σχήμα 3.7: Αναστροφή δυαδικού ψηφίου μεγέθους 8 Πηγή: [26]

### 3.7 Υπορουτίνα FFT

Εφόσον έχουμε εξετάσει τις βασικές αρχές του αλγορίθμου FFT θα δούμε πως αυτές μπορούν να γίνουν πρόγραμμα. Το παρακάτω πρόγραμμα είναι μια ρουτίνας γραμμένη σε Java η οποία υλοποιεί τον FFT σε μια διάσταση για πλήθος στοιχείων που είναι δύναμη του 2. Περισσότερες πληροφορίες στην βιβλιογραφία [28],[26].

```
public static void four1(double[] data, int n,int isign) {
    int nn,mmax,m,j,istep,i;
    double wtemp,wr,wpr,wpi,wi,theta,tempr,tempi;
    if (n<2 || (n&(n-1))!= 0) throw new IllegalArgumentException("n must be power
        of 2 in four1");
    nn = n << 1;
    j = 1;
    for (i=1;i<nn;i+=2) {
        if (j > i) {
            swap(data,j-1,i-1);
            swap(data,j,i);
        }
        m=n;
        while (m >= 2 && j > m) {
            j -= m;
            m >>= 1;
        }
    }
}
```

```

    j += m;
}
mmax=2;
while (nn > mmax) {
    istep=mmax << 1;
    theta=isign*(6.28318530717959/mmax);
    wtemp=Math.sin(0.5*theta);
    wpr = -2.0*wtemp*wtemp;
    wpi=Math.sin(theta);
    wr=1.0;
    wi=0.0;
    for (m=1;m<mmax;m+=2) {
        for (i=m;i<=nn;i+=istep) {
            j=i+mmax;
            tempr=wr*data[j-1]-wi*data[j];
            tempi=wr*data[j]+wi*data[j-1];
            data[j-1]=data[i-1]-tempr;
            data[j]=data[i]-tempi;
            data[i-1] += tempr;
            data[i] += tempi;
        }
        wr=(wtemp=wr)*wpr-wi*wpi+wr;
        wi=wi*wpr+wtemp*wpi+wi;
    }
    mmax=istep;
}
}
}

```

Οι παράμετροι που μπαίνουν ως είσοδοι στην υπορουτίνα είναι ο αριθμός των στοιχείων  $n(=N)$  ένας πίνακας `data` με πλήθος στοιχείων  $[0, \dots, 2 * n - 1]$  και ένας ακέραιος αριθμός `isign` ο οποίος λαμβάνει τιμές μόνο 1 ή -1 και συμβολίζει στην ουσία τον μιγαδικό αριθμό  $i$  ή  $-i$  ανάλογα με την τιμή που του θέτουμε κάθε φορά. Αν `isign = -1` τότε η υπορουτίνα υπολογίζει τον αντίστροφο μετασχηματισμό Fourier χωρίς όμως τον παράγοντα κανονικοποίησης  $1/N$  ο οποίος μπορεί να μπει από την χρήστη μετά την υπορουτίνα.

Ο μονοδιάστατος πίνακας `data` περιέχει τα δεδομένα που βάζουμε ως είσοδο και τα οποία θέλουμε να μετασχηματίσουμε. Παρατηρούμε όμως ότι ο πίνακας αυτός έχει  $2n$  στοιχεία. Αυτό συμβαίνει γιατί κάθε μιγαδικός αριθμός καταλαμβάνει δύο θέσεις στην μνήμη του πίνακα `data`. Η μια θέση αναφέρεται στο πραγματικό του μέρος και η άλλη στο φανταστικό του μέρος. Δηλαδή `data[0]` είναι το πραγματικό μέρος του πρώτου μιγαδικού  $f_0$  και `data[1]` το μιγαδικό του μέρους ( $f_0 = data[0] + i \times data[1]$ ). Με αυτό τον τρόπο ο πίνακας εκτείνεται μέχρι το στοιχείο `data[2*n-1]` το οποίο θα είναι το φανταστικό μέρος του μιγαδικού στοιχείου  $f_{N-1}$ . Εικόνα 3.8.

Η υπορουτίνα επιστρέφει της τιμές στο πεδίο των συχνοτήτων πάλι με το πραγματικό και το φανταστικό κομμάτι να αποθηκεύονται σε διαδοχικές τιμές του πίνακα. Η πρώτη θετική μη μηδενική τιμή αποθηκεύεται στις θέσεις του πίνακα `data[2], data[3]` ενώ αντίστοιχα η μικρότερη αρνητική τιμή αποθηκεύεται στις θέσεις `data[2*n-2], data[2*n-1]`. Η θετικές τιμές αυξάνονται από καταλαμβάνοντας τις `data[4], data[5]` μέχρι τη θέση `data[n-2], data[n-1]` και αντίστοιχα οι αρνητικές τιμές αποθηκεύονται στις θέσεις `data[2*n-4], data[2*n-3]` έως και τη θέση `data[n+2], data[n+3]`. Οι μεσαίες



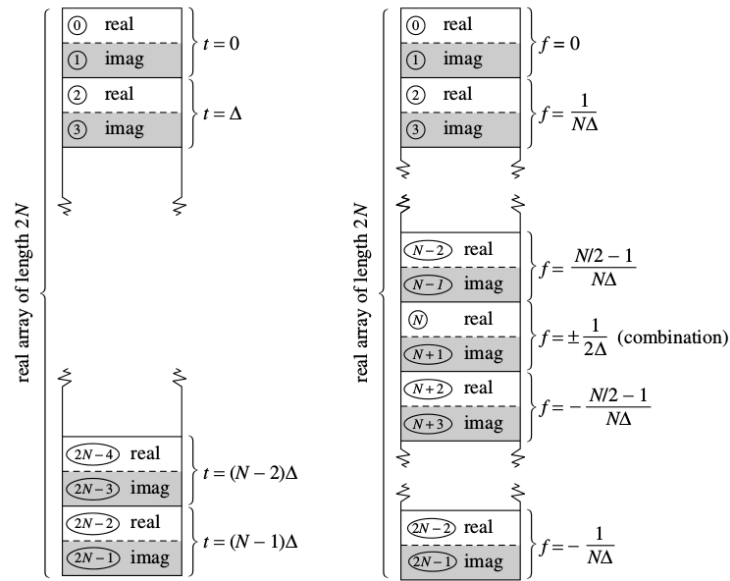
θέσεις  $\text{data}[n]$ ,  $\text{data}[n+1]$  αντιστοιχούν στην συχνότητα Nyquist  $f = \frac{1}{2\Delta}$ .

Στην εικόνα 3.8 παρατηρούμε τον πίνακα που εισάγαμε στην υπορουτίνα και αντίστοιχα τον πίνακα που λαμβάνουμε στην έξοδο. Ο πρώτος πίνακας περιέχει τις τιμές της δειγματοληπτημένης συνάρτησης που αντιστοιχούν σε

$$t = 0, \Delta, \dots, (N-1)\Delta \quad (3.19)$$

όπου  $\Delta$  η τιμή διαχωρισμού των δειγμάτων. Στην έξοδο λαμβάνουμε τις τιμές στον χώρο του Fourier οι οποίες αντιστοιχούν σε

$$f = 0, \frac{1}{N\Delta}, \dots, \pm \frac{1}{2\Delta}, \dots, -\frac{1}{N\Delta} \quad (3.20)$$



Σχήμα 3.8: Διανύσματα εισόδου-εξόδου FFT. Το διάνυσμα εισόδου περιέχει  $N$  μιγαδικούς αριθμούς. Ο κάθε ο αριθμός αποτελεί ένα σημείο της συνάρτησης για την οποία θέλουμε να κάνουμε τον μετασχηματισμό Fourier. Κάθε μιγαδικός είναι αποθηκευμένος σε δύο διαδοχικές θέσεις. Η έξοδος θα αντιστοιχεί σε έναν πίνακα με  $N$  μιγαδικούς αριθμούς όπου κάθε ένας αντιστοιχεί σε  $N$  συχνότητες. Ο κάθε μιγαδικός αντιστοιχεί όπως και πριν σε δύο διαδοχικές τιμές του πίνακα. Πηγή: [26]

### 3.8 Παραλλαγές του FFT

Υπάρχουν πολλές διαφορετικές παραλλαγές του αλγόριθμου FFT. Οι διαδικασίες που είδαμε παραπάνω ονομάζονται Cooley-Tukey FFT ή αποδεκατισμός-χρόνου (decimation in time). Αρχικά ο αλγόριθμος αναδιατάσσει τα δεδομένα εισόδου σε σειρά ανάστροφου δυαδικού (bit-reverse order) όπως περιγράψαμε προηγουμένως και μετά υπολογίζει το μετασχηματισμό τους με  $\log_2 N$  υπολογισμούς. Είναι όμως πιθανών να γίνουν πρώτα οι υπολογισμοί των δεδομένων εισόδου και μετά η αναδιάταξη όρων σε σειρά ανάστροφου δυαδικού. Η αναδιάταξη αυτή δηλαδή θα συμβεί στο πεδίο των συχνοτήτων. Ο αλγόριθμος αυτός καλείται αποδεκατισμός-συχνότητας (decimation in frequency) ή Sande-Tukey FFT. Για μερικές εφαρμογές η διαδικασία αναστροφής δυαδικού ψηφίου μπορεί να αποφευχθεί. Για παράδειγμα σε εφαρμογές όπου κάνουμε μετασχηματισμό Fourier προκύπτουν οι ανακατεμένες τιμές κάνουμε πράξεις με αυτές και μετά επιστρέφουμε πίσω με έναν ανάστροφο μετασχηματισμό Fourier.

Άλλοι αλγόριθμοι FFT υποδιαιρούν το αρχικό σύνολο δεδομένων μεγέθους  $N$  όχι μέχρι τη μονάδα αλλά μέχρι μια άλλη δύναμη του 2. Για παράδειγμα  $N=4$  ή  $N=8$ . Έπειτα εκμεταλλεύονται συμμετρίες για τα συγκεκριμένα  $N$ . Για παράδειγμα για  $N=4$  οι τριγωνομετρικοί αριθμοί ημίτονου και συνημιτόνου προκύπτουν όλοι  $\pm 1$  και 0 οπότε πολύ πολλαπλασιασμοί απλοποιούνται.

Επίσης υπάρχουν αλγόριθμοι FFT όπου το σύνολο των δεδομένων έχουν μέγεθος  $N$  το οποίο δεν είναι δύναμη του 2. Οι αλγόριθμοι αυτοί λειτουργούν με παρόμοιο τρόπο, εφαρμόζοντας το λήμμα Danielson-Lanczos διαιρώντας το πρόβλημα σε μικρότερα, αλλά όχι με ένα παράγοντα 2 αλλά με οποιοδήποτε μικρό πρώτο αριθμό ο οποίος διαιρεί το  $N$ . Αν όμως το  $N$  είναι πρώτος αριθμός (διαιρείται με τον εαυτό του και την μονάδα) τότε δεν μπορεί να γίνει υποδιαίρεση του προβλήματος και έτσι ο μετασχηματισμός γίνεται πολύ αργός της τάξης του  $N^2$  και όχι της τάξης του  $N \log_2 N$ .

#### 3.8.1 FFT για 2 και για περισσότερες διαστάσεις

Ανάλογα με την περίπτωση του γρήγορου μετασχηματισμού στην μια διάσταση μπορούμε να εργαστούμε και σε περισσότερες διαστάσεις.

Έστω μια μιγαδική συνάρτηση  $h(k_1, k_2)$  ορισμένη σε ένα δύο διαστάσεων πλέγμα  $0 < k_1 \leq N_1 + 1$ ,  $0 < k_2 \leq N_2 + 1$ . Προκύπτει ότι ο δις-διαστατός διακριτός μετασχηματισμός Fourier θα δίνεται από την σχέση 3.21. Η είσοδος του μετασχηματισμού Fourier θα είναι ένας δις-διαστατός πίνακας όπου τα στοιχεία  $k_1$  θα αντιστοιχούν στις γραμμές και τα  $k_2$  στις στήλες.

$$H(n_1, n_2) = \sum_{k_2=0}^{N_2-1} \sum_{k_1=0}^{N_1-1} e^{(2\pi i k_2 n_2)/N_2} e^{(2\pi i k_1 n_1)/N_1} h(k_1, k_2) \quad (3.21)$$

Η παραπάνω έκφραση μπορεί να γραφεί στη μορφή:

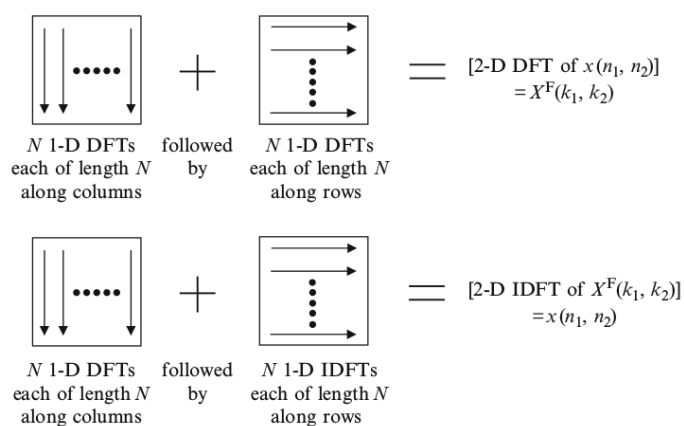
$$H(n_1, n_2) = \sum_{k_2=0}^{N_2-1} e^{(2\pi i k_2 n_2)/N_2} \left[ \sum_{k_1=0}^{N_1-1} e^{(2\pi i k_1 n_1)/N_1} h(k_1, k_2) \right] \quad (3.22)$$

Όπου όρος στις αγκύλες αντιστοιχεί στον μόνο-διαστατό μετασχηματισμό Fourier της γραμμής  $k_2$  και μπορεί να υπολογιστεί με τις μεθόδους που έχουν αναφερθεί παραπάνω. Παρατηρούμε ότι ο FFT στις δύο διαστάσεις προκύπτει υπολογίζοντας διαδοχικά τον μόνο-διαστατό FFT κάθε δείκτη. Συμβολικά μπορούμε να γράψουμε.

$$H(n_1, n_2) = \text{FFT ανα γραμμή}(\text{FFT ανα στήλη}[h(k_1, k_2)])$$

$$H(n_1, n_2) = \text{FFT ανα στήλη}(\text{FFT ανα γραμμή}[h(k_1, k_2)])$$

Αντίστοιχα μπορούμε να υπολογίσουμε τον μετασχηματισμό Fourier για περισσότερες από δύο διαστάσεις. Ο μετασχηματισμός Fourier σε δύο διαστάσεις εφαρμόζεται στην επεξεργασία εικόνας και βίντεο. Μια εικόνα συνήθως μπορεί να αναπαρασταθεί σε ένα δύο διαστάσεων διάνυσμα της χρωματικής έντασης κάθε pixel. Χρησιμοποιώντας τον FFT μπορεί να γίνει επεξεργασία της εικόνας π.χ φιλτράρισμα για υψηλής ή χαμηλής συχνότητας.



Σχήμα 3.9: 2-D FFT , 2-D IFFT

Αν ο πίνακας έχει διαστάσεις  $(N \times N)$  τότε ο FFT του πίνακα μπορεί να προκύψει με  $2N$  μόνο-διάστατους FFT.



## Κεφάλαιο 4

# Εφαρμογή του FFT στην επίλυση της εξίσωσης Laplace

### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται μια αριθμητική μέθοδος για την επίλυση της εξίσωσης Laplace για επίπεδη γεωμετρία όπως σε έναν ανιχνευτή τύπου pixel ή strip. Μέσω της μεθόδου αυτής υπολογίζονται οι χωρητικότητες. Οι χωρητικότητες αυτές είναι πολύ σημαντικές στον σχεδιασμό των ανιχνευτών καθώς σχετίζονται με το θόρυβο και τα φαινόμενα crosstalk ανάμεσα στα pixel ή τα strip του ανιχνευτή όπως εξετάσαμε στο κεφάλαιο 2. Η γνώση των χωρητικότητας ανάλογα με τη γεωμετρία είναι πολύ χρήσιμη για το βέλτιστο σχεδιασμό του ανιχνευτή καθώς και των εξωτερικών ηλεκτρονικών. Η μέθοδος αυτή περιγράφεται αναλυτικά στην βιβλιογραφία [1], [2]

### 4.2 Αριθμητική επίλυση της εξίσωσης Laplace

#### 4.2.1 Ανιχνευτές με επίπεδη γεωμετρία τύπου pixel

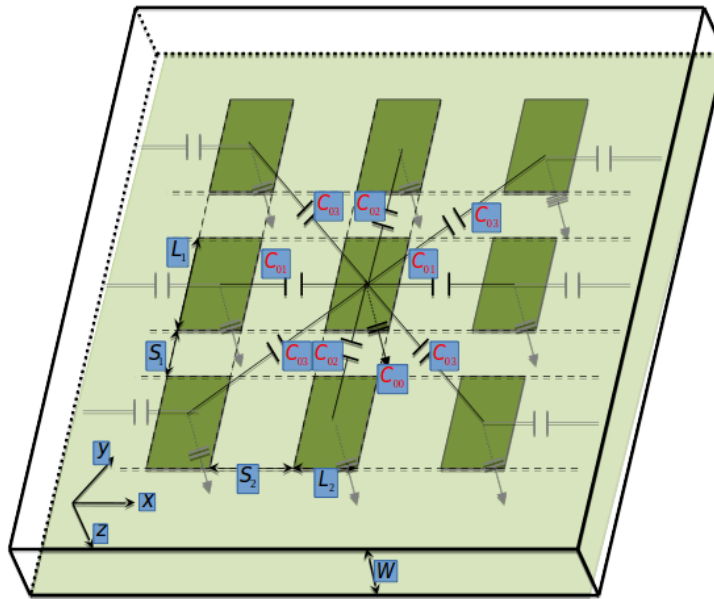
Στη μέθοδο αυτή γίνεται αριθμητική επίλυση της εξίσωσης Laplace στις 3 διαστάσεις. Θεωρούμε μια επίπεδη γεωμετρία ανιχνευτή με 9 pixel εικόνα 4.1. Οι δύο διαστάσεις θεωρούνται παράλληλες με την επιφάνεια των διόδων και χωρίζονται σε διακριτά σημεία, ενώ η τρίτη θεωρείται κάθετη με την επιφάνεια του ανιχνευτή και συνεχής. Θεωρούμε ότι στη μια επιφάνεια έχουν δημιουργηθεί οι δίοδοι σε μορφή ορθογώνιων pixel, ενώ η άλλη επιφάνεια είναι συνεχής. Επίσης το πάχος των pixel είναι αρκετά μικρό σε σχέση με τη περιοχή απογύμνωσης η οποία θεωρούμε ότι καλύπτει όλο τον όγκο του ανιχνευτή. Η χωρητικότητες που μας ενδιαφέρει να υπολογίσουμε είναι η χωρητικότητα μεταξύ του κάθε pixel και του υποστρώματος  $C_{00}$  καθώς και οι χωρητικότητες μεταξύ των γειτονικών pixel  $C_{01}$  και των διαγωνίων  $C_{02}$ . Οι χωρητικότητες σχετίζονται άμεσα με τη γεωμετρία του ανιχνευτή.

Για να υπολογίσουμε τις χωρητικότητες αυτές επιλύουμε την εξίσωση Poisson με τις κατάλληλες συνοριακές συνθήκες. Για απλούστευση θεωρούμε ότι ο όγκος καθώς και η επιφάνεια του ανιχνευτή δεν έχουν στατικά φορτία. Σε πραγματικές συνθήκες υπάρχουν φορτία και στην περίπτωση του πλήρους απογυμνωμένου ανιχνευτή τα οποία οφείλονται στα ιόντα που υπάρχουν στις περιοχές με περίσσεια ηλεκτρονίων και οπών καθώς και σε ατέλειες του υλικού. Παρόλα αυτά τα στατικά φορτία δεν συνεισφέρουν στον υπολογισμό της χωρητικότητας ενός πλήρους απογυμνωμένου α-

νιχνευτή καθώς δίνουν ένα ηλεκτρικό πεδίο που είναι ανεξάρτητο της τάσης πόλωσης. Άρα χωρίς βλάβη της γενικότητας μπορούν να θεωρηθούν μηδέν σε όλο τον όγκο του ανιχνευτή.

Η συγκεκριμένη μέθοδος αφορά την ιδανική περίπτωση όπου ο ανιχνευτής είναι απαλλαγμένος από φορτία Άρα η επίλυση της εξίσωσης Poisson περιορίζεται στην εξίσωση Laplace

$$\nabla^2 V(x, y, z) = 0 \quad (4.1)$$



Σχήμα 4.1: Διάταξη pixel με χωρητικότητες ανάμεσα σε γειτονικά pixel της ίδιας σειράς  $C_{01}$ , διαγώνια  $C_{02}$  και  $C_{00}$  μεταξύ pixel και υποστρώματος.  $W$  είναι το πάχος του υποστρώματος,  $S_1$  και  $S_2$  είναι η απόσταση των γειτονικών pixel για τον  $x$  και για τον  $y$  άξονα αντίστοιχα,  $L_1$  και  $L_2$  το πλάτος του κάθε pixel για τον  $x$  και για τον  $y$  άξονα αντίστοιχα.

Ως συνοριακές συνθήκες θεωρούμε την επιφάνεια που μας ενδιαφέρει ότι έχει δυναμικό 1 και γειώνουμε τις άλλες επιφάνειες. Δηλαδή:

1. Κεντρικό pixel  $V_{22} = 1$
2. Πίσω επιφάνεια γειωμένη  $V_b = 0$
3. Γειτονικά pixel γειωμένα  $V_{11} = V_{12} = V_{13} = V_{21} = V_{23} = V_{31} = V_{32} = V_{33} = 0$

Επίσης στις περιοχές που δεν καλύπτονται από pixel η συνοριακή συνοριακή συνθήκη προκύπτει από νόμο του Gauss:

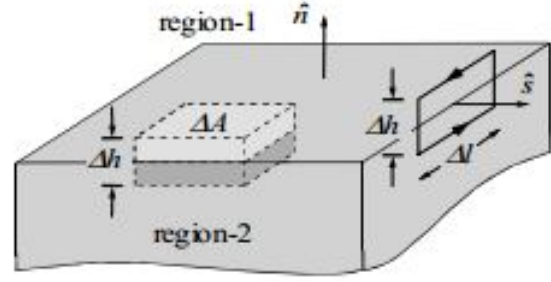
$$\lim_{\Delta h \rightarrow 0} \iint_V \vec{D} d\vec{S} = \hat{n} (\vec{D}_s - \vec{D}_a)$$

$$\hat{n} (\vec{D}_s - \vec{D}_a) = \sigma$$

$$\hat{n} \cdot (\varepsilon_s \vec{E}_s - \varepsilon_a \vec{E}_a) = \sigma$$

Όπου  $\varepsilon_s, \varepsilon_a$  η διηλεκτρική σταθερά για τον ημιαγωγό και για τον εξωτερικό χορό αντίστοιχα. Αρχικά υποθέσαμε ότι δεν υπάρχουν στατικά φορτία στις επιφάνειες του ανιχνευτή που δεν καλύπτονται από pixel άρα τελικά προκύπτει

$$\varepsilon_s E_s - \varepsilon_a E_a = 0 \quad (4.2)$$



Σχήμα 4.2: Συνοριακές συνθήκες στην επιφάνεια ενός διηλεκτρικού μέσου. Το ηλεκτρικό πεδίο θα πρέπει να είναι συνεχές Πηγή [29]

Οι άξονες έχουν τη φορά που φαίνεται στο σχήμα 4.1. Η επιφάνεια για  $z=0$  αντιστοιχεί στην επιφάνεια των pixel εκεί δηλαδή που έχουν δημιουργηθεί οι δίοδοι.

Για την επίλυση της εξίσωσης Laplace εφαρμόζουμε έναν δύο διαστάσεων μετασχηματισμό Fourier στις διαστάσεις  $x, y$ . Προκύπτει έτσι η παρακάτω διαφορική εξίσωση 4.3

$$\nabla^2 V = 0 \xrightarrow{\mathcal{F}}$$

$$\frac{\partial^2 V(k_x, k_y, z)}{\partial z^2} = (k_x^2 + k_y^2) V(k_x, k_y, z) \quad (4.3)$$

όπου :

- $V(k_x, k_y, z)$  το δυναμικό στο χώρο του Fourier
- $k_x, k_y$ , αντιστοιχούν στις συντεταγμένες του χώρου Fourier

Έστω  $F(k_x, k_y) = \sqrt{k_x^2 + k_y^2}$ . Τότε η εξίσωση 4.3 έχει λύση της μορφής

$$V(k_x, k_y, z) = A(k_x, k_y) e^{-F(k_x, k_y)z} + B(k_x, k_y) e^{F(k_x, k_y)z} \quad (4.4)$$

Διαφορίζοντας τη 4.4 για  $z=0$ . Προκύπτει

$$E_z(k_x, k_y, 0) = -F(k_x, k_y) [B(k_x, k_y) - A(k_x, k_y)] \quad (4.5)$$

Οι συνοριακές συνθήκες για την πίσω πλευρά επιβάλλουν ότι  $V(k_x, k_y, W) = 0$  τότε από τις εξισώσεις 4.4, 4.5 προκύπτει

$$E_z(k_x, k_y, 0) = F(k_x, k_y) \frac{1 + e^{-2F(k_x, k_y)W}}{1 - e^{-2F(k_x, k_y)W}} V(k_x, k_y, 0) \Leftrightarrow \quad (4.6)$$

$$V(k_x, k_y, 0) = \frac{E_z(k_x, k_y, 0)}{F(k_x, k_y) \frac{1 + e^{-2F(k_x, k_y)W}}{1 - e^{-2F(k_x, k_y)W}}} \quad (4.7)$$

Η σχέση 4.6 συνδέει το ηλεκτρικό πεδίο στο χώρο Fourier με το δυναμικό στο χώρο Fourier.

Η εξίσωση Laplace μπορεί να λυθεί μέσω μιας αυτοσυνεπής μεθόδου.

Αρχικά ορίζουμε μια αρχική τιμή του δυναμικού στον χώρο. Αυτή μπορεί να είναι μια γραμμική παρεμβολή μεταξύ του κεντρικού pixel και των γειτονικών. Τα pixel καθώς και η πίσω πλευρά θα έχουν δυναμικό όπως ορίστηκε από της συνοριακές συνθήκες. Αρά θεωρούμε :

- Κεντρικό pixel  $V_{22} = 1$
- Πίσω επιφάνεια γειωμένη  $V_b = 0$
- Γειτονικά pixel γειωμένα  $V_{11} = V_{12} = V_{13} = V_{21} = V_{23} = V_{31} = V_{32} = V_{33} = 0$

Ενώ ορίζουμε μια συνάρτηση μεταξύ των pixel η οποία να ακολουθεί τις αρχικές συνθήκες και να είναι συμβατή από πλευράς φυσικής . Αυτή θα μπορούσε να είναι μια συνάρτηση της μορφής:

$$V(x, y, 0) = \frac{1}{1+r} \quad \text{όπου} \quad r = \sqrt{x^2 + y^2} \quad (4.8)$$

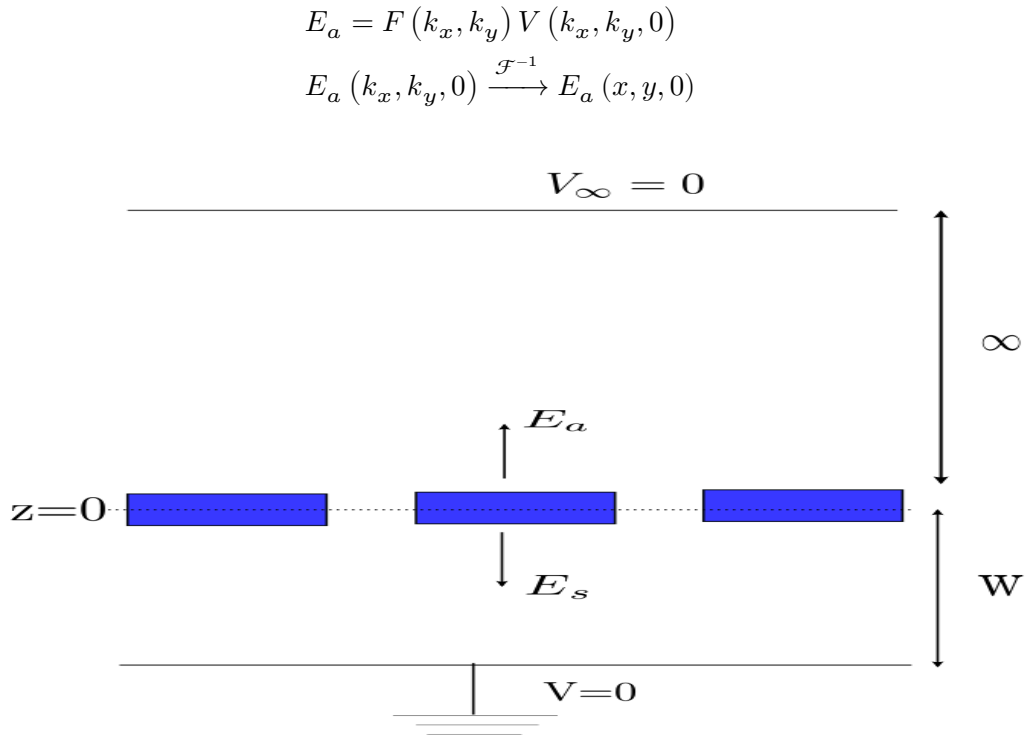
Η όποια συμβαδίζει με της συνθήκες του προβλήματος αφού αν θεωρήσουμε ότι  $r = 0$  είναι η θέση του κεντρικού pixel προκύπτει  $V = 1$  ενώ παράλληλα εκφράζει την πτώση του δυναμικού καθώς απομακρυνόμαστε από το κεντρικό pixel.

Εφαρμόζοντας ένα δύο διαστάσεων μετασχηματισμό Fourier στην αρχική συνάρτηση υπολογίζουμε το δυναμικό στο χώρο των συχνοτήτων. Έπειτα μπορούμε να υπολογίσουμε το πεδίο στο εσωτερικό του ανιχνευτή από την σχέση 4.6 και εφαρμόζοντας σε αυτή ανάστροφο μετασχηματισμό Fourier.

$$\begin{aligned} V(k_x, k_y, 0) &= \mathcal{F}_{xy}[V(x, y, 0)] \Rightarrow \\ E_s(k_x, k_y, 0) &= F(k_x, k_y) \frac{1 + e^{-2F(k_x, k_y)W}}{1 - e^{-2F(k_x, k_y)W}} V(k_x, k_y, 0) \Rightarrow \\ E_s(k_x, k_y, 0) &\xrightarrow{\mathcal{F}^{-1}} E_s(x, y, 0) \end{aligned}$$

Επειδή η μέθοδος είναι γενική και δεν έχει καθοριστεί ακόμα ποια περιοχή είναι η περιοχή που αντιστοιχεί σε πυρίτιο και ποια θεωρείται η εξωτερική περιοχή, μπορούμε να υπολογίσουμε το πεδίο στον εξωτερικό χώρο από τη σχέση 4.6 θεωρώντας  $W \rightarrow \infty$  και εφαρμόζοντας ανάστροφο μετασχηματισμό Fourier. Το ότι έχουμε πυρίτιο στην περιοχή από 0 έως  $w$  και αντίστοιχα έχουμε αέρα στον υπόλοιπο χώρο καθορίζεται μόνο από την διηλεκτρική σταθερά η οποία υπεισέρχεται αργότερα στον υπολογισμό του φορτίου και της χωρητικότητας. Άρα μπορούμε να υπολογίσουμε από την σχέση 4.6 το ηλεκτρικό πεδίο σε έναν χώρο από το 0 έως  $w$  και αντίστοιχα σε μία περιοχή από 0 έως το  $\infty$  όπου στο άπειρο θεωρούμε ( $V_\infty = 0$ ), όπως κάναμε τον υπολογισμό του ηλεκτρικού πεδίου μέσα στην περιοχή από 0 έως  $w$  η οποία πρακτικά θεωρείται η περιοχή του ανιχνευτή. Θεωρούμε δηλαδή μια "απείρου μήκους" περιοχή που εκτείνεται από την επιφάνεια των διόδων μέχρι το άπειρο και υπολογίζουμε σε αυτό το ηλεκτρικό πεδίο από τη σχέση 4.6. Υπολογίζουμε με αυτό τον τρόπο το ηλεκτρικό πεδίο στο εξωτερικό χώρο που αργότερα με την προσθήκη της διηλεκτρικής σταθεράς του αέρα στον υπολογισμό του φορτίου και της χωρητικότητας ορίζουμε την περιοχή του αέρα. Προκύπτουν οι σχέσεις:





Σχήμα 4.3: Περιοχή από 0 έως  $w$  και περιοχή από 0 έως  $\infty$  στις δύο διαστάσεις. Υπολογίζουμε το ηλεκτρικό πεδίο και στις δύο περιοχές από την σχέση 4.6.

Εφόσον έχουμε υπολογίσει το ηλεκτρικό πεδίο στον εξωτερικό χώρο, μπορούμε να υπολογίσουμε το ηλεκτρικό πεδίο στο χώρο που δεν καλύπτεται από τις διόδους μέσω της συνοριακής συνθήκης 4.2.

$$E_s = E_a \frac{\epsilon_a}{\epsilon_s} \quad (4.9)$$

Το πεδίο στις περιοχές που καλύπτονται από τα pixel παραμένει υπολογισμένο όπως προηγουμένως. Από τις νέες τιμές του ηλεκτρικού πεδίου που προκύπτουν μπορούμε να υπολογίσουμε το δυναμικό. Εφαρμόζουμε ένα μετασχηματισμό Fourier στις νέες τιμές. Ύστερα υπολογίζουμε το δυναμικό από τη σχέση 4.7 στο χώρο Fourier και με ανάστροφο μετασχηματισμό καταλήγουμε σε μια νέα τιμή για το δυναμικό  $V'(x, y, 0)$ . Το δυναμικό στα pixel παραμένει όπως και προηγουμένως σύμφωνα με τις συνοριακές συνθήκες (κεντρικό pixel  $V = 1$  και  $V = 0$  στα υπόλοιπα).

$$E'(k_x, k_y, 0) = \mathcal{F}_{xy}[E'(x, y, 0)] \Rightarrow$$

$$V'(k_x, k_y, 0) = \frac{E'_z(k_x, k_y, 0)}{F(k_x, k_y) \frac{1+e^{-2F(k_x, k_y)W}}{1-e^{-2F(k_x, k_y)W}}} \Rightarrow$$

$$V'(k_x, k_y, 0) \xrightarrow{\mathcal{F}^{-1}} V'(x, y, 0)$$

Το συνολικό δυναμικό θα είναι ένας γραμμικός συνδυασμός της αρχικής τιμής και της νέας τιμής που προέκυψε από τις συνοριακές συνθήκες. Με συντελεστές οποιαδήποτε τιμή θα μπορέσει να επιφέρει σύγκλιση. Δηλαδή ισχύει μια σχέση της μορφής.

$$V'' = (1 - f)V + fV' \quad (4.10)$$

Όπου το  $f$  είναι μια σταθερά από 0 έως 1. Επιλέγουμε την  $f$  κατάλληλα ώστε η μέθοδος να συγκλίνει. Για παράδειγμα θα μπορούσαμε να θέσουμε  $f = 0.2$  έτσι ώστε

$$V'' = 0,8V + 0,2V' \quad (4.11)$$

Επαναλαμβάνουμε τη παραπάνω διαδικασία θεωρώντας κάθε φορά ως αρχική τιμή το συνολικό δυναμικό  $V''$  μέχρι να υπάρξει σύγκλιση. Δηλαδή να ισχύει η σχέση:

$$|V'' - V| = \varepsilon \quad (4.12)$$

Όπου  $\varepsilon$  είναι ένας αρκετά μικρός αριθμός  $\varepsilon \approx 10^{-3}$

Εφόσον έχει επιτευχθεί σύγκλιση έχει γίνει μια εκτίμηση του δυναμικού καθώς και του ηλεκτρικού πεδίου στο εσωτερικό χώρο του ανιχνευτή και στον εξωτερικό.

Οι συνοριακές συνθήκες πάνω στην επιφάνεια στη περιοχή που καλύπτεται από pixel δίνουν μία πυκνότητα φορτίου. Δηλαδή ισχύει

$$\varepsilon_s E_s + \varepsilon_a E_a = \sigma_f \quad (4.13)$$

Ολοκληρώνοντας πάνω σε όλη την επιφάνεια κάθε pixel προκύπτει το συνολικό φορτίο  $Q$ . Λόγω του ότι θέλουμε ο ανιχνευτής σαν σύνολο να είναι ηλεκτρικά ουδέτερος υποθέτουμε ότι το φορτίο αυτό επάγεται και στην πίσω πλευρά του ανιχνευτή  $Q_b = -Q$ . Το κεντρικό pixel επάγει φορτίο όχι μόνο στη πίσω πλευρά αλλά και στα άλλα pixel κατά συνέπεια η κατανομή του ηλεκτρικού πεδίου καθώς και η χωρητικότητα του pixel με το υπόστρωμα διαφέρει από την περίπτωση ενός απομονωμένου pixel.

$$Q = \iint \sigma_f dS = \iint \varepsilon_s E_s + \varepsilon_a E_a dS \quad (4.14)$$

Το κεντρικό pixel με την πίσω πλευρά σχηματίζει έναν πυκνωτή όπου στους οπλισμούς του πυκνωτή υπάρχει φορτίο  $Q$  ενώ η τάση θεωρείται μονάδα. Η χωρητικότητα του πυκνωτή θα είναι αριθμητικά ίση με το φορτίο. Αφού  $C = Q/V$  και έχουμε υποθέσει ότι  $V = 1$  στο κεντρικό pixel. Ομοίως το συνολικό φορτίο που επάγεται σε κάθε pixel σχηματίζει με το κεντρικό έναν άλλο πυκνωτή. Το φορτίο που επάγεται θα είναι αριθμητικά ίσο με τη χωρητικότητα του πυκνωτή σε σχέση με το κεντρικό pixel.

Με αυτό τρόπο υπολογίζουμε τις χωρητικότητες του κεντρικού pixel σε σχέση με το υπόστρωμα (C0) καθώς και τις χωρητικότητες των γειτονικών pixel σε σχέση με το κεντρικό pixel (C11,C12,C13,C21,C23,C31,C32,C33). Τα pixel που βρίσκονται σε συμμετρικές θέσεις σε σχέση με το κεντρικό αναμένεται να έχουν ίσες χωρητικότητες. Για παράδειγμα το pixel που βρίσκεται ακριβώς πάνω από το κεντρικό στον άξονα  $y$  είναι συμμετρικό με το ακριβώς κάτω από το κεντρικό στον άξονα  $y$  επομένως αναμένεται να έχουν σχεδόν ίσες χωρητικότητες (C12=C32). Όμως λόγω αριθμητικού σφάλματος οι τιμές αυτές διαφέρουν ελάχιστα. Για τον λόγο αυτό υπολογίζουμε τη μέση τιμή των χωρητικότητων για τα pixel που είναι συμμετρικά ως προς το κεντρικό. Δηλαδή:

Τα γειτονικά pixel που βρίσκονται στην ευθεία παράλληλα με τον  $x$  άξονα είναι συμμετρικά μεταξύ τους

$$C_{01x} = \frac{C_{21} + C_{23}}{2} \quad (4.15)$$

Τα γειτονικά pixel που βρίσκονται στην ευθεία παράλληλα με τον  $y$  άξονα είναι συμμετρικά μεταξύ τους

$$C_{01y} = \frac{C_{12} + C_{32}}{2} \quad (4.16)$$

Τα διαγώνια γειτονικά pixel είναι όλα συμμετρικά ως προς το κεντρικό

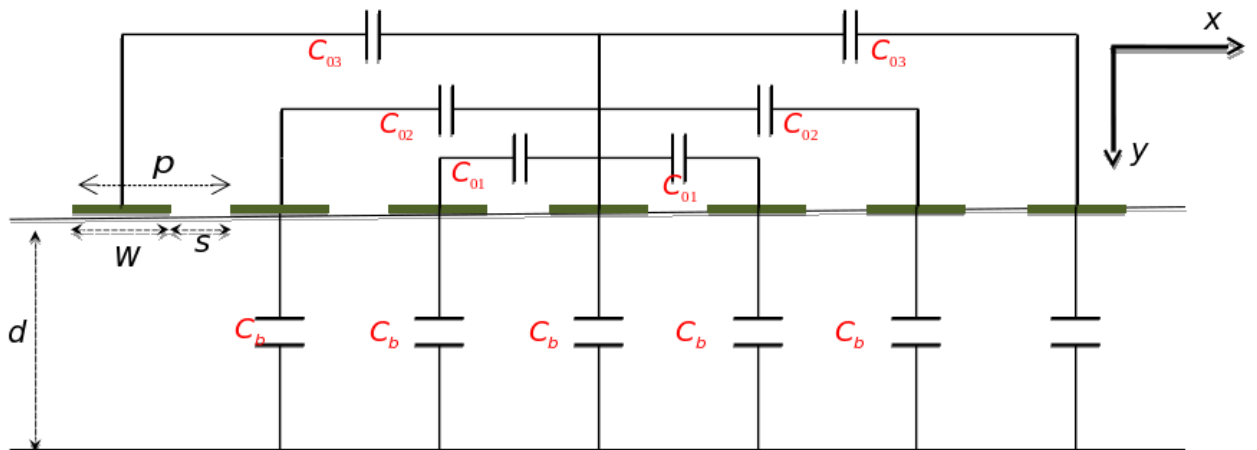
$$C_{02} = \frac{C_{11} + C_{13} + C_{31} + C_{33}}{4} \quad (4.17)$$

Στην περίπτωση που τα pixel θεωρηθούν τετράγωνα τότε τα γειτονικά pixel που βρίσκονται στην ευθεία είτε παράλληλα με το  $x$  είτε με το  $y$  είναι συμμετρικά μεταξύ τους. Έτσι θα μπορούσαμε να πάρουμε μια μέση τιμή και για τα 4

$$C_{01} = \frac{C_{12} + C_{21} + C_{23} + C_{32}}{4} \quad (4.18)$$

#### 4.2.2 Ανιχνευτές με επίπεδη γεωμετρία τύπου strip

Με παρόμοιο τρόπο μπορούμε να λύσουμε την εξίσωση Laplace για διάταξη ανιχνευτή μικρο-λωρίδων. Μια τέτοια διάταξη φαίνεται στην εικόνα 4.4.



Σχήμα 4.4: Διάταξη μικρο-λωρίδων (strip) με χωρητικότητες ανάμεσα σε γειτονικές λωρίδες.  $C_{01}$  οι χωρητικότητες μεταξύ του κεντρικού και των πρώτων γειτονικών λωρίδων,  $C_{02}$  μεταξύ των κεντρικών και των δεύτερων γειτονικών λωρίδων,  $C_{03}$  μεταξύ του κεντρικού και των τρίτων γειτονικών λωρίδων,  $C_b$  η χωρητικότητα μεταξύ της κάθε λωρίδας και της πίσω πλευράς,  $W$  είναι το πλάτος της κάθε λωρίδας,  $S$  η απόσταση μεταξύ των λωρίδων,  $d$  το πάχος του ανιχνευτή.

Όπως και στη προηγούμενη περίπτωση θεωρούμε ότι δεν υπάρχουν στατικά φορτία στον όγκο του ανιχνευτή. Επίσης λόγω συμμετρίας το πρόβλημα απλοποιείται στις δύο διαστάσεις αφού κατά μήκος των λωρίδων θα ισχύουν τα ίδια φαινόμενα. Η εξίσωση Laplace στις δύο διαστάσεις μπορεί να γραφεί στη μορφή

$$\nabla^2 V(x, z) = 0 \quad (4.19)$$

Οι αντίστοιχες συνοριακές συνθήκες θα είναι:

$$V_L(x, 0) = V_0 \quad \} \quad \text{κεντρική μικρο-λωρίδα} \quad (4.20)$$

$$= 0 \quad \} \quad \text{γειτονικές μικρο-λωρίδες} \quad (4.21)$$

$$\varepsilon_\alpha \varepsilon_0 \frac{\partial V_L}{\partial z} \Big|_{z=0^+} - \varepsilon_\alpha \varepsilon_0 \frac{\partial V_L}{\partial z} \Big|_{z=0^-} = 0 \quad \} \quad \text{μεταξύ δύο μικρο-λωρίδων} \quad (4.22)$$

Οι άξονες  $x, z$  έχουν τη φορά που φαίνεται στο σχήμα 4.4. Η επιφάνεια για  $z = 0$  αντιστοιχεί στην επιφάνεια των μικρο-λωρίδων. Ο άξονας  $x$  χωρίζονται διακριτά σημεία ενώ ο  $z$  θεωρείται συνεχής, Εφαρμόζοντας ένα μετασχηματισμό Fourier στη διεύθυνση  $x$  προκύπτει η παρακάτω διαφορική εξίσωση.

$$\begin{aligned} \nabla^2 V(x, z) &= 0 \xrightarrow{\mathcal{F}} \\ \frac{\partial^2 V(k_x, z)}{\partial z^2} &= k_x^2 V(k_x, z) \end{aligned} \quad (4.23)$$

Όπου  $V_L(k_x, z)$  το δυναμικό στο χώρο του Fourier και  $k_x$  οι αντίστοιχες συντεταγμένες στο χώρο του Fourier. Η παραπάνω διαφορική εξίσωση θα έχει λύσεις της μορφής

$$V_L(k_x, z) = A e^{-k_x z} + B e^{k_x z} \quad (4.24)$$

Όπου  $V_L(k_x, z)$  το δυναμικό στο χώρο του Fourier και  $k_x$  οι αντίστοιχες συντεταγμένες στο χώρο του Fourier. Εφαρμόζοντας τις συνοριακές συνθήκες 4.34 στη 4.24 προκύπτει

$$z = 0 \quad V_L = A + B \quad (4.25)$$

$$z = W \quad 0 = A e^{-k_x W} + B e^{k_x W} \quad \frac{A}{B} = -e^{2k_x W} \quad (4.26)$$

Όπου προκύπτει ότι:

$$B = \frac{V_L e^{-k_x W}}{e^{-k_x W} - e^{k_x W}} \quad A = \frac{V_L e^{k_x W}}{e^{-k_x W} - e^{k_x W}} \quad (4.27)$$

Άρα τελικά η λύση της διαφορική εξίσωσης θα έχει τη μορφή

$$V_L(k_x, z) = V_L(k_x) \frac{\sinh[k_x(w-z)]}{\sinh(k_x w)} \quad (4.28)$$

Διαφορίζοντας την 4.28 προκύπτει η σχέση

$$E_{zL} = k_x V_L(k_x) \frac{\cosh[w-z]}{\sinh(k_x w)} \quad (4.29)$$

Όπου για  $z = 0$  προκύπτει ότι

$$E_{zL} = k_x V_L(k_x) \frac{e^{k_x w} + e^{-k_x w}}{e^{k_x w} - e^{-k_x w}} \quad (4.30)$$

Όπως και στην περίπτωση των pixel ακολουθούμε μια αυτοσυνεπή μέθοδο. Αρχικά ορίζουμε μια αρχική τιμή για το δυναμικό. Ως αρχική τιμή στο χώρο που δεν

καλύπτεται από λωρίδες ορίζουμε μια συνάρτηση της μορφής 4.31 σύμφωνα με τις συνθήκες του προβλήματος.

$$V = 1 - \frac{x}{L} \quad x = [0, \dots, L] \quad (4.31)$$

Όπου το  $L$  είναι η απόσταση μεταξύ των μικρο-λωρίδων. Στην περιοχή των λωρίδων ορίζουμε το δυναμικό σύμφωνα με τις συνοριακές συνθήκες που ορίσαμε προηγουμένως

$$V_L(x, 0) = 1 \quad \} \quad \text{κεντρική μικρο-λωρίδα} \quad (4.32)$$

$$= 0 \quad \} \quad \text{γειτονικές μικρο-λωρίδες} \quad (4.33)$$

$$\varepsilon_\alpha \varepsilon_0 E_s - \varepsilon_\alpha \varepsilon_0 E_a = 0 \quad \} \quad \text{μεταξύ δύο μικρο-λωρίδων} \quad (4.34)$$

Ακολουθώντας επαναληπτικά την ίδια μέθοδο που ακολουθήσαμε στα pixel μέχρι η μέθοδος να συγκλίνει καταφέρνουμε να υπολογίσουμε το φορτίο στη κεντρική λωρίδα καθώς και αυτό που επάγεται στην πίσω πλευρά και στις γειτονικές λωρίδες. Το φορτίο θα είναι αριθμητικό ίσο με τη χωρητικότητα της κεντρικής λωρίδας σε σχέση με το υπόστρωμα και αντίστοιχα το φορτίο που επάγεται στις γειτονικές λωρίδες θα είναι αριθμητικά ίσο με τη χωρητικότητα σε σχέση με τη κεντρική λωρίδα.

Οι γειτονικές λωρίδες που βρίσκονται σε συμμετρικές θέσεις σε σχέση με τη κεντρική αναμένουμε να έχουν τις ίδιες τιμές στη χωρητικότητα αφού το επαγόμενο φορτίο θα είναι σχεδόν ίσο. Λόγω όμως αριθμητικού σφάλματος αυτές οι τιμές μπορεί να διαφέρουν ελάχιστα. Για το λόγο αυτό υπολογίζουμε τη μέση τιμή των χωρητικότητων για τις συμμετρικές λωρίδες.

Αν για παράδειγμα έχουμε 5 λωρίδες με την 3 να είναι η κεντρική. Και έχουμε υπολογίσει τις χωρητικότητες για τη κεντρική ( $C_3$ ), της αμέσως γειτονικές ( $C_2, C_4$ ) και τις επόμενες γειτονικές ( $C_1, C_5$ ) τότε οι μέσες τιμές θα είναι

$$C_{01} = \frac{C_2 + C_4}{2} \quad \text{αμέσως γειτονικές} \quad (4.35)$$

$$C_{02} = \frac{C_2 + C_4}{2} \quad \text{επόμενες γειτονικές} \quad (4.36)$$

### 4.3 Αλγόριθμος επίλυσης εξίσωσης Laplace

Εφόσον ορίσαμε τη μεθοδολογία της αριθμητικής επίλυσης της εξίσωσης Laplace, θα ορίσουμε σε αυτή την ενότητα έναν αλγόριθμό ώστε να είναι δυνατόν αυτή η μέθοδος να υλοποιηθεί σε ένα πρόγραμμα το οποίο θα υπολογίζει της χωρητικότητες. Για να υλοποιήσουμε τη παραπάνω μέθοδο σε ένα πρόγραμμα θα πρέπει να ακολουθήσουμε τα εξής βήματα.

1. Αρχικά όπως κάθε πρόβλημα ηλεκτροστατικής σχεδιάζουμε το σχήμα του προβλήματος. Θα ορίσουμε έναν πίνακα δύο διαστάσεων ( $V[i, j]$ ) ο οποίος θα αναπαριστά την επιφάνεια των pixel και πάνω σε αυτόν θα γίνουν οι υπολογισμοί. Μέσα στον πίνακα αυτό ορίζουμε υποπίνακες οι οποίοι θα αντιστοιχούν στη θέση κάθε pixel.

Ορίζουμε αρχικά μια παράμετρο με όνομα (*resolution*). Η μεταβλητή αυτή καθορίζει την ανάλυση με την οποία θα κάνουμε τους υπολογισμούς καθώς και το

μέγεθος του πίνακα. Δηλαδή το πλήθος των σημείων που θα χωριστεί η επιφάνεια των pixel ώστε να γίνουν οι υπολογισμοί. Μεγάλο *resolution* θα μας δώσει πολλά σημεία και μεγαλύτερη ανάλυση στον υπολογισμό όμως μπορεί να κάνει το πρόγραμμα πιο αργό και πιο απαιτητικό από θέμα επεξεργασίας. Ορίζουμε επίσης τις μεταβλητές που καθορίζουν τη γεωμετρία του προβλήματος δηλαδή:

- $W$  το πάχος του ανιχνευτή
- $l_x$  το πλάτος του κάθε pixel παράλληλα με τον άξονα  $x$
- $l_y$  το πλάτος του κάθε pixel παράλληλα με τον άξονα  $y$
- $s_x$  η απόσταση μεταξύ των pixel παράλληλα με τον άξονα  $x$
- $s_y$  η απόσταση του κάθε pixel παράλληλα με τον άξονα  $y$

Προκύπτουν 6 μεταβλητές ( $res, l_x, l_y, s_x, s_y, w$ ) οι οποίες καθορίζουν τη γεωμετρία του προβλήματος. Βάζοντας διάφορες τιμές σε αυτές τις μεταβλητές θα μπορούμε να υπολογίσουμε τη χωρητικότητα ανάλογα με τη γεωμετρία και να εκτιμήσουμε ποια γεωμετρία δίνει μικρότερη χωρητικότητα.

Για να "σχεδιάσουμε" τα pixel μέσα στον πίνακα ορίζουμε 6 μεταβλητές  $A1, A2, A3, A4, A5, A6$  για τον άξονα  $x$  και 6 για τον άξονα  $y$  ( $B1, B2, B3, B4, B5, B6$ ). Οι μεταβλητές αυτές καθορίζουν τα όρια των υπό-πινάκων που αντιστοιχούν στα pixel. Εικόνα 4.5.

Είναι δηλαδή οι στήλες και οι γραμμές του πίνακα που ο χώρος ανάμεσα τους θα έχει τις τιμές που αντιστοιχούν σε pixel. Στις περιοχές αυτές του πίνακα θα βάλουμε τις αρχικές συνθήκες που αντιστοιχούν στα pixel.

2. Αφού προσομοιώσαμε την γεωμετρία σε μορφή δισδιάστατου πίνακα. Καθορίζουμε τις αρχικές τιμές του προβλήματος όπως ορίστηκαν στην προηγούμενη ενότητα. Βάζουμε στην περιοχή του πίνακα που αντιστοιχεί στο κεντρικό pixel ( $V[i][j] = 1$ ) και στα γειτονικά pixel ( $V[i][j] = 0$ ) ενώ στις ενδιάμεσες περιοχές ορίζουμε τις τιμές που αντιστοιχούν στην αρχική τιμή ( $V = \frac{1}{1+r}$ ). Οι απόσταση  $r$  καθορίζεται από τη θέση που βρίσκεται κάθε στοιχείο του πίνακα σε σχέση με τα όρια του κεντρικού pixel, εικόνα 4.6. Ένα στοιχείο  $(i, j)$  του πίνακα ( $V[i][j]$ ) θα έχει τιμή:

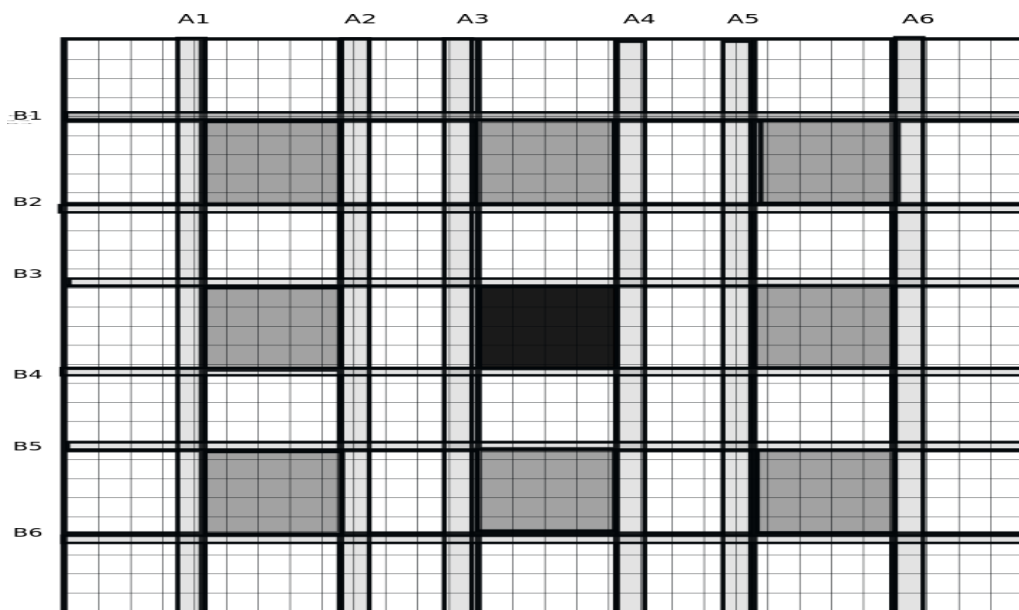
$$V[i][j] = \frac{1}{1+L} \quad \text{όπου} \quad L = \sqrt{(i - A3 + 1)^2 + (j - B3 + 1)^2} \quad (4.37)$$

Ορίζουμε με αυτό τον τρόπο μια διακριτή συνάρτηση όπου  $i = k\Delta$  και  $j = k\Delta$  με  $k = 0, 1, 2, \dots, N-1$  και  $\Delta = 1$

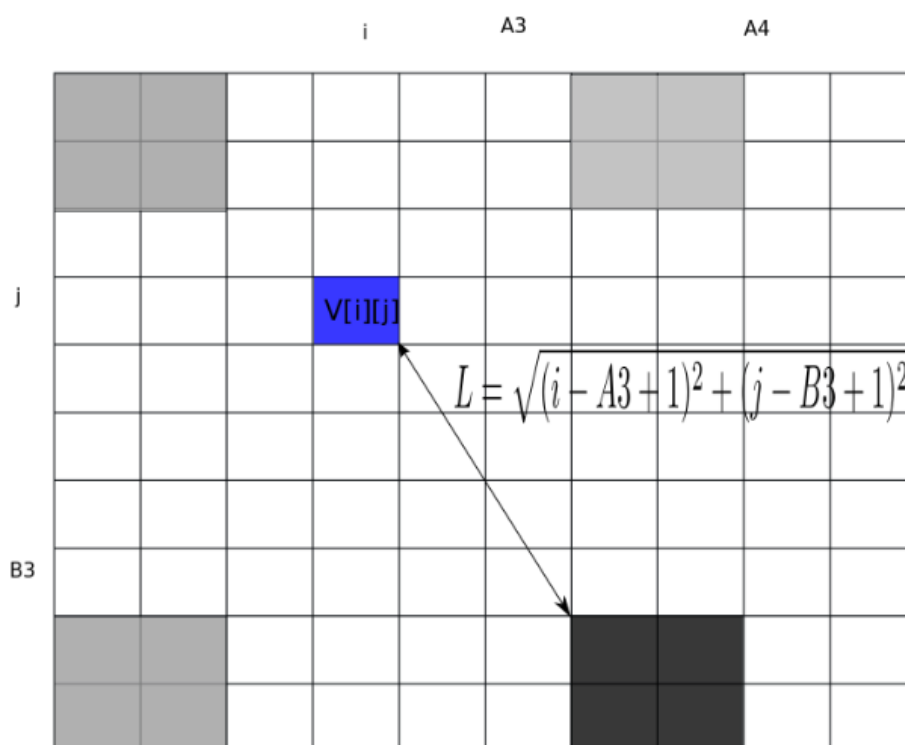
3. Ορίζουμε έναν μονοδιάστατο πίνακα  $F[i]$  όπου αποθηκεύουμε σε αυτόν τις τιμές των συντελεστών Fourier  $k_x, k_y$ . Οι τιμές αυτές και στους δύο άξονες θα είναι ίδιες. Για τις τιμές αυτές ισχύει  $k_x = k_y = 2\pi f_i$  όπου :

$$f_i = \frac{i}{N\Delta} \quad \text{και} \quad n = -\frac{N}{2}, \dots, \frac{N}{2} \quad (4.38)$$

Οι τιμές  $-\frac{N}{2}, \frac{N}{2}$  αντιστοιχούν στο κατώτερο και στο ανώτερο όριο της συχνότητας Nyquist. Όπως περιγράψαμε στο κεφάλαιο 3. Εικόνα 3.8



Σχήμα 4.5: Πίνακας δύο διαστάσεων. Μέσα στον πίνακα αυτό ορίζουμε τις θέσεις κάθε pixel. Με τις στήλες ( A1.....A6) και ( B1.....B6) καθορίζονται τα νοητά όρια κάθε pixel. Οι σκιασμένες περιοχές αντιστοιχούν στις θέσεις pixel.



Σχήμα 4.6: Η τιμή του δυναμικού ενός στοιχείου του πίνακα καθορίζεται από την θέση του στοιχείου σε σχέση με την θέση του υπό-πίνακα ο οποίος αντιστοιχεί στο κεντρικό pixel.

Στην περίπτωση μας που έχουμε ένα διδιάστατο πίνακα  $res \times res$  ο δείκτης των γραμμών και αντίστοιχα των στηλών του πίνακα θα έχει τιμές  $i = 0, 1, 2, \dots, res-1$  με  $\Delta = 1$ . Άρα η συχνότητα θα έχει τιμές

$$f = 0, \frac{1}{res}, \frac{2}{res}, \dots, \frac{res/2-1}{res}, \frac{1}{2}, \dots, \frac{res/2-1}{res}, \dots, -\frac{2}{res}, -\frac{1}{res} \quad (4.39)$$

Κάτι που αντιστοιχεί σε

$$k_y = k_x = 0, 2\pi \frac{1}{res}, \dots, 2\pi \frac{1}{2}, \dots, -2\pi \frac{1}{res} \quad (4.40)$$

Ο πίνακας  $F[i]$  θα έχει τιμές

$$F[i] = 0, 2\pi \frac{1}{res}, \dots, 2\pi \frac{1}{2}, \dots, -2\pi \frac{1}{res} \quad \text{όπου } i = 0, 1, 2, \dots, res-1 \quad (4.41)$$

4. Εφόσον έχουμε ορίσει την γεωμετρία και τις αρχικές τιμές. Εφαρμόζουμε έναν διδιάστατο μετασχηματισμό Fourier στο πίνακα ( $V[i][j]$ ) με μία υπορουτίνα όπως έχει περιγραφεί στο κεφάλαιο 3.

$$V[i][j] = FFT2D(V[i][j]) \quad (4.42)$$

5. Από το δυναμικό υπολογίζουμε το πεδίο στο εξωτερικό χώρο από τη σχέση 4.6 για  $W \rightarrow \infty$  και το δυναμικό στο εσωτερικό χώρο. Προκύπτουν δύο νέοι πίνακες με τις τιμές του πεδίου στον εξωτερικό χώρο ( $ES[i][j]$ ) και το πεδίο στο εσωτερικό ( $E1[i][j]$ ) οι οποίες θα βρίσκονται στο χώρο Fourier.

$$ES[i][j] = V[i][j] * R$$

$$E1[i][j] = V[i][j](1.0/d) \quad \text{για } R = 0$$

$$E1[i][j] = V[i][j] * R * (1.0 + \exp(-2.0 * R * d)) / (1.0 - \text{Math.exp}(-2.0 * R * d)) \quad \text{για } R > 0$$

Όπου  $R = \sqrt{F^2[i] + F^2[j]}$  αντιστοιχεί στη συνάρτηση  $F(k_x, k_y) = \sqrt{k_x^2 + k_y^2}$

6. Κάνουμε έναν αντίστροφο μετασχηματισμό Fourier ώστε να προκύψουν οι τιμές του ηλεκτρικού πεδίου

$$ES[i][j] = \frac{FFT2D^{-1}(ES[i][j])}{N}$$

$$E1[i][j] = \frac{FFT2D^{-1}(E1[i][j])}{N}$$

Όπου  $N = res * res$  σταθερά κανονικοποίησης του μετασχηματισμού Fourier.

7. Έχοντας υπολογίσει τις τιμές του πεδίου μπορούμε να υπολογίσουμε τις τιμές των χωρητικότητων σε κάθε pixel από τη σχέση. όπου η ολοκλήρωση μπορεί να γίνει αθροίζοντας σε όλα τα στοιχεία που αντιστοιχούν σε pixel. Θα προκύψουν 9 καινούργιες μεταβλητές ( $c0.c1, c2, c3, c4, c5, c6, c7, c8$ ).



8. Επειδή το πεδίο θα πρέπει να είναι συνεχές στην επιφάνεια που δεν καλύπτεται από pixel (δεν υπάρχουν στατικά φορτία) υπολογίζουμε τις νέες τιμές του πεδίου στις περιοχές που δεν καλύπτονται από pixel από τη σχέση 4.9. Ενώ κρατάμε τις τιμές στις περιοχές που αντιστοιχούν σε pixel όπως υπολογίστηκαν προηγουμένως. Προκύπτει τελικά ένας καινούργιος πίνακας  $E1[i][j]$ .
9. Εφαρμόζουμε ένα ευθύ μετασχηματισμό Fourier στο  $E1[i][j]$

$$E1[i][j] = FFT2D(E1[i][j]) \quad (4.43)$$

10. Από τη σχέση 4.7 υπολογίζουμε το καινούργιο δυναμικό ( $V'[i][j]$ ) στον χώρο του Fourier
11. Εφαρμόζουμε έναν ανάστροφο μετασχηματισμό Fourier στο  $V[i][j]$

$$V[i][j] = \frac{FFT2D^{-1}(V[i][j])}{N}$$

12. Στις περιοχές που καλύπτονται από pixel επιστρέφουμε τις τιμές στις αρχικές συνθήκες ( $V'[i][j]=1$ ) κεντρικό pixel και ( $V'[i][j]=0$ ) γειτονικά.
13. Ορίζουμε μια σχέση υπέρθεσης της αρχικής τιμής και της νέας.

$$V[i][j] = 0.8 * V[i][j] + 0.2 * V'[i][j] \quad (4.44)$$

14. Ελέγχουμε αν επιτυγχάνεται σύγκλιση της μεθόδου.

$$|V[i][j] - V'[i][j]| < \varepsilon \quad (4.45)$$

Αν δεν επιτυγχάνεται σύγκλιση επιστρέφουμε στο βήμα 3. Επαναλαμβάνουμε τα βήματα 3 έως 14 μέχρι να επιτευχθεί σύγκλιση. Αν συγκλίνει τότε έχει γίνει μια καλή εκτίμηση του πεδίου και των χωρητικότητας οπότε προχωράμε στα επόμενα βήματα.

15. Υπολογίζουμε τις μέσες τιμές των χωρητικότητας για τα συμμετρικά ως προς το κεντρικό pixel για να εξομαλυνθεί το αριθμητικό σφάλμα.

$$\begin{aligned} C01x &= (C4 + C5)/(-2.0) \\ C01y &= (C2 + C7)/(-2.0) \\ C02 &= (C1 + C3 + C6 + C8)/(-4.0) \end{aligned}$$

Το πρόσημο μείον προκύπτει από το ότι οι τιμές των χωρητικότητας των γειτονικών pixel σε σχέση με το κεντρικό έχουν αρνητική τιμή και αυτό οφείλεται στο ότι η χωρητικότητα είναι αριθμητικά ίση με το φορτίο. Στα γειτονικά pixel υπάρχει αρνητικό επαγόμενο φορτίο  $-Q$  οπότε και η χωρητικότητα η οποία προκύπτει από το φορτίο θα έχει αρνητικό πρόσημο.

16. Υπολογίζουμε την ολική τιμή της χωρητικότητας για το κεντρικό pixel πολλαπλασιάζοντας με την διηλεκτρική σταθερά  $\epsilon_0$  η οποία έχει παραληφθεί από τους υπολογισμούς.

Ο μετασχηματισμός Fourier που μπορούμε να χρησιμοποιήσουμε θα είναι μια υπορουτίνα όπως περιγράφηκε στο κεφάλαιο 3. Λόγω συμμετρίας του προβλήματος θα μπορούσαμε να χρησιμοποιήσουμε σχετικά εύκολα οποιοδήποτε αλγόριθμο διαιρεί τα δεδομένα με το 2. Οπότε θα μπορούσαμε να επιλέξουμε ο πίνακας στον οποίο θα θέσουμε τις αρχικές τιμές να είναι  $(N \times N)$  όπου το  $N$  είναι δύναμη το δύο. Όπως είπαμε η μεταβλητή *res* (resolution) καθορίζει και το μέγεθος του πίνακα δηλαδή ο πίνακας θα έχει μέγεθος στην ουσία  $(res \times res)$ . Άρα αν χρησιμοποιήσουμε ένα αλγόριθμο που διαιρεί τα δεδομένα στα δύο μπορούμε να θέσουμε στη μεταβλητή (*resolution*) τις τιμές (π.χ 512 , 1024 , 2048).

Θα μπορούσαμε να χρησιμοποιήσουμε επίσης και οποιοδήποτε άλλο αλγόριθμο ο οποίος δεν απαιτεί την χρήση δεδομένων με δύναμη του 2. Όμως επειδή η συμμετρία του προβλήματος μας το επιτρέπει μπορούμε με μεγαλύτερη ευκολία να χρησιμοποιήσουμε έναν αλγόριθμο της δύναμης του 2.

Επίσης ο πίνακας που θα ορίσουμε τις αρχικές συνθήκες ( $V[i][j]$ ) θα έχει αρχικά πραγματικές τιμές. Όμως ο μετασχηματισμός Fourier θα μας δώσει μιγαδικές τιμές τις οποίες θα πρέπει να τις αποθηκεύσουμε με κάποιο τρόπο. Ένας τρόπος είναι να ορίσουμε έναν τρισδιάστατο πίνακα ( $V[i][j][k]$ ). Όπου το  $i$  και το  $j$  θα παίρνουν τιμές από το 1 μέχρι τη τιμή της μεταβλητής *res* ενώ το  $k$  θα παίρνει μόνο δυο τιμές 0, 1 ή 1, 2. Ανάλογα με το πώς ορίζει η γλώσσα προγραμματισμού που θα χρησιμοποιήσουμε την αρχή της αρίθμησης ενός πίνακα.

Με αυτό τον τρόπο προκύπτουν δύο διδιάστατοι πίνακες όπου στον ( $V[i][j][0]$ ) ορίζουμε τις πραγματικές τιμές ενώ στον ( $V[i][j][1]$ ) ορίζουμε τις φανταστικές τιμές. Άρα μπορούμε εξαρχής να ορίσουμε έναν πίνακα  $V[i][j][k]$  και επειδή οι αρχικές τιμές θα είναι πραγματικές να ορίσουμε τον πίνακα που θα αποθηκεύονται οι φανταστικές ( $V[i][j][1]$ ) να είναι παντού μηδέν.

Ο μετασχηματισμός Fourier ως είσοδο χρησιμοποιεί έναν μονοδιάστατο πίνακα με συνολικά  $(2 * res * res)$  στοιχεία ( $data[0, \dots, (2 \times res \times res - 1)]$ ) όπως αναφέρθηκε και στο κεφάλαιο 3. Οι τιμές του πίνακα παίρνουν διαδοχικά τις πραγματικές τιμές και τις φανταστικές. Δηλαδή στο πρώτο στοιχείο του πίνακα θα μπει η τιμή ( $data[0] = V[0][0][0]$ ) ενώ στο δεύτερο στοιχείο του πίνακα θα μπει η τιμή ( $data[1] = V[0][0][1]$ ) και με αυτό τρόπο γεμίζουμε διαδοχικά τον πίνακα (*data*). Πάνω σε αυτόν τον πίνακα θα γίνει όλη η διαδικασία του γρήγορου μετασχηματισμού Fourier. Τα αποτελέσματα θα αποθηκευτούν πάλι στον πίνακα (*data*). Άρα ως έξοδο από τον μετασχηματισμό Fourier θα πάρουμε τον πίνακα (*data*) με τις καινούργιες τιμές που προέκυψαν από τον μετασχηματισμό.

Έπειτα θα πρέπει να ξεχωρίσουμε πάλι τις πραγματικές και τις φανταστικές τιμές οι οποίες θα είναι σε διαδοχικές θέσεις του πίνακα του (*data*) και να τις αποθηκεύσουμε σε ένα πίνακα π.χ ( $FV[i][j][k]$ ). Για παράδειγμα το πρώτο στοιχείο του πίνακα θα αποθηκευθεί στη θέση ( $FV[0][0][0] = data[0]$ ) ενώ το αμέσως επόμενο στη θέση ( $FV[0][0][1] = data[1]$ ).

Με παρόμοιο τρόπο εργαζόμαστε και στη περίπτωση των μικρο-λωρίδων με την διαφορά ότι ο πίνακας είναι μονοδιάστατος και αντίστοιχα το πρόγραμμα οπου κάνει το μετασχηματισμό Fourier.

## Κεφάλαιο 5

# Πρόγραμμα γρήγορου υπολογισμού χωρητικότητας σε Java

### 5.1 Εισαγωγή

Στο παρόν κεφάλαιο γίνεται εφαρμογή της αριθμητικής επίλυσης της εξίσωσης Laplace με χρήση του μετασχηματισμού Fourier και όσον αναφέρθηκαν στα προηγούμενα κεφάλαια και η υλοποίησή τους σε πρόγραμμα για τον γρήγορο υπολογισμό των χωρητικότητας. Η υλοποίηση των προγραμμάτων έγινε στα πλαίσια της παρούσας διπλωματικής στο Ε.Κ.Ε.Φ.Ε Δημόκριτος.

### 5.2 Πρόγραμμα υπολογισμού χωρητικότητας σε Java

Παρακάτω παρουσιάζουμε το πρόγραμμα που υλοποιήθηκε σε Java. Το πρόγραμμα υπολογίζει τις χωρητικότητες για διάταξη τύπου strip ή pixel. Αποτελείται από τρεις κλάσεις pixel.java, strip.java και cap.java. Οι κλάσεις pixel.java και strip.java είναι υπεύθυνες για την εφαρμογή της αριθμητικής μεθόδου και τον υπολογισμό των χωρητικότητας ενώ η cap.java υλοποιεί Frame το οποίο κάνει πιο εύκολη την διεπαφή του χρήστη με το πρόγραμμα.

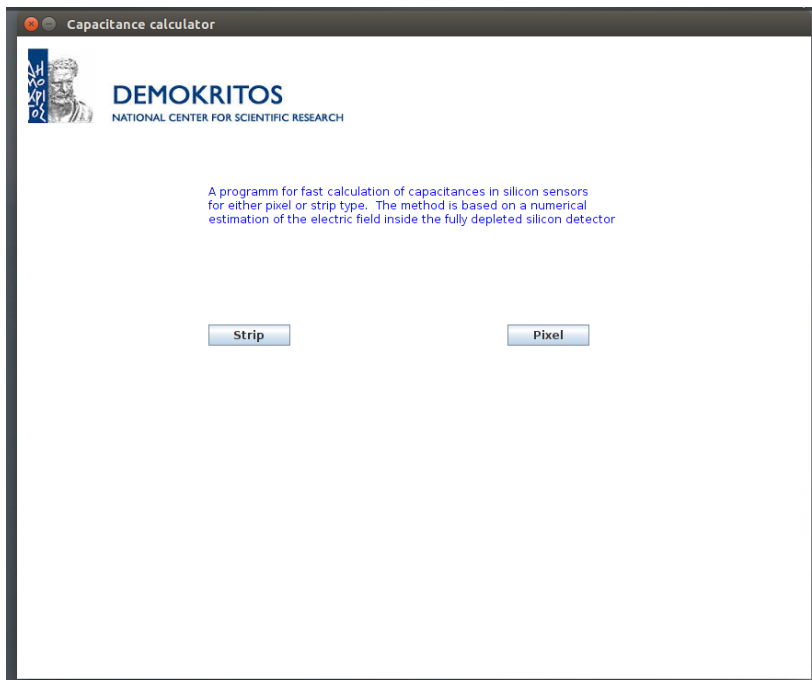
Αρχικά μπορούμε να επιλέξουμε την διάταξη για την οποία θέλουμε να κάνουμε τους υπολογισμούς. Μπορούμε να επιλέξουμε pixel ή strip πατώντας το αντίστοιχο κουμπί εικόνα 5.1.

Αν επιλέξουμε διάταξη pixel εμφανίζεται το παρακάτω παράθυρο (εικόνα 5.2). Στο παράθυρο αυτό καθορίζουμε την γεωμετρία κάθε pixel την απόσταση μεταξύ τους και το πάχος του ανιχνευτή θέτοντας τιμές στις διάφορες μεταβλητές. Αυτό μπορεί να γίνει επιλέγοντας "Inputs" και θέτοντας τις τιμές που θέλουμε στις αντίστοιχες μεταβλητές. Αν θέλουμε να επαναλάβουμε τους υπολογισμούς πολλές φορές για διάφορες διατάξεις μπορούμε να επιλέξουμε να εισάγουμε τις τιμές που θέλουμε να πάρουν οι μεταβλητές από κάποιο txt αρχείο επιλέγοντας "Inputs from file".

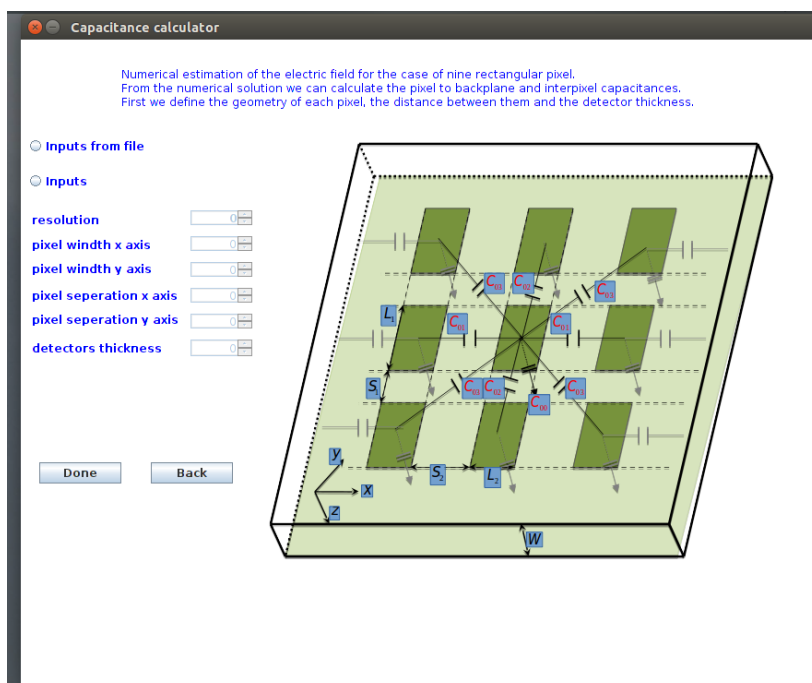
Αν επιλέξουμε διάταξη τύπου strip εμφανίζεται το παρακάτω παράθυρο (εικόνα 5.3) στο οποίο εισάγουμε τις τιμές για τις οποίες θέλουμε να γίνουν οι υπολογισμοί όπως και προηγουμένως.

Αφού καθορίσουμε τις τιμές για τις οποίες θέλουμε να γίνουν οι υπολογισμοί πατάμε το κουμπί "Done". Εμφανίζεται το παράθυρο εικόνα 5.4 στο οποίο θα εμφανίζονται τα αποτελέσματα. Πατώντας το κουμπί "Start" ξεκινάει ο υπολογισμός εικόνα 5.5. Σε κάθε κύκλο υπολογίζονται το φορτίο στο κεντρικό pixel ή strip καθώς και το επαγό-

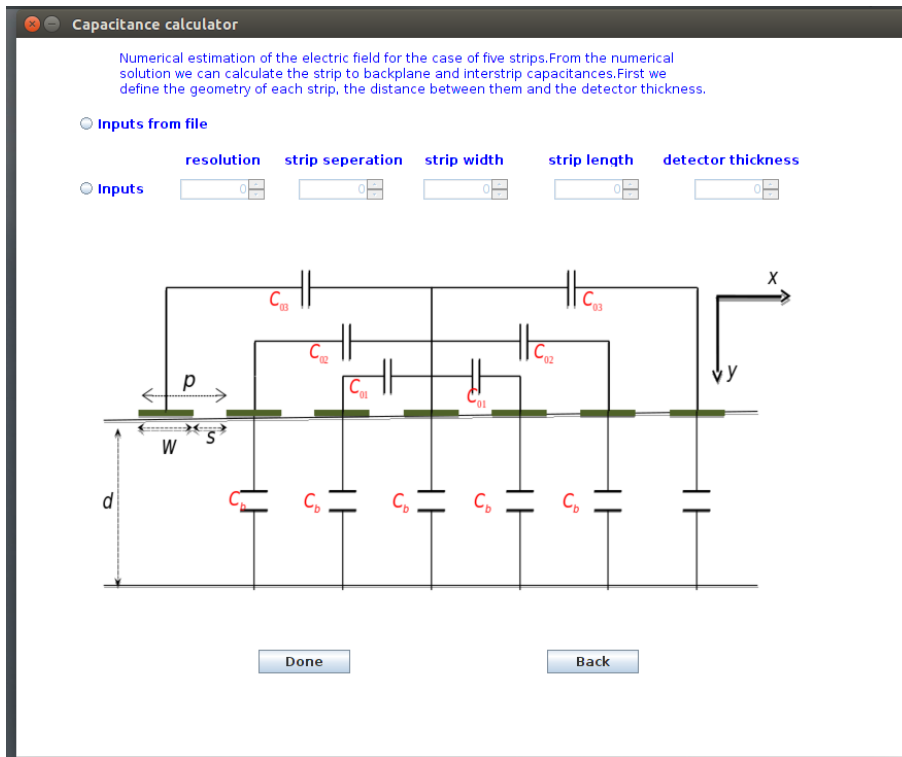
μενο φορτίο στα γειτονικά pixels ή strips. Το φορτίο θα είναι αριθμητικά ίσο με τη χωρητικότητα. Στο τέλος εφόσον ο αλγόριθμός συγκλίνει παρουσιάζονται τα τελικά αποτελέσματα εικόνα 5.6. Μπορούμε να σταματήσουμε την διαδικασία πατώντας το κουμπί "Stop", να καθαρίσουμε το παράθυρο πατώντας "Clear" και να μεταβούμε στα προηγούμενα παράθυρα πατώντας "Back".



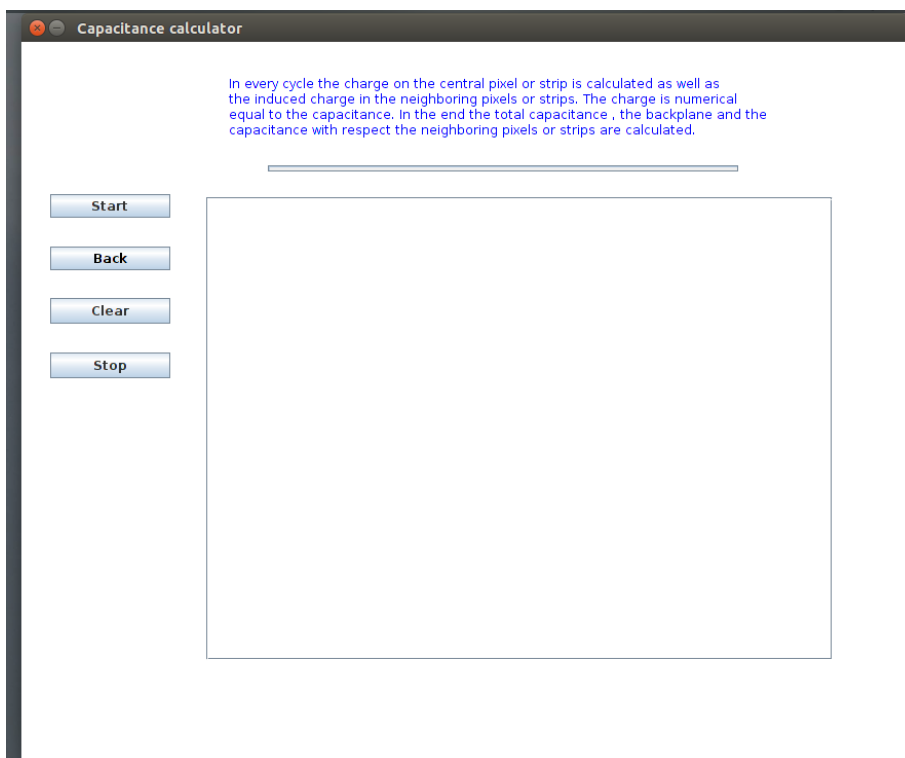
Σχήμα 5.1:



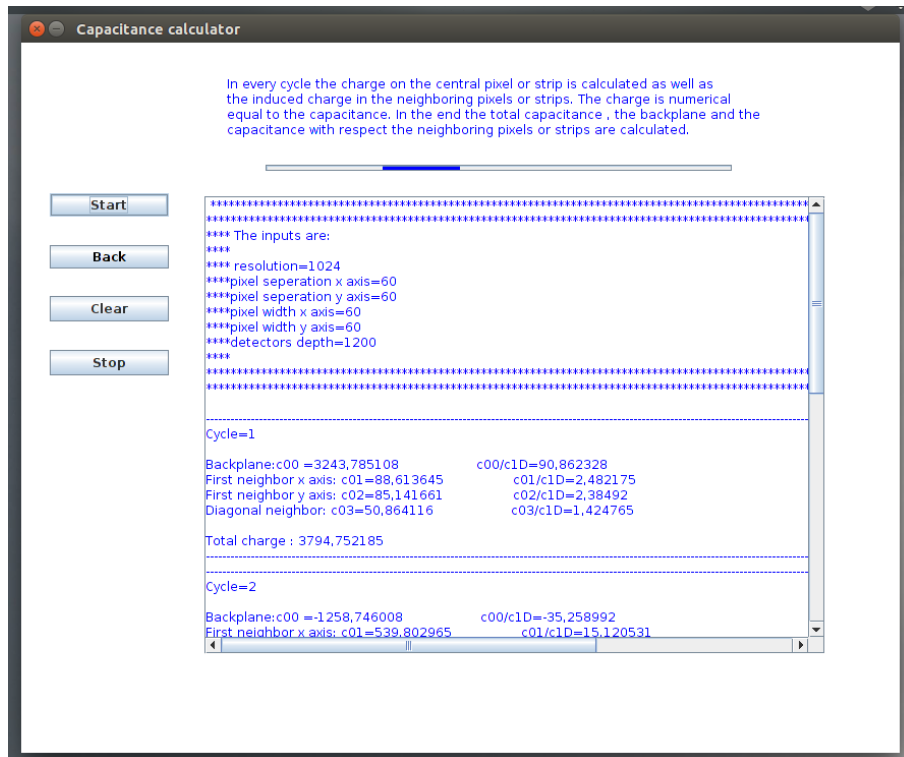
Σχήμα 5.2:



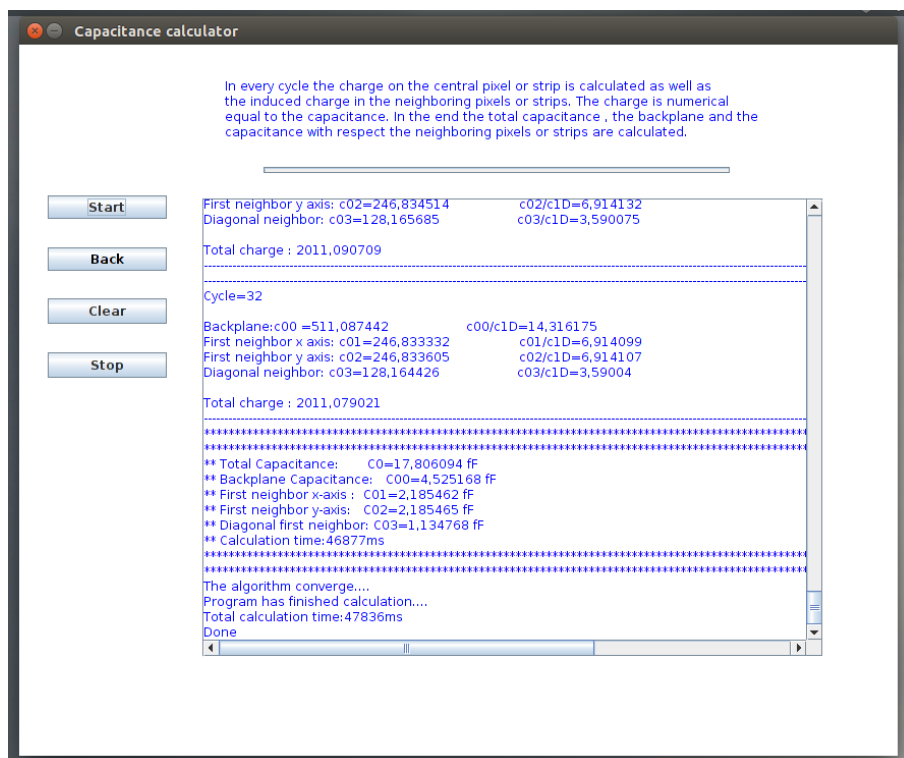
Σχήμα 5.3:



Σχήμα 5.4:



Σχήμα 5.5:



Σχήμα 5.6:

## Κεφάλαιο 6

# Αποτελέσματα της μεθόδου για διάφορες γεωμετρίες

### 6.1 Εισαγωγή

Στο παρόν κεφάλαιο εξετάζουμε τα αποτελέσματα που προέκυψαν για διάφορες γεωμετρίες μέσω της αριθμητικής μεθόδου που εξετάσαμε στο κεφάλαιο 4. Οι αναλογίες διαστάσεων των pixel και των strip που χρησιμοποιήθηκαν αντιστοιχούν στις διαστάσεις που έχουν επιλεγεί ως οι πιθανότερες για την φάση II της αναβάθμισης του ανιχνευτή τροχιών του πειράματος CMS η οποία θα ολοκληρωθεί το 2024. Η σωστή εκτίμηση των χωρητικότητων οδηγεί στο βέλτιστο σχεδιασμό τόσο του ανιχνευτή όσο και των εξωτερικών ηλεκτρονικών.

### 6.2 Διάταξη τύπου pixel

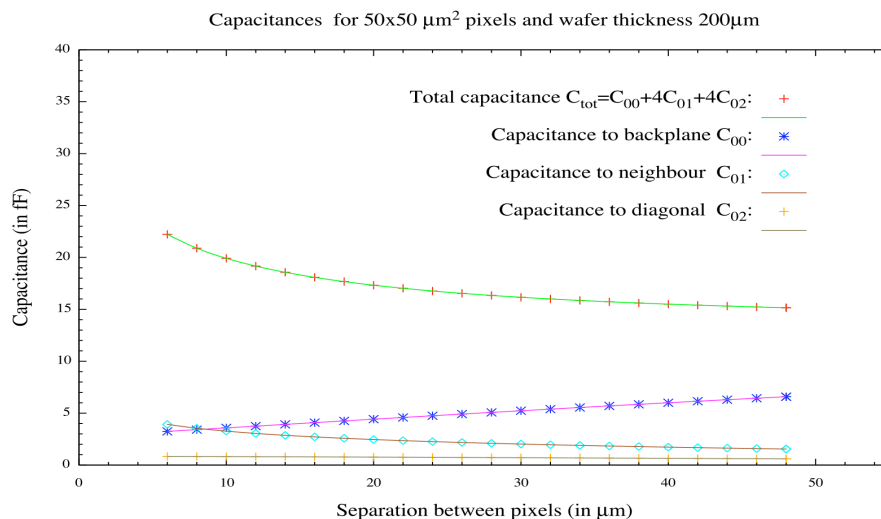
Οι γεωμετρίες των pixel για τα οποία έγιναν οι μετρήσεις είναι:

1.  $50 \times 50 \mu\text{m}$
2.  $100 \times 100 \mu\text{m}$
3.  $25 \times 100 \mu\text{m}$
4.  $50 \times 100 \mu\text{m}$

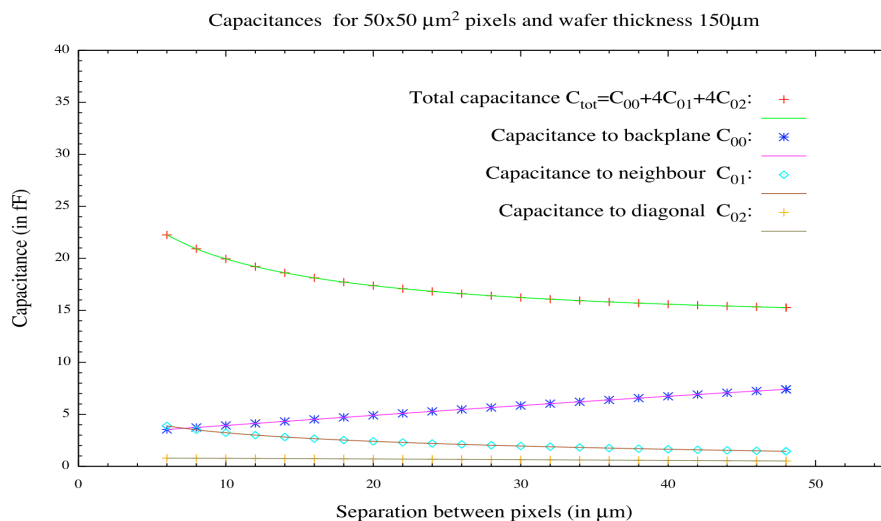
Για τη κάθε μια από τις παραπάνω γεωμετρίες pixel το πάχος του ανιχνευτή μεταβάλετε σύμφωνα με τις παρακάτω τιμές.

1.  $d = 100 \mu\text{m}$
2.  $d = 150 \mu\text{m}$
3.  $d = 200 \mu\text{m}$

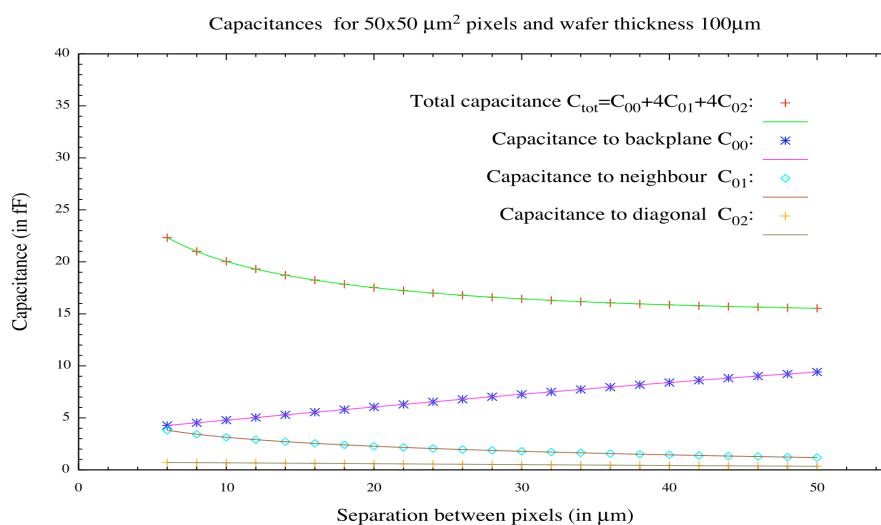
Για κάθε ένα σύνολο από τις παραπάνω μετρήσεις υπολογίστηκαν η συνολική χωρητικότητα, η χωρητικότητα σε σχέση με τα γειτονικά pixel και η χωρητικότητα σε σχέση με το πίσω επίπεδο για διαφορετικές τιμές της απόστασης μεταξύ των pixel. Προκύπτουν τα παρακάτω διαγράμματα.



(α')



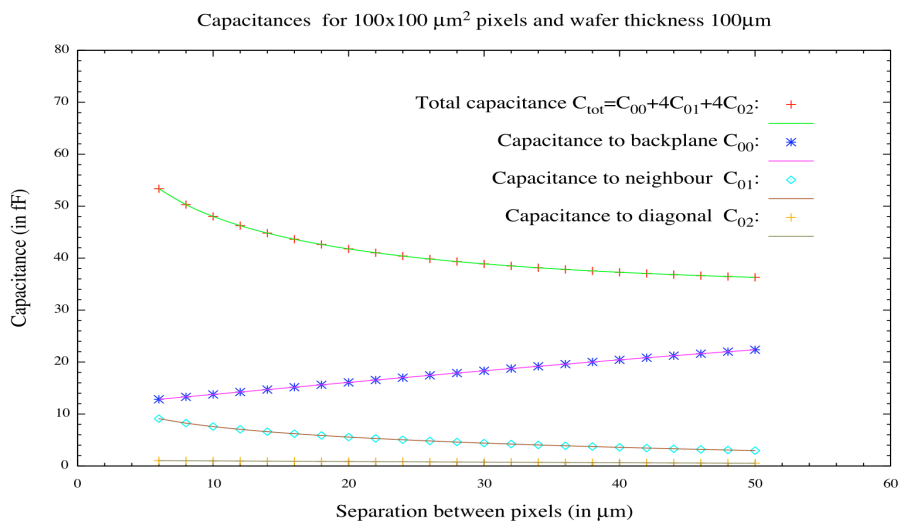
(β')



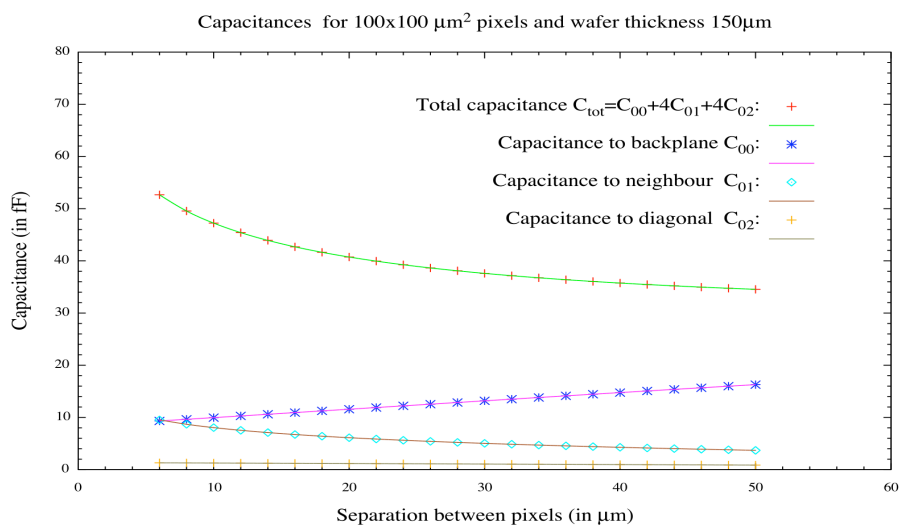
(γ')

Σχήμα 6.1: Γεωμετρία pixel  $50 \times 50$   
 Γεωμετρία pixel  $50 \times 50 \mu\text{m}$  για πάχη ανιχνευτή  $100, 150, 200 \mu\text{m}$

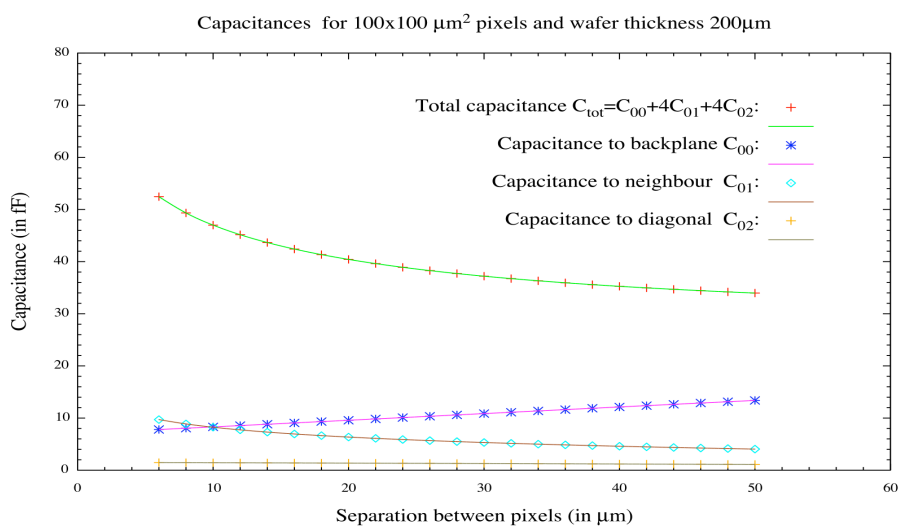




(α')

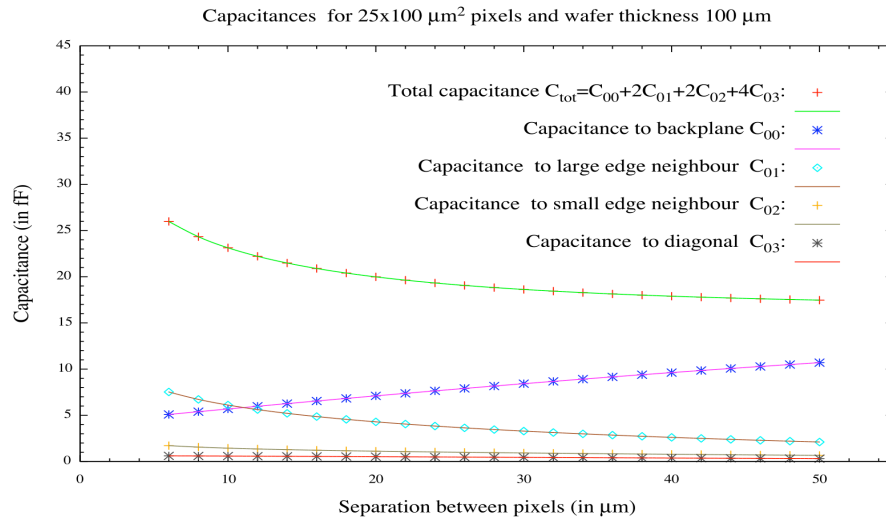


(β')

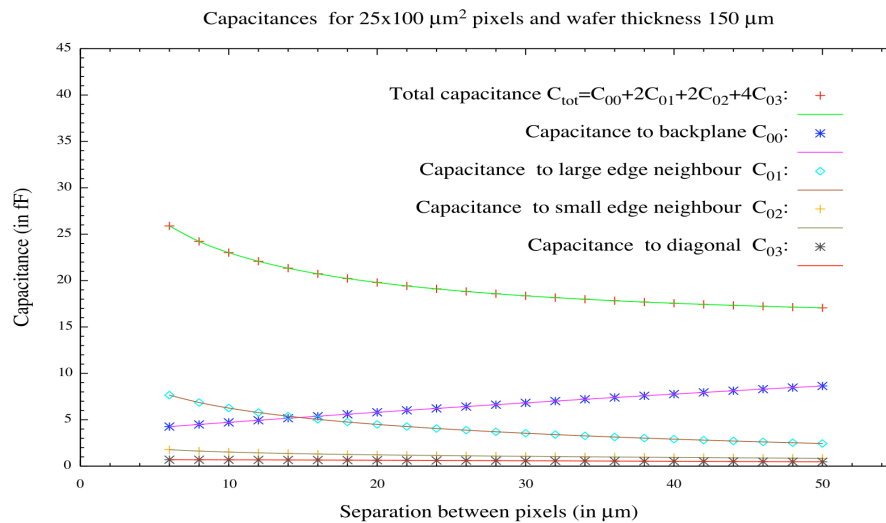


(γ')

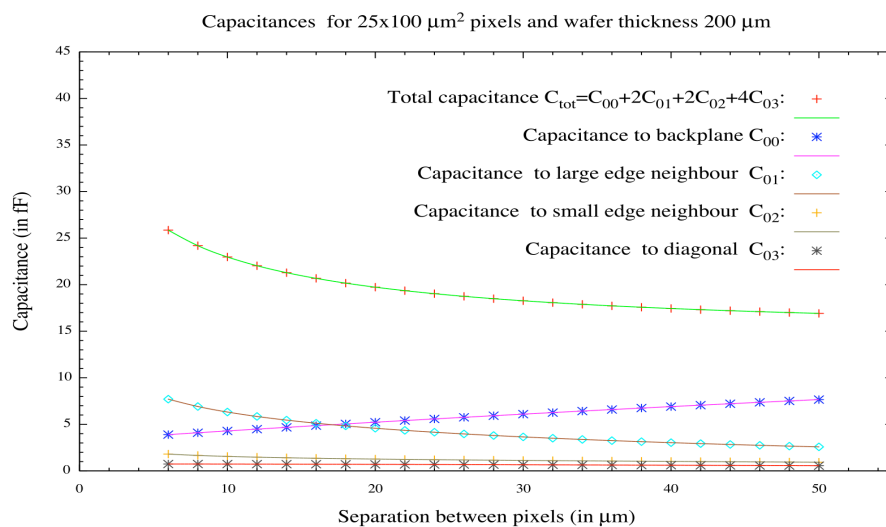
Σχήμα 6.2: Γεωμετρία pixel  $100 \times 100$   
Γεωμετρία pixel  $100 \times 100 \mu\text{m}$  για πάχη ανιευτή  $100, 150, 200 \mu\text{m}$



(α')

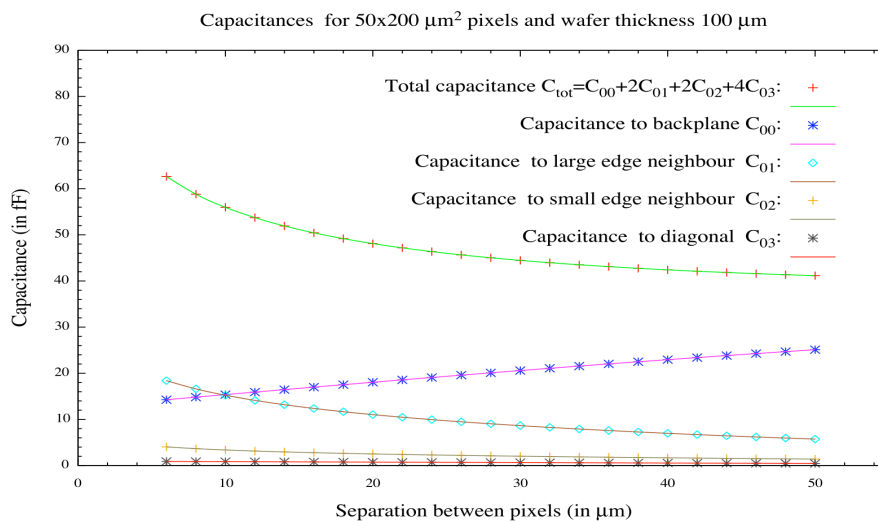


(β')

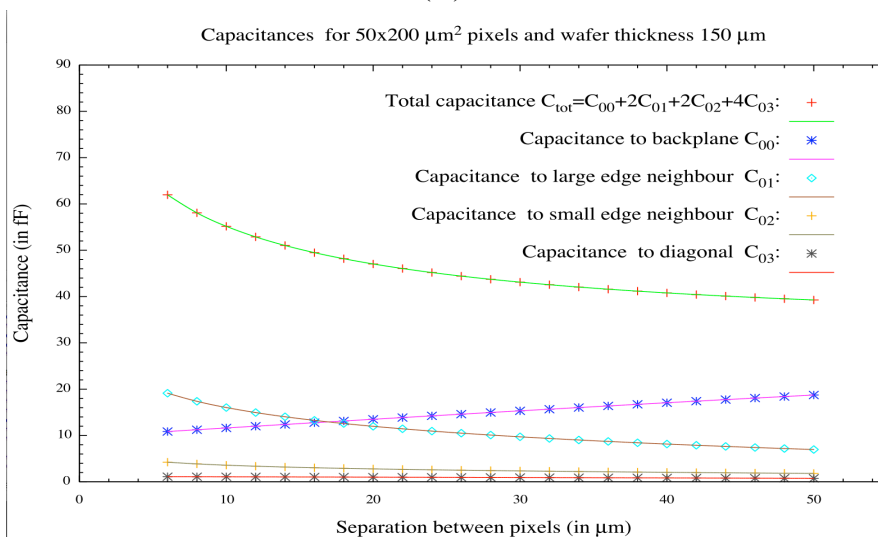


(γ')

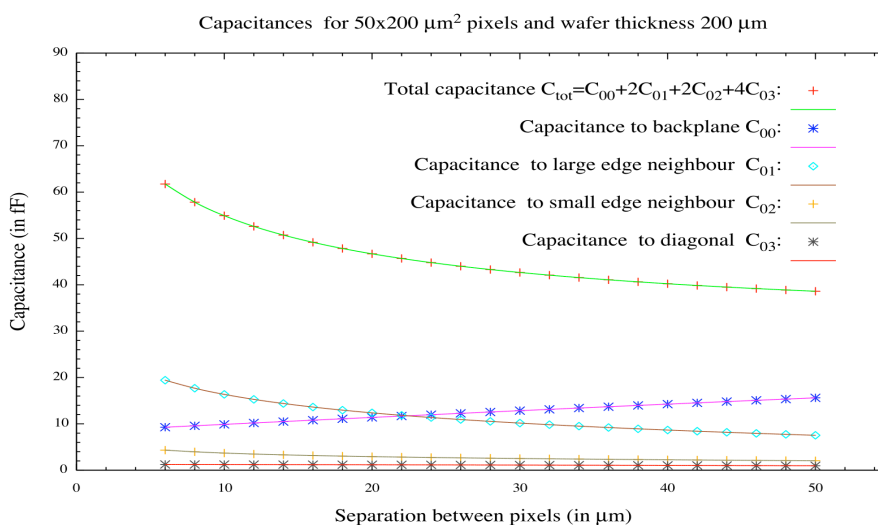
Σχήμα 6.3: Γεωμετρία pixel  $25 \times 100$   
 Γεωμετρία pixel  $25 \times 100 \mu\text{m}$  για πάχη ανιευτή  $100, 150, 200 \mu\text{m}$



(α')



(β')



(γ')

Σχήμα 6.4: Γεωμετρία pixel  $50 \times 200$   
Γεωμετρία pixel  $50 \times 200 \mu\text{m}$  για πάχη ανιχνευτή  $100, 150, 200 \mu\text{m}$

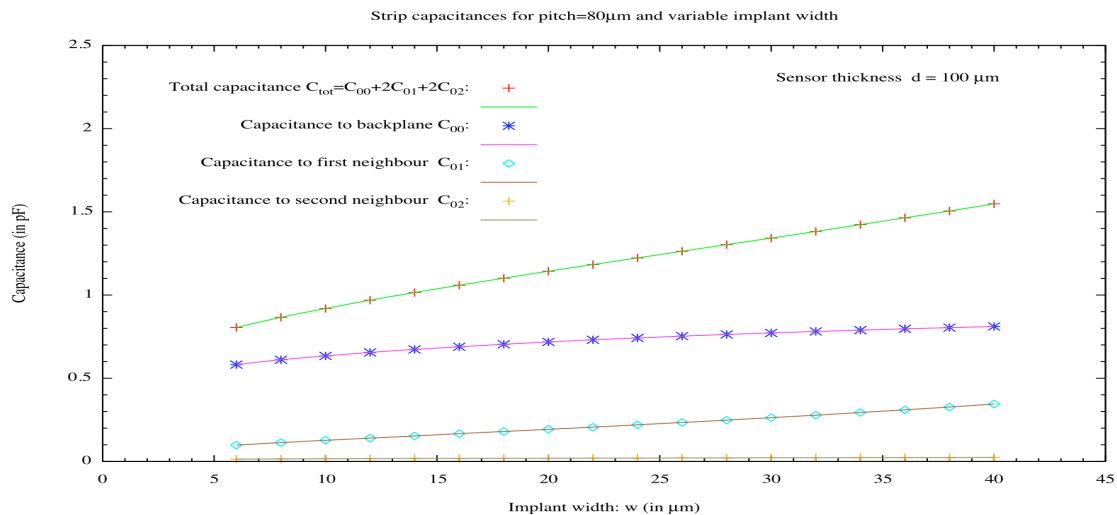
Παρατηρούμε ότι η συνολική χωρητικότητα μειώνεται όσο αυξάνεται η απόσταση, όπως και οι χωρητικότητες σε σχέση με τα γειτονικά pixel. Αρχικά για μικρές αποστάσεις μειώνεται με αρκετά γρήγορο ρυθμό, ενώ σε πιο μεγάλες αποστάσεις μειώνεται με μικρότερο ρυθμό, κάτι που ήταν αναμενόμενο αφού η χωρητικότητα εξαρτάται από το αντίστροφο της απόστασης ( $C = \epsilon \epsilon_0 \frac{A}{d}$ ).

Επίσης παρατηρούμε ότι η χωρητικότητα σε σχέση με το πίσω επίπεδο ( $C_{00}$ ) αυξάνεται με την απόσταση μεταξύ των pixel. Αυτό οφείλεται στο ότι η χωρητικότητα ( $C_{00}$ ) εξαρτάται από την διαφορά  $C_{00} = C_{tot} - 4C_{01} - 4C_{02}$  η οποία μειώνεται όσο αυξάνεται η απόσταση αφού η συνεισφορά των γειτονικών pixel γίνεται μικρότερη.

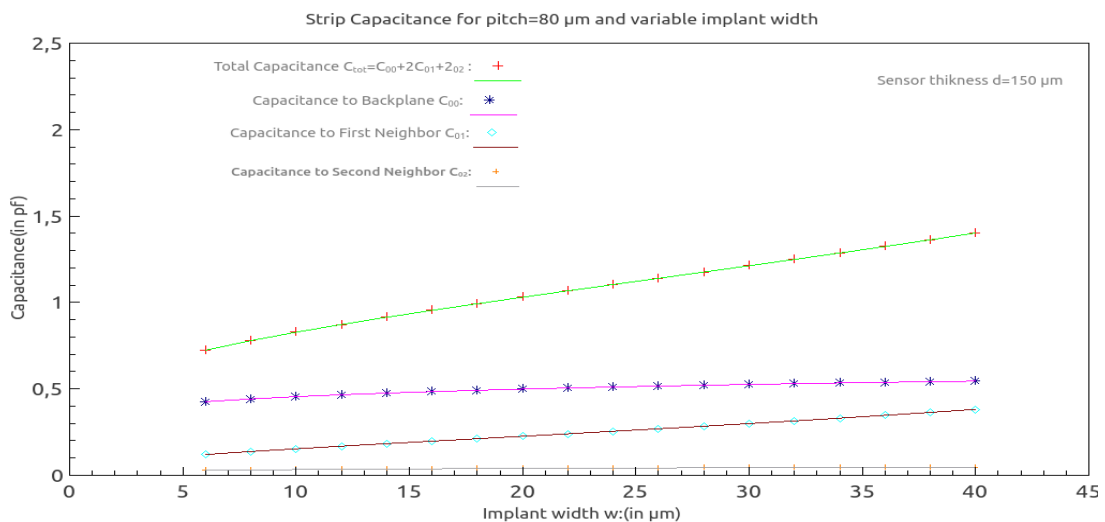
### 6.3 Διάταξη τύπου strip

Με τον ίδιο τρόπο υπολογίζουμε τις χωρητικότητες για επίπεδη γεωμετρία τύπου strip. Σε αυτή την περίπτωση κρατάμε σταθερή την απόσταση μεταξύ των λωρίδων και μεταβάλλουμε σταδιακά το πλάτος των λωρίδων ξεκινώντας από  $w=6$  έως  $40 \mu\text{m}$  και αυξάνοντας κατά  $2 \mu\text{m}$ . Οι υπολογισμοί έγιναν για μια τιμή του πάχους του ανιχνευτή  $d=100, d=150, d=200, \mu\text{m}$  διαγράμματα 6.5.

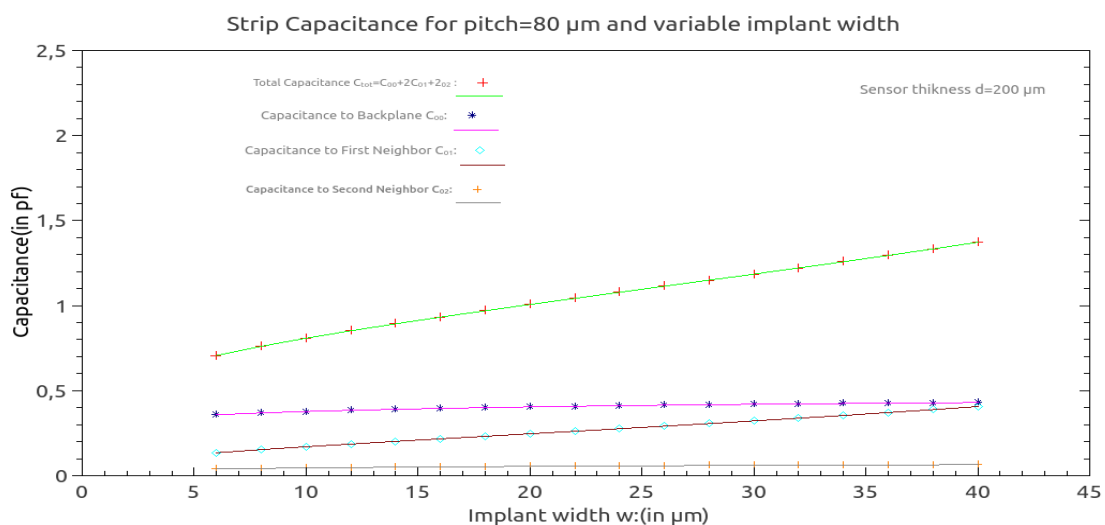
Παρατηρούμε ότι η συνολική χωρητικότητα καθώς και οι χωρητικότητες σε σχέση με τις γειτονικές λωρίδες αυξάνεται με το πλάτος κάθε λωρίδας εφόσον η απόσταση μεταξύ τους παραμένει σταθερή.



(α')



(β')



(γ')

Σχήμα 6.5: Γεωμετρία strip. Διαγράμματα χωρητικότητας για πάχη ανιχνευτή 100,150,200  $\mu\text{m}$  ως προς το πλάτος κάθε λωρίδας.

Υπολογισμός χωρητικότητας για γεωμετρία strip κρατώντας σταθερή τη μεταβλητή pitch 80  $\mu\text{m}$  (πλάτος λωρίδας + απόσταση μεταξύ τους). Μεταβάλεται το πλάτος καθώς και η απόσταση κρατώντας το άθροισμα τους σταθερό.

## 6.4 Συμπεράσματα

Οι χωρητικότητες που δημιουργούνται μεταξύ των διόδων του ανιχνευτή και μεταξύ των διόδων και του πίσω επιπέδου καθώς και το ρεύμα διαρροής καθορίζουν σε μεγάλο βαθμό τον λόγο σήματος προς θόρυβο ενός ανιχνευτή πυριτίου καθώς και την διακριτική τους ικανότητα, όπως αναφέραμε στο κεφάλαιο 2.

Με βάση την αριθμητική μέθοδο που περιγράψαμε στο κεφάλαιο 4 και περιγράφεται αναλυτικά στην βιβλιογραφία [1], [2] αναπτύξαμε ένα πρόγραμμα σε Java. Με το πρόγραμμα αυτό γίνεται γρήγορος υπολογισμός των χωρητικότητων ανάλογα με την γεωμετρία. Το 3 διαστάσεων πρόβλημα της επίλυσης της εξίσωσης Laplace σε ένα πλήρους απογυμνωμένο ανιχνευτή pixel απλοποιείται σε ένα δύο διαστάσεων πρόβλημα χρησιμοποιώντας το μετασχηματισμό Fourier και αντίστοιχα το 2 δύο διαστάσεων πρόβλημα σε ανιχνευτές strip απλοποιείται στη μια διάσταση. Με αυτό τον τρόπο καταφέρνουμε να υπολογίσουμε τις χωρητικότητες ανάλογα με τη γεωμετρία του ανιχνευτή σε μικρό χρονικό διάστημα.

Στον παρακάτω πίνακα φαίνεται ο χρόνος που χρειάστηκε το πρόγραμμα για διαφορετικές τιμές της μεταβλητής resolution η οποία καθορίζει και το μέγεθος του πίνακα με τον οποίο γίνονται οι υπολογισμοί. Παρατηρούμε για διάταξη τύπου pixel με  $resolution = 2048$  το οποίο αντιστοιχεί σε  $2048 \times 2048 = 4194304$  δεδομένα χρειάζονται  $207,163 \text{ sec} \approx 3,5 \text{ min}$ . Για διατάξεις τύπου strip όπου ο πίνακας είναι μονοδιάστατος ο χρόνος υπολογισμού είναι ιδιαίτερα μικρός.

Συμπεραίνουμε ότι με τη μέθοδο που αναφέραμε και το πρόγραμμα που υλοποιήσαμε μπορούμε να έχουμε μια αρκετά καλή εκτίμηση των χωρητικότητων σε σχέση με τη γεωμετρία του ανιχνευτή σε πολύ μικρό χρονικό διάστημα.

Ο υπολογισμός των χωρητικότητων μπορεί να γίνει και με άλλα προγράμματα προσομοίωσης όπως το TCAD-sentaurus <http://www.synopsys.com/Tools/TCAD/Pages/default.aspx>, όμως η διαδικασία αυτή απαιτεί περισσότερο χρόνο.

	resolution	Χρόνος (sec)
pixel	512	12,134
	1024	56,760
	2048	207,163
strip	1024	0,248
	16384	0,618
	65536	2,142
	524288	13,670

# Βιβλιογραφία

- [1] Kavadias S., Misiakos K., Loukas D., *Calculation of pixel detector capacitances through three dimensional numerical solution of the Laplace equation*. IEEE Transactions on (Volume:41 , Issue: 2 ) 397 - 401 1994.
- [2] Mousa J., Kyriakis A. , Loukas D. , *Potential Profile, Capacitances and Charge Collection in the Silicon Detector of the CMS Preshower*. N.C.S.R Demokritos Institute of Nuclear Physics. Greece, Agia Paraskevi, 2000
- [3] Spieler Helmuth, *Semiconductor Detector Systems*. Oxford University Press. Pinole, California, 2005.
- [4] Grupen Claus ,Shwartz Borris *Particle Detectors*. Cambridge University Press, United States of America, Second Edition 2008.
- [5] Knoll Glenn F, *Radiation Detection and Measurement*. Cambridge University Press, United States of America, Fourth Edition 2010.
- [6] Hall G., *Silicon detector and signal processing experiment*. Second ICFA school of instrumentation in elementary particle physics, Imperial College, Blackett Laboratory, London U.K. 1989.
- [7] Moser Hans-Günther , *Silicon detector systems in high energy physics*. Progress in Particle and Nuclear Physics 63 186–237 Max-Planck-Institut für Physik, Föhringer-Ring 6, D-80805 Munich, Germany, 2009 .
- [8] Hartmann Frank , *Silicon tracking detectors in high-energy physics*. Nuclear Instruments and Methods in Physics Research A 666 25–46 Institut für Experimentelle Kernphysik, KIT, Karlsruhe, Germany 2012.
- [9] Nomerotski Andrei , *Silicon detectors for tracking and vertexing*. Nuclear Instruments and Methods in Physics Research A 598 33–40 Particle Physics, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, UK 2009
- [10] Azzurri Paolo , *The CMS Silicon Strip Tracker*. Nuclear Instruments and Methods in Physics Research A 598 33–40 Scuola Normale Superiore, piazza dei Cavalieri 7, 56100 Pisa, Italy, Journal of Physics: Conference Series 41, 2006
- [11] C.M.S Collaboration *Technical proposal for the upgrade of the C.M.S. detector through 2020*. 2012
- [12] D. Abbaneo, D. Bortoletto, J. Christiansen, A. Dierlamm, T. Hreus, T. Liu, A. Mussgiller, A. Onnela, F. Palla, M. Pesaresi, N. Pozzobon, A. Ryd, G. Sguazzoni, L. Skinnari, K. Stenson, A. Tricomi, F. Vasey, S. Viret (... et al....) *Phase II Technical proposal - Tracker chapter*. 2014

- [13] Parashae N., *CMS Pixel Detector Upgrade*. CMS Collaboration, USA 20011.
- [14] CMS Collaboration *The CMS experiment at the CERN LHC*. Published by Institute of Physics Publishing and Sissa 2008
- [15] Veszpremi Viktor, *Operation and performance of the CMS tracker*. Wigner Research Centre for Physics, 1525 Budapest, P.O.Box 49, Hungary 2014.
- [16] The CMS Collaboration, *Description and performance of track and primary-vertex reconstruction with the CMS tracker*. 2014.
- [17] Turchetta R , *Spatial resolution of silicon microstrip detectors*. Nuclear Instruments and Methods in Physics Research A 335 (1993) 44-58, North-Holland, 1993
- [18] Ph. Bloch a , Y.-H. Chang b , A.E. Chen b , A. Cheremukhin c , N. Egorov d , A. Go b , S. Golubkov d , I. Golutvin c , S.R. Hou b,1 , K. Konjgov d , Y. Kozlov d , A. Kyriakis e , W.T. Lin b , D. Loukas e , A. Markou e , J. Mousa e , A. Peisert a , \*, A. Sidorov d , E. Tsoi e , N. Zamiatin c , E. Zubarev c *Silicon sensors for the CMS preshower detector*. a CERN, Geneva, Switzerland, b NCU, Chung-Li, Taiwan, c JINR, Dubna, Russia, d RIMST/ELMA, Zelenograd, Russia, e NCSR Demokritos, Athens, Greece, Nuclear Instruments and Methods in Physics Research A 479 (2002) 265–277 2002.
- [19] Bloch Ph., Peisert A., Chang Y.H., Chen, S. Hou A.E., Lin W.T., *Investigation of the charge collection for strongly irradiated Silicon strip detectors of the CMS ECAL Preshower*. Albuquerque, New Mexico 87131 U.S.A. 2000.
- [20] Gorne Grant, Hoferkamp Martin , Santistevan Geno, Seidel Sally, *Capacitance of Silicon Pixels*. Albuquerque, New Mexico 87131 U.S.A.
- [21] Cattaneo Paolo, *Capacitance calculation in a microstrip detector and its applications to signal processing*. Max-Planck-Institut für Physik und Astrophysik, Werner-Heisenberg-Institut für Physik, P. O. Box 401212, München, FRG 1990.
- [22] Cattaneo Paolo, *Capacitances in micro-strip detectors: a conformal mapping approach*. INFN Pavia, Via Bassi 6, Pavia, I-27100, Italy
- [23] Barberis, Cartiglia , LeVier, et al, *Capacitances in silicon microstrip detectors*. Nuclear Instruments and Methods in Physics Research A 342 90-95 Max-Planck-Institut für Physik, Föhringer-Ring 6, D-80805 Munich, North-Holland, 1994.
- [24] Rancoita, P. G. *Silicon detectors and elementary particle physics*. E P Division, CERN, CH-1211 Geneva 23, Switzerland 1983.
- [25] Rao K.R. , Kim D.N. , Hwang J.J. *Fast Fourier Transform: Algorithms and Applications*. Springer, Signals and Communication Technology, 2010
- [26] Press William H., Teukolsky Saul A., Vetterling William T., Flannery Brian P. , *Numerical Recipes*. Cambridge University Press Third Edition 2007.
- [27] Frigo Matteo , Johnson Steven G., *The Design and Implementation of FFTW3*. Published in Proc. IEEE, vol. 93, no. 2, pp. 216–231 , 2005.
- [28] Ramirez Robert W., *The FFT Fundamentals and Concepts*. SPS Information group, Tektronix, 1975.



- 
- [29] Jackson J.D., *Classical Electrodynamics*, Berkeley,California Second Edition, 1974.
- [30] Ramirez Robert W., *Effective Java*. Sun Microsystems, Second Edition 2008.
- [31] Enckel Bruce, *Thinking in java*. Fourth Edition 2006.



## Παράρτημα Α΄

# Αναλυτικά ο κώδικας του προγράμματος σε Java

Στην ενότητα αυτή παραθέτουμε τον κώδικα σε Java με τον οποίο υλοποιείται το πρόγραμμα για τον υπολογισμό των χωρητικότητων που περιγράψαμε στο κεφάλαιο 5. Το πρόγραμμα αποτελείται από τρεις κλάσεις pixel.java , strip.java , cap.java. Η πρώτη κλάση pixel.java είναι υπεύθυνη για τον υπολογισμό των χωρητικότητων σε περίπτωση επίπεδης γεωμετρίας τύπου pixel. Ενώ αντίστοιχα η κλάση strip.java είναι υπεύθυνη για τον υπολογισμό χωρητικότητων σε περίπτωση επίπεδης γεωμετρίας τύπου strip. Η κλάση cap.java υλοποιεί ένα Frame το οποίο κάνει πιο εύκολη την αλληλεπίδραση του χρήστη με το πρόγραμμα. Στην κλάση cap.java ο χρήστης θέτει τις τιμές εισόδου και έπειτα για αυτές τις τιμές καλείται η αντίστοιχη κλάση pixel.java, ή strip.java ανάλογα με την επιλογή του χρήστη και γίνονται οι υπολογισμοί.

Για τον μετασχηματισμό Fourier χρησιμοποιήθηκε η βιβλιοθήκη JTransforms η οποία είναι μια βιβλιοθήκη γραμμένη εξολοκλήρου σε java για την υλοποίηση του γρήγορου μετασχηματισμού Fourier. Η βιβλιοθήκη JTransforms είναι ανοιχτού κώδικα και μπορούμε ελεύθερα να την κατεβάσουμε και να τη χρησιμοποιήσουμε από την σελίδα <https://sites.google.com/site/piotrwendykier/software/jtransforms>. Η βιβλιοθήκη αυτή συνδυάζει όλους τους γνωστούς αλγόριθμους FFT και ανάλογα με το αριθμό των τιμών που επιλέγουμε να εισάγουμε επιλέγεται ο βέλτιστος αλγόριθμος. Μπορούμε να επιλέξουμε οποιαδήποτε πλήθος δεδομένων όμως αν το πλήθος είναι δύναμη του 2 τότε ο χρόνος υπολογισμού μειώνεται δραστηκιά.

Η κατασκευή του προγράμματος έγινε με το Eclipse IDE.

### A΄.0.1 Κλάση pixel.java

```
//*****//
//      LAPLACE SOLVER FOR FINITE DEPTH      //
//      STRUCTURE OF 9 RECTANGULAR JUNCTIONS //
//      INCLUDING THE CONTRIBUTION OF AIR.     //
//      SEPTEMBER 2014.                       //
//*****//
```

```
import java.io.*;
import java.math.RoundingMode;
```

```

import javax.swing.JTextArea;
import org.jtransforms.fft.DoubleFFT_2D;
import java.text.DecimalFormat;

public class pixel {
//*****//
// Definition of variables //
//*****//
    private static int A1,A2,A3,A4,A5,A6;
    private static int B1,B2,B3,B4,B5,B6;
    private static int res,lx,ly,sx,sy,d,cycle;
    private static int dim;
    private static boolean stop;

    private static double[][][] V;
    private static double[][][] V1;
    private static double[][][] fV;
    private static double[][][] ES;
    private static double[][][] E1;
    private static double[][][] fE1;
    private static double[][][] fES;
    private static double[][][] fV1;
// private static int[] mn=new int[2];
    private static double[] data;
    private static double[] Fk;
    private static double c0,c1,c2,c3,c4,c5,c6,c7,c8;
    private static double c02,c00,c1D,c00D,c02D, c01x,c01y,c01Dx,c01Dy;
    private static final double esi=11.9;
    private static final double eair=1.0;
    private static double total,backplane,neighborx,neighbory,diagonal;
    private static final double evacume=8.854*Math.pow(10, -3);
    static DecimalFormat df;

    public pixel(int resolution,int width_x_axis,int width_y_axis,
        int seperation_x_axis,int seperatio_y_axis
        ,int detector_thickness){

        res=resolution;
        lx=width_x_axis;
        ly=width_y_axis;
        sx=seperation_x_axis;
        sy=seperatio_y_axis;
        d=detector_thickness;
        dim=res;
        V=new double[dim][dim][2]; // Voltage initial guess
        V1=new double[dim][dim][2]; // Voltage new estimation
        fV=new double[dim][dim][2]; // Voltage Fourier space
        ES=new double[dim][dim][2]; // Electric field ambient
        E1=new double[dim][dim][2]; // Electric field inside silicon
        fES= new double[dim][dim][2]; // Electric field ambient Fourier space
        fE1=new double[dim][dim][2]; // Electric field silicon Fourier space
        fV1=new double[dim][dim][2]; // Voltage new estimation Fourier space
        data= new double[2*dim*dim]; // data is a real array of length twice the product

```

```

        // of these lengths, in which the data are stored as
        //in a multidimensional complex array:
        // real and imaginary parts of each element are in
        // consecutive locations
Fk=new double[dim];    // Fourier coefficients
}

public static void grindplacement(){
    //*****//
    // Placement of pixels in the grid //
    //*****//

    //nn[0]=res;
    //nn[1]=res;
    // 3 central pixel horizontal bounds
    A3=(res-lx)/2-1;
    A4=(res+lx)/2;
    // 3 upper pixel horizontal bounds
    A2=A3-sx+1;
    A1=A2-lx-1;
    // 3 down pixel horizontal bounds
    A5=A4+sx-1;
    A6=A5+lx+1;

    // 3 central pixel vertical bounds
    B3=(res-ly)/2-1;
    B4=(res+ly)/2;
    // 3 upper pixel vertical bounds
    B2=B3-sy+1;
    B1=B2-ly-1;
    // 3 down pixel vertical bounds
    B5=B4+sy-1;
    B6=B5+ly+1;

}

public static double[] kVectors(){
    //*****//
    //Calculation of k vectors. Fourier coefficients //
    //*****//
    double [] A=new double[(int)res];
    for(int i=1; i<=(res/2+1); i++){
        A[i-1]=(double) 2.0*Math.PI*(i-1)/res;;
    }
    int k=1;
    for(int i=(int)(res/2+2); i<=res; i++){
        A[i-1]=A[res/2-k];
        k++;
    }
    return A;

}

public static void pixelcapacitance(JTextArea Outputs){

```

```

Fk=kVectors();
    DecimalFormat df = new DecimalFormat("0.0000");
    df.setRoundingMode(RoundingMode.HALF_UP);
//Printing outputs in file
try{
    PrintStream out = new PrintStream(new FileOutputStream("pixel_output.txt",
        true));
    System.setOut(out);
    }
catch(FileNotFoundException e){

}

// Contractor of the DoubleFFT_2D for jtransforms library
DoubleFFT_2D fftDo = new DoubleFFT_2D(res,res);

//Time calculation

//call the method for placing the pixels postion in the grid
grindplacement();

System.out.println("*****"
+
"*****"+'\n');
System.out.println("*****"
+
"*****"+'\n');
System.out.println("****"+" "+'\n');
System.out.println("****"+" "+"The inputs are:"+'\n');
System.out.println("****"+" "+" "+'\n');
System.out.println("****"+" "+"resolution="+res+'\n'+ "****"
+"pixel seperation x axis="+sx+'\n'+ "****"+"pixel seperation y
axis="+sy+'\n'+ "****"
+"pixel width x axis="+lx+'\n'+ "****"+"pixel width y
axis="+ly+'\n'+ "****"+"detectors depth="+d+'\n');
System.out.println("****"+" "+" "+'\n');
System.out.println("*****"
+
"*****"+'\n');
System.out.println("*****"
+
"*****"+'\n');
System.out.println('\n');

Outputs.append("*****"
+
"*****"+'\n');
Outputs.append("*****"
+
"*****"+'\n');
Outputs.append("****"+" "+"The inputs are:"+'\n');
Outputs.append("****"+" "+" "+'\n');
Outputs.append("****"+" "+"resolution="+res+'\n'+ "****"

```

```

+"pixel seperation x axis="+sx+'\n'+ "****"+"pixel seperation y
  axis="+sy+'\n'+ "****"
+"pixel width x axis="+lx+'\n'+ "****"+"pixel width y
  axis="+ly+'\n'+ "****"+"detectors depth="+d+'\n');
Outputs.append("****" + " " + '\n');
Outputs.append("*****"
+
"*****" + '\n');
Outputs.append("*****"
+
"*****" + '\n');
Outputs.append(" " + '\n');

//*****//
// Hypothetical solution on surface //
//
//*****//
for(int i=0; i<res; i++){
  for(int j=0; j<res; j++){
    V[i][j][1]=0.0;
    //kentriko pixel V=1
    if((i>A3)&&(i<A4)&&(j>B3)&&(j<B4)){
      V[i][j][0]=1.0;
    }
    //periohx 1 pixel 1
    else if((i>=0)&&(i<=A3-1)&&(j>=0)&&(j<=B3)){
      double x=Math.sqrt(Math.pow((i-A3),2)+Math.pow((j-B3),2));
      V[i][j][0]=(double) 1.0/(1+x);
      if((i>=A1+1)&&(i<=A2-1)&&(j>=B1+1)&&(j<=B2-1)){
        V[i][j][0]=0.0;
      }
    }
    // periohx 3 pixel 3
    else if((i>=0)&&(i<=A3)&&(j>=B4)&&(j<res)){
      double x=Math.sqrt(Math.pow((i-A3),2)+Math.pow((j-B4),2));
      V[i][j][0]=(double) 1.0/(1+x);
      if((i>=A1+1)&&(i<=A2-1)&&(j>=B5+1)&&(j<=B6-1)){
        V[i][j][0]=0.0;
      }
    }
    // periohx 7 pixel 7
    else if((i>=A4+1)&&(i<res)&&(j>=0)&&(j<=B3)){
      double x=Math.sqrt(Math.pow((i-A4),2)+Math.pow((j-B3),2));
      V[i][j][0]=(double) 1.0/(1+x);
      if((i>=A5+1)&&(i<=A6-1)&&(j>=B1+1)&&(j<=B2-1)){
        V[i][j][0]=0.0;
      }
    }
    //periohx 9 pixel 9
    else if((i>=A4+1)&&(i<res)&&(j>=B4+1)&&(j<res)){
      double x=Math.sqrt(Math.pow((i-A4),2)+Math.pow((j-B4),2));
      V[i][j][0]=(double) 1.0/(1+x);
      if((i>=A5+1)&&(i<=A6-1)&&(j>=B5+1)&&(j<=B6-1)){
        V[i][j][0]=0.0;
      }
    }
  }
}

```

```

    }
    // perioxh 2 pixel 2
    else if((i>=0)&&(i<=A3)&&(j>=B3+1)&&(j<=B4)){
        double x=A3-i;
        V[i][j][0]=(double) 1.0/(1+x);
        if((i>=A1+1)&&(i<=A2-1)&&(j>=B3+1)&&(j<=B4-1)){
            V[i][j][0]=0.0;
        }
    }

    //perioxh 4 pixel 4
    else if((i>=A3+1)&&(i<=A4)&&(j>=0)&&(j<=B3)){
        double x=B3-j;
        V[i][j][0]=(double) 1.0/(1+x);
        if((i>=A3+1)&&(i<=A4-1)&&(j>=B1+1)&&(j<=B2-1)){
            V[i][j][0]=0.0;
        }
    }

    //perioxh 6 pixel 6
    else if((i>=A3+1)&&(i<=A4)&&(j>=B4+1)&&(j<res)){
        double x=j-B4;
        V[i][j][0]=(double) 1.0/(1+x);
        if((i>=A3+1)&&(i<=A4-1)&&(j>=B5+1)&&(j<=B6-1)){
            V[i][j][0]=0.0;
        }
    }

    //perioxh 8 pixel 8
    else if((i>=A4+1)&&(i<res)&&(j>=B3+1)&&(j<=B4)){
        double x=i-A4;
        V[i][j][0]=(double) 1.0/(1+x);
        if((i>=A5+1)&&(i<=A6-1)&&(j>=B3+1)&&(j<=B4-1)){
            V[i][j][0]=0.0;
        }
    }
}
}

//*****//
//*****//

boolean flag=true;    // check convergence
cycle=1;

long t1 =System.currentTimeMillis();
while(flag){

    stop=false;
    checkStop(stop);
    if(stop){break;}
}

```



```

//*****//
// Fourier Transform of hypothetical potential V           //
// FV potential in Fourier space                          //
//*****//

int m=0;
for(int i=0; i<res; i++){
  for(int j=0; j<res; j++){
    data[m]=V[i][j][0];
    data[m+1]=V[i][j][1];
    m+=2;
  }
}
//fourn(data,nm,1);
fftDo.complexForward(data);

for(int i=1; i<=2*res*res-1;i+=2){
  int l= (int) (i/(2*res)+1);

  int j=(int) (i-(l-1)*(2*res));

  int k=(int) ((j-1)/2+1);

  fV[k-1][l-1][0]=data[i-1];
  fV[k-1][l-1][1]=data[i];
}

Fk=kVectors();      // Set array of k vectors

//*****//
// From FV we calculate the electric field in Fourier space for the ambient
// (fES) and silicon (fE1) //
// for the ambient (fES) and silicon (fE1)
//
//*****//

for(int i=0; i<res; i++){
  for(int j=0; j<res; j++){
    double R=Math.sqrt(Math.pow(Fk[i], 2)+Math.pow(Fk[j], 2)); //R=sqrt(kx^2+ky^2)

    fES[i][j][0]=(double)fV[i][j][0]*R;
    fES[i][j][1]=(double)fV[i][j][1]*R;

    if(R==0.0){
      fE1[i][j][0]=(double)fV[i][j][0]*(1.0/d);
      fE1[i][j][1]=(double)fV[i][j][1]*(1.0/d);
    }
    else if(R>0.0){
      fE1[i][j][0]=(double)fV[i][j][0]*R*(1.0+Math.exp(-2.0*R*d))/(1.0-Math.exp(-2.0*R*d));

```

```

        fE1[i][j][1]=(double)fV[i][j][1]*R*(1.0+Math.exp(-2.0*R*d))/
            (1.0-Math.exp(-2.0*R*d));

    }

}

//*****//
// Inverse Fourier transform for fE1 and fES          //
// E1 ,ES is the electric field                       //
//*****//

m=0;
for(int i=0; i<res; i++){
    for(int j=0; j<res; j++){
        data[m]=fES[i][j][0];
        data[m+1]=fES[i][j][1];
        m+=2;
    }
}

//fourn(data,nn,-1);
fftDo.complexInverse(data, false);
for(int i=1; i<=2*res*res;i+=2){
    int l=i/(2*res)+1;
    int j=i-(l-1)*(2*res);
    int k=(j-1)/2+1;
    ES[k-1][l-1][0]=(double)data[i-1]*(1.0/(res*res)); //normalization
    ES[k-1][l-1][1]=(double)data[i]*(1.0/(res*res)); //normalization
}
m=0;
for(int i=0; i<res; i++){
    for(int j=0; j<res; j++){
        data[m]=fE1[i][j][0];
        data[m+1]=fE1[i][j][1];
        m+=2;
    }
}
//fourn(data,nn,-1);
fftDo.complexInverse(data, false);
for(int i=1; i<=2*res*res;i+=2){
    int l=(int)i/(2*res)+1;
    int j=(int)i-(l-1)*(2*res);
    int k=(int)(j-1)/2+1;
    E1[k-1][l-1][0]=(double)data[i-1]*(1.0/(res*res));
    E1[k-1][l-1][1]=(double)data[i]*(1.0/(res*res));
}

//*****//
// Calculation of the total charge in the regions covered by junction and
//

```

```

// Setting electric displacement continuous in the front surface outside of the
// junctions //
//*****//

c0=0.0;c1=0.0;c2=0.0;c3=0.0;c4=0.0;c5=0.0;c6=0.0;c7=0.0;c8=0.0;
for(int i=0; i<res; i++){
  for(int j=0; j<res; j++){
    if((i>A3)&&(i<A4)&&(j>B3)&&(j<B4)){
      c0=(double)(c0+esi*E1[i][j][0]+eair*ES[i][j][0]);
    }

    if((i>A1)&&(i<A2)&&(j>B1)&&(j<B2)){
      c1=(double)(c1+esi*E1[i][j][0]+eair*ES[i][j][0]);
    }
    if((i>A3)&&(i<A4)&&(j>B1)&&(j<B2)){
      c2=(double)(c2+esi*E1[i][j][0]+eair*ES[i][j][0]);
    }
    if((i>A5)&&(i<A6)&&(j>B1)&&(j<B2)){
      c3=(double)(c3+esi*E1[i][j][0]+eair*ES[i][j][0]);
    }
    if((i>A1)&&(i<A2)&&(j>B3)&&(j<B4)){
      c4=(double)(c4+esi*E1[i][j][0]+eair*ES[i][j][0]);
    }
    if((i>A5)&&(i<A6)&&(j>B3)&&(j<B4)){
      c5=(double)(c5+esi*E1[i][j][0]+eair*ES[i][j][0]);
    }
    if((i>A1)&&(i<A2)&&(j>B5)&&(j<B6)){
      c6=(double)(c6+esi*E1[i][j][0]+eair*ES[i][j][0]);
    }
    if((i>A3)&&(i<A4)&&(j>B5)&&(j<B6)){
      c7=(double)(c7+esi*E1[i][j][0]+eair*ES[i][j][0]);
    }
    if((i>A5)&&(i<A6)&&(j>B5)&&(j<B6)){
      c8=(double)(c8+esi*E1[i][j][0]+eair*ES[i][j][0]);
    }
  }

  if(((i<=A1)||((i>=A2)&&(i<=A3))
      ||((i>=A4)&&(i<=A5))||
      (i>=A6))||((j<=B1)||((j>=B2)
      &&(j<=B3))||((j>=B4)&&
      (j<=B5))||((j>=B6))) {

      E1[i][j][0]=(double)eair*ES[i][j][0]/esi;
      E1[i][j][1]=(double)eair*ES[i][j][1]/esi;
  }
}
}

//*****//
// Fourier transform of the new estimated electric field //
// fE1 is now the new estimated field //
//*****//

```

```

m=0;
for(int i=0; i<res; i++){
    for(int j=0; j<res; j++){
        data[m]=(double)E1[i][j][0];
        data[m+1]=(double)E1[i][j][1];
        m+=2;
    }
}
//fourn(data,nn,1);
fftDo.complexForward(data);
for(int i=1; i<=2*res*res;i+=2){
    int l=i/(2*res)+1;
    int j=i-(l-1)*(2*res);
    int k=(j-1)/2+1;
    fE1[k-1][l-1][0]=data[i-1];
    fE1[k-1][l-1][1]=data[i];
}

//*****//
// From fE1 we calculate the new estimated potential //
//*****//

for(int i=0; i<res; i++){
    for(int j=0; j<res; j++){
        double R=Math.sqrt(Math.pow(Fk[i], 2)+Math.pow(Fk[j], 2));

        if(R==0.0){
            fV1[i][j][0]=(double)fE1[i][j][0]*d;
            fV1[i][j][1]=(double)fE1[i][j][1]*d;
        }
        else if(R>0.0){
            fV1[i][j][0]=(double)(fE1[i][j][0]
                /(R*(1.0+Math.exp(-2.0*R*d))/(1.0-Math.exp(-2.0*R*d))));
            fV1[i][j][1]=(double)(fE1[i][j][1]
                /(R*(1.0+Math.exp(-2.0*R*d))/(1.0-Math.exp(-2.0*R*d))));
        }
    }
}

//*****//
// Inverse Fourier transform for fV1 //
//*****//

m=0;
for(int i=0; i<res; i++){
    for(int j=0; j<res; j++){
        data[m]=fV1[i][j][0];
        data[m+1]=fV1[i][j][1];
        m+=2;
    }
}

```

```

    }
}
//fourier(data,nn,-1);
    fftDo.complexInverse(data, false);
    for(int i=1; i<=2*res*res;i+=2){
        int l=i/(2*res)+1;
        int j=i-(l-1)*(2*res);
        int k=(j-1)/2+1;
        V1[k-1][l-1][0]=data[i-1]*(1.0/(res*res));
        V1[k-1][l-1][1]=data[i]*(1.0/(res*res));
    }

flag=false; //set flag false to check convergence

//*****//
// V1 is the new value for voltage.We suppose that the solution is and //
// superposition of the initial hypothetical value and V1 //
// Voltage=(1-f)V+fV1 //
// where f is a coefficient between 0 and 1 //
// Then we check the difference between (Voltage-V). Also we take care //
// of voltage to be 1 in the central junction and 0 to the other junctions. //
//*****//

for(int i=0; i<res; i++){
    for(int j=0; j<res; j++){
        V[i][j][0]=(double)(0.8*V[i][j][0]+0.2*V1[i][j][0]);
        V[i][j][1]=(double)(0.8*V[i][j][1]+0.2*V1[i][j][1]);
        //pixel 1
        if((i>A1)&&(i<A2)&&(j>B1)&&(j<B2)){
            V[i][j][0]=(double)0.0;
            V[i][j][1]=(double)0.0;
        }
        //pixel 4
        if((i>A3)&&(i<A4)&&(j>B1)&&(j<B2)){
            V[i][j][0]=(double)0.0;
            V[i][j][1]=(double)0.0;
        }
        //pixel 7
        if((i>A5)&&(i<A6)&&(j>B1)&&(j<B2)){
            V[i][j][0]=(double)0.0;
            V[i][j][1]=(double)0.0;
        }
        //pixel 8
        if((i>A5)&&(i<A6)&&(j>B3)&&(j<B4)){
            V[i][j][0]=(double)0.0;
            V[i][j][1]=(double)0.0;
        }
        //pixel 2
        if((i>A1)&&(i<A2)&&(j>B3)&&(j<B4)){
            V[i][j][0]=(double)0.0;
            V[i][j][1]=(double)0.0;
        }
        //pixel 3
        if((i>A1)&&(i<A2)&&(j>B5)&&(j<B6)){

```

```

    V[i][j][0]=(double)0.0;
    V[i][j][1]=(double)0.0;
}
//pixel 6
if((i>A3)&&(i<A4)&&(j>B5)&&(j<B6)){
    V[i][j][0]=(double)0.0;
    V[i][j][1]=(double)0.0;
}
//pixel 9
if((i>A5)&&(i<A6)&&(j>B5)&&(j<B6)){
    V[i][j][0]=(double)0.0;
    V[i][j][1]=(double)0.0;
}
//kentriko pixel 5
if((i>A3)&&(i<A4)&&(j>B3)&&(j<B4)){
    V[i][j][0]=(double)1.0;
    V[i][j][1]=(double)0.0;
}

if(Math.abs(V[i][j][0]-V1[i][j][0])>0.001){// check of convergence if V-V1<0.01
    flag=true
                                // and the loop stops.

    flag=true;
}
}
}
c01x=(double) (c4+c5)/2.0*(-1.0);
c01y=(double) (c2+c7)/2.0*(-1.0);
c02=(double) (c1+c3+c6+c8)/4.0*(-1.0);
c00=(double) (c0-2.0*c01x-2.0*c01y-4.0*c02);
total=c0*evacume;
backplane=c00* evacume;
neighborx=c01x*evacume;
neighbory=c01y*evacume;
diagonal=c02*evacume;
// If you want to normalize to C1D ***
c1D=(double)esi*lx*ly/d;
c00D=c00/c1D;
c01Dx=c01x/c1D;
c01Dy=c01y/c1D;

    c02D=c02/c1D;
//*** end of normalization ***
    checkStop(stop);
    if(stop){break;}

System.out.println("-----+
+
-----"+'\n');
System.out.println("Cycle"+"="+cycle+" "+'\n');
System.out.println(" "+'\n');
System.out.println("Backplane:"+c00+" "+df.format(c00)+
    "+c00/c1D="+df.format(c00D)+" "+'\n' );
System.out.println("First neighbor x axis:"+
    "+c01"+"="+df.format(c01x)+
    "+c01/c1D="+df.format(c01Dx)+" "+'\n' );

```

```

System.out.println("First neighbor y axis:"+" "+c02+"="+df.format(c01y)+"
    "+"c02/c1D="+df.format(c01Dy)+" '+'\n' );
System.out.println("Diagonal neighbor:"+" "+c03+"="+df.format(c02)+"
    "+"c03/c1D="+df.format(c02D)+" '+'\n' );
System.out.println(" '+'\n');
System.out.println("Total charge :"+" "+df.format(c0)+'\n');
System.out.println("-----"
    +
    "-----"+"\n');

Outputs.append("-----"
    +
    "-----"+"\n');
Outputs.append("Cycle"+"="+cycle+" '+'\n');
Outputs.append(" "+"'\n');
Outputs.append("Backplane:"+"c00"+" "+df.format(c00)+"
    "+"c00/c1D="+df.format(c00D)+" '+'\n' );
Outputs.append("First neighbor x axis:"+" "+c01+"="+df.format(c01x)+"
    "+"c01/c1D="+df.format(c01Dx)+" '+'\n' );
Outputs.append("First neighbor y axis:"+" "+c02+"="+df.format(c01y)+"
    "+"c02/c1D="+df.format(c01Dy)+" '+'\n' );
Outputs.append("Diagonal neighbor:"+" "+c03+"="+df.format(c02)+"
    "+"c03/c1D="+df.format(c02D)+" '+'\n' );
Outputs.append(" '+'\n');
Outputs.append("Total charge :"+" "+df.format(c0)+'\n');
Outputs.append("-----"
    +
    "-----"+"\n');

if(cycle>1000){
    break;
}
cycle++;
}

    long t2 =System.currentTimeMillis();
System.out.println("*****"
    +
    "*****"+"\n');
System.out.println("*****"
    +
    "*****"+"\n');
System.out.println("**"+" "+"Total Capacitance:"+" "+C0="+df.format(total)+"
    "+"fF"+"\n');
System.out.println("**"+" "+"Backplane Capacitance:"+"
    "+"C00="+df.format(backplane)+" "+"fF"+"\n');
System.out.println("**"+" "+"First neighbor x-axis :"+"
    "+"C01="+df.format(neighborx)+" "+"fF"+"\n');
System.out.println("**"+" "+"First neighbor y-axis:"+"
    "+"C02="+df.format(neighborx)+" "+"fF"+"\n');
System.out.println("**"+" "+"Diagonal first neighbor:"+"
    "+"C03="+df.format(diagonal)+" "+"fF"+"\n');
System.out.println("**"+" "+"Calculation time:"+(t2-t1)+"+"ms"+"\n');

```

```

System.out.println("*****"
+
    "*****"+'\n');
System.out.println("*****"
+
    "*****"+'\n');
if(stop){System.out.println("Programm stopped..");}
if(!stop){System.out.println("The algorithm converge..."+'\n');
    System.out.println("Program has finished calculation..."+'\n');
}
if(cycle>1000){
    System.out.println("No convergence has reached until 1000 cycle ");
System.out.println("Program stopped..");
}

Outputs.append("*****"
+
    "*****"+'\n');
Outputs.append("*****"
+
    "*****"+'\n');
Outputs.append("**" + "Total Capacitance:" + " C0=" + df.format(total) +
    "fF"+'\n');
Outputs.append("**" + "Backplane Capacitance:" +
    "C00=" + df.format(backplane) + "fF"+'\n');
Outputs.append("**" + "First neighbor x-axis :"+
    "C01=" + df.format(neighborx) + "fF"+'\n');
Outputs.append("**" + "First neighbor y-axis:" +
    "C02=" + df.format(neighbory) + "fF"+'\n');
Outputs.append("**" + "Diagonal first neighbor:" +
    "C03=" + df.format(diagonal) + "fF"+'\n');
Outputs.append("**" + "Calculation time:" + (t2-t1) + "ms"+'\n');
Outputs.append("*****"
+
    "*****"+'\n');
Outputs.append("*****"
+
    "*****"+'\n');
if(stop){Outputs.append("Programm stopped..");}
if(!stop){Outputs.append("The algorithm converge..."+'\n');
    Outputs.append("Program has finished calculation..."+'\n');
}
if(cycle>1000){
    Outputs.append("No convergence has reached until 1000 cycle ");
Outputs.append("Program stopped..");
}
}

public static String OutputsToString()
{ DecimalFormat df = new DecimalFormat("0.0000");
    df.setRoundingMode(RoundingMode.HALF_UP);
    String s=df.format(total) + " " + df.format(backplane) + " " + df.format(neighborx) +
        " " + df.format(neighbory) + " " + df.format(diagonal);
}

```



```

return s;
}

public static void checkStop(boolean s)
{
    stop=s;
}

/*
public final static void swap(double x[], int a, int b) {
    double t = x[a];
    x[a] = x[b];
    x[b] = t;
}
/*
public static void fourn(double[] data, int[] nn ,int isign) {
/*fourier ndim.h
Replaces data by its ndim-dimensional discrete Fourier transform, if isign is
input as 1.
nn[0..ndim-1] is an integer array containing the lengths of each dimension
(number of complex
values), which must all be powers of 2. data is a real array of length twice the
product
of these lengths, in which the data are stored as in a multidimensional complex
array: real and
imaginary parts of each element are in consecutive locations, and the rightmost
index of the
array increases most rapidly as one proceeds along data. For a two-dimensional
array, this is
equivalent to storing the array by rows. If isign is input as 1, data is
replaced by its inverse
transform times the product of the lengths of all dimensions.*/
/*int idim,i1,i2,i3,i2rev,i3rev,ip1,ip2,ip3,ifp1,ifp2;
int ibit,k1,k2,n,nprev,nrem,ntot=1,ndim=nn.length;
double tempi,temp,theta,wi,wpi,wpr,wr,wtemp;
for (idim=0;idim<ndim;idim++) ntot *= nn[idim]; //Total no. of complex values.
if ((ntot<2)||((ntot&(ntot-1))!=0)) throw new IllegalArgumentException("must
have powers of 2 in fourn");

nprev=1;
for (idim=ndim-1;idim>=0;idim--) { //Main loop over the dimensions.
n=nn[idim];
nrem=ntot/(n*nprev);
ip1=nprev << 1;
ip2=ip1*n;
ip3=ip2*nrem;
i2rev=0;
for (i2=0;i2<ip2;i2+=ip1) { //This is the bit-reversal sec
if(i2 < i2rev) {// tion of the routine.
//630 Chapter 12. Fast Fourier Transform
for (i1=i2;i1<i2+ip1-1;i1+=2) {
for (i3=i1;i3<ip3;i3+=ip2) {
i3rev=i2rev+i3-i2;

```

```

swap(data,i3, i3rev);
swap(data,i3+1,i3rev+1);
}
}
}
ibit=ip2 >> 1;
while (ibit >= ip1 && i2rev+1 > ibit) {
i2rev -= ibit;
ibit >>= 1;
}
i2rev += ibit;
}
ifp1=ip1; //Here begins the Danielson-
//Lanczos section of the
//routine.
while (ifp1 < ip2) {
ifp2=ifp1 << 1;
theta=isign*2*Math.PI/(ifp2/ip1);// Initialize for the trigonomet
wtemp=Math.sin(0.5*theta);          // ric recurrence.
wpr= -2.0*wtemp*wtemp;
wpi=Math.sin(theta);
wr=1.0;
wi=0.0;
for (i3=0;i3<ifp1;i3+=ip1) {
for (i1=i3;i1<i3+ip1-1;i1+=2) {
for (i2=i1;i2<ip3;i2+=ifp2) {
k1=i2; //Danielson-Lanczos formula:
k2=k1+ifp1;
tempr=wr*data[k2]-wi*data[k2+1];
tempi=wr*data[k2+1]+wi*data[k2];
data[k2]=data[k1]-tempr;
data[k2+1]=data[k1+1]-tempi;
data[k1] += tempr;
data[k1+1] += tempi;
}
}
wr=(wtemp=wr)*wpr-wi*wpi+wr;// Trigonometric recurrence.
wi=wi*wpr+wtemp*wpi+wi;
}
ifp1=ifp2;
}
nprev *= n;
}
}

```

```

public static void writeTextFile(String fileName, String s) {
    FileWriter output = null;
    try {
        output = new FileWriter(fileName);
        BufferedWriter writer = new BufferedWriter(output);
        writer.write(s);
    }
}

```

```
    } catch (Exception e) {
        throw new RuntimeException(e);
    } finally {
        if (output != null) {
            try {
                output.close();
            } catch (IOException e) {
                // Ignore issues during closing
            }
        }
    }
}

}

}*/
}
```

---

## A'.0.2 Κλάση strip.java

---

```

//*****//
//      LAPLACE SOLVER FOR FINITE DEPTH      //
//      STRUCTURE OF 5 PARALLEL STRIPS      //
//      INCLUDING THE CONTRIBUTION OF AIR.    //
//      2-D Calculation.                    //
//      SEPTEMBER 2014.                    //
//*****//

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.PrintStream;
import java.text.DecimalFormat;
import javax.swing.JTextArea;
import org.jtransforms.fft.DoubleFFT_1D;
import java.math.RoundingMode;
public class strip {

    //*****//
    // Definition of variables                //
    //*****//
    private static int dim;
    private static final double esi=11.9,eair=1.0,e0=8.854*Math.pow(10, -6);
    private static int lj, pitch,d,len;
    private static int res;
    private static int A1,A2,A3,A4,A5,A6,A7,A8,A9,A10;
    private static double V[] [] ;
    private static double V1[] [] ;
    private static double ES[] [] ;
    private static double E1[] [] ;
    private static double data[];
    private static double fES[];
    private static double fE1[];
    static PrintStream out;

    private static double[] Fk;
    private static double c1,c2,c3,c4,c5,c01,c02,c00,c1D,c00D,c01D,c02D,
    total_cap,cap_first_neighbour_pF,cap_second_neighbour_pF;
    private static double cap_strip2back_pF;
    private static boolean stop;

    public strip(int resolution, int width,int seperation,int detector_thickness,
        int length){
        res=resolution;
        lj=width;
        pitch=seperation;
        d=detector_thickness;
        len=length;
        dim=res;
        V =new double[dim] [2];
        V1=new double[dim] [2];
        ES=new double [dim] [2];
        E1=new double [dim] [2];

```

```

    data=new double[2*dim];
    fES=new double[2*dim];
    fE1=new double[2*dim];

    Fk=new double[2*dim];

}

public static void stripcapacitance(JTextArea Outputs){

    DecimalFormat df = new DecimalFormat("0.0000");
    df.setRoundingMode(RoundingMode.HALF_UP);

    try{
        out = new PrintStream(new FileOutputStream("strip_outputs.txt", true));
        System.setOut(out);
    }
    catch(FileNotFoundException e){

    }

    DoubleFFT_1D fftDo = new DoubleFFT_1D(res);

    gridplacement();
    System.out.println("*****");
    System.out.println("*****");
    System.out.println("****"+" "+'\n');
    System.out.println("****"+" "+"The inputs are:"+"\n");
    System.out.println("****"+" "+'\n');
    System.out.println("****"+" "+"resolution="+res+'\n'+ "****"+" "+"strip
        seperation="+pitch+'\n'+ "****"+" "+"strip width="+lj+'\n'
            + "****"+" "+"dectoror
                width="+d+'\n'+ "****"+" "+"strip
                    length="+len+'\n');
    System.out.println("****"+" "+'\n');
    System.out.println("*****");
    System.out.println("*****");
    System.out.println(" "+'\n');

    Outputs.append("*****");
    Outputs.append("*****");
    Outputs.append("****"+" "+'\n');
    Outputs.append("****"+" "+"The inputs are:"+"\n");
    Outputs.append("****"+" "+'\n');
    Outputs.append("****"+" "+"resolution="+res+'\n'+ "****"+" "+"strip
        seperation="+pitch+'\n'+ "****"+" "+"strip width="+lj+'\n'
            + "****"+" "+"dectoror
                width="+d+'\n'+ "****"+" "+"strip
                    length="+len+'\n');

```

```

Outputs.append("****"+" "+'\n');
Outputs.append("*****");
Outputs.append("*****");
Outputs.append(" "+'\n');

//*****//
// Hypothetical solution.Set voltage central strip 1, //
// the first neighbor and the second neighbor strips to 0 //
//*****//

for(int i=0; i<res; i++){
    V[i][1]=(double)0.0;
    V[i][0]=(double) 0.0;
if ((i>A5)&&(i<A6)){
    V[i][0]=(double)1.0;
}
if((i>=A4-1)&&(i<=A5+1)){
    V[i][0]=(double) (i-(A4-1))/((A5+1)-(A4-1));
}
if((i>=A6-1)&&(i<=A7+1)){
    V[i][0]=(double) ((A7+1)-i)/((A7+1)-(A6-1));
}
}
}
//*****//

int cycle=1;
boolean flag=true; // check convergence

long t1 =System.currentTimeMillis();

while(flag==true){

    //loop will continue until convergence is reached (until flag=false)
    stop=false;
    checkStop(stop);
    if(stop){break;}

    //*****//
    // Fourier transform of surface potential V //
    //*****//

    int m=0;
    for(int i=0; i<res; i++){
        data[m]=(double)V[i][0];
        data[m+1]=(double)V[i][1];
        m+=2;
    }

    //four1(data,res,1);
    fftDo.complexForward(data); // call method complexForward from the object
    fftDo for FFT

```

```

Fk=kVectors(); //Set array of k vectors

    //*****//
    // From potential in Fourier space we can calculate //
    // the electric field in the ambient and in silicon at //
    // Fourier space //
    //*****//

m=0;
for(int i=0; i<res; i++){
    fES[m]=(double)data[m]*Fk[i];
    fES[m+1]=(double)data[m+1]*Fk[i];

    if(Fk[i]==0.0){
        fE1[m]=(double)data[m]*(1.0/d);
        fE1[m+1]=(double)data[m+1]*(1.0/d);
    }
    else if(Fk[i]>0.0){
fE1[m]=(double) data[m]*Fk[i]*(1.0+Math.exp(-2.0*Fk[i]*d))/(1.0-Math.exp(-2.0*Fk[i]*d));
fE1[m+1]=(double) data[m+1]*Fk[i]*(1.0+Math.exp(-2.0*Fk[i]*d))/
        (1.0-Math.exp(-2.0*Fk[i]*d));

    }
    else{
fE1[m]=(double) data[m]*Fk[i]*
        ((double)1.0+Math.exp(2.0*Fk[i]*d))/
        (double)(Math.exp(2.0*Fk[i]*d)-1.0);
fE1[m+1]=(double) data[m+1]*Fk[i]*
        ((double)1.0+Math.exp(2.0*Fk[i]*d))/
        (double)(Math.exp(2.0*Fk[i]*d)-1.0);

    }
    m+=2;
}

//*****//
// Inverse Fourier transform will convert fES, fE1 in //
// electric field ES E1 //
//*****//

fftDo.complexInverse(fES, false);
//four1(fES,res,-1);
m=0;
for(int i=0; i<res; i++){
    //normaliztion
    ES[i][0]=(double)fES[m]*(1.0/res);
    ES[i][1]=(double)fES[m+1]*(1.0/res);
    m+=2;
}

fftDo.complexInverse(fE1, false);
//four1(fE1,res,-1);
m=0;

```

```

for(int i=0; i<res; i++){
    //normaliztion
    E1[i][0]=(double)fE1[m]*(1.0/res);
    E1[i][1]=(double)fE1[m+1]*(1.0/res);
    m+=2;
}

//*****//
//    CALCULATION OF SUMs(FIELD*SURF.),WHICH IS DESIGNATED //
//    BY Cn AND IS THE TOTAL CHARGE ON THE APPROPRIATE //
//    SURFACE DIVIDED BY THE DIELECTRIC CONSTANT OF SILICON. //
//    WE HAVE SUMMATION ONLY ON SQUARES BECAUSE //
//    OUTSIDE OF THE SQUARES CHARGE IS ZERO. //
//*****//

c1=0.0;c2=0.0;c3=0.0;c4=0.0;c5=0.0;
for(int i=0; i<res; i++){
    if((i>=A1+1)&&(i<=A2-1)){
        c1=(double)(c1+esi*E1[i][0]+eair*ES[i][0]);
    }
    if((i>=A3+1)&&(i<=A4-1)){
        c2=(double)(c2+esi*E1[i][0]+eair*ES[i][0]);
    }
    if((i>=A5+1)&&(i<=A6-1)){
        c3=(double)(c3+esi*E1[i][0]+eair*ES[i][0]);
    }
    if((i>=A7+1)&&(i<=A8-1)){
        c4=(double)(c4+esi*E1[i][0]+eair*ES[i][0]);
    }
    if((i>=A9+1)&&(i<=A10-1)){
        c5=(double)(c5+esi*E1[i][0]+eair*ES[i][0]);
    }

    if(((i<=A1)||((i>=A2)&&(i<=A3))||((i>=A4)&&(i<=A5))
        ||((i>=A6)&&(i<=A7))||((i>=A8)&&(i<=A9))||((i>=A10))))
    {
        E1[i][0]=(double)eair*ES[i][0]/esi;
        E1[i][1]=(double)eair*ES[i][1]/esi;
    }
}

//*****//
//    Fourier transform of electric field E1 //
//    and calculation of the new value V1 //
//*****//

```



```

m=0;
for(int i=0; i<res; i++){
    data[m]=(double)E1[i][0];
    data[m+1]=(double)E1[i][1];
    m+=2;
}

//four1(data,res,1);
fftDo.complexForward(data);
m=0;
for(int i=0; i<res; i++){
    double R=Fk[i];
    if(R==0.0){
        data[m]=(double)data[m]*d;
        data[m+1]=(double)data[m+1]*d;
    }
    else if(R>0.0){
        data[m]=(double)(data[m]/(R*(1.0+Math.exp(-2.0*R*d))/(1.0-Math.exp(-2.0*R*d))));
        data[m+1]=(double)(data[m+1]/(R*(1.0+Math.exp(-2.0*R*d))/(1.0-Math.exp(-2.0*R*d))));
    }
    else{
        data[m]=(double)(data[m]/(R*(1.+Math.exp(2.0*R*d))/(Math.exp(2.0*R*d)-1.0)));
        data[m+1]=(double)(data[m+1]/(R*(1.0+Math.exp(2.0*R*d))/(Math.exp(2.0*R*d)-1.0)));
    }
    m+=2;
}

// four1(data,res,-1);
fftDo.complexInverse(data, false);
m=0;
for(int i=0; i<res; i++){
    V1[i][0]=(double)data[m]*(1.0/res);
    V1[i][1]=(double)data[m+1]*(1.0/res);
    m+=2;
}

flag=false; //flag false to check convergence

//*****
// V1 is the new value for voltage.We suppose that the solution is and
//
// superposition of the initial hypothetical value and V1
//
//          Voltage=(1-f)V+fV1
//
// where f is a coefficient between 0 and 1
//
// Then we check the difference between (Voltage-V). Also we take care
//

```

```

// of voltage to be 1 in the central junction and 0 to the other junctions.
//
//*****

for(int i=0; i<res; i++){
V[i][0]=(double)(0.9*V[i][0]+0.1*V1[i][0]);
V[i][1]=(double)(0.9*V[i][1]+0.1*V1[i][1]);
if((i>=A1+1)&&(i<=A2-1)){
    V[i][0]=(double)0.0;
    V[i][1]=(double)0.0;
}
if((i>=A3+1)&&(i<=A4-1)){
    V[i][0]=(double)0.0;
    V[i][1]=(double)0.0;
}
if((i>=A5+1)&&(i<=A6-1)){
    V[i][0]=(double)1.0;
    V[i][1]=(double)0.0;
}
if((i>=A7+1)&&(i<=A8-1)){
    V[i][0]=(double)0.0;
    V[i][1]=(double)0.0;
}
if((i>=A9+1)&&(i<=A10-1)){
    V[i][0]=(double)0.0;
    V[i][1]=(double)0.0;
}

if(Math.abs(V[i][0]-V1[i][0])>0.001){
    flag=true;
}

}

c01=(double)(c2+c4)/2.0*(-1.0);
c02=(double)(c1+c5)/2.0*(-1.0);
c00=(double)(c3-2.0*(c01+c02));
c1D=(double)lj*esi/d;
c01D=(double)c01/c1D;
c02D=(double)c02/c1D;
c00D=(double)c00/c1D;
total_cap=(double)c3*e0*len;
cap_strip2back_pF=(double)c00*e0*len;
cap_first_neighbour_pF=(double)c01*e0*len;
cap_second_neighbour_pF=(double)c02*e0*len;

checkStop(stop);
    if(stop){break;}

Outputs.append("-----");
Outputs.append("Cycle"+"="+cycle+'\n');
Outputs.append(" "+"'\n');

```

```

Outputs.append ("Backplane:"+" "+df.format(c00)+"
                "+"C00/C1D="+df.format(c00D)+'\n');
Outputs.append ("Fist neighbor:"+" "+df.format(c01)+"
                "+"C01/C1D="+df.format(c01D)+'\n');
Outputs.append ("Second neighbor:"+" "+df.format(c02)+"
                "+"C02/C1D="+df.format(c02D)+'\n');
Outputs.append(" "+'\n');
Outputs.append("Total charge:"+" "+df.format(c3)+'\n');
Outputs.append("-----");

System.out.println("-----");
System.out.println("Cycle"+"="+cycle+'\n');
System.out.println(" "+'\n');
System.out.println ("Backplane:"+" "+df.format(c00)+"
                    "+"C00/C1D="+df.format(c00D)+'\n');
System.out.println ("Fist neighbor:"+" "+df.format(c01)+"
                    "+"C01/C1D="+df.format(c01D)+'\n');
System.out.println ("Second neighbor:"+" "+df.format(c02)+"
                    "+"C02/C1D="+df.format(c02D)+'\n');
System.out.println(" "+'\n');
System.out.println("Total charge:"+" "+df.format(c3)+'\n');
System.out.println("-----");

if(cycle>1000){
    break;
}

cycle++;
}

    total_cap=c3*e0*len;
    cap_strip2back_pF=c00*e0*len;
    cap_first_neighbour_pF=c01*e0*len;
    cap_second_neighbour_pF=c02*e0*len;

    long t2 =System.currentTimeMillis();

Outputs.append(" "+'\n'+ " "+'\n'+ " "+'\n');
Outputs.append("*****");
Outputs.append("*****");
Outputs.append("**"+ " "+ "Total Capacitance:"+" "+df.format(total_cap)+"
                "+"pF"+'\n');
Outputs.append("**"+ " "+ "Capacitance firt neighbour:"+"
                "+df.format(cap_first_neighbour_pF)+" "+pF"+'\n');
Outputs.append("**"+ " "+ "Capacitance second neighbour:"+"
                "+df.format(cap_second_neighbour_pF)+" "+pF"+'\n');
Outputs.append("**"+ " "+ "Capacitance strip to backplane:"+"
                "+df.format(cap_strip2back_pF)+" "+pF"+'\n');
Outputs.append("**"+ " "+ "Calculation time:"+(t2-t1)+" "+ms'+'\n');
Outputs.append("*****");
Outputs.append("*****");
if(stop){Outputs.append("Programm stopped..");}

```

```

    if(!stop){Outputs.append("The algorithm converge..."+'\n');
        Outputs.append("Program has finished calculation..."+'\n');
    }
    if(cycle>1000){
        Outputs.append("No convergence has reached until 1000 cycle ");
        Outputs.append("Program stopped..");
    }

System.out.println(" "+'\n'+ " "+'\n'+ " "+'\n');
System.out.println("*****");
System.out.println("*****");
System.out.println("**"+ " "+ "Total Capacitance:"+"
    "+df.format(total_cap)+" "+pF+'\n');
System.out.println("**"+ " "+ "Capacitance firt neighbour:"+"
    "+df.format(cap_first_neighbour_pF)+" "+pF+'\n');
System.out.println("**"+ " "+ "Capacitance second neighbour:"+"
    "+df.format(cap_second_neighbour_pF)+" "+pF+'\n');
System.out.println("**"+ " "+ "Calculation time:"+(t2-t1)+" "+ms+'\n');
System.out.println("*****");
System.out.println("*****");
if(stop){System.out.println("Programm stopped..");}
if(!stop){System.out.println("The algorithm converge..."+'\n');
    System.out.println("Program has finished calculation..."+'\n');
    }
    if(cycle>1000){
        System.out.println("No convergence has reached until 1000 cycle ");
        System.out.println("Program stopped..");
    }
    }
    out.close();
}

public static String OutputsToString()
{
    DecimalFormat df = new DecimalFormat("0.0000");
    df.setRoundingMode(RoundingMode.HALF_UP);
    String s=df.format(total_cap)+" "+df.format(cap_strip2back_pF)+"
        "+df.format(cap_first_neighbour_pF)+" "+df.format(cap_second_neighbour_pF);
    return s;
}

public static void gridplacement(){
    //*****//
    // placement of strips on the grid' //
    //*****//
    dim=res;
    A5=(int)(res-lj)/2-1;
    A4=A5-pitch+1;
    A3=A4-lj-1;
    A2=A3-pitch+1;
    A1=A2-lj-1;
    A6=(int)(res+lj)/2;
    A7=A6+pitch-1;
    A8=A7+lj+1;
    A9=A8+pitch-1;
}

```

```

    A10=A9+lj+1;
}

public static double[] kVectors(){
//*****//
//Calculation of k vectors. Fourier coefficients //
//*****//
    double [] A=new double[res];
    for(int i=1; i<=(res/2+1); i++){
        A[i-1]=(double) 2.0*Math.PI*(i-1)/res;;
    }
    int k=1;
    for(int i=(int)(res/2+2); i<=res; i++){
        A[i-1]=(double) -1.0*A[res/2-k];
        k++;
    }
    return A;
}

public static void checkStop(boolean s)
{
    stop=s;
}

/*
public final static void swap(double x[], int a, int b) {
    double t = x[a];
    x[a] = x[b];
    x[b] = t;
}

public static void four1(double[] data, int n,int isign) {
    int nn,mmax,m,j,istep,i;
    double wtemp,wr,wpr,wpi,wi,theta,temp,tempi;
    if (n<2 || (n&(n-1))!= 0) throw new IllegalArgumentException("n must be power
        of 2 in four1");
    nn = n << 1;
    j = 1;
    for (i=1;i<nn;i+=2) {
        if (j > i) {
            swap(data,j-1,i-1);
            swap(data,j,i);
        }
        m=n;
        while (m >= 2 && j > m) {
            j -= m;
            m >>= 1;
        }
        j += m;
    }
    mmax=2;
}

```

```
while (nn > mmax) {
  istep=mmax << 1;
  theta=isign*(6.28318530717959/mmax);
  wtemp=Math.sin(0.5*theta);
  wpr = -2.0*wtemp*wtemp;
  wpi=Math.sin(theta);
  wr=1.0;
  wi=0.0;
  for (m=1;m<mmax;m+=2) {
    for (i=m;i<=nn;i+=istep) {
      j=i+mmax;
      tempr=wr*data[j-1]-wi*data[j];
      tempi=wr*data[j]+wi*data[j-1];
      data[j-1]=data[i-1]-tempr;
      data[j]=data[i]-tempi;
      data[i-1] += tempr;
      data[i] += tempi;
    }
    wr=(wtemp=wr)*wpr-wi*wpi+wr;
    wi=wi*wpr+wtemp*wpi+wi;
  }
  mmax=istep;
}
}*/

}
```

---

### A.0.3 Κλάση cap.java

---

```
//*****//
//      This class implements the JFrame      //
//      SEPTEMBER 2014.                        //
//*****//

import java.util.ArrayList;
import javax.imageio.ImageIO;
import javax.swing.JFrame;
import javax.swing.JPanel;

import java.awt.CardLayout;
import java.awt.Toolkit;

import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JRadioButton;
import javax.swing.SwingWorker;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.image.BufferedImage;

import javax.swing.JProgressBar;
import javax.swing.JLabel;

import java.beans.PropertyChangeEvent;
import java.beans.PropertyChangeListener;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStream;

import java.util.List;
import java.util.Scanner;

import javax.swing.JSpinner;
import javax.swing.SpinnerNumberModel;

import java.awt.Color;

import javax.swing.SwingConstants;
```

```
public class cap extends JPanel
implements
PropertyChangeListener {

    /**
    *
    */
    private static final long serialVersionUID = 1L;
    private JProgressBar progressBar;

    JFileChooser chooser;

    boolean done ;

    private JFrame frmCapacitanceCalculator;
    private JPanel panelCapcitanace ;
    private JPanel panelStrip ;
    private JPanel panelPixel ;
    private JPanel panelDone;
    private int res,lx,ly,sx,sy,d;
    private int res2,lj,pitch,len,d2;
    private String select;
    private String select2;
    private ArrayList<String> lines, Output_lines;
    private boolean stop;
    JTextArea Outputs = new JTextArea();

    SwingWorker<Void,Void> worker;
    //private JPanel strip;
    //private JPanel pixel;
    public JButton btnStop = new JButton("Stop");
    static PrintStream out2;

    /**
    * Launch the application.
    */
    public static void main(String[] args) {

        javax.swing.SwingUtilities.invokeLater(new Runnable() {

            public void run() {
                try {
                    cap window = new cap();
                    window.frmCapacitanceCalculator.setVisible(true);

                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```



```
    }
  });
}

/**
 * Create the application.
 */

public cap() {

    initialize();

}

/**
 * Initialize the contents of the frame.
 */
private void initialize() {

    stop=false;

    lines=new ArrayList<String>();
    Output_lines=new ArrayList<String>();
    frmCapacitanceCalculator = new JFrame("Capacitance calculation");
    frmCapacitanceCalculator.setTitle("Capacitance calculator");
    frmCapacitanceCalculator.setForeground(Color.WHITE);
    frmCapacitanceCalculator.setBackground(new Color(64, 64, 64));
    frmCapacitanceCalculator.getContentPane().setBackground(Color.black);
    frmCapacitanceCalculator.setResizable(false);
    frmCapacitanceCalculator.setBounds(100, 100, 863, 698);
    frmCapacitanceCalculator.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frmCapacitanceCalculator.getContentPane().setLayout(new CardLayout(0, 0));

    // panels
    /*******
    panelCapcittance = new JPanel();
    panelCapcittance.setBackground(Color.WHITE);
    frmCapacitanceCalculator.getContentPane().add(panelCapcittance,
        "name_14409002124862");
    panelCapcittance.setLayout(null);

    panelStrip = new JPanel();
    panelStrip.setBackground(Color.WHITE);
    frmCapacitanceCalculator.getContentPane().add(panelStrip,
        "name_14412539338863");
    panelStrip.setLayout(null);

    panelPixel = new JPanel();
    panelPixel.setBackground(Color.WHITE);
```

```

frmCapacitanceCalculator.getContentPane().add(panelPixel,
    "name_14416447323044");
panelPixel.setLayout(null);

panelDone = new JPanel();
panelDone.setBackground(Color.WHITE);
frmCapacitanceCalculator.getContentPane().add(panelDone,
    "name_28504729000154");
panelDone.setLayout(null);
//*****

progressBar = new JProgressBar();
progressBar.setBounds(240, 120, 458, 6);
progressBar.setForeground(Color.BLUE);
panelDone.add(progressBar);

    // spinners and labels for panel pixel
//*****
final JSpinner spinner_res = new JSpinner();
spinner_res.setModel(new SpinnerNumberModel(new Integer(0), new Integer(0),
    null, new Integer(1)));
spinner_res.setBounds(184, 185, 67, 16);
panelPixel.add(spinner_res);

final JSpinner spinner_ly = new JSpinner();
spinner_ly.setBounds(184, 241, 67, 16);
panelPixel.add(spinner_ly);

final JSpinner spinner_sx = new JSpinner();
spinner_sx.setBounds(184, 269, 67, 16);
panelPixel.add(spinner_sx);

final JSpinner spinner_sy = new JSpinner();
spinner_sy.setBounds(184, 297, 67, 16);
panelPixel.add(spinner_sy);

final JSpinner spinner_lx = new JSpinner();
spinner_lx.setBounds(184, 213, 67, 16);
panelPixel.add(spinner_lx);

final JSpinner spinner_d = new JSpinner();
spinner_d.setBounds(184, 325, 67, 18);
panelPixel.add(spinner_d);

JLabel lblNewLabel = new JLabel("resolution");
lblNewLabel.setForeground(Color.BLUE);
lblNewLabel.setHorizontalAlignment(SwingConstants.LEFT);
lblNewLabel.setLabelFor(spinner_res);
lblNewLabel.setBounds(12, 188, 97, 15);
panelPixel.add(lblNewLabel);

JLabel lblLx = new JLabel("pixel windth x axis");
lblLx.setForeground(Color.BLUE);

```

```
lblLx.setLabelFor(spinner_lx);
lblLx.setHorizontalAlignment(SwingConstants.LEFT);
lblLx.setBounds(12, 215, 143, 14);
panelPixel.add(lblLx);

JLabel lblSy = new JLabel("pixel windth y axis");
lblSy.setForeground(Color.BLUE);
lblSy.setLabelFor(spinner_ly);
lblSy.setHorizontalAlignment(SwingConstants.LEFT);
lblSy.setBounds(12, 241, 143, 14);
panelPixel.add(lblSy);

JLabel lblSx = new JLabel("pixel seperation x axis");
lblSx.setForeground(Color.BLUE);
lblSx.setLabelFor(spinner_sx);
lblSx.setHorizontalAlignment(SwingConstants.LEFT);
lblSx.setBounds(12, 270, 158, 14);
panelPixel.add(lblSx);

JLabel lblSy_1 = new JLabel("pixel seperation y axis");
lblSy_1.setForeground(Color.BLUE);
lblSy_1.setLabelFor(spinner_sy);
lblSy_1.setHorizontalAlignment(SwingConstants.LEFT);
lblSy_1.setBounds(12, 296, 172, 14);
panelPixel.add(lblSy_1);

JLabel lblWindth = new JLabel("detectors thickness");
lblWindth.setForeground(Color.BLUE);
lblWindth.setLabelFor(spinner_d);
lblWindth.setHorizontalAlignment(SwingConstants.LEFT);
lblWindth.setVerticalAlignment(SwingConstants.TOP);
lblWindth.setBounds(12, 327, 158, 14);
panelPixel.add(lblWindth);

    spinner_res.setEnabled(false);
    spinner_ly.setEnabled(false);
    spinner_sx.setEnabled(false);
    spinner_sy.setEnabled(false);
    spinner_lx.setEnabled(false);
    spinner_d.setEnabled(false);

//*****

    btnStop.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            stop=true;
        }
    });
```

```
// radio buttons for panel pixel
//*****
JRadioButton Inputs = new JRadioButton("Inputs");
Inputs.setForeground(Color.BLUE);
Inputs.setBackground(Color.WHITE);
Inputs.setBounds(6, 137, 103, 32);
panelPixel.add(Inputs);

JRadioButton InputsFromFile = new JRadioButton("Inputs from file");
InputsFromFile.setForeground(Color.BLUE);
InputsFromFile.setBackground(Color.WHITE);
InputsFromFile.setBounds(6, 99, 245, 32);
panelPixel.add(InputsFromFile);

ButtonGroup group = new ButtonGroup();

group.add(InputsFromFile);
group.add(Inputs);

Inputs.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        spinner_res.setEnabled(true);
        spinner_ly.setEnabled(true);
        spinner_sx.setEnabled(true);
        spinner_sy.setEnabled(true);
        spinner_lx.setEnabled(true);
        spinner_d.setEnabled(true);
        select="inputs";
    }
});

InputsFromFile.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        select="inputs from file";

        spinner_res.setEnabled(false);
        spinner_ly.setEnabled(false);
        spinner_sx.setEnabled(false);
        spinner_sy.setEnabled(false);
        spinner_lx.setEnabled(false);
        spinner_d.setEnabled(false);

        FileReader fileReader=null;
        chooser=new JFileChooser();
        int result=chooser.showOpenDialog(null);

        // System.out.println(result+" "+JFileChooser.APPROVE_OPTION);

        if(result==JFileChooser.APPROVE_OPTION){

            File file=chooser.getSelectedFile();
            String s=chooser.getSelectedFile().getPath();
```

```
try {
    fileReader = new FileReader(file);

} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    JOptionPane.showMessageDialog(frmCapacitanceCalculator,"file not
        found",s,JOptionPane.ERROR_MESSAGE ,null );
}

BufferedReader reader = new BufferedReader(fileReader);
String line = "";

try {

    lines.clear();
    while ((line = reader.readLine()) != null) {
        if(line!= null){
            Scanner sc=new Scanner(line);
            if(sc.hasNextInt()){
                lines.add(line);
                sc.close();
            }
        }
    }
}
// System.out.println(lines_with_variables);

} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    JOptionPane.showMessageDialog(frmCapacitanceCalculator,"file not
        found",s,JOptionPane.ERROR_MESSAGE ,null );
}

} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    JOptionPane.showMessageDialog(frmCapacitanceCalculator,"file not
        found",s,JOptionPane.ERROR_MESSAGE ,null );
}

}

finally {

    try {
        fileReader.close();
    } catch (IOException e) {
        // Ignore issues during closing
    }
}

}
}
```

```

);

//*****

// done and back button panel pixel
//*****
JButton DonePixel = new JButton("Done");
DonePixel.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        if(select.equalsIgnoreCase("inputs")){
            res = Integer.parseInt(spinner_res.getValue().toString());
            lx = Integer.parseInt(spinner_lx.getValue().toString());
            ly = Integer.parseInt(spinner_ly.getValue().toString());
            sx = Integer.parseInt(spinner_sx.getValue().toString());
            sy = Integer.parseInt(spinner_sy.getValue().toString());
            d= Integer.parseInt(spinner_d.getValue().toString());
            lines.clear();
            lines.add(res+" "+lx+" "+ly+" "+sx+" "+sy+" "+d);

        }

        panelDone.setVisible(true);
        panelPixel.setVisible(false);
    }
});
DonePixel.setBounds(20, 458, 89, 23);
panelPixel.add(DonePixel);

JButton BackPixel = new JButton("Back");
BackPixel.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        panelPixel.setVisible(false);
        panelCapcitance.setVisible(true);
    }
});
BackPixel.setBounds(141, 458, 89, 23);
panelPixel.add(BackPixel);
//*****

//images and labels panel pixel
//*****
final String IMG_PATH2 = "/images/pixels.png";

ImageIcon icon2=null;

try {
    BufferedImage img =
        ImageIO.read(cap.class.getResource(IMG_PATH2));

    icon2 = new ImageIcon(img);

```

```

    } catch (IOException e) {
        e.printStackTrace();
    }

    JLabel lblNewLabel_4 = new JLabel(icon2);
    lblNewLabel_4.setVerticalAlignment(SwingConstants.TOP);
    lblNewLabel_4.setBounds(269, 111, 559, 477);
    panelPixel.add(lblNewLabel_4);

    JTextArea txtrNumericalAssessmentOf = new JTextArea();
    txtrNumericalAssessmentOf.setForeground(Color.BLUE);
    txtrNumericalAssessmentOf.setEditable(false);
    txtrNumericalAssessmentOf.setText("Numerical estimation of the electric
        field for the case of nine rectangular pixel.\nFrom the numerical
        solution we can calculate the pixel to backplane and interpixel
        capacitances.\nFirst we define the geometry of each pixel, the
        distance between them and the detector thickness. ");
    txtrNumericalAssessmentOf.setBounds(109, 30, 669, 68);
    panelPixel.add(txtrNumericalAssessmentOf);

//*****

    // buttons strip pixel for capacitance panel
//*****

    JButton stripButton = new JButton("Strip");
    stripButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            select2="strip";

            panelStrip.setVisible(true);
            panelCapcittance.setVisible(false);
        }
    });
    stripButton.setBounds(208, 313, 89, 23);
    panelCapcittance.add(stripButton);

    JButton pixelButton = new JButton("Pixel");
    pixelButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            select2="pixel";

            panelPixel.setVisible(true);
            panelCapcittance.setVisible(false);
        }
    });
    pixelButton.setBounds(533, 313, 89, 23);
    panelCapcittance.add(pixelButton);
//*****

```

```

// image capacitance panel
//*****
final String IMG_PATH = "/images/dhmokritos.jpg";

ImageIcon icon=null;

try {
    BufferedImage img = ImageIO.read(cap.class.getResource(IMG_PATH));
    icon = new ImageIcon(img);

    } catch (IOException e) {
    e.printStackTrace();
    }

JLabel lblNewLabel_1 = new JLabel(icon);

lblNewLabel_1.setBounds(12, 12, 405, 83);
panelCapcittance.add(lblNewLabel_1);

JTextArea txtrAProgrammFor = new JTextArea();
txtrAProgrammFor.setTabSize(14);
txtrAProgrammFor.setEditable(false);
txtrAProgrammFor.setForeground(Color.BLUE);
txtrAProgrammFor.setText("A programm for fast calculation of capacitances
    in silicon sensors \nfor either pixel or strip type. The method is
    based on a numerical \nestimation of the electric field inside the
    fully depleted silicon detector\n");
txtrAProgrammFor.setBounds(208, 161, 460, 83);
panelCapcittance.add(txtrAProgrammFor);
//*****

//Spinners and labels for strip panel

final JSpinner spinner_res2 = new JSpinner();
spinner_res2.setBounds(159, 137, 82, 20);
panelStrip.add(spinner_res2);

final JSpinner spinner_lj = new JSpinner();
spinner_lj.setBounds(395, 137, 82, 20);
panelStrip.add(spinner_lj);

final JSpinner spinner_pitch = new JSpinner();
spinner_pitch.setBounds(274, 137, 82, 20);
panelStrip.add(spinner_pitch);

final JSpinner spinner_d2 = new JSpinner();
spinner_d2.setBounds(658, 137, 82, 20);
panelStrip.add(spinner_d2);

final JSpinner spinner_len = new JSpinner();

```



```
spinner_len.setBounds(522, 137, 82, 20);
panelStrip.add(spinner_len);

JLabel lblResolution = new JLabel("resolution");
lblResolution.setForeground(Color.BLUE);
lblResolution.setHorizontalAlignment(SwingConstants.CENTER);
lblResolution.setLabelFor(spinner_res2);
lblResolution.setBounds(159, 111, 82, 14);
panelStrip.add(lblResolution);

JLabel lblLj = new JLabel("strip width");
lblLj.setForeground(Color.BLUE);
lblLj.setHorizontalAlignment(SwingConstants.CENTER);
lblLj.setLabelFor(spinner_lj);
lblLj.setBounds(382, 111, 105, 14);
panelStrip.add(lblLj);

JLabel lblPitch = new JLabel("strip seperation");
lblPitch.setForeground(Color.BLUE);
lblPitch.setHorizontalAlignment(SwingConstants.CENTER);
lblPitch.setLabelFor(spinner_d2);
lblPitch.setBounds(251, 111, 132, 14);
panelStrip.add(lblPitch);

JLabel lblWindth_1 = new JLabel("detector thickness");
lblWindth_1.setForeground(Color.BLUE);
lblWindth_1.setHorizontalAlignment(SwingConstants.CENTER);
lblWindth_1.setLabelFor(spinner_d2);
lblWindth_1.setBounds(622, 111, 142, 14);
panelStrip.add(lblWindth_1);

JLabel lblLength = new JLabel("strip length");
lblLength.setForeground(Color.BLUE);
lblLength.setHorizontalAlignment(SwingConstants.CENTER);
lblLength.setLabelFor(spinner_len);
lblLength.setBounds(505, 111, 105, 14);
panelStrip.add(lblLength);

    spinner_res2.setEnabled(false);
    spinner_lj.setEnabled(false);
    spinner_pitch.setEnabled(false);
    spinner_d2.setEnabled(false);
    spinner_len.setEnabled(false);
```

```
//*****
```

```
//radio buttons for strip panel
```

```
JRadioButton InputsStrip = new JRadioButton("Inputs");
InputsStrip.setForeground(Color.BLUE);
InputsStrip.setBackground(Color.WHITE);
InputsStrip.setBounds(58, 135, 72, 23);
panelStrip.add(InputsStrip);

JRadioButton InputsFromFileStrip = new JRadioButton("Inputs from
file");
InputsFromFileStrip.setForeground(Color.BLUE);
InputsFromFileStrip.setBackground(Color.WHITE);
InputsFromFileStrip.setBounds(58, 72, 153, 23);
panelStrip.add(InputsFromFileStrip);

ButtonGroup group2 = new ButtonGroup();
group2.add(InputsFromFileStrip);
group2.add(InputsStrip);

InputsStrip.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        select="inputs Strip";

        spinner_res2.setEnabled(true);
        spinner_lj.setEnabled(true);
        spinner_pitch.setEnabled(true);
        spinner_d2.setEnabled(true);
        spinner_len.setEnabled(true);
    }
});

InputsFromFileStrip.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        select="inputs from file Strip";

        spinner_res.setEnabled(false);
        spinner_lj.setEnabled(false);
        spinner_pitch.setEnabled(false);
        spinner_d2.setEnabled(false);
        spinner_len.setEnabled(false);
    }
});

FileReader fileReader=null;
chooser=new JFileChooser();
int result=chooser.showOpenDialog(null);

// System.out.println(result+" "+JFileChooser.APPROVE_OPTION);

if(result==JFileChooser.APPROVE_OPTION){
```

```
File file=chooser.getSelectedFile();
String s=chooser.getSelectedFile().getPath();

try {
    fileReader = new FileReader(file);

} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    JOptionPane.showMessageDialog(frmCapacitanceCalculator, "file
        not found",s,JOptionPane.ERROR_MESSAGE ,null );
}

BufferedReader reader = new BufferedReader(fileReader);
String line = "";

try {

    lines.clear();
    while ((line = reader.readLine()) != null) {
        if(line!= null){
            Scanner sc= new Scanner(line);
            if(sc.hasNextInt()){
                lines.add(line);
            }
            sc.close();
        }
    }

}

// System.out.println(lines_with_variables);

} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    JOptionPane.showMessageDialog(frmCapacitanceCalculator, "file
        not found",s,JOptionPane.ERROR_MESSAGE ,null );
}

} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    JOptionPane.showMessageDialog(frmCapacitanceCalculator, "file
        not found",s,JOptionPane.ERROR_MESSAGE ,null );
}

}

finally {

    try {
        fileReader.close();
    } catch (IOException e) {
        // Ignore issues during closing
    }
}
```

```

        }

    }
}
);

//*****

//button strip back panel strip
//*****
JButton DoneStrip = new JButton("Done");
DoneStrip.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {

        if(select.equalsIgnoreCase("inputs Strip")){
            res2 = Integer.parseInt(spinner_res2.getValue().toString());
            lj = Integer.parseInt(spinner_lj.getValue().toString());
            pitch =
                Integer.parseInt(spinner_pitch.getValue().toString());
            d2 = Integer.parseInt(spinner_d2.getValue().toString());
            len = Integer.parseInt(spinner_len.getValue().toString());
            lines.clear();
            lines.add(res2+" "+lj+" "+pitch+" "+d2+" "+len);
        }
        panelDone.setVisible(true);
        panelStrip.setVisible(false);

    }
}
);

DoneStrip.setBounds(235, 594, 89, 23);
panelStrip.add(DoneStrip);

JButton BackStrip = new JButton("Back");
BackStrip.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        panelStrip.setVisible(false);
        panelCapcitance.setVisible(true);
    }
});
BackStrip.setBounds(515, 594, 89, 23);
panelStrip.add(BackStrip);

final String IMG_PATH3 = "/images/strips.png";

ImageIcon icon3=null;

```

```

try {
    BufferedImage img =
        ImageIO.read(cap.class.getResource(IMG_PATH3));

    icon3 = new ImageIcon(img);

    } catch (IOException e) {
    e.printStackTrace();
    }

JLabel lblNewLabel_2 = new JLabel(icon3);
lblNewLabel_2.setVerticalAlignment(SwingConstants.TOP);
lblNewLabel_2.setBounds(29, 199, 786, 391);
panelStrip.add(lblNewLabel_2);

JTextArea txtrNumericalEstimationOf = new JTextArea();
txtrNumericalEstimationOf.setText("Numerical estimation of the
    electric field for the case of five strips.From the
    numerical\nsolutio we can calculate the strip to backplane and
    interstrip capacitances.First we\ndefine the geometry of each
    strip, the distance between them and the detector thickness. ");
txtrNumericalEstimationOf.setForeground(Color.BLUE);
txtrNumericalEstimationOf.setEditable(false);
txtrNumericalEstimationOf.setBounds(100, 12, 626, 62);
panelStrip.add(txtrNumericalEstimationOf);

JButton btnNewButton_1 = new JButton("Back");
btnNewButton_1.setBounds(28, 199, 117, 23);
btnNewButton_1.setForeground(Color.BLACK);
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        if(select2.equalsIgnoreCase("pixel")){
            panelDone.setVisible(false);
            panelPixel.setVisible(true);
            Outputs.setText("");
            stop=false;
        }
        else if(select2.equalsIgnoreCase("strip")){
            panelDone.setVisible(false);
            panelStrip.setVisible(true);
            Outputs.setText("");
            stop=false;
        }
    }
});
panelDone.add(btnNewButton_1);
//*****

```

```

//Buttons start, back,stop,clear for panel done
//*****
JButton btnStart = new JButton("Start");
btnStart.setBounds(28, 148, 117, 23);
btnStart.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        stop=false;
        progressBar.setIndeterminate(true);
        start();
    }
});
panelDone.add(btnStart);

JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(180, 151, 610, 450);
panelDone.add(scrollPane);
scrollPane.setViewportView(Outputs);
Outputs.setForeground(Color.BLUE);

JButton btnClear = new JButton("Clear");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        Outputs.setText(" ");
        stop=false;
    }
});
btnClear.setBounds(28, 249, 117, 25);
panelDone.add(btnClear);

btnStop = new JButton("Stop");
btnStop.setBounds(28, 302, 117, 25);
panelDone.add(btnStop);

JTextArea txtrInEveryCycle = new JTextArea();
txtrInEveryCycle.setText("In every cycle the charge on the central
    pixel or strip is calculated as well as\nthe induced charge in the
    neighboring pixels or strips. The charge is numerical \nequal to
    the capacitance. In the end the total capacitance , the backplane
    and the\ncapacitance with respect the neighboring pixels or strips
    are calculated. ");
txtrInEveryCycle.setEditable(false);
txtrInEveryCycle.setForeground(Color.BLUE);
txtrInEveryCycle.setBounds(202, 33, 538, 75);
panelDone.add(txtrInEveryCycle);
scrollPane.createVerticalScrollBar();
scrollPane.createHorizontalScrollBar();
//*****

}

public void stop()
{

```

```
}

public void start(){

    // worker.addPropertyChangeListener(this);

    System.out.println("starting");

    worker=new SwingWorker<Void,Void>(){

        private pixel p;
        private strip s;

        @Override
        protected void done() {
            // TODO Auto-generated method stub
            super.done();
            Toolkit.getDefaultToolkit().beep();
            progressBar.setIndeterminate(false);
            if(!stop){Outputs.append("Done");}
            worker.isDone();
        }

        @Override
        protected void process(List<Void> arg0) {
            // TODO Auto-generated method stub
            super.process(arg0);
        }

        @Override
        protected Void doInBackground() throws Exception {
            // TODO Auto-generated method stub
            if(select2.equalsIgnoreCase("pixel")){
                long t1 =System.currentTimeMillis();
                for(String line : lines){
                    Scanner sc = new Scanner(line);
                    if(sc.hasNextInt()==true){
                        res=sc.nextInt();
                        lx=sc.nextInt();
                        ly=sc.nextInt();
                        sx=sc.nextInt();
                        sy=sc.nextInt();
                        d=sc.nextInt();
                    }
                    p=new pixel(res, lx, ly, sx, sy, d);
                    btnStop.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                            stop=true;
                            pixel.checkStop(stop);
                            worker.cancel(true);
                        }
                    });
                }
                pixel.pixelcapacitance(Outputs);
                sc.close();
                if(stop){break;}
            }
        }
    };
}
```

```

        Output_lines.add(pixel.OutputsToString());
    }

    try{
        out2 = new PrintStream(new
            FileOutputStream("pixel_outputs_summarize.txt", true));
        System.setOut(out2);
    }
    catch(FileNotFoundException e){

    }

    long t2 =System.currentTimeMillis();
    Outputs.append("Total calculation time:"+(t2-t1)+" "+"ms"+'\n');
    Outputs.append(" "+"'\n');
    Outputs.append(" "+"'\n');
    Outputs.append(" "+"'\n');
    Outputs.append(" "+"'\n');
    Outputs.append("Summarize");
    Outputs.append(" "+"'\n');
    Outputs.append("resolution"+" "+" "+" "+"strip width x axis"+"
        "+" "+" "+"strip width y axis"+" "+" "+"
        " "+"strip seperation x axis"+" "+" "+" "+"strip seperation y
        axis"+" "+" "+" "+"detector thickness"+" "+" "+" ");
    Outputs.append("Total Capacitance"+" "+" "+" "+"Capacitance to
        Backpalne"+" "+" "+"
        " "+"Capacitance First Neighbor x axis"+" "+" "+" "+"Capacitance
        First Neighbor y axis"+" "+" "+"
        " "+"Capacitance Diagonal Neighbor"+" "+" "+" "+"'\n');
    System.out.print("resolution"+" "+" "+" "+"strip width x axis"+"
        "+" "+" "+"strip width y axis"+" "+" "+"
        " "+"strip seperation x axis"+" "+" "+" "+"strip
        seperation y axis"+" "+" "+" "+"detector thickness"+"
        "+" "+" ");
    System.out.print("Total Capacitance"+" "+" "+" "+"Capacitance
        to Backpalne"+" "+" "+"
        " "+"Capacitance First Neighbor x axis"+" "+" "+"
        "+"Capacitance First Neighbor y axis"+" "+" "+"
        " "+"Capacitance Diagonal Neighbor"+" "+" "+" "+"'\n');

    int index=0;
    for(String line : lines){
        Scanner sc = new Scanner(line);
        Outputs.append(sc.nextInt()+" "+" "+" "+sc.nextInt()
            +" "+" "+" "+sc.nextInt()+" "+" "+" "+sc.nextInt()+"
            "+" "+" "+sc.nextInt()+" "+" "+" "+sc.nextInt()+"
            "+" "+" ");
        String line2=Output_lines.get(index);
        Scanner sc2=new Scanner(line2);
        Outputs.append(sc2.nextDouble()+" "+" "+" "+sc2.nextDouble()
            +" "+" "+" "+sc2.nextDouble()+" "+" "+"
            "+sc2.nextDouble()+" "+" "+" "+sc2.nextDouble()+'\n');
        index++;
        sc.close();
        sc2.close();
    }
}

```



```

    }
    int index2=0;
    for(String line : lines){
        Scanner sc = new Scanner(line);
        System.out.print(sc.nextInt()+" "+", "+" "+sc.nextInt()
            +" "+", "+" "+sc.nextInt()+" "+", "+" "+sc.nextInt()+"
                "+" "+", "+" "+sc.nextInt()+" "+", "+" " );
        String line2=Output_lines.get(index2);
        Scanner sc2=new Scanner(line2);
        System.out.print(sc2.nextDouble()+" "+", "+" "+sc2.nextDouble()
            +" "+", "+" "+sc2.nextDouble()+" "+", "+" "
                "+sc2.nextDouble()+"'\n'");
        index2++;
        sc.close();
        sc2.close();
    }
}
else if(select2.equalsIgnoreCase("strip"))
{

    long t1 =System.currentTimeMillis();
    for(String line : lines){
        Scanner sc = new Scanner(line);
        if(sc.hasNextInt()==true){
            res2=sc.nextInt();
            lj=sc.nextInt();
            pitch=sc.nextInt();
            d2=sc.nextInt();
            len=sc.nextInt();
        }

        s=new strip(res2, lj, pitch, d2, len);
        btnStop.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                stop=true;
                strip.checkStop(stop);
                worker.cancel(true);
            }
        });
        strip.stripcapacitance(Outputs);
        sc.close();
        if(stop){break;}
        Output_lines.add(strip.OutputsToString());

    }

    try{
        out2 = new PrintStream(new
            FileOutputStream("strip_outputs_summarize.txt", true));
        System.setOut(out2);
    }
    catch(FileNotFoundException e){

```

```

    }
    long t2 =System.currentTimeMillis();
    Outputs.append("Total calculation time:"+(t2-t1)+" "+"ms"+"'\n');
    Outputs.append(" "+"'\n');
    Outputs.append(" "+"'\n');
    Outputs.append(" "+"'\n');
    Outputs.append(" "+"'\n');
    Outputs.append("Summarize");
    Outputs.append(" "+"'\n');
    Outputs.append("resolution"+" "+" "+" "+"strip width"+" "+" "+" "+"
        "+"strip seperation"+" "+" "+" "+"
        " "+"detectors thickness"+" "+" "+" "+"length"+" "+" "+" "+" ");
    Outputs.append("Total Capacitance"+" "+" "+" "+"Capacitance to
        Backpalne"+" "+" "+" "+"
        " "+"Capacitance First Neighbor"+" "+" "+" "+"Capacitance Second
        Neighbor"+" "+" "+"'\n');
    System.out.print("resolution"+" "+" "+" "+"strip width"+"
        "+" "+" "+"strip seperation"+" "+" "+" "+"
        " "+"detectors thickness"+" "+" "+" "+"length"+" "+" "+" "+"
        ");
    System.out.print("Total Capacitance"+" "+" "+" "+"Capacitance to
        Backpalne"+" "+" "+" "+"
        " "+"Capacitance First Neighbor"+" "+" "+" "+"Capacitance
        Second Neighbor"+" "+" "+"'\n');

    int index=0;
    for(String line : lines){
        Scanner sc = new Scanner(line);
        Outputs.append(sc.nextInt()+" "+" "+" "+sc.nextInt()
            +" "+" "+" "+sc.nextInt()+" "+" "+" "+sc.nextInt()+"
            "+" "+" "+sc.nextInt()+" "+" "+" ");
        String line2=Output_lines.get(index);
        Scanner sc2=new Scanner(line2);
        Outputs.append(sc2.nextDouble()+" "+" "+" "+sc2.nextDouble()
            +" "+" "+" "+sc2.nextDouble()+" "+" "+" "+"
            "+sc2.nextDouble()+"'\n');
        index++;
        sc.close();
        sc2.close();
    }
    int index2=0;
    for(String line : lines){
        Scanner sc = new Scanner(line);
        System.out.print(sc.nextInt()+" "+" "+" "+sc.nextInt()
            +" "+" "+" "+sc.nextInt()+" "+" "+" "+sc.nextInt()+"
            "+" "+" "+sc.nextInt()+" "+" "+" ");
        String line2=Output_lines.get(index2);
        Scanner sc2=new Scanner(line2);
        System.out.print(sc2.nextDouble()+" "+" "+" "+sc2.nextDouble()
            +" "+" "+" "+sc2.nextDouble()+" "+" "+" "+"
            "+sc2.nextDouble()+"'\n');
        index2++;
        sc.close();
        sc2.close();
    }

```

---

```
        }

        }
        out2.close();
        return null;
    }

};
worker.execute();
}

public void propertyChange(PropertyChangeEvent evt) {
    // TODO Auto-generated method stub
    progressBar.setIndeterminate(true);
    if ("progress" == evt.getPropertyName()) {
        int progress = (Integer) evt.getNewValue();
        progressBar.setValue(progress);
        Outputs.append(String.format(
            "Completed %d%% of worker.\n", worker.getProgress()));
    }
}
}
```

---