

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

Εργαστήριο Μικροεπεξεργαστών και Υλικού



**«Υλοποίηση σε αναδιατασσόμενη λογική
αλγορίθμου υπολογισμού βέλτιστης πορείας
ιπτάμενου αντικειμένου για μείωση ηχητικής
πίεσης»**

Δημήτριος Π. Κοντός

Επιτροπή:

Παπαευσταθίου Ιωάννης, Επίκουρος Καθηγητής (Επιβλέπων)

Δόλλας Απόστολος, Καθηγητής

Πνευματικάτος Διονύσιος, Αναπληρωτής Καθηγητής

Χανιά 2009

Περιεχόμενα

Περίληψη.....	5
Ευχαριστίες.....	7
1. Ευρύτερο πλαίσιο	10
1.1 Εισαγωγή.....	10
1.2 Το πρόβλημα	11
2. Ο υπολογισμός του ηχητικού πεδίου.....	14
2.1 Το μαθηματικό μοντέλο	14
2.2 Η αριθμητική προσέγγιση του μοντέλου (CNPE μέθοδος)	15
2.3 Μετασχηματισμός χωρικής περιοχής	16
2.4 Τριδιαγώνιος αλγόριθμος, μέθοδος LU	17
2.5 Ο κώδικας.....	17
2.6 Βασικά βήματα του αλγορίθμου.....	18
3. Εισαγωγή στη σχεδίαση	21
3.1 Υλικό και λογισμικό που χρησιμοποιήθηκε	21
3.2 Περιγραφή πυρήνων κινητής υποδιαστολής	21
3.3 Χρήση δομικών πόρων από τους πυρήνες κινητής υποδιαστολής.....	23
3.4 Επιλογή FPGA	26
4. Σχεδίαση και υλοποίηση	28
4.1 Συναρτήσεις.....	28
4.2 Τμήματα.....	30
4.3 Σχεδίαση και υλοποίηση του τμήματος μνήμης.....	33
4.4 Σχεδίαση τμημάτων	35
4.4.1 Πρώτη μέθοδος (αναλυτική μορφή).....	35
4.4.2 Δεύτερη μέθοδος (FSM & RF).....	38
4.4.3 Συνδιασμός των δύο μεθόδων	44

4.5	Υλοποίηση τμημάτων	45
4.5.1	Πρώτη υλοποίηση (πρώτη μέθοδος)	45
4.5.2	Δεύτερη υλοποίηση (πρώτη και δεύτερη μέθοδος)	45
4.5.3	Τρίτη Υλοποίηση (πρώτη και δεύτερη μέθοδος, μόνο λογική).....	46
4.5.4	Τέταρτη υλοποίηση (δεύτερη μέθοδος).....	47
4.6	Επιβεβαίωση αποτελεσμάτων	48
4.7	Πρόταση για υλοποίηση	48
5.	Συμπεράσματα.....	50
5.1	Μέτρηση χρόνου εκτέλεσης σε H/Y	50
5.2	Σύγκριση συχνοτήτων και κύκλων ρολογιού	52
5.3	Σύγκριση δέσμευσης πόρων.....	53
5.4	Τελική σύγκριση	54
6.	Επίλογος και μελλοντική προοπτική.....	58
6.1	Επίλογος.....	58
6.2	Μελλοντική προοπτική	58
7.	Αναφορές.....	59

Περίληψη

Ο αεροδυναμικός θόρυβος που παράγεται από τους στροφείς ελικοπτέρων και κάθε ιπτάμενου αντικειμένου έχει μελετηθεί λεπτομερώς στο παρελθόν. Παρά τις προσπάθειες για βελτιωμένα αεροδυναμικά σχέδια που μειώνουν την ηχητική εκπομπή, ο θόρυβος παραμένει ένα πρόβλημα για κάθε πολιτική και στρατιωτική αποστολή επειδή διαδίδεται σε μεγάλες αποστάσεις στην ατμόσφαιρα. Λαμβάνοντας υπόψιν τα ατμοσφαιρικά και γεωγραφικά δεδομένα είναι δυνατόν να υπολογιστεί η βέλτιστη πορεία πτήσης ιπτάμενου αντικειμένου για ελαχιστοποίηση του θορύβου.

Μεταξύ των μαθηματικών μοντέλων, που έχουν χρησιμοποιηθεί για την περιγραφή της διάδοσης ήχου στην ατμόσφαιρα, μία ικανοποιητική προσέγγιση δίνεται από Παραβολική Διαφορική Εξίσωση (ΠΔΕ). Η λύση της εξίσωσης αυτής δίνει την ένταση του ηχητικού πεδίου για μεγάλες αποστάσεις. Ο υπολογισμός της λύσης γίνεται σταδιακά, κατά μήκος της κατεύθυνσης της διάδοσης. Αυτό επιτρέπει να συμπεριληφθούν και άλλα χαρακτηριστικά του προβλήματος, όπως οι τοπικές ατμοσφαιρικές συνθήκες και η γεωμορφολογία.

Για την επίλυση της ΠΔΕ στον Η/Υ υπάρχουν πολλές αξιόλογες μέθοδοι, μία από αυτές είναι η CNPE (Crank-Nicholson for the Parabolic Equation). Η μέθοδος προκύπτει από την προσέγγιση της ΠΔΕ με σχήματα Πεπερασμένων Διαφορών. Είναι θεωρητικά τεκμηριωμένη και έχει αποδειχθεί ευσταθής και ελεγχόμενης ακρίβειας, για την επίλυση της ΠΔΕ σε υπολογιστικό περιβάλλον, τόσο σε δύο όσο και τρεις χωρικές διαστάσεις.

Όπως όλα τα σχήματα πεπερασμένων διαφορών έτσι και η μέθοδος CNPE οδηγεί τελικά σε ένα γραμμικό σύστημα, με αγνώστους την τιμή της έντασης του ηχητικού πεδίου πάνω στους κόμβους διακριτοποίησης. Ο βασικός πίνακας αυτού του συστήματος έχει μη μηδενικά στοιχεία μόνο στις τρεις διαγωνίους, είναι δηλαδή “αραιός”, οπότε μπορεί να επιλυθεί γρήγορα με ειδικούς αλγορίθμους.

Η συγκεκριμένη εργασία επικεντρώνεται στον τριδιαγώνιο αλγόριθμο και ειδικότερα στα συστήματα που προκύπτουν από την Crank-Nicholson διακριτοποίηση 2ου βαθμού για την επίλυση της παραβολικής εξίσωσης. Οι τριδιαγώνιοι αλγόριθμοι γενικά δεν δέχονται παραλληλία σε όλα τα επίπεδα, κάτι που δυσκολεύει την υλοποίηση σε αναδιατασσόμενη λογική.

Η ανάπτυξη της τεχνολογίας αναδιατασσίμενης λογικής επιτρέπει την υλοποίηση συστημάτων που είναι όχι μόνο γρήγορα αλλά και απόλυτα ακριβή.

Μέρος των εξισώσεων σχεδιάστηκαν σε υλικό και διάφοροι μέθοδοι εξετάσθηκαν. Έγιναν επίσης υλοποιήσεις για όλες τις διαφορετικές μεθόδους και παραμέτρους με σκοπό να επιτευχθεί η μεγαλύτερη δυνατή παραλληλία και ταχύτητα με λογικό κόστος.

Ας σημειωθεί ότι τα αποτελέσματα αυτής της εργασίας μπορούν να χρησιμοποιηθούν σε οποιεσδήποτε εφαρμογές μοντελοποιούνται από την ΠΔΕ, όπως σε προβλήματα υπολογισμού της κατανομής θερμότητας και την διάχυση υγρών ρίπων.

Ευχαριστίες

Η εργασία αυτή εκπονήθηκε στο Εργαστήριο Μικροεπεξεργαστών και Υλικού του Πολυτεχνείου Κρήτης υπό την επίβλεψη του Επίκουρου Καθηγητή κ. Ιωάννη Παπαευσταθίου, κατά τη διάρκεια του ακαδημαϊκού έτους 2008-2009.

Θα ήθελα πρωτίστως να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Ιωάννη Παπαευσταθίου για την υποστήριξη του και την εμπιστοσύνη που έδειξε στις δυνατότητες μου. Η καθοδήγηση και οι συμβουλές του αποτελούν την βάση αυτής της εργασίας.

Θα ήθελα να ευχαριστήσω επίσης τον Καθηγητή κ. Απόστολο Δόλλα και τον Αναπληρωτή Καθηγητή κ. Διονύσιο Πνευματικάτο όχι μόνο σαν μέλη της εξεταστικής επιτροπής αλλά και σαν καθηγητές μου στη διάρκεια των προπτυχιακών μου σπουδών.

Ευχαριστώ επίσης τον Δρ. Χρήστο Αρβανίτη που διέθεσε όχι μόνο τον κώδικά του ώστε να υλοποιηθεί αυτή η διπλωματική εργασία αλλά και την πολύτιμη συμβουλή και βοήθεια σε κάθε στάδιο της υλοποίησης.

Θα ήθελα να πω ένα μεγάλο ευχαριστώ στον Μεταπτυχιακό Φοιτητή και φίλο μου Γιώργο Χατζηπαρασκευά χωρίς τη βοήθεια του οποίου δεν θα είχε ολοκληρωθεί αυτή η εργασία. Στάθηκε δίπλα μου προσφέροντας την αμέριστη καθοδήγησή του σε κάθε μου βήμα.

Ευχαριστώ πολύ και όλα τα μέλη του Εργαστηρίου Μικροεπεξεργαστών και Υλικού που όλο αυτό τον καιρό μου προσέφεραν όχι μόνο την βοήθεια αλλά και την φιλία τους.

Ένα μεγάλο ευχαριστώ αξίζει σε όλους τους συναδέλφους και φίλους που με στήριξαν με την φιλία τους όχι μόνο στις προπτυχιακές μου σπουδές αλλά και στη ζωή μου. Αγγέλικα, Βαγγέλη, Γεωργία, Γιώργο, Νίκο, Πέγκυ, Πολυχρόνη, Σταυρούλα και Φαίη σας ευχαριστώ για την ανεκτίμητη αγάπη σας. Μια μεγάλη συγνώμη για όλους όσους δεν ανέφερα προσωπικά. Σας ευχαριστώ πολύ όλους.

Τέλος θα ήθελα να πω ένα μεγάλο ευχαριστώ στην οικογένειά μου. Οι γονείς μου, Παντελής και Γιώτα, και τα αδέρφια μου, Βασίλης και Ανδριάννα, είναι η πηγή της δύναμης μου σε ό,τι κάνω στη ζωή μου. Δεν υπάρχουν λόγια να περιγράψουν την αγάπη και την ευγνωμοσύνη μου.

*στους γονείς μου, Πάντελή και Τιώτα,
και στα αδέρφια μου, Βασίλη και Ανδριάνα*

1. Ευρύτερο πλαίσιο

Η συγκεκριμένη διπλωματική εργασία εντάσσεται στο ευρύτερο πλαίσιο της μελέτης «Βέλτιστη τροχιά ηχητικής πηγής υπεράνω τοπογραφίας» [1] .

Παρακάτω αναφέρονται κάποια στοιχεία για την υφιστάμενη τεχνολογία και τεχνογνωσία, τους στόχους της αναφερόμενης μελέτης και αναλύεται περισσότερο το τμήμα που απασχολεί την συγκεκριμένη διπλωματική εργασία.

1.1 Εισαγωγή

Ο αεροδυναμικός θόρυβος που παράγεται από τους στροφείς ελικοπτέρων έχει μελετηθεί λεπτομερώς στο παρελθόν. Παρά τις προσπάθειες για βελτιωμένα αεροδυναμικά σχέδια που μειώνουν την ηχητική εκπομπή, ο θόρυβος παραμένει ένα πρόβλημα για τις πολιτικές και στρατιωτικές αποστολές επειδή διαδίδεται με μικρή μείωση στην ατμόσφαιρα. Κατά συνέπεια, ο θόρυβος που παραλαμβάνεται στις μακρινές θέσεις δίνει την προειδοποίηση για το πλησιάζων όχημα πολύ πριν να είναι ορατό στο σημείο υποδοχής. Η πρόβλεψη της μεγάλης ακτίνας διάδοσης θορύβου σε ανώμαλο έδαφος και η επιλογή της βέλτιστης πορείας πτήσης ιπτάμενων αντικειμένων που παράγουν το κατώτατο επίπεδο θορύβου στο σημείο υποδοχής είναι επομένως σημαντικά στοιχεία για την επιτυχία οποιασδήποτε αποστολής.

Η μεγάλη ακτίνα διάδοσης θορύβου σε ανώμαλο έδαφος περιγράφεται από τις γραμμικές εξισώσεις Euler [5]. Εντούτοις, η αριθμητική λύση αυτών των εξισώσεων είναι υπολογιστικά απαιτητική ακόμη και για περιοχές μέτριου μεγέθους, επειδή απαιτεί την εφαρμογή αριθμητικών μεθόδων υψηλής τάξης ακρίβειας στο χώρο και το χρόνο. Από την άλλη πλευρά, η εφαρμογή των αποδοτικών αλλά προσεγγιστικών μεθόδων ακτίνας για τη διάδοση θορύβου δεν παράγει ακριβείς προβλέψεις για τα ηχητικά κύματα που διαδίδονται σε ανώμαλο έδαφος, ανομοιογενές περιβάλλον, και με ισχυρές κλίσεις αέρα. Η κατανόηση των μηχανισμών διάδοσης θορύβου στην ατμόσφαιρα έχει προσελκύσει σημαντική προσοχή που οφείλεται στο μεγάλο κοινωνικοοικονομικό αντίκτυπο του περιβαλλοντικού θορύβου. Η ατμόσφαιρα είναι ένα ανομοιογενές κινούμενο μέσο όπου οι άνεμοι, η ατμοσφαιρική αναταραχή, και οι παραλλαγές θερμοκρασίας οδηγούν σε μια, ποικίλου εύρους και ύψους, ταχύτητα ήχου στον αέρα. Η

σημαντική παραλλαγή της ταχύτητας του ήχου στην ατμόσφαιρα σε συνδυασμό με την ανομοιομορφία του εδάφους προσθέτει περισσότερες δυσκολίες στη πρόβλεψη θορύβου μεγάλης ακτίνας.

Για την σχεδόν οριζόντια διάδοση θορύβου σε μια ανομοιογενή ατμόσφαιρα, η εξίσωση Helmholtz [6] παράγει μια παραβολική εξίσωση η οποία έχει χρησιμοποιηθεί αποτελεσματικά στο παρελθόν για να μοντελοποιήσει τη διάδοση μεγάλης ακτίνας. Πιο συγκεκριμένα, η τυποποιημένη προσέγγιση παραβολικής εξίσωσης μικρής γωνίας που εισήχθη από τον Tappert [7], η τριτοτάξια παραβολική προσέγγιση που εισήχθη από τον Claerbout [7], και η Παραβολική Εξίσωση “Green’s function” [9] χρησιμοποιήθηκαν για να μοντελοποιήσουν την διάδοση θορύβου στην ατμόσφαιρα με αξονική συμμετρία. Επιπλέον, αυτές οι μέθοδοι συνδυάστηκαν με πρότυπα ατμοσφαιρικής αναταραχής για να μελετηθούν τα αποτελέσματα της ατμοσφαιρικής αναταραχής στην διάδοση θορύβου. Πραγματοποιήθηκαν επίσης αριθμητικές προσομοιώσεις της ακουστικής διάδοσης σε μια ατμόσφαιρα με ταχύτητα ήχου εξαρτώμενη από το ύψος και με επιφάνειες με ποικίλη σύνθετη αντίσταση. Πρόσφατα, τα αριθμητικά πρότυπα βασισμένα στην παραβολική προσέγγιση για την διάδοση του ήχου στην ατμόσφαιρα διασυνδέθηκαν με ένα γεωγραφικό σύστημα πληροφοριών (Geographic Information System, GIS) για την πρόβλεψη της ηχητικής διάδοσης σε ανώμαλο έδαφος.

Η ανάλυση των παραπάνω μεθόδων δεν αποτελεί στόχο της παρούσας εργασίας η οποία επικεντρώνεται στην επιλυσιμότητα της CNPE μεθόδου που παρουσιάζεται εκτενέστερα παρακάτω.

1.2 Το πρόβλημα

Για να παρουσιασθεί η στρατηγική βελτιστοποίησης που χρησιμοποιήθηκε, θα περιγραφεί το πρόβλημα στις δύο διαστάσεις. Τα βασικά στοιχεία αυτής της διαδικασίας καταγράφονται στο Σχήμα 1.2-1. Η ζητούμενη τροχιά της ηχητικής πηγής (ελικόπτερο), πρέπει να επιλεγεί έτσι ώστε, ο επίγειος παρατηρητής (receiver) να έχει την μικρότερη απολαβή ήχου, σε όλη την διάρκεια της πτήσης από το αρχικό σημείο (κάτω αριστερά) έως το τελικό σημείο (κορυφή λόφου).

Για να βρεθεί η ζητούμενη τροχιά, διαιρείται αρχικά η οριζόντια απόσταση μεταξύ του αρχικού και τελικού σημείου σε N μικρότερα τμήματα οπότε προκύπτουν οι

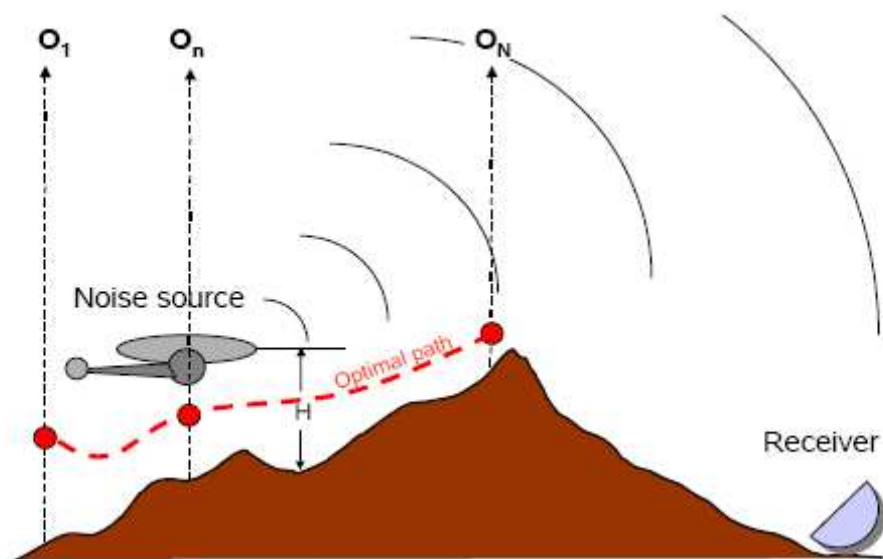
οριζόντιες θέσεις $O_1, O_2, \dots, O_N$. Υποθέτοντας ότι μόνο στις θέσεις αυτές επιτρέπεται στην πηγή να αλλάζει ύψος πτήσης (ενώ μεταξύ οποιονδήποτε δύο διαδοχικών θέσεων O_k, O_{k+1} αυτή κινείται οριζόντια) το αρχικό πρόβλημα αναγάγεται στον υπολογισμό του ύψους $Z_1, Z_2, \dots, Z_N$ της πηγής πάνω στις θέσεις $O_1, O_2, \dots, O_N$. Με αυτή την θεώρηση, η ζητούμενη τροχιά προσεγγίζεται από μία κατά τμήματα σταθερή καμπύλη (ασυνεχή πάνω στις οριζόντιες θέσεις), που προκύπτει επιλύοντας επαναλαμβανόμενες φορές ένα πρόβλημα της μορφής:

«Δεδομένου ότι η πηγή ήχου βρίσκεται σε γνωστή οριζόντια θέση O_k , ζητείται το ύψος της Z_k ώστε ο παρατηρητής να έχει την μικρότερη απολαβή ήχου».

Για την λύση αυτού του προβλήματος, υπολογίζεται το πεδίο της έντασης του ήχου που παράγει η πηγή, θέτοντας την σε διαφορετικά ύψη $Z_{k1}, Z_{k2}, \dots, Z_{kM}$. Από αυτά επιλέγεται το ύψος Z_k για το οποίο ο παρατηρητής έχει την μικρότερη απολαβή.

Συνοψίζοντας, προκειμένου να επιλυθεί το αρχικό πρόβλημα με αυτήν την στρατηγική, απαιτούνται $N \times M$ υπολογισμοί του πεδίου έντασης που παράγει η πηγή, όταν τίθεται στις θέσεις $(O_1, Z_{11}), (O_1, Z_{12}), \dots, (O_1, Z_{1M}), (O_2, Z_{21}), (O_2, Z_{22}), \dots, (O_2, Z_{2M}), \dots, (O_N, Z_{N1}), (O_N, Z_{N2}), \dots, (O_N, Z_{NM})$. Άρα για την επίλυση απαιτείται ένας οικονομικός και ακριβής υπολογισμός του ηχητικού πεδίου που δημιουργεί μία πηγή ήχου γνωστών χαρακτηριστικών (συχνότητα, ένταση ήχου) και θέσης (οριζόντιο μήκος, ύψος) πάνω από μία μη-επίπεδη τοπογραφία εδάφους.

Σε τρεις διαστάσεις, το πρόβλημα βελτιστοποίησης αντιμετωπίζεται με ανάλογο τρόπο, ωστόσο ο όγκος των πράξεων που απαιτούνται αυξάνει με τετραγωνικό ρυθμό.



Σχήμα 1.2-1. Σχηματική παρουσίαση διάδοσης ήχου μεγάλης ακτίνας ενός ελικοπτήρου που πετάει σε ύψος H .

2. Ο υπολογισμός του ηχητικού πεδίου

2.1 Το μαθηματικό μοντέλο

Η συνήθης παραβολική εξίσωση (ΠΕ) εισήχθη από τον Tappert σαν ένα μοντέλο που θα περιγράψει διάδοση ήχου περιορισμένης γωνίας στην υποθαλάσσια ακουστική. Χρησιμοποιώντας την προσέγγιση $\phi(r, z) = u(r, z)H(r)$, όπου $H(r)$ είναι η εξίσωση Hankel πρώτου τύπου και μηδενικής τάξης προκύπτει η εξής Ελλειπτική Μερική Διαφορική Εξίσωση

$$[\partial_{rr} + (1/r)\partial_r + \partial_{zz} + k^2(r, z)]\phi = 0 \quad (1)$$

για οριζόντια απόσταση r και βάθος z .

Αγνοώντας τον όρο u_{rr} προκύπτει η ακόλουθη προσέγγιση.

$$u_r = \frac{ik_0}{2}[n^2(r, z) - 1]u + \frac{i}{2k_0}u_{zz} \quad (2)$$

Η Σχέση (2) είναι η Παραβολική Μερική Διαφορική Εξίσωση που θα προσεγγισθεί αριθμητικά. Ο προσδιορισμός του πεδίου έντασης $u(r, z)$ του ήχου, πάνω από μία ορθογώνια περιοχή, περιγράφεται από την Μερική Διαφορική Εξίσωση (ΜΔΕ) της Σχέσης (1). Η ΠΕ της Σχέσης (2) που τελικά επιλύεται, είναι μία καλή προσέγγιση της Σχέσης (1). Γενικά είναι προτιμότερο να επιλύει κανείς παραβολικές ΜΔΕ αντί ελλειπτικών ή υπερβολικών για λόγους ευστάθειας. Στις παραβολικές ΜΔΕ οι στρογγυλοποιήσεις που θα γίνουν στον υπολογιστή κατά την διάρκεια των υπολογισμών, επιδρούν μόνο στην ακρίβεια και όχι στον αλγόριθμο.

Αναλυτικότερα για τις παραπάνω σχέσεις, η οριζόντια απόσταση από την πηγή συμβολίζεται με r και η κατακόρυφη συμβολίζεται με z . Στην περίπτωση υγρού περιβάλλοντος το z αναφέρεται ως «βάθος» (από την στάθμη επιφανείας). Για αέριο περιβάλλον όμως το z συμβολίζει το «ύψος» (από το έδαφος). Η ένταση του ηχητικού πεδίου στο σημείο (r, z) συμβολίζεται με $u(r, z)$ και η συχνότητα της πηγής με f . Η τοπική ταχύτητα ήχου στο μέσο διάδοσης συμβολίζεται με c και ενδέχεται να είναι διαφορετική από σημείο σε σημείο λόγω διαφορετικής θερμοκρασίας. Ο κυματάρριθμος συμβολίζεται με k ($k = \frac{2\pi f}{c}$) και η ταχύτητα ήχου

αναφοράς με c_0 (πρόκειται για μία σταθερή τιμή αναφοράς). Τέλος ο δείκτης διάθλασης συμβολίζεται με n ($n = \frac{c_0}{c}$).

Στη συνέχεια για λόγους απλοποίησης η Σχέση (2) γράφεται

$$u_r = \alpha(r, z)u + \beta u_{zz} \quad (3)$$

όπου $\alpha(r, z) = ik_0[n^2(r, z) - 1]/2$ και $\beta = i/2k_0$. Επίσης εισάγονται οι συνοριακές συνθήκες που θα χρησιμοποιηθούν για την επίλυση.

$$u(r, 0) = u(r, z_{\max}) = 0 \quad (4)$$

2.2 Η αριθμητική προσέγγιση του μοντέλου (CNPE μέθοδος)

Γενικά για την επίλυση της διαφορικής εξίσωσης πάνω από μια περιοχή, πρέπει να είναι γνωστή η τιμή της λύσης στα σύνορα που οριοθετούν την περιοχή. Με τις συνθήκες της Σχέσης (4) θεωρείται ότι ακριβώς πάνω από το έδαφος η ένταση του ήχου είναι μηδέν λόγω απόλυτης διάχυσης. Με άλλου τύπου συνθήκες θα περιγραφόταν η ανάκλαση ή συνδυασμός ανάκλασης και διάχυσης.

Έστω ότι αναζητείται η αριθμητική λύση της u που ικανοποιεί την (3) πάνω από μια ορθογώνια χωρική περιοχή. Επειδή αναζητούνται προσεγγίσεις με την CNPE μέθοδο, υποθέτουμε ότι η χωρική περιοχή διαιρείται ομοιόμορφα σε μικρότερα ορθογώνια οριζόντιας πλευράς Δr και κάθετης πλευράς Δz , χρησιμοποιώντας κατάλληλο πλέγμα. Η λύση u της (3) πάνω στους κόμβους αυτού του πλέγματος θα προσεγγισθεί χρησιμοποιώντας όρους δευτέρου βαθμού (Crank-Nikolson 2^{ου} βαθμού).

Συμβολίζοντας με U_k^n την προσεγγιστική τιμή που έχει η ζητούμενη λύση u στον κόμβο $(n \Delta r, k \Delta z)$, (και αν για λόγους απλότητας συμβολισθεί η u_r σαν $R(u)$) τότε από την σχέση (3) συμπεραίνεται η εξής προσεγγιστική σχέση CN-2 (Crank-Nikolson 2^{ου} βαθμού)

$$\left[U - \frac{\Delta r}{2} R\right]_k^{n+1} = \left[U + \frac{\Delta r}{2} R\right]_k^n \quad (5)$$

οπότε, με άγνωστους τις προσεγγιστικές τιμές U_k^n πάνω στους κόμβους διακριτοποίησης, παίρνουμε την εξής «τριδιαγώνια σχέση»:

$$a_l U_{k-1}^{n+1} + b_l U_k^{n+1} + a_r U_{k+1}^{n+1} = a_r U_{k-1}^n + b_r U_k^n + a_l U_{k+1}^n \quad (6)$$

με τους CN-2 συντελεστές

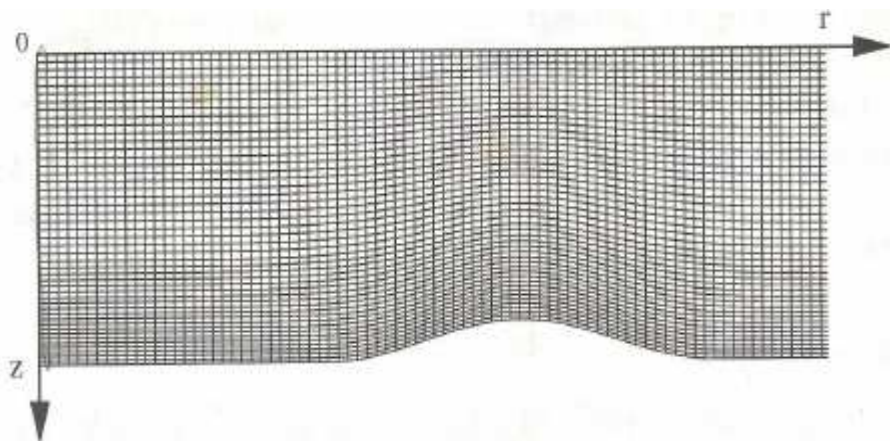
$$\begin{aligned} a_l &= -\beta h_r h_z \\ b_l &= 1 - \alpha h_r + 2\beta h_r h_z \\ a_r &= \beta h_r h_z \\ b_r &= 1 + \alpha h_r + 2\beta h_r h_z \end{aligned} \quad (7)$$

όπου $h_r = \Delta r / 2$ και $h_z = 1 / (\Delta z)^2$.

Οπότε, ο υπολογισμός της προσεγγιστικής λύσης ανάγεται στην επίλυση τού γραμμικού συστήματος (6). Λόγω της αραιής μορφής του, αυτό το σύστημα επιλύεται εύκολα με τον τριδιαγώνιο αλγόριθμο.

2.3 Μετασχηματισμός χωρικής περιοχής

Οι αριθμητικές λύσεις της υποθαλάσσιας διάδοσης θορύβου στις διανυσματικές περιοχές λαμβάνονται χρησιμοποιώντας τους κατάλληλους μετασχηματισμούς των εκάστοτε συντεταγμένων (Σχήμα 2.3-1). Αυτοί οι μετασχηματισμοί αντιστοιχούν τη φυσική περιοχή, η οποία χωρίζεται κατά ανομοιόμορφα διαστήματα, σε μια ορθογώνια περιοχή μη-ομοιόμορφης διαμέρισης, που αναφέρεται ως υπολογιστική περιοχή. Η αριθμητική λύση εκτελείται στο μη-ομοιόμορφο πλέγμα χρησιμοποιώντας τις συγκεκριμένες μεθόδους και τύπους.



Σχήμα 2.3-1. Μη ορθογώνια περιοχή και ο μετασχηματισμός της.

2.4 Τριδιαγώνιος αλγόριθμος, μέθοδος LU

Από τη Σχέση (6) παράγεται ένα τριδιαγώνιο σύστημα με αγνώστους την λύση της ΠΕ της Σχέσης (3) πάνω στους κόμβους διακριτοποίησης (δηλαδή πάνω στα σημεία $(r, z)_{ij}$ του πλέγματος που χωρίστηκε η ορθογώνια χωρική περιοχή). Η επίλυση αυτού του συστήματος στον κώδικα που μελετάται γίνεται με τον τριδιαγώνιο αλγόριθμο [16] (μία παραλλαγή της μεθόδου LU που χρησιμοποιείται όταν ο πίνακας ενός συστήματος είναι παντού μηδέν εκτός από την κεντρική διαγώνιο και τις υποδιαγωνίους πάνω και κάτω από αυτήν).

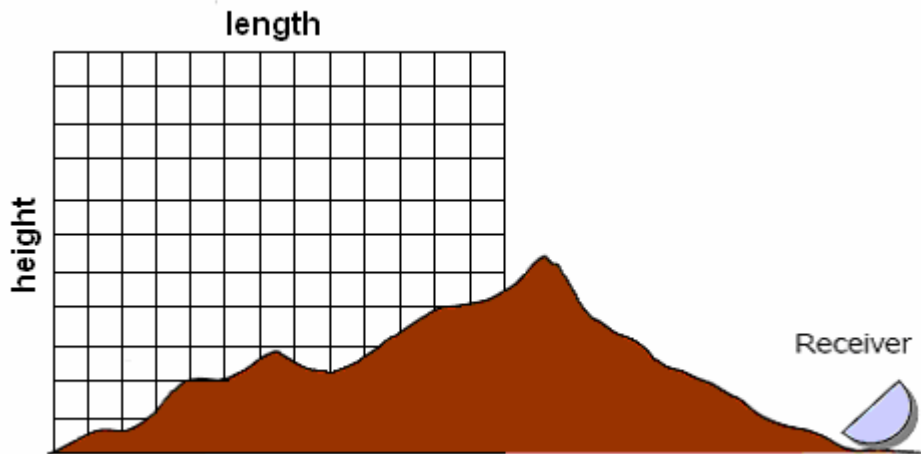
2.5 Ο κώδικας

Για την επαλήθευση των αποτελεσμάτων που δίνονται από την αναδιατασσόμενη λογική, χρησιμοποιήθηκε ο κώδικας του Δρ. Χ. Αρβανίτη που υλοποιεί την μέθοδο CN-2 σε γλώσσα C. Ακολουθεί μια σύντομη περιγραφή του κώδικα, τόσο για να παρουσιαστεί η λειτουργία του όσο και να γίνει σαφές ποιες ιδιαιτερότητες παρουσιάζει όσο αφορά την σχεδίαση του σε αναδιατασσόμενη λογική.

Η συγκεκριμένη εφαρμογή (που παράχθηκε από ένα σύνολο αρχείων της γλώσσας C) δέχεται σαν είσοδο τη θέση του παρατηρητή, το ύψος του πομπού και τα τοπογραφικά στοιχεία μιας περιοχής και επιστρέφει σαν αποτέλεσμα την μέγιστη λαμβανόμενη ηχητική πίεση που αντιλαμβάνεται ο παρατηρητής. Δίνοντας σαν είσοδο στην εφαρμογή διάφορες τιμές για το ύψος του πομπού είναι δυνατόν να προσδιοριστεί το ύψος στο οποίο η ηχητική πίεση είναι και η ελάχιστη στο σημείο που βρίσκεται ο παρατηρητής. Επαναλαμβάνοντας τη διαδικασία για κάθε σημείο του οριζόντιου άξονα μπορεί να σχεδιαστεί μια πορεία στον δισδιάστατο χώρο η οποία θα παράγει την μικρότερη δυνατή ηχητική πίεση στον συγκεκριμένο παρατηρητή. Οι επαναλήψεις του αλγόριθμου αυξάνονται όσο αυξάνεται ο αριθμός των δυνατών θέσεων του πομπού (και απαιτείται αρκετός χρόνος και υπολογιστική ισχύ) κάτι που ώθησε και αυτήν την μελέτη για την υλοποίηση/επιτάχυνση του με χρήση υλικού.

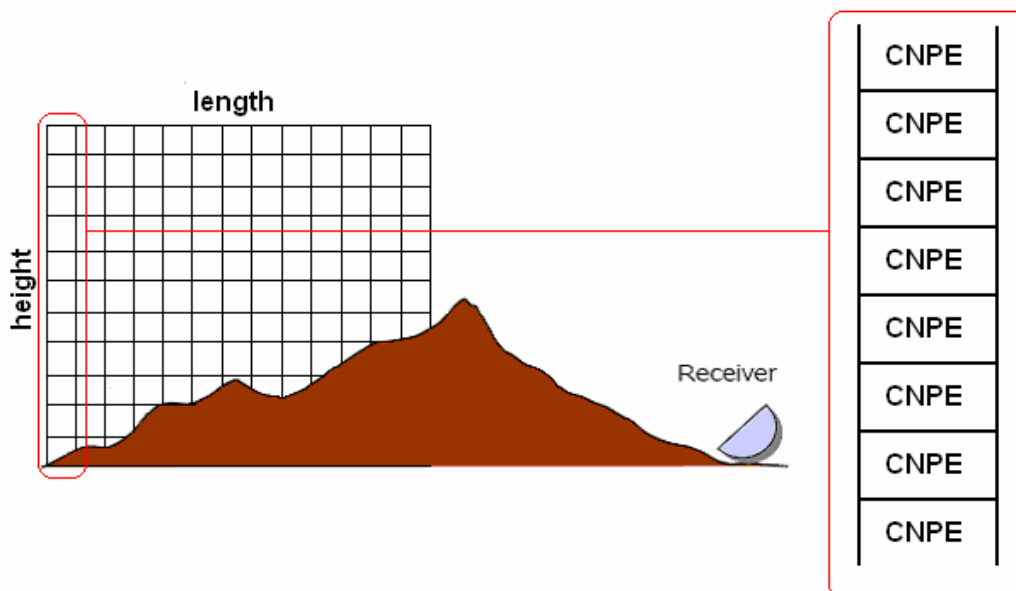
2.6 Βασικά βήματα του αλγορίθμου

1° Βήμα: Δεδομένου ενός αρχείου τοπογραφίας εδάφους η χωρική περιοχή χωρίζεται σε τμήματα (με χρήση του μετασχηματισμού χωρικής περιοχής) (Σχήμα 2.6-1).



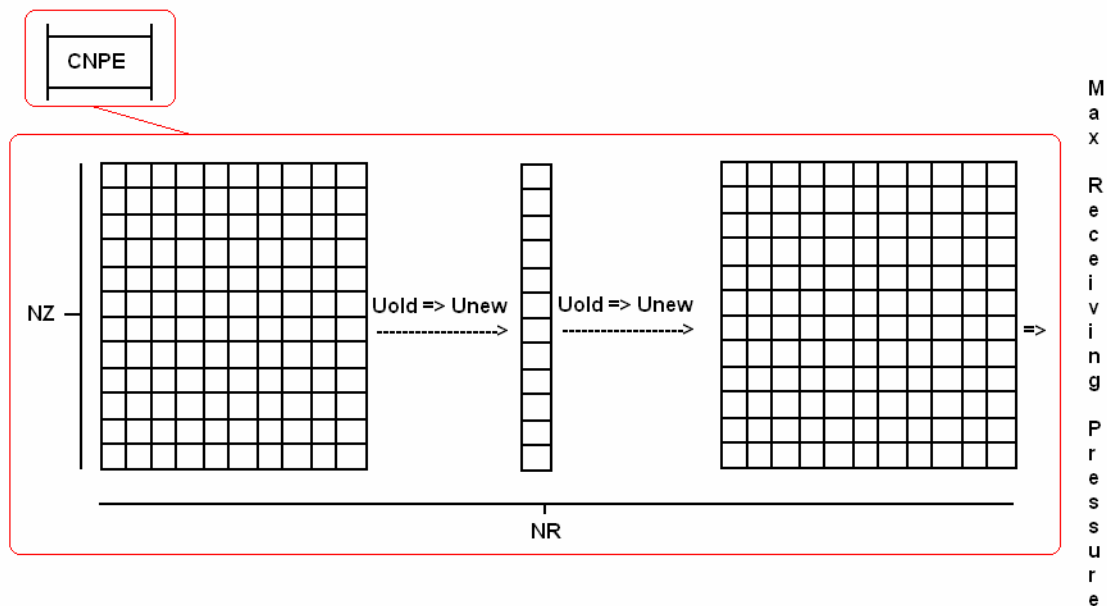
Σχήμα 2.6-1. Ο δέκτης (receiver) που θέλουμε να λάβει τη μικρότερη ηχητική πίεση και τα τμήματα που χωρίζεται το δισδιάστατο περιβάλλον.

2° Βήμα: Για διάφορα ύψη, και για κάθε συντεταγμένη του οριζόντιου άξονα καλείται ο CNPE αλγόριθμος (Σχήμα 2.6-2).



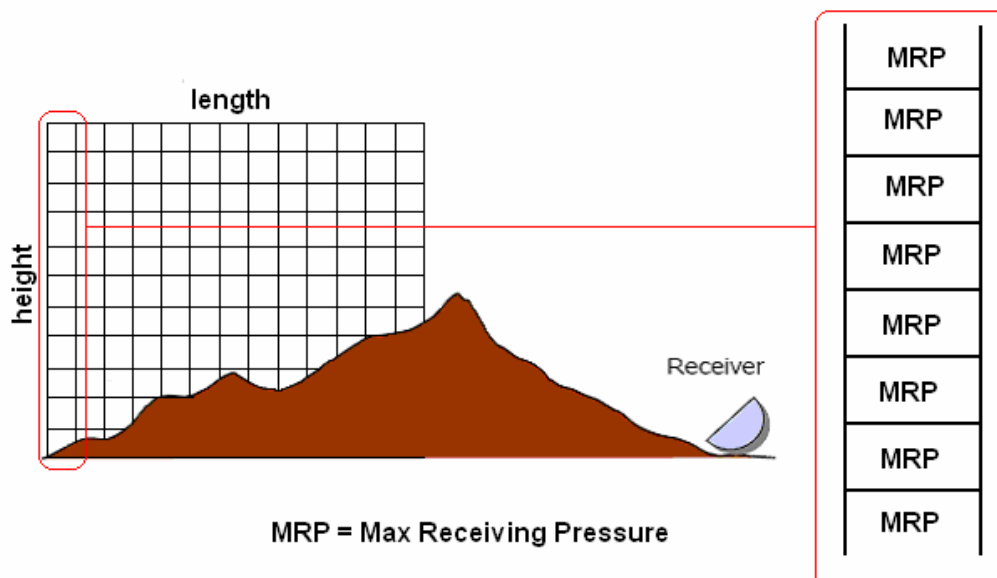
Σχήμα 2.6-2. Ο CNPE αλγόριθμος για όλα τα ύψη.

3^ο Βήμα: Για κάθε κλήση του αλγορίθμου επιλύεται το τριδιαγώνιο σύστημα και υπολογίζεται η μέγιστη ηχητική πίεση που ασκείται στον δέκτη (Σχήμα 2.6-3).



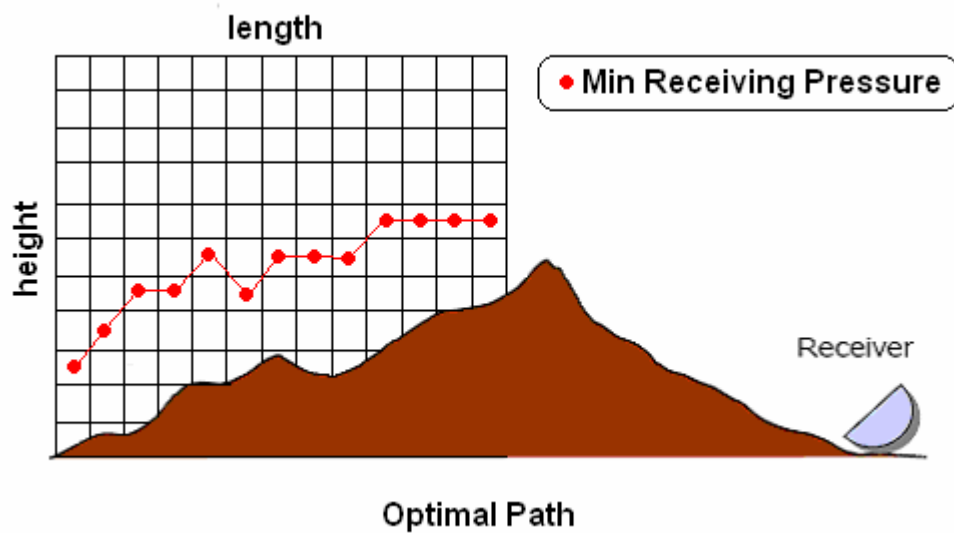
Σχήμα 2.6-3. Ο υπολογισμός της μέγιστης ηχητικής πίεσης για ένα μόνο τμήμα.

4^ο Βήμα: Δημιουργείται το διάνυσμα μέγιστων πιέσεων για όλα τα ύψη (Σχήμα 2.6-4).



Σχήμα 2.6-4: Ο μέγιστος θόρυβος που προκαλείται σε κάθε ύψος για τον δέκτη.

5^ο Βήμα: Εντοπίζεται το ύψος που προκαλεί την μικρότερη ηχητική πίεση στο δέκτη ($\text{Min}(\text{MRP}_1, \text{MRP}_2, \dots, \text{MRP}_N)$). Επαναλαμβάνοντας την ίδια διαδικασία για κάθε σημείο του οριζώντιου άξονα μπορεί να σχεδιαστεί η βέλτιστη πορεία πτήσης (Σχήμα 2.6-5).



Σχήμα 2.6-5. Τα ύψη με την μικρότερη ηχητική πίεση στο δέκτη και η βέλτιστη πορεία πτήσης.

Το υπολογιστικά αργό κομμάτι του αλγορίθμου είναι η μέθοδος CN-2.

3. Εισαγωγή στη σχεδίαση

3.1 Υλικό και λογισμικό που χρησιμοποιήθηκε

Για την διεκπεραίωση της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκαν διάφοροι Η/Υ που έφεραν επεξεργαστές Intel Celeron, Intel Pentium 4, και AMD. Τα λειτουργικά συστήματα που χρησιμοποιήθηκαν ήταν Windows XP Professional Service Pack 2 32bit, Ubuntu Linux 7.10 και Ubuntu Linux 8.04.

Για την σχεδίαση του συστήματος σε αναδιατασσόμενη λογική χρησιμοποιήθηκε το πρόγραμμα ISE 10.1.01 WebPACK της εταιρείας Xilinx και οι πυρήνες κινητής υποδιαστολής που προσφέρει η εταιρεία ενώ για την προσομοίωση του χρησιμοποιήθηκε το πρόγραμμα Modelsim 6.2c.

Ο αλγόριθμος σε C μεταγλωττίστηκε με τον GCC compiler. Οι χρόνοι εκτέλεσης του αλγορίθμου μετρήθηκαν σε έναν Η/Υ Intel Pentium 4 με CPU 2,66 GHz και 1024 Mb RAM με λειτουργικό σύστημα Ubuntu Linux 8.04 με τη χρήση του προγράμματος Intel Vtune Performance Analyzer.

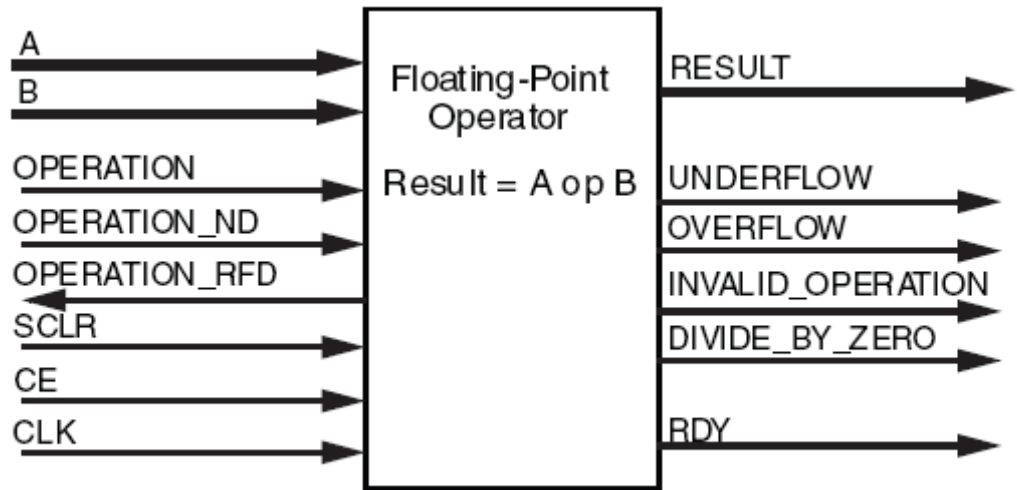
3.2 Περιγραφή πυρήνων κινητής υποδιαστολής

Ο πυρήνας κινητής υποδιαστολής (Floating Point Cell, FPC) [3] που χρησιμοποιήθηκε παρέχει στους σχεδιαστές τα μέσα για να εκτελεσθούν πράξεις αριθμητικής κινητής υποδιαστολής σε μία FPGA. Ο πυρήνας μπορεί να προσαρμοστεί για λειτουργία απλής η διπλής ακρίβειας, για συγκεκριμένο λανθάνοντα χρόνο (το διάστημα που μεσολαβεί μεταξύ δεδομένων και αποτελέσματος, latency) και ως προς τη διεπαφή (interface).

Ο πυρήνας υποστηρίζει διαδικασίες πολλαπλασιασμού, πρόσθεσης/αφαίρεσης, διαίρεσης, υπολογισμού τετραγωνικής ρίζας, σύγκρισης, μετατροπής μεταξύ διάφορων τύπων αριθμητικής κινητής υποδιαστολής, μετατροπής από αριθμητική αναπαράσταση κινητής υποδιαστολής σε αριθμητική αναπαράσταση σταθερής υποδιαστολής και το αντίστροφο. Επιπλέον συμμορφώνεται πλήρως με τα πρότυπα IEEE-754 (με μόνο ελάχιστες τεκμηριωμένες αποκλίσεις).

Η λειτουργία κάθε πυρήνα διευκρινίζεται όταν αυτός παράγεται και κάθε παραλλαγή έχει μια κοινή διεπαφή. Αυτή η διεπαφή παρουσιάζεται στο Σχήμα 3.2-

1. Όταν ένας χρήστης επιλέγει μια λειτουργία που απαιτεί μόνο έναν τελεστέο, η είσοδος B παραλείπεται.



DS335_01_021508

Σχήμα 3.2-1. Σχηματικό διάγραμμα του γενικού δυαδικού πυρήνα κινητής υποδιαστολής.

Μερικά από τα βασικά σήματα αναλύονται παρακάτω:

A: Είσοδος τελεστέου A.

B: Είσοδος τελεστέου B.

OPERATION: Το σήμα OPERATION, όσο αφορά την εργασία, είναι παρόν στον πυρήνα πρόσθεσης/αφαίρεσης. Η μορφή είναι δυαδική και κωδικοποιείται ως 000000 για πρόσθεση και 000001 για αφαίρεση.

OPERATION_ND: Το σήμα OPERATION_ND πρέπει να ενεργοποιηθεί όταν οι τελεστέοι ισχύουν στις εισαγωγές A και B. Το μη ενεργοποιημένο OPERATION_ND αποτρέπει την έναρξη των νέων διαδικασιών και την επόμενη ενεργοποίηση του σήματος RDY.

RDY: Το σήμα RDY ενεργοποιείται από τον πυρήνα για να δείξει ότι η έξοδος είναι έγκυρη.

Σημαντικό στοιχείο στην σχεδίαση κάθε πυρήνα είναι η επιλογή του λανθάνοντος χρόνου (το διάστημα που μεσολαβεί μεταξύ δεδομένων και αποτελέσματος, latency). Το διάστημα αυτό μπορεί να τεθεί μεταξύ μηδέν και μιας μέγιστης αξίας που εξαρτάται από τις παραμέτρους που επιλέγονται.

Επίσης για την υλοποίηση κάθε πυρήνα απαιτείται ένας αριθμός δομικών μονάδων (πόρων) που προσφέρει η κάθε FPGA. Οι πόροι αυτοί είναι ένα σύνολο από LUTs (Look Up Tables), FFs (Flip Flops) και DSP48Es. Η επιλογή του λανθάνοντος χρόνου (latency) και της επιθυμητής συχνότητας λειτουργίας του κάθε πυρήνα προσδιορίζουν και το απαιτούμενο ποσοστό των πόρων που δεσμεύονται. Η χρήση των πόρων LUTs και FFs και η μέγιστη συχνότητα μειώνονται με τη μείωση του λανθάνοντος χρόνου. Η ελαχιστοποίηση του λανθάνοντος χρόνου ελαχιστοποιεί τους απαιτούμενους πόρους.

3.3 Χρήση δομικών πόρων από τους πυρήνες κινητής υποδιαστολής

Ο πυρήνας κινητής υποδιαστολής (FPC) απαιτεί έναν αριθμό δομικών μονάδων (πόρων) που παρέχει μια FPGA. Οι πόροι αυτοί είναι ένα σύνολο από LUTs (Look Up Tables), FFs (Flip Flops) και DSP48Es. Η επιλογή του λανθάνοντος χρόνου και της επιθυμητής συχνότητας λειτουργίας του κάθε πυρήνα προσδιορίζουν και το απαιτούμενο ποσοστό των πόρων που δεσμεύονται.

Πρακτικά κάθε σχεδίαση ενός συστήματος με αναδιατασσόμενη λογική προϋποθέτει και συγκεκριμένη επιλογή παραμέτρων για τον τρόπο σχεδίασης του πυρήνα. Οι πέντε προαναφερθείσες παράμετροι (LUTs, FFs, DSP48Es, λανθάνων χρόνος και συχνότητα λειτουργίας) πρέπει να προσαρμοστούν με τέτοιο τρόπο ώστε τελικά να ικανοποιούν τις εκάστοτε σχεδιαστικές ανάγκες, δηλαδή κατά περίπτωση να εντοπίζεται η χρυσή τομή ανάμεσα στην υψηλή συχνότητα λειτουργίας του συστήματος, την ελάχιστη χρήση πόρων και την τελική ταχύτητα του αλγορίθμου, που είναι και το ζητούμενο.

Μια υψηλή συχνότητα λειτουργίας είναι πάντα επιθυμητή, όμως αυτό κοστίζει υπερβολικά σε πόρους και χρόνο, δεν επιτρέπει την χρήση αρκετών παράλληλων συστημάτων στην ίδια FPGA και κατά συνέπεια δεν πετυχαίνει τα επιθυμητά αποτελέσματα στην ταχύτητα του αλγορίθμου.

Στον αντίποδα, μια σχεδίαση με ελάχιστη χρήση πόρων μειώνει τη συχνότητα λειτουργίας τόσο πολύ που καθιστά το όλο σύστημα τελικά ασύμφορο.

Τελικά σε κάθε σχεδίαση χρειάζεται να λαμβάνονται υπόψιν οι προαναφερθείσες παράμετροι για να θεωρηθεί όσο το δυνατόν βέλτιστη. Και επειδή η συγκεκριμένη σχεδίαση έχει ως βάση τους πυρήνες κινητής υποδιαστολής οι ίδιοι πρέπει να

σχεδιασθούν κάνοντας βέλτιστη χρήση των πόρων ώστε να ικανοποιούνται οι τελικές σχεδιαστικές απαιτήσεις (παραλληλία, ταχύτητα κλπ.).

Η εταιρεία Xilinx παρέχει κάποιες εκτιμήσεις πόρων, λανθάνοντος χρόνου και συχνότητας λειτουργίας για τους πυρήνες της, οι οποίοι στην περίπτωση μας είναι πυρήνες πολλαπλασιασμού, πρόσθεσης/αφαίρεσης και διαίρεσης. Οι εκτιμήσεις αυτές ελέγχθησαν πειραματικά και συμπληρώθηκαν από μετρήσεις για τις περιπτώσεις που δεν καλύπτονταν.

Στους Πίνακες 3.3-1 και 3.3-2 φαίνονται αναλυτικά οι σχέσεις των πέντε βασικών σχεδιαστικών παραμέτρων για κάθε πυρήνα.

Target Device: xc5vsx240t-2ff1738 - Design Tool: ISE 10.1.01 - WebPACK												
Resource Estimate for Floating-Point Modules												
Multiplier												
No Usage: Logic Only	latency 2	latency 3	latency 4	latency 5	latency 6	latency 7	latency 8					
LUTs (Look Up Tables)	664	701	594	625	625	635	690					
FFs (Flip-Flops)	145	244	433	524	592	635	690					
Speed (MHz)	92	172	247	253	313	316	334					
Medium Usage: 1 x DSP48E	latency 2	latency 3	latency 4	latency 5	latency 6	latency 7	latency 8					
LUTs (Look Up Tables)	242	250	255	262	259	260	262					
FFs (Flip-Flops)	53	92	133	237	240	327	379					
Speed (MHz)	96	192	188	280	284	327	379					
Full Usage: 2 x DSP48E	latency 2	latency 3	latency 4	latency 5	latency 6	latency 7	latency 8					
LUTs (Look Up Tables)	80	88	101	96	102	99	122					
FFs (Flip-Flops)	53	88	101	77	120	149	173					
Speed (MHz)	110	216	236	307	381	383	410					
Max Usage: 3 x DSP48E	latency 2	latency 3	latency 4	latency 5	latency 6							
LUTs (Look Up Tables)	102	106	106	86	97							
FFs (Flip-Flops)	52	59	75	111	114							
Speed (MHz)	176	231	275	359	410							
Adder/Subtractor												
No Usage: Logic Only	latency 2	latency 3	latency 4	latency 5	latency 6	latency 7	latency 8	latency 9	latency 10	latency 11	latency 12	
LUTs (Look Up Tables)	396	396	395	407	388	390	384	416	419	427	435	
FFs (Flip-Flops)	76	144	188	231	293	367	404	454	488	528	557	
Speed (MHz)	105	175	227	231	293	312	330	365	398	427	435	
Full Usage: 2 x DSP48E	latency 2	latency 3	latency 4	latency 5	latency 6	latency 7	latency 8	latency 9	latency 10	latency 11		
LUTs (Look Up Tables)	255	256	255	248	248	241	234	236	241	240		
FFs (Flip-Flops)	86	86	142	140	207	241	274	284	309	337		
Speed (MHz)	102	175	190	209	279	280	274	365	366	410		

Πίνακας 3.3-1. Επιλογή σχεδιαστικών παραμέτρων για πυρήνες πολλαπλασιασμού και πρόσθεσης/αφαίρεσης.

Divider						
No Usage: Logic Only	latency 5	latency 10	latency 12	latency 13	latency 14	latency 15
LUTs (Look Up Tables)	779	782	785	775	771	784
FFs (Flip-Flops)	264	511	512	512	709	710
Speed (MHz)	80	176	176	176	255	255

Πίνακας 3.3-2. Επιλογή σχεδιαστικών παραμέτρων για πυρήνες διαίρεσης.

Όπως προκύπτει και από τους παραπάνω πίνακες μπορούν να γίνουν κάποιες γενικές αλλά ιδιαίτερα σημαντικές παρατηρήσεις. Όσο αυξάνει ο λανθάνων χρόνος αυξάνει χαρακτηριστικά και η συχνότητα λειτουργίας. Επίσης αυξάνει και η χρήση των LUTs και πολύ περισσότερο των FFs. Η χρήση περισσότερων πόρων DSP48Es βελτιώνει την συχνότητα λειτουργίας του πυρήνα και μειώνει αισθητά την χρήση των LUTs και FFs. Όλα τα παραπάνω πρέπει να ληφθούν υπόψιν τόσο στην σχεδίαση/υλοποίηση του τελικού συστήματος όσο και στην επιλογή της κατάλληλης FPGA (που θα παρέχει τους πόρους).

3.4 Επιλογή FPGA

Η επιλογή μιας συγκεκριμένης FPGA σχετίζεται άμεσα με τις ιδιαιτερότητες του υπό σχεδίαση αλγορίθμου καθώς και τις προδιαγραφές του συστήματος (ταχύτητα, κατανάλωση ισχύος, ελάχιστη χρήση πόρων). Το ζητούμενο στην παρούσα εργασία είναι η μέγιστη ταχύτητα. Οι πυρήνες κινητής υποδιαστολής απαιτούν μεγάλο αριθμό LUTs, FFs και DSP48Es. Πρέπει να χρησιμοποιηθούν όσο το δυνατόν περισσότεροι τέτοιοι πυρήνες στη σχεδίαση με σκοπό την βελτιστοποίηση της ταχύτητας και της παραλληλίας του συστήματος. Οι παραπάνω λόγοι επιβάλλουν μια FPGA από την οικογένεια Virtex-5 της εταιρείας Xilinx [4].

Παρακάτω παρουσιάζεται ένας πίνακας των ιδιαίτερων χαρακτηριστικών των FPGA της οικογένειας Virtex-5.

Device	Configurable Logic Blocks (CLBs)			DSP48E Slices ⁽²⁾	Block RAM Blocks			CMTs ⁽⁴⁾	PowerPC Processor Blocks	Endpoint Blocks for PCI Express	Ethernet MACs ⁽⁵⁾	Max RocketIO Transceivers ⁽⁶⁾		Total I/O Banks ⁽⁸⁾	Max User I/O ⁽⁷⁾
	Array (Row x Col)	Virtex-5 Slices ⁽¹⁾	Max Distributed RAM (Kb)		18 Kb ⁽³⁾	36 Kb	Max (Kb)					GTP	GTX		
XC5VLX30	80 x 30	4,800	320	32	64	32	1,152	2	N/A	N/A	N/A	N/A	N/A	13	400
XC5VLX50	120 x 30	7,200	480	48	96	48	1,728	6	N/A	N/A	N/A	N/A	N/A	17	560
XC5VLX85	120 x 54	12,960	840	48	192	96	3,456	6	N/A	N/A	N/A	N/A	N/A	17	560
XC5VLX110	160 x 54	17,280	1,120	64	256	128	4,608	6	N/A	N/A	N/A	N/A	N/A	23	800
XC5VLX155	160 x 76	24,320	1,640	128	384	192	6,912	6	N/A	N/A	N/A	N/A	N/A	23	800
XC5VLX220	160 x 108	34,560	2,280	128	384	192	6,912	6	N/A	N/A	N/A	N/A	N/A	23	800
XC5VLX330	240 x 108	51,840	3,420	192	576	288	10,368	6	N/A	N/A	N/A	N/A	N/A	33	1,200
XC5VLX20T	60 x 26	3,120	210	24	52	26	936	1	N/A	1	2	4	N/A	7	172
XC5VLX30T	80 x 30	4,800	320	32	72	36	1,296	2	N/A	1	4	8	N/A	12	360
XC5VLX50T	120 x 30	7,200	480	48	120	60	2,160	6	N/A	1	4	12	N/A	15	480
XC5VLX85T	120 x 54	12,960	840	48	216	108	3,888	6	N/A	1	4	12	N/A	15	480
XC5VLX110T	160 x 54	17,280	1,120	64	296	148	5,328	6	N/A	1	4	16	N/A	20	680
XC5VLX155T	160 x 76	24,320	1,640	128	424	212	7,632	6	N/A	1	4	16	N/A	20	680
XC5VLX220T	160 x 108	34,560	2,280	128	424	212	7,632	6	N/A	1	4	16	N/A	20	680
XC5VLX330T	240 x 108	51,840	3,420	192	648	324	11,664	6	N/A	1	4	24	N/A	27	960
XC5VSX35T	80 x 34	5,440	520	192	168	84	3,024	2	N/A	1	4	8	N/A	12	360
XC5VSX50T	120 x 34	8,160	780	288	264	132	4,752	6	N/A	1	4	12	N/A	15	480
XC5VSX95T	160 x 46	14,720	1,520	640	488	244	8,784	6	N/A	1	4	16	N/A	19	640
XC5VSX240T	240 x 78	37,440	4,200	1,056	1,032	516	18,576	6	N/A	1	4	24	N/A	27	960
XC5VTX150T	200 x 58	23,200	1,500	80	456	228	8,208	6	N/A	1	4	N/A	40	20	680
XC5VTX240T	240 x 78	37,440	2,400	96	648	324	11,664	6	N/A	1	4	N/A	48	20	680
XC5VFX30T	80 x 38	5,120	380	64	136	68	2,448	2	1	1	4	N/A	8	12	360
XC5VFX70T	160 x 38	11,200	820	128	296	148	5,328	6	1	3	4	N/A	16	19	640
XC5VFX100T	160 x 56	16,000	1,240	256	456	228	8,208	6	2	3	4	N/A	16	20	680
XC5VFX130T	200 x 56	20,480	1,580	320	596	298	10,728	6	2	3	6	N/A	20	24	840
XC5VFX200T	240 x 68	30,720	2,280	384	912	456	16,416	6	2	4	8	N/A	24	27	960

Notes:

1. Virtex-5 slices are organized differently from previous generations. Each Virtex-5 slice contains four LUTs and four flip-flops (previously it was two LUTs and two flip-flops.)
2. Each DSP48E slice contains a 25 x 18 multiplier, an adder, and an accumulator.
3. Block RAMs are fundamentally 36 Kbits in size. Each block can also be used as two independent 18-Kbit blocks.
4. Each Clock Management Tile (CMT) contains two DCMs and one PLL.
5. This table lists separate Ethernet MACs per device.
6. RocketIO GTP transceivers are designed to run from 100 Mb/s to 3.75 Gb/s. RocketIO GTX transceivers are designed to run from 150 Mb/s to 6.5 Gb/s.
7. This number does not include RocketIO transceivers.
8. Includes configuration Bank 0.

Πίνακας 3.4-1. Πίνακας με τα χαρακτηριστικά των FPGA της οικογένειας Virtex-5

Δεδομένου ότι ο αλγόριθμος απαρτίζεται κυρίως από πράξεις κινητής υποδιαστολής και οι πυρήνες της Xilinx που υλοποιούν αυτές τις πράξεις απαιτούν μεγάλο αριθμό πόρων (LUTs, FFs, DSP48Es) επιλέχθηκε το μοντέλο XC5VSX240T το οποίο, όπως φαίνεται και στον παραπάνω πίνακα, προσφέρει (μέχρι τη στιγμή που γράφεται αυτό το κείμενο) τον μεγαλύτερο αριθμό Look Up Tables (4 x 37,440), Flip Flops (4 x 37,440) και DSP48Es (1,056).

4. Σχεδίαση και υλοποίηση

Διάφοροι σχεδιαστικοί μέθοδοι δοκιμάστηκαν. Τελικά, και λόγω της ιδιαιτερότητας του αλγορίθμου, επελέγησαν δύο μέθοδοι και ο συνδυασμός τους.

4.1 Συναρτήσεις

Ο κεντρικός κώδικας του CNPE αλγορίθμου είναι ο CNPE.c, ο οποίος εκτελεί την μέθοδο CN-2. Αν και οι δύο αλγόριθμοι, CNPE και GFPE, έχουν σχεδιαστεί για αριθμούς διπλής ακρίβειας (double precision) η σχεδίαση σε υλικό, μέχρι τη στιγμή που γράφεται το κείμενο, προϋποθέτει αριθμητική απλής ακρίβειας (single precision). Ο CNPE αλγόριθμος είναι αυτός που ανταποκρίνεται εξίσου καλά και σε αριθμητική μονής ακρίβειας και αυτός είναι και ο βασικός λόγος για την επιλογή του να σχεδιαστεί σε αναδιατασσόμενη λογική.

Από τον CNPE αλγόριθμο ενδιαφέρον παρουσιάζει η CNPE συνάρτηση, μέρος της οποίας τελικά επελέγη για να σχεδιαστεί με αναδιατασσόμενη λογική. Όταν καλείται αυτή η συνάρτηση ξεκινάει μια επανάληψη για NR-1 φορές (όπου στον κώδικα NR = 1005, οριζόντιος άξονας) και σε κάθε επανάληψη καλούνται με τη σειρά τους τρεις κύριες συναρτήσεις (rhs, amat, trid). Οι συναρτήσεις αυτές υλοποιούν μια επανάληψη για NZ-1 φορές (όπου στον κώδικα NZ = 501, κάθετος άξονας) όπου σε κάθε επανάληψη υλοποιείται ένας αριθμός πράξεων (πολλαπλασιασμοί, προσθέσεις, αφαιρέσεις και διαιρέσεις) μιγαδικών (πραγματικό και φανταστικό μέρος) αριθμών (κινητής υποδιαστολής) μονής ακρίβειας. Μόνη διαφοροποίηση αποτελούν οι συνοριακές συνθήκες (δηλαδή το πρώτο και το τελευταίο στοιχείο των εκάστοτε διανυσμάτων) οι οποίες κατά περίπτωση αντιμετωπίζονται διαφορετικά. Μια σχηματική αναπαράσταση των παραπάνω σε ψευδογλώσσα φαίνεται στο Σχήμα 4.1-1.

```
CNPE( for( i=1; i<NR; i++) ){  
  
    bc()  
    rhs( {for( j=0; j<NZ; j++) } )  
    bc()  
    amat( {for( j=0; j<NZ; j++) } )  
    trid( {for( j=0; j<NZ; j++)  
        |         for( j=0; j<NZ; j++) } )  
    bc()  
  
}
```

Σχήμα 4.1-1. Η συνάρτηση CNPE περιέχει επανάληψη για NR φορές, οι συναρτήσεις rhs και amat για NZ φορές ενώ η trid περιέχει δύο επαναλήψεις για NZ φορές. Η συνάρτηση bc (boundary conditions) δεν περιέχει επανάληψη και ικανοποιεί απλά τις εκάστοτε συνοριακές συνθήκες.

Αρχικά φαίνεται ότι για την υλοποίηση του αλγορίθμου απαιτείται ένας μεγάλος αριθμός επαναλήψεων. Έτσι, αν και οι απαιτούμενες πράξεις φαίνονται λίγες, οι διπλές επαναλήψεις που προκύπτουν καθιστούν τον κώδικα υπολογιστικά «ακριβό». Παρ' όλα αυτά, όσο μεγάλος και να είναι ο αριθμός των απαιτούμενων πράξεων, κατά βάση είναι τελικά ίδιες και επαναλαμβανόμενες. Επίσης η μορφή των πράξεων είναι ίδια παντού, δηλαδή πράξεις αριθμών κινητής υποδιαστολής απλής ακρίβειας. Από την άλλη πλευρά όμως, και με μια πιο ενδελεχή ματιά στον κώδικα, προκύπτει ότι κάθε βήμα/συνάρτηση του αλγορίθμου εξαρτάται από το προηγούμενο. Δηλαδή, στις περισσότερες περιπτώσεις, κάθε βήμα/συνάρτηση απαιτεί ένα προηγούμενο αποτέλεσμα για να μπορεί να υλοποιηθεί. Περισσότερες λεπτομέρειες για την αλληλεξάρτηση των διάφορων πράξεων σε κάθε βήμα παρατίθενται παρακάτω.

Στην σχεδίαση του παραπάνω αλγορίθμου λοιπόν σε αναδιατασσόμενη λογική οφείλουν να ληφθούν υπόψιν οι παραπάνω ιδιαιτερότητες, δηλαδή η επανάληψη όμοιων πράξεων, οι αριθμοί κινητής υποδιαστολής απλής ακρίβειας καθώς και η εξάρτηση κάθε «βήματος» από το προηγούμενό του.

4.2 Τμήματα

Η επανάληψη που προκύπτει από την CNPE συνάρτηση δεν υλοποιείται άμεσα σε υλικό. Πρακτικά σχεδιάζεται ένα σύστημα που λύνει την συνάρτηση CNPE για μία επανάληψη και ανά συγκεκριμένους κύκλους τροφοδοτείται με νέα δεδομένα. Το ίδιο ισχύει και για τις εσωτερικές συναρτήσεις. Κάθε λύση που προκύπτει είναι για ένα συγκεκριμένο ζευγάρι ij . Κάθε επιλύτης «ξεκινάει» για ένα δεδομένο i (που αντιστοιχεί σε ένα συγκεκριμένο σημείο στον οριζόντιο άξονα) και επαναλαμβάνεται για j από 0 μέχρι NZ . Τα διαφορετικά j -βήματα δεν είναι ανεξάρτητα μεταξύ τους, επειδή υπάρχει άμεση αλληλεξάρτηση δεδομένων μεταξύ των $j-1$, j και $j+1$ στοιχείων.

Μια παρουσίαση των αριθμητικών πράξεων για τις τρεις βασικές συναρτήσεις ($amat$, rhs , $trid$) καθώς και των υποσυναρτήσεών τους φαίνεται στα Σχήματα 4.2-1, 4.2-2, 4.2-3.

```
void amat(){
    for(0, NZ, ++){
        cca.re= hr * ca[i][j].re, cca.im= hr * ca[i][j].im;
        ccb.re= hr2 * cb[i][j].re, ccb.im= hr2 * cb[i][j].im;
        ccc.re= hr * cc[i][j].re, ccc.im= hr * cc[i][j].im;
        al[j].re= ccb.re - ccc.re, al[j].im= ccb.im - ccc.im;
        ad[j].re= 1.- cca.re + 2. * ccc.re, ad[j].im= - cca.im + 2. * ccc.im;
        au[j].re= - ccb.re - ccc.re, au[j].im= - ccb.im - ccc.im;
    }
}
```

Σχήμα 4.2-1. Η συνάρτηση $amat$ περιέχει μια σειρά από πράξεις πολλαπλασιασμού, πρόσθεσης και αφαίρεσης αριθμών κινητής υποδιαστολής.

```
void rhs(){
    for(0, NZ, ++){
        cca.re = ca[im1][j].re*hr, cca.im = ca[im1][j].im*hr;
        ccb.re = cb[im1][j].re*hr2, ccb.im = cb[im1][j].im*hr2;
        ccc.re = cc[im1][j].re*hr, ccc.im = cc[im1][j].im*hr;
        bl.re = - ccb.re + ccc.re, bl.im= - ccb.im + ccc.im;
        bm.re = 1.+ cca.re - 2.*ccc.re, bm.im= cca.im - 2.*ccc.im;
        bu.re = ccb.re + ccc.re, bu.im= ccb.im + ccc.im;
        ZmulZ( bl, u[j-1], cmp1), ZmulZ( bm, u[j], cmp2), ZmulZ( bu, u[j+1], cmp3);
        uu[j].re=cmp1.re+cmp2.re+cmp3.re, uu[j].im=cmp1.im+cmp2.im+cmp3.im;
    }
}
```

Σχήμα 4.2-2. Η συνάρτηση rhs περιέχει μια σειρά από πράξεις πολλαπλασιασμού, πρόσθεσης και αφαίρεσης αριθμών κινητής υποδιαστολής.

```
void trid(){
    for(0,NZ,++){
        ZdivZ( &au[k-1], &bb, &c[k]);
        ZsubZmulZ( &am[k], &al[k], &c[k], &bb);
        ZsubZmulZ( &b[k], &al[k], &u[k-1], &cmp1);
        ZdivZ( &cmp1, &bb, &u[k]);
    }

    for(NZ,1,--){
        ZmulZ( c[k], u[k], cmp2);
        u2[k-1].re = u2[k-1].re - cmp2.re;
        u2[k-1].im = u2[k-1].im - cmp2.im;
    }
}
```

Part 1

Part 2

Σχήμα 4.2-3. Η συνάρτηση trid περιέχει μια σειρά από πράξεις πολλαπλασιασμού, πρόσθεσης και αφαίρεσης αριθμών κινητής υποδιαστολής.

```
ZmulZ(){
    Z3.re=Z1.re*Z2.re - Z1.im*Z2.im;
    Z3.im=Z1.re*Z2.im + Z1.im*Z2.re;
}

Znorm(){
    x=Z->re;
    y=Z->im;
    return sqrt( x*x + y*y);
}

ZdivZ(){
    x= Z2.re * Z2.re + Z2.im * Z2.im ;
    Z3.re= (Z1.re * Z2.re + Z1.im * Z2.im ) / x;
    Z3.im= (-Z1.re * Z2.im + Z1.im * Z2.re ) / x;
}

ZsubZmulZ(){
    Z3->re=Z->re - (Z1->re*Z2->re - Z1->im*Z2->im);
    Z3->im=Z->im - (Z1->re*Z2->im + Z1->im*Z2->re);
}
```

Σχήμα 4.2-4. Βοηθητικές συναρτήσεις που καλούνται από τις rhs, amat και trid.

Στις πράξεις προστίθεται και η διαίρεση.

Όπως φαίνεται και στο Σχήμα 4.2-3 η συνάρτηση trid αποτελείται από δύο μέρη. Στο πρώτο μέρος παράγεται το j στοιχείο ξεκινώντας από το μηδέν μέχρι το NZ. Στο δεύτερο μέρος η επανάληψη ξεκινάει από το NZ, μειώνει μέχρι το 1 και κάνοντας κάθε φορά χρήση του εκάστοτε j στοιχείου παράγει το j-1. Κάτι τέτοιο, όσον αφορά τη σχεδίαση σε υλικό, απαιτεί ένα στάδιο προσωρινής αποθήκευσης

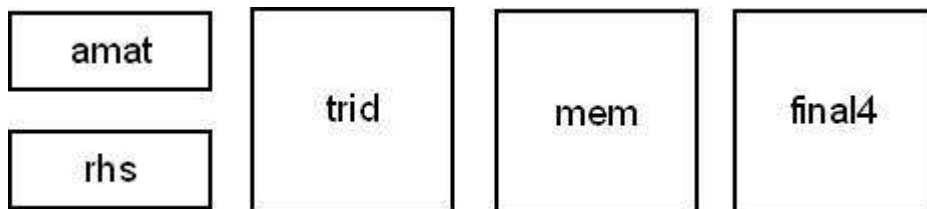
των διανυσμάτων u και c σε μνήμη στο πρώτο μέρος ώστε να μπορέσει στο επόμενο μέρος να χρησιμοποιηθούν για την αντίστροφη επανάληψη (δεύτερο μέρος). Έτσι η συνάρτηση `trid` χωρίζεται σε δύο τμήματα (`trid` και `final4`), όπως φαίνεται στο Σχήμα 4.2-5 που ανάμεσά τους παρεμβάλλεται το στάδιο της προσωρινής αποθήκευσης σε μνήμη.

```
void trid(){
    for(0,NZ,++)
        ZdivZ( &au[k-1], &bb, &c[k]);
        ZsubZmulZ( &am[k], &al[k], &c[k], &bb);
        ZsubZmulZ( &b[k], &al[k], &u[k-1], &cmp1);
        ZdivZ( &cmp1, &bb, &u[k]);
    }

void final4(){
    for(NZ,1,--)
        ZmulZ( c[k], u[k], cmp2);
        u2[k-1].re = u2[k-1].re - cmp2.re;
        u2[k-1].im = u2[k-1].im - cmp2.im;
    }
```

Σχήμα 4.2-5. Οι συναρτήσεις `trid` και `final4` μετά την διάσπαση της αρχικής `trid`.

Όπως φαίνεται και από τον κώδικα, κάθε συνάρτηση απαιτεί αποτέλεσμα από την προηγούμενη της και με τη σειρά της καθυστερεί κάποιους κύκλους για να παράξει η ίδια αποτέλεσμα. Αυτό αποκλείει κάθε περίπτωση παραλληλίας, τουλάχιστον σε αυτό το σημείο. Μοναδική εξαίρεση αποτελούν οι συναρτήσεις `amat` και `rhs`, οι οποίες, όντας ανεξάρτητες μεταξύ τους, μπορούν να δουλέψουν ταυτόχρονα και με τη σειρά τους να τροφοδοτήσουν την `trid` με τιμές. Το σχηματικό διάγραμμα της σχεδίασης στο οποίο φαίνονται τα διαφορετικά τμήματα φαίνεται στο Σχήμα 4.2-6.



Σχήμα 4.2-6. Τα τμήματα της σχεδίασης του αλγορίθμου με μόνη δυνατή παραλληλία στις πρώτες δύο συναρτήσεις (`amat` και `rhs`).

Κάθε τμήμα από αυτά που παρουσιάζονται στο Σχήμα 4.2-6 είναι ένα σύνολο από πράξεις κινητής υποδιαστολής οι οποίες υλοποιήθηκαν με χρήση λογικής και

πυρήνων κινητής υποδιαστολής. Το μόνο τμήμα που διαφέρει και η σχεδίαση του έγινε με χρήση μνημών και λογικής από την αρχή είναι το τμήμα της μνήμης mem.

Λόγω της σειριακής φύσης του αλγορίθμου (στα τριδιαγώνια συστήματα κάθε βήμα εξαρτάται από το προηγούμενο και δημιουργεί το επόμενο), δεν είναι εφικτή η χρήση παραλληλίας ή εσωτερικής διοχέτευσης (pipelining) σε κάθε στάδιο, η παρούσα σχεδίαση επικεντρώνεται σε δύο στόχους. Πρώτος στόχος είναι η επιτάχυνση του συστήματος με σκοπό την όσο το δυνατόν πιο γρήγορη δημιουργία κάθε διανύσματος (σταθερό i , αυξανόμενο j). Δεύτερος στόχος είναι ο περιορισμός των πόρων που απαιτεί ο κάθε επιλύτης έτσι ώστε στο συνολικό σχέδιο να μπορούν να χρησιμοποιηθούν παράλληλα όσο το δυνατόν περισσότεροι επιλύτες (σε αυτό το επίπεδο η παραλληλία είναι εφικτή) για διαφορετικά «ύψη».

4.3 Σχεδίαση και υλοποίηση του τμήματος μνήμης

Όπως προαναφέρθηκε τα περισσότερα τμήματα του αλγορίθμου είναι σχεδιαστικά παρόμοια, καθώς απαιτούν απλά έναν αριθμό αλληλοεξαρτώμενων πράξεων. Διαφοροποίηση παρουσιάζει το τμήμα της μνήμης, το οποίο σχεδιάστηκε ξεχωριστά.

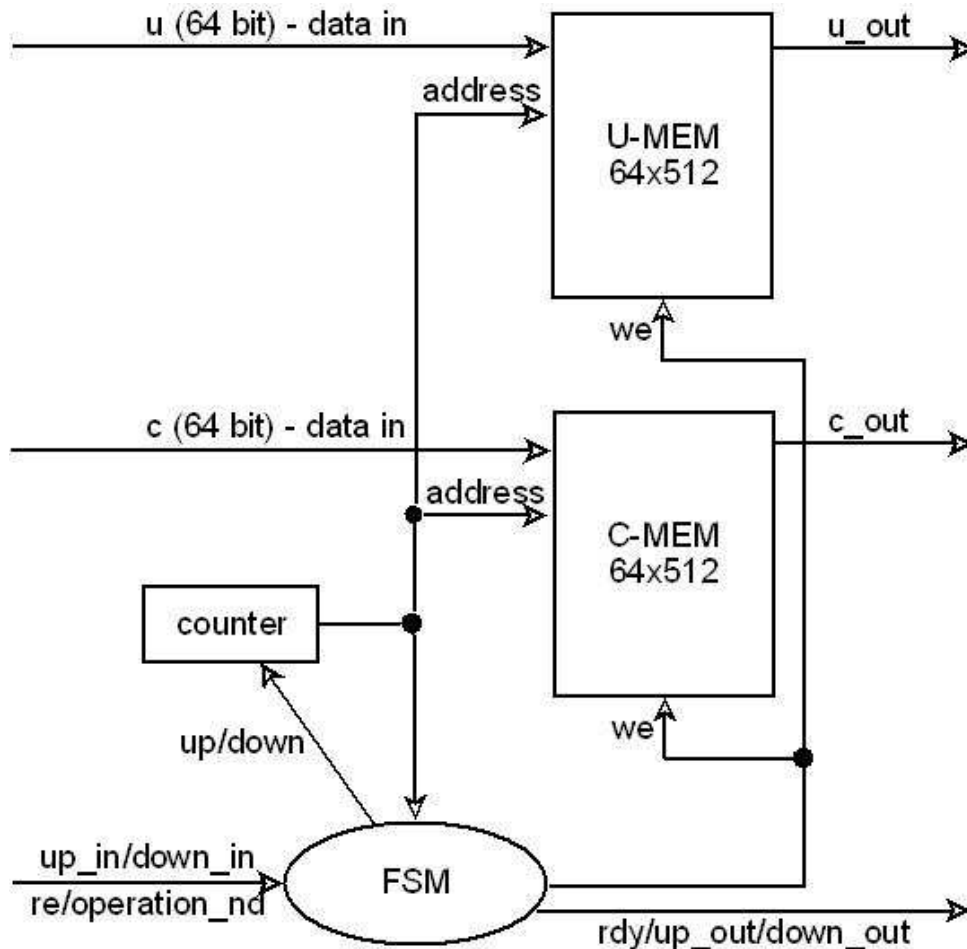
Σκοπός του συγκεκριμένου τμήματος είναι να «αντιστρέφει» τη σειρά με την οποία λαμβάνει δεδομένα από τα τμήματα που προηγούνται (Σχήμα 4.3-1). Τα δεδομένα εισάγονται σε τυχαίους κύκλους και μόνο ένα σήμα να επιβεβαιώνει την εγκυρότητά τους.

$$\begin{array}{l} \text{input} : \{u(511).re, u(511).im\} \Rightarrow \{u(510).re, u(510).im\} \Rightarrow \dots \Rightarrow \{u(1).re, u(1).im\} \Rightarrow \{u(0).re, u(0).im\} \Rightarrow \\ \text{output} : \{u(0).re, u(0).im\} \Rightarrow \{u(1).re, u(1).im\} \Rightarrow \dots \Rightarrow \{u(510).re, u(510).im\} \Rightarrow \{u(511).re, u(511).im\} \Rightarrow \end{array}$$

Σχήμα 4.3-1. Ο τρόπος εισαγωγής των δεδομένων (πρώτη σειρά) και ο τρόπος εξαγωγής τους (δεύτερη σειρά) του τμήματος mem.

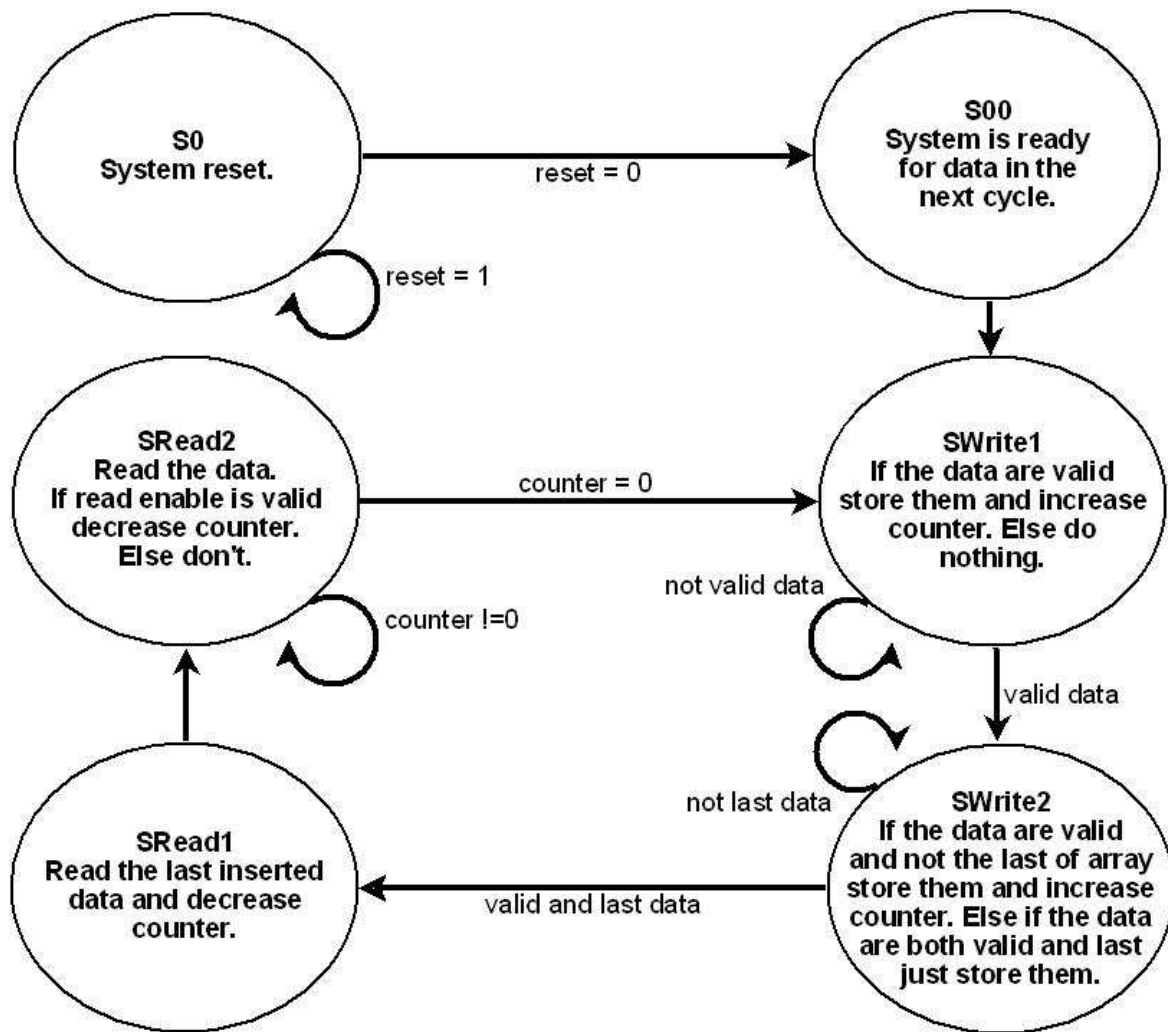
Χρησιμοποιήθηκαν δύο μνήμες 64 επί 512 bit (κάθε αριθμός κινητής υποδιαστολής έχει μήκος 32 bit και κάθε διάνυσμα έχει πραγματικό και φανταστικό μέρος, $32+32=64$ bit) και ένας μετρητής με δυνατότητα αύξησης και μείωσης της εξόδου. Η λειτουργία όλων των στοιχείων συγχρονίζεται από μία μηχανή πεπερασμένων καταστάσεων (FSM) η οποία δεχόμενη ανα διαστήματα τα σωστά

σήματα ενεργοποίησης αυξομειώνει την τιμή του μετρητή, ενεργοποιεί τα σήματα εγγραφής των μνημών και φροντίζει να μεταβαίνει στα κατάλληλα στάδια ανάγνωσης ή εγγραφής (Σχήμα 4.3-2).



Σχήμα 4.3-2. Οι δύο μνήμες της σχεδίασης (U-MEM, C-MEM), ο μετρητής (counter) και η FSM συγχρονισμού λειτουργίας.

Πιο συγκεκριμένα, και όπως φαίνεται στο Σχήμα 4.3-3 μόλις ενεργοποιηθεί το σύστημα η FSM ελέγχει τα εισερχόμενα δεδομένα. Αν συνοδεύονται από σήμα εγκυρότητας και αρχής διανύσματος καταγράφονται στη θέση μηδέν των μνημών και αυξάνεται ο μετρητής κατά ένα. Κάθε έγκυρο ζευγάρι δεδομένων τοποθετείται στη μνήμη κατά αύξουσα σειρά. Αν παρατηρηθούν έγκυρα μεν δεδομένα αλλά συνοδεύονται από σήμα τέλους διανύσματος, η FSM περνάει από το στάδιο της εγγραφής στο στάδιο της ανάγνωσης. Τότε, όποτε ζητούνται δεδομένα (από το σήμα ενεργοποίησης ανάγνωσης) επιστρέφονται με αντίστροφη σειρά (η τιμή του μετρητή μειώνεται) σε σχέση με εκείνη που ελήφθησαν (last in, first out). Μόλις η τιμή του μετρητή γίνεται μηδέν η FSM επιστρέφει σε κατάσταση αναμονής.



Σχήμα 4.3-3. Η λειτουργία της FSM για το τμήμα mem.

4.4 Σχεδίαση τμημάτων

Τα υπόλοιπα τμήματα της σχεδίασης (rhs, amat, trid, final4) σχεδιάστηκαν με δύο τρόπους.

4.4.1 Πρώτη μέθοδος (αναλυτική μορφή)

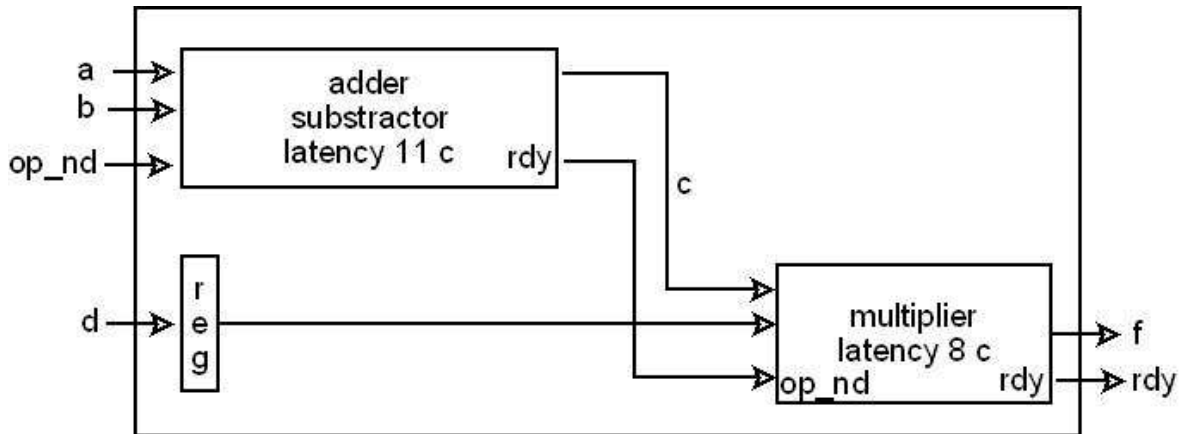
Η πρώτη μέθοδος περιλαμβάνει αναλυτική (structural) μορφή σχεδίασης κάνοντας χρήση όλων των απαιτούμενων πυρήνων κινητής υποδιαστολής (floating point core, FPC) καθώς και καταχωρητών που απαιτούνταν για την καθυστέρηση διάδοσης σημάτων. Η συγκεκριμένη μέθοδος υλοποίησης συνοψίζεται στο παρακάτω παράδειγμα.

Παράδειγμα: Έστω ότι χρειάζεται να γίνουν οι παρακάτω πράξεις

$$a + b = c$$

$$c * d = f$$

όπου τα a,b και d είναι γνωστά και f το ζητούμενο. Η σχεδίαση με αναλυτική μορφή φαίνεται στο Σχήμα 4.4.1-1.

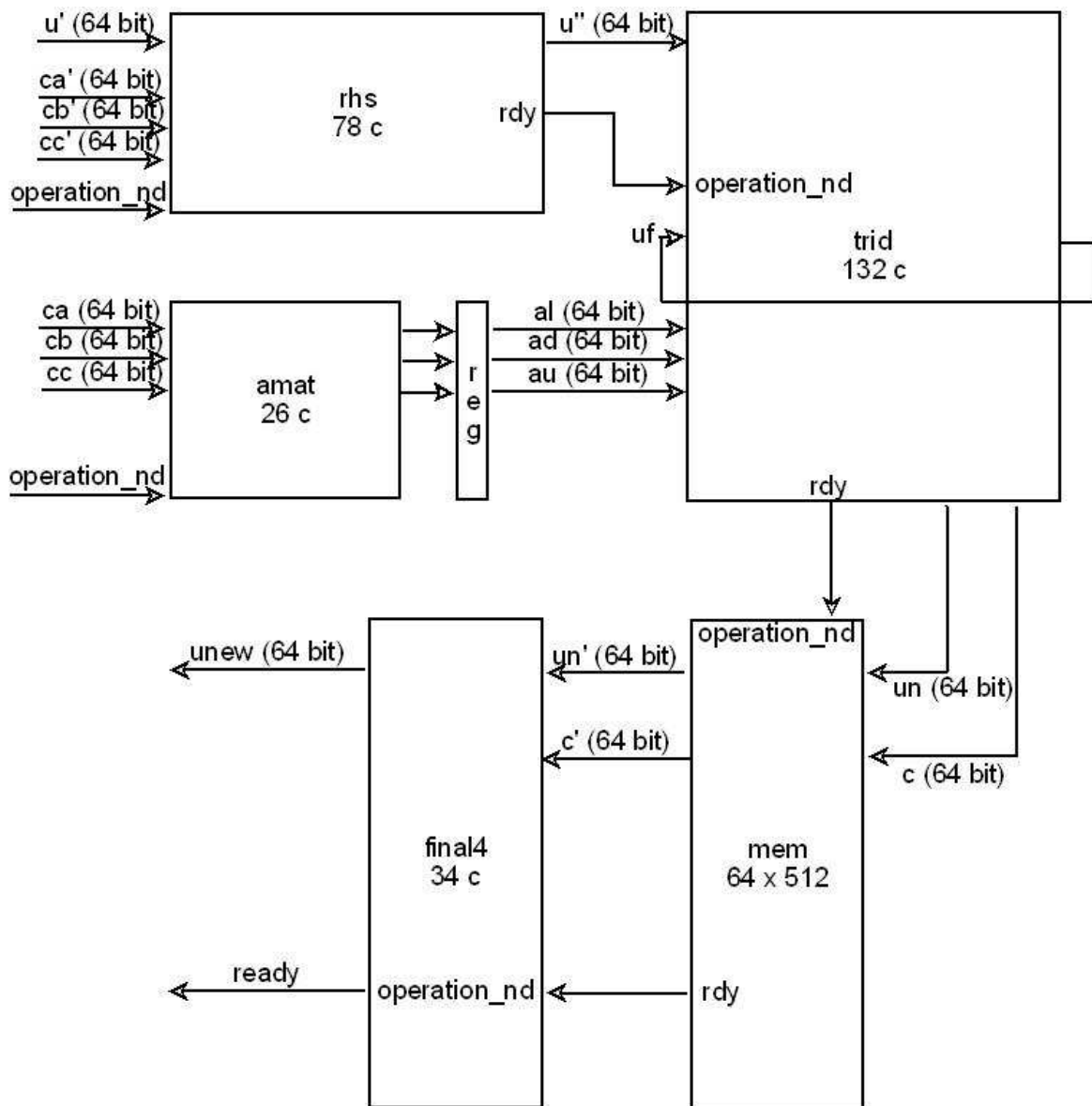


Σχήμα 4.4.1-1. Η σχεδίαση των πράξεων του παραδείγματος.

Ακολουθώντας την λογική του παραπάνω παραδείγματος κάθε συνάρτηση είναι δυνατόν να αναπαρασταθεί με αναλυτικό τρόπο επειδή αποτελείται από ένα σύνολο πράξεων (πολλαπλασιασμός, πρόσθεση, αφαίρεση, διαίρεση). Δεν παρατίθεται αναλυτικά το σχέδιο για την υλοποίηση κάθε συνάρτησης καθώς η φιλοσοφία παραμένει ακριβώς η ίδια (εξ'ολοκλήρου σχεδίαση με FPC και καταχωρητές καθυστέρησης) με την μόνη διαφορά ότι μεγαλώνει ο αριθμός των απαιτούμενων πράξεων.

Οι διεπαφές που παρουσιάζονται στους FPC διατηρούνται και στα ανώτερα επίπεδα της σχεδίασης. Στο παράδειγμα του Σχήματος 4.4.1-1 και όπως γνωρίζουμε και από το Κεφάλαιο 3.2, κάθε πυρήνας έχει ένα σήμα εγκυρότητας δεδομένων εισόδου (operation_nd) και ένα σήμα εγκυρότητας δεδομένων εξόδου (rdy). Έτσι το σήμα έγκυρης εξόδου από τον αθροιστή αποτελεί και σήμα έγκυρης εισόδου στον πολλαπλασιαστή και κατά συνέπεια μετά από 19 κύκλους (λανθάνων χρόνος: $8+11=19$ κύκλοι) προκύπτει η ζητούμενη τιμή του f συνοδευόμενη και από το σήμα έγκυρης εξόδου του πολλαπλασιαστή. Με τη ίδια λογική κάθε ανώτερο επίπεδο σχεδίασης διατηρεί τα ίδια σήματα έτσι ώστε η τελική σχεδίαση όλων των τμημάτων/συναρτήσεων να διατηρεί τις διεπαφές των

FPC και κατά συνέπεια να είναι δυνατός ο συνδυασμός τους όπως φαίνεται στο Σχήμα 4.4.1-2.



Σχήμα 4.4.1-2. Το τελικό σχέδιο. Φαίνεται πως το σήμα rdy κάθε τμήματος γίνεται το $operation_nd$ του επόμενου.

Αξιολογώντας την παραπάνω σχεδίαση, όσο αξιόπιστη και σχεδιαστικά απλή μπορεί να φαίνεται, στην περίπτωση του συγκεκριμένου αλγορίθμου κρίνεται μη αποδοτική. Κάνοντας χρήση της παραπάνω αναλυτικής σχεδίασης δίνεται η δυνατότητα στους FPC να αποδώσουν τη μέγιστη δυνατή (βάση τιμών της Xilinx) συχνότητα λειτουργίας καθώς και να λειτουργούν παρέχοντας την δυνατότητα για πλήρη εσωτερική διοχέτευση (pipelining). Δηλαδή από τη στιγμή που θα αρχίσει ο κάθε πυρήνας να λειτουργεί για την κάθε ομάδα τιμών είναι έτοιμος στον αμέσως

επόμενο κύκλο να δεχτεί την επόμενη. Στην συγκεκριμένη περίπτωση του αλγορίθμου που μελετάται στην παρούσα διπλωματική εργασία μια τέτοια χρήση δεν είναι εφικτή καθώς κάθε στάδιο/βήμα (όπως αναλύθηκε σε προηγούμενο κεφάλαιο) χρειάζεται δεδομένα εισόδου από προηγούμενα στάδια καθώς και σε κάποιες περιπτώσεις ανατροφοδότηση στην είσοδο από τα ίδια δεδομένα εξόδου του. Το γεγονός αυτό καθιστά την σχεδίαση αυτή κατά περίπτωση ασύμφορη αφού όχι μόνο το μεγαλύτερο μέρος του υλικού μένει σχεδόν ανεκμετάλλευτο (στο διάστημα που μεσολαβεί μέχρι την τροφοδότηση της σχεδίασης με νέα δεδομένα) αλλά και για την υλοποίηση απαιτείται πολύ μεγάλος αριθμός FPC και κατά συνέπεια πόρων (LUTs, FFs, DSP48Es) της FPGA. Έτσι ο στόχος για όσο το δυνατόν περισσότερων επιλυτών που καταναλώνουν ελάχιστους πόρους δεν μπορεί να επιτευχθεί.

4.4.2 Δεύτερη μέθοδος (FSM & RF)

Ο δεύτερος τρόπος σχεδίασης περιλαμβάνει και αυτός την χρήση FPC για τις πράξεις κινητής υποδιαστολής αλλά υλοποιείται με χρήση FSM και ενός αρχείου καταχωρητών. Σκοπός αυτής της υλοποίησης είναι η ελαχιστοποίηση της χρήσης των πόρων της FPGA αφού για κάθε στάδιο δεσμεύεται μια μονάδα για κάθε είδος πράξης.

Σαν πρώτο βήμα αποθηκεύονται στο αρχείο καταχωρητών οι αρχικές τιμές που απαιτεί κάθε συνάρτηση. Έπειτα και αφού έχει σχεδιαστεί ένα συγκεκριμένο χρονοπρόγραμμα μοναδικό για κάθε συνάρτηση η FSM αναλαμβάνει τον συγχρονισμό. Συνδέοντας σε κάθε κύκλο τις κατάλληλες εξόδους του αρχείου καταχωρητών με τις εισόδους των FPC και τις εξόδους των FPC με τις κατάλληλες εισόδους στο αρχείο καταχωρητών επιτυγχάνεται επαναχρησιμοποίηση των μονάδων στο μέγιστο βαθμό. Έτσι αφού ξεκινήσει να γίνεται μια πράξη σε έναν κύκλο στον επόμενο αντί ο πυρήνας να μείνει αχρησιμοποίητος υλοποιεί μια άλλη πράξη που χρειάζεται, η οποία δεν έχει εξάρτηση με την πρώτη. Η συγκεκριμένη μέθοδος υλοποίησης αναλύεται καλύτερα στο παρακάτω παράδειγμα.

Παράδειγμα: Έστω ότι χρειάζεται να γίνουν οι παρακάτω πράξεις

$$(a \cdot b) - c = x$$

$$(d + c) \cdot (b + a) = y$$

$$x \cdot y = z$$

όπου τα a, b, c, d είναι γνωστά και x, y, z τα ζητούμενα. Θεωρούμε ότι διατίθεται μία μονάδα πολλαπλασιασμού και μία πρόσθεσης/αφαίρεσης με λανθάνοντα χρόνο 2 και 4 κύκλους αντίστοιχα.

Οι παραπάνω σχέσεις κάνοντας χρήση βοηθητικών μεταβλητών t_1, t_2, t_3 , μετασχηματίζονται σε

$$a \cdot b = t_1$$

$$t_1 - c = x$$

$$d + c = t_2$$

$$b + a = t_3$$

$$t_2 \cdot t_3 = y$$

$$x \cdot y = z$$

Το χρονοπρόγραμμα για το συγκεκριμένο παράδειγμα αναλύεται στον Πίνακα 4.4.2-1.

	mul_in A	mul_in B	mul_out	add_in A	add_in B	add_out
1	a	b		d(+)	c	
2				b(+)	a	
3			t1			
4				t1(-)	c	t2
5						t3
6	t2	t3				
7						
8			y			x
9	x	y				
10						
11			z			

Πίνακας 4.4.2-1. Το χρονοπρόγραμμα με το σύνολο των πράξεων. Φαίνεται σε ποιον κύκλο ξεκινάνε οι πράξεις και σε ποιόν κύκλο επιστρέφονται τα αποτελέσματα.

Κατόπιν σχεδιάζοντας αρχείο καταχωρητών 5 θέσεων, οι αρχικές τιμές a, b, c, d τοποθετούνται στις θέσεις από 0 έως 3 αντίστοιχα, όπως φαίνεται τον Πίνακα 4.4.2-2.

Register File	
0	a
1	b
2	c
3	d
4	

Πίνακας 4.4.2-2. Το αρχείο καταχωρητών αρχικά.

Στη συνέχεια οι μεταβλητές στο χρονοπρόγραμμα αντικαθίστανται με θέσεις μνήμης στο αρχείο καταχωρητών.

	mul_in A	mul_in B	mul_out	add_in A	add_in B	add_out
1	RF(0)	RF(1)		RF(3)	RF(2)	
2				RF(1)	RF(0)	
3			RF(4)			
4				RF(4)	RF(2)	RF(0)
5						RF(1)
6	RF(0)	RF(1)				
7						
8			RF(3)			RF(2)
9	RF(2)	RF(3)				
10						
11			RF(4)			

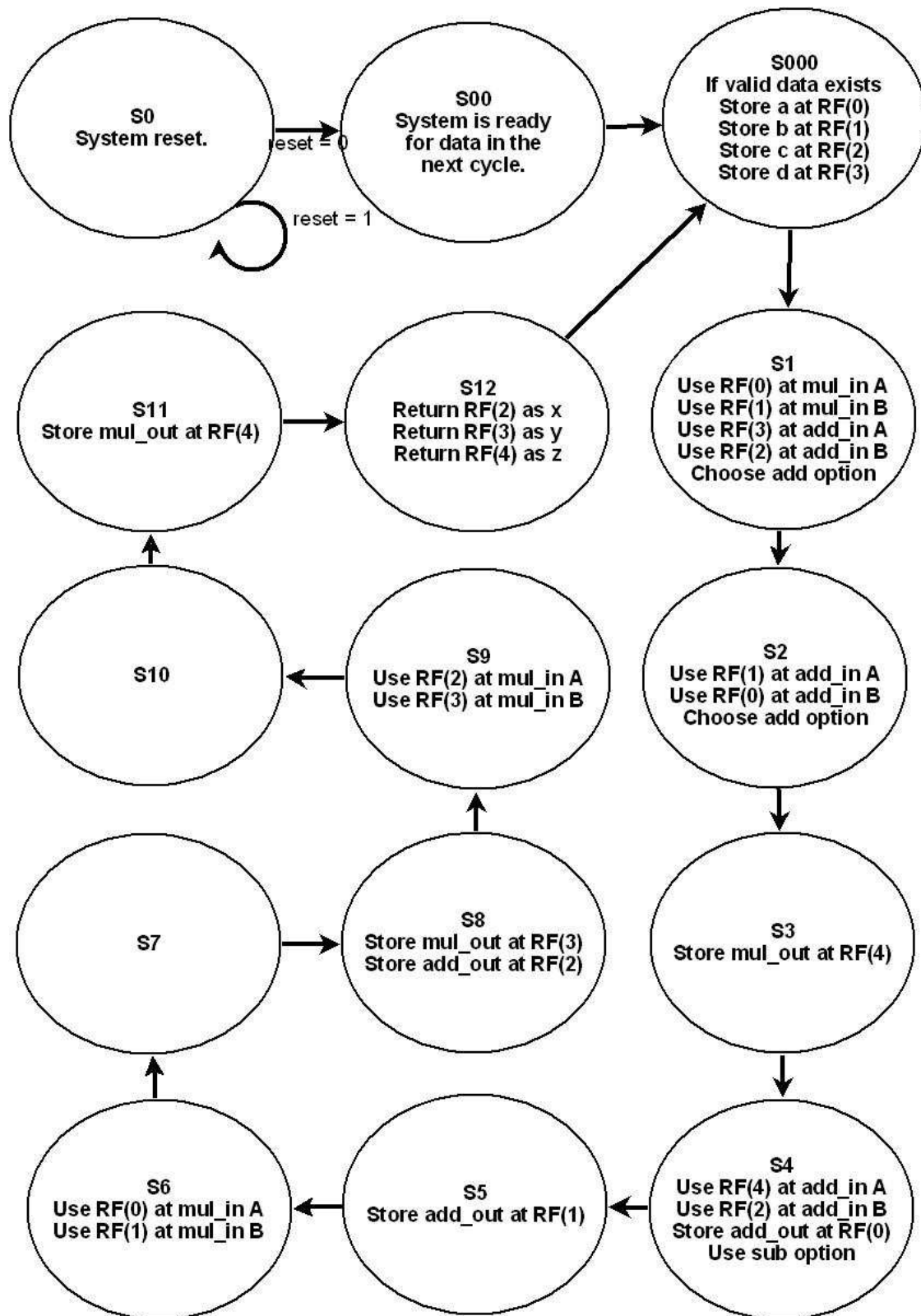
Πίνακας 4.4.2-3. Το χρονοπρόγραμμα με το σύνολο των πράξεων. Οι μεταβλητές έχουν αντικατασταθεί από θέσεις μνήμης.

Τελικά οι τιμές των x, y, z θα βρίσκονται στο αρχείο καταχωρητών στις θέσεις μνήμης 2, 3, 4 αντίστοιχα.

Register File		
0	a	t2
1	b	t3
2	c	X
3	d	Y
4	t1	Z

Πίνακας 4.4.2-4. Το αρχείο καταχωρητών τελικά. Οι παλιές καταχωρήσεις έχουν πανωγραφεί.

Η δουλεία της FSM είναι να συγχρονίσει όλη αυτήν την διαδικασία. Στο συγκριμένο παράδειγμα η μορφή της φαίνεται στο Σχήμα 4.4.2-5.



Σχήμα 4.4.2-5. Η FSM της δεύτερης μεθόδου. Η διαδικασία είναι παρόμοια για κάθε σύνολο πράξεων.

Έχοντας πάντα υπόψιν τον αριθμό πυρήνων FP που χρειάζεται και τον λανθάνοντα χρόνο τους (latency) μπορεί να δημιουργηθεί το κατάλληλο χρονοπρόγραμμα πράξεων οποιασδήποτε συνάρτησης. Κατόπιν επιλέγοντας το κατάλληλο μέγεθος για αρχείο καταχωρητών και υπολογίζοντας τον απαιτούμενο αριθμό κύκλων προκύπτει η αντιστοιχία των τιμών του χρονοπρογράμματος με θέσεις μνήμης του αρχείου καταχωρητών. Έχοντας σαν κανόνα την προτεραιότητα της μεγαλύτερης «αλυσίδας» εξαρτώμενων υπολογισμών οι κενές θέσεις που προκύπτουν στο χρονοπρόγραμμα μπορούν να συμπληρωθούν με τις υπόλοιπες πράξεις.

Αξιολογώντας την παραπάνω σχεδίαση προκύπτει ότι το σχεδιαζόμενο σύστημα είναι ταχύτερο και απαιτεί λιγότερους πόρους σε σύγκριση με την προηγούμενη μέθοδο σχεδίασης. Επίσης επιτυγχάνεται η μέγιστη δυνατή επαναχρησιμοποίηση των FPCs. Από την άλλη πλευρά η συχνότητα λειτουργίας του συστήματος μειώνεται αισθητά (λόγω της FSM πολλών κύκλων) χωρίς όμως τελικά αυτό να καθιστά την σχεδίαση ασύμφορη. Ένα ακόμα μειονέκτημα είναι ότι η χρήση μιας τέτοιας διαδικασίας για την υλοποίηση μιας συνάρτησης κάθε άλλο παρά απλή είναι καθώς οποιαδήποτε αλλαγή στους σχεδιαστικούς παράγοντες (λανθάνων χρόνος, αριθμός FPC) απαιτεί τον πλήρη επανασχεδιασμό όλου του συστήματος.

4.4.3 Συνδυασμός των δύο μεθόδων

Μία επιπλέον επιλογή σχεδίασης αποτελεί ο συνδυασμός των δύο παραπάνω. Δηλαδή το κάθε τμήμα σχεδιάζεται κατά περίπτωση είτε με αναλυτική μορφή είτε με χρήση FSM και αρχείου καταχωρητών. Μια τέτοια σχεδίαση συνδυάζει τα ιδιαίτερα χαρακτηριστικά της κάθε περίπτωσης. Πρέπει να ληφθεί υπόψιν ότι αν ένα τμήμα έχει σχεδιαστεί με την πρώτη μέθοδο με σκοπό τη μέγιστη συχνότητα λειτουργίας, ενώ κάποιο άλλο με τη δεύτερο μέθοδο (και συνεπώς λειτουργεί με μικρότερη συχνότητα) η μέγιστη συχνότητα λειτουργίας που θα επιτύχει το σύστημα θα ταυτίζεται με εκείνη του «αργού» τμήματος.

4.5 Υλοποίηση τμημάτων

4.5.1 Πρώτη υλοποίηση (πρώτη μέθοδος)

Για την πρώτη υλοποίηση του συστήματος χρησιμοποιήθηκε η πρώτη μέθοδος που αναφέρεται στο Κεφάλαιο 4.2.1. Χρησιμοποιήθηκαν επίσης πυρήνες κινητής υποδιαστολής οι οποίοι σχεδιάστηκαν με χρήση του εργαλείου Xilinx Core Generator. Πιο συγκεκριμένα (βλέπε Κεφάλαια 3.2, 3.3) ο πολλαπλασιαστής σχεδιάστηκε με λανθάνοντα χρόνο 8 κύκλων και χρήση 2 μονάδων DSP48E (χρειάστηκε και χρήση πόρων LUTs και FFs). Ο αθροιστής/αφαιρέτης σχεδιάστηκε με λανθάνοντα χρόνο 11 κύκλων και χρήση 2 μονάδων DSP48E (χρειάστηκε και χρήση πόρων LUTs και FFs). Ο διαιρέτης με λανθάνοντα χρόνο 28 κύκλων και η υλοποίηση του έγινε μόνο με χρήση πόρων LUTs και FFs (logic only).

Επίσης ανάμεσα στα διάφορα στάδια καθώς και όπου χρειαζόταν καθυστέρηση ή προσωρινή αποθήκευση κάποιων τιμών χρησιμοποιήθηκαν καταχωρητές.

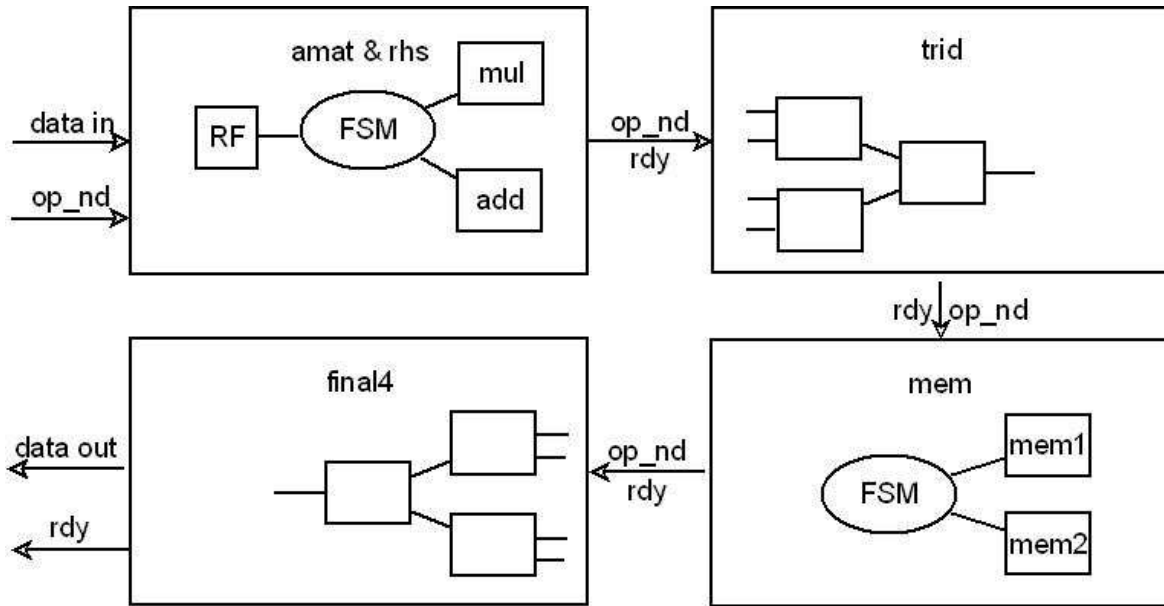
Η συγκεκριμένη υλοποίηση δούλεψε με συχνότητα λειτουργίας 400MHz, όσο περίπου είναι και η συχνότητα λειτουργίας που επιτρέπουν οι προδιαγραφές των συγκεκριμένων πυρήνων κινητής υποδιαστολής (FPCs).

4.5.2 Δεύτερη υλοποίηση (πρώτη και δεύτερη μέθοδος)

Για την συγκεκριμένη υλοποίηση χρησιμοποιήθηκε συνδυασμός των δύο σχεδιαστικών μεθόδων όπως αυτές αναλύονται στα Κεφάλαια 4.4.1 και 4.4.2

Χρησιμοποιήθηκαν επίσης πυρήνες κινητής υποδιαστολής και καταχωρητές με τις ίδιες προδιαγραφές και χαρακτηριστικά της πρώτης υλοποίησης.

Οι συναρτήσεις `amat` και `rhs` συγχωνεύτηκαν σε μία και η επίλυση των απαιτούμενων πράξεων υλοποιήθηκε με χρήση της δεύτερης μεθόδου (ένας πολλαπλασιαστής, ένας προσθέτης/αφαιρέτης, FSM, αρχείο καταχωρητών). Οι συναρτήσεις `trid` και `final4` υλοποιήθηκαν με την πρώτη μέθοδο και με αναλυτική σχεδίαση. Η παραπάνω υλοποίηση παρουσιάζεται στο Σχήμα 4.5.2-1.



Σχήμα 4.5.2-1. Σχηματική αναπαράσταση της δεύτερης υλοποίησης.

Η συχνότητα λειτουργίας του κάθε τμήματος παρουσιάζεται στον Πίνακα 4.5.2-2.

Design Parts	Frequency (MHz)
amat & rhs	251
trid	403
mem	397
final4	403

Πίνακας 4.5.2-2. Οι συχνότητες λειτουργίας των τμημάτων της δεύτερης υλοποίησης.

Η μέγιστη συχνότητα λειτουργίας στην οποία μπορεί να λειτουργήσει το σύστημα είναι η μικρότερη που επιτυγχάνεται από το σύνολο των τμημάτων η οποία είναι 251 MHz (Πίνακας 4.5.2-2).

4.5.3 Τρίτη υλοποίηση (πρώτη και δεύτερη μέθοδος, μόνο λογική)

Η συγκεκριμένη υλοποίηση είναι ακριβώς ίδια με την δεύτερη υλοποίηση με την μόνη διαφορά ότι οι FPC υλοποιήθηκαν κάνοντας χρήση μόνο λογικών πόρων LUTs και FFs χωρίς να γίνει χρήση των DSP48Es.

Η μέγιστη συχνότητα λειτουργίας του συστήματος είναι και σε αυτή την περίπτωση 251MHz.

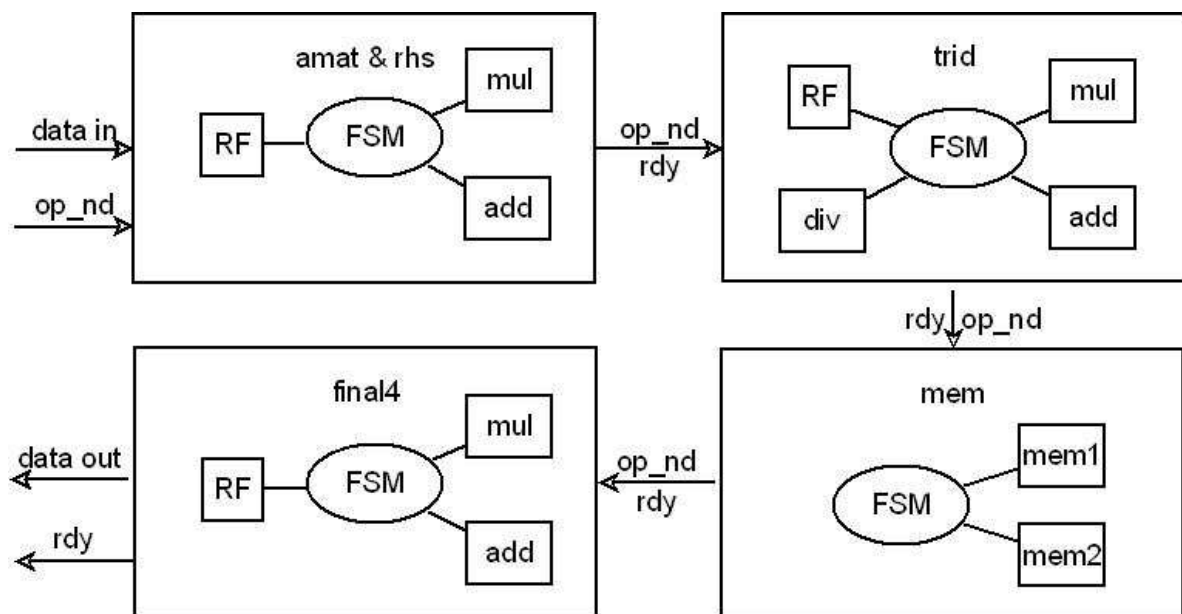
4.5.4 Τέταρτη υλοποίηση (δεύτερη μέθοδος)

Για την συγκεκριμένη υλοποίηση χρησιμοποιήθηκε η δεύτερη σχεδιαστική μέθοδος, δηλαδή χρήση FSM και αρχείων καταχωρητών αποκλειστικά.

Χρησιμοποιήθηκαν επίσης πυρήνες κινητής υποδιαστολής οι οποίοι σχεδιάστηκαν με το εργαλείο Xilinx Core Generator. Πιο συγκεκριμένα ο πολλαπλασιαστής σχεδιάστηκε με λανθάνοντα χρόνο 3 κύκλων και χρήση 2 μονάδων DSP48E (χρειάστηκε και χρήση πόρων LUTs και FFs). Ο αθροιστής/αφαιρέτης σχεδιάστηκε με λανθάνοντα χρόνο 5 κύκλων και χρήση 2 μονάδων DSP48E (χρειάστηκε και χρήση πόρων LUTs και FFs). Ο διαιρέτης με λανθάνοντα χρόνο 14 κύκλων και η υλοποίηση του έγινε μόνο με χρήση πόρων LUTs και FFs (logic only).

Επίσης ανάμεσα στα διάφορα στάδια καθώς και όπου χρειαζόταν καθυστέρηση ή προσωρινή αποθήκευση κάποιων τιμών χρησιμοποιήθηκαν καταχωρητές.

Οι συναρτήσεις *amat* και *rhs* συγχωνεύτηκαν σε μία και η επίλυση των απαιτούμενων πράξεων υλοποιήθηκε με χρήση της δεύτερης μεθόδου (ένας πολλαπλασιαστής, ένας προσθέτης/αφαιρέτης, FSM, αρχείο καταχωρητών). Οι συναρτήσεις *trid* και *final4* υλοποιήθηκαν με τον ίδιο τρόπο. Η παραπάνω υλοποίηση παρουσιάζεται στο Σχήμα 4.5.4-1.



Σχήμα 4.5.4-1. Σχηματική αναπαράσταση της τέταρτης υλοποίησης.

Όλο το σύστημα τελικά δουλεύει με μέγιστη συχνότητα λειτουργίας 111 MHz κάνοντας όμως χρήση των λιγότερων μονάδων FPC σε σχέση με τις προηγούμενες υλοποιήσεις (συνολικά 6 FPCs).

4.6 Επιβεβαίωση αποτελεσμάτων

Τα αποτελέσματα των υλοποιήσεων συγκρίθηκαν με αυτά του αλγορίθμου και επιβεβαιώθηκε η εγκυρότητά τους. Ελέγχθηκαν τόσο τα αποτελέσματα των συναρτήσεων ξεχωριστά όσο και η συνολική λειτουργία του συστήματος.

Αρχικά στον προσομοιωτή λειτουργίας του συστήματος (Modelsim 6.2c) ενσωματώθηκαν οι κατάλληλες βιβλιοθήκες για την μετατροπή των εξόδων του συστήματος από δυαδική αναπαράσταση 32 bit κινητής υποδιαστολής σε πραγματικούς αριθμούς (float).

Τα δυαδικά αποτελέσματα γράφονταν επίσης σε αρχείο και μετατρέπονταν σε δεκαδικούς πραγματικούς με χρήση ειδικής εφαρμογής.

Επίσης, απομονώθηκαν οι συναρτήσεις από την εφαρμογή του Η/Υ και τα αποτελέσματά τους ταυτοποιήθηκαν με τα αντίστοιχα υλοποιημένα σε υλικό τμήματα. Και στις δύο περιπτώσεις τα αποτελέσματα εγγράφονταν σε αρχεία τα οποία και τελικά ταυτοποιούνταν με χρήση εφαρμογής σύγκρισης.

Το τελικό διάνυσμα που παράγει η CNPE εφαρμογή αφού εγγράφηκε σε αρχείο ταυτοποιήθηκε με το αντίστοιχο διάνυσμα που παράχθηκε από την προσομοίωση με υλικό με χρήση εφαρμογής σύγκρισης.

4.7 Πρόταση για υλοποίηση

Ο CNPE αλγόριθμος απαιτεί συνολικά έναν αριθμό από πράξεις οι οποίες πιο συγκεκριμένα είναι 48 πολλαπλασιασμοί, 42 προσθέσεις/αφαιρέσεις και 4 διαιρέσεις για τον υπολογισμό κάθε τιμής του διανύσματος.

Μια προτεινόμενη υλοποίηση (μη υπολογίζοντας τους πόρους που απαιτούνται) είναι μια σχεδίαση παρόμοια με την 1^η υλοποίηση με χρήση της 1^{ης} μεθόδου ενός συστήματος που σε κάθε κύκλο ρολογιού υλοποιεί τις ίδιες πράξεις για διαφορετικό και ανεξάρτητο ύψος. Π.χ. για έναν αριθμό από N ύψη χρησιμοποιείται το ίδιο υλικό (και το pipelining που προσφέρει) για να υπολογίζει ξεχωριστά διανύσματα με tomasulo-like συγχρονισμό. Οι απαιτήσεις σε μνήμη και

σε λογική αυξάνουν, δηλαδή απαιτούνται περισσότεροι καταχωρητές και N , αντί για ένα, τμήματα μνημών ($N \cdot (2 \cdot 64 \cdot 512)$ bit) τα οποία θα μπορούσαν να συγχρονίζονται με την round-robin μέθοδο και να επικοινωνούν με την υπόλοιπη σχεδίαση (δεν υπολογίζεται το κόστος σε συχνότητα αυτής της διεπαφής).

Με την συγκεκριμένη υλοποίηση, όσο πολύπλοκη και να είναι, είναι δυνατή όχι μόνο η μέγιστη επαναχρησιμοποίηση του υλικού (σε κάθε κύκλο υλοποιούνται όλες οι πράξεις για διάφορα ύψη) αλλά και η λειτουργία του συστήματος με τη μέγιστη δυνατή συχνότητα για την 1^η υλοποίηση, δηλαδή τα 400 MHz.

5. Συμπεράσματα

Για την επιλογή της βέλτιστης υλοποίησης λήφθηκε υπόψιν η συχνότητα λειτουργίας και οι απαιτούμενοι πόροι και τελικά η επιτάχυνση του αλγόριθμου σε αναδιατασσόμενη λογική σε σχέση με τον H/Y.

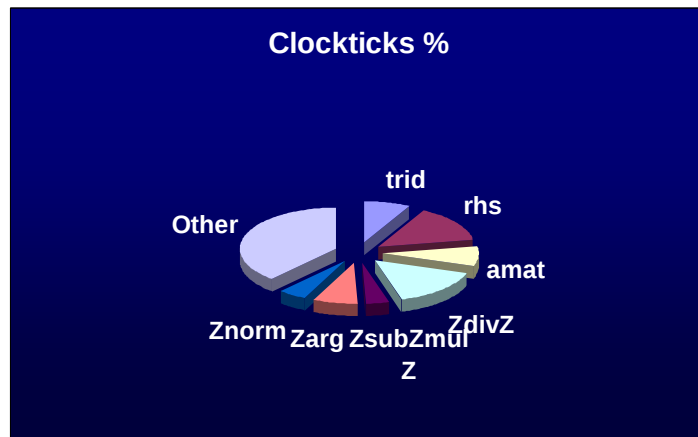
5.1 Μετρήσεις χρόνου εκτέλεσης σε H/Y

Οι χρόνοι εκτέλεσης του αλγορίθμου μετρήθηκαν σε H/Y με επεξεργαστή Intel Pentium 4 2,66 GHz και 1024 Mb RAM με λειτουργικό Ubuntu Linux 8.04. Οι μετρήσεις έγιναν με τη χρήση του προγράμματος Intel Vtune Performance Analyzer.

Σημαντικό στοιχείο των μετρήσεων αποτελεί ο αριθμός κύκλων ρολογιού (clockticks) που απαιτούνταν για την εκτέλεση της κάθε συνάρτησης. Ο μέσος όρος των κύκλων που απαιτεί η κάθε συνάρτηση παρουσιάζεται στον Πίνακα 5.1-1 και το Διάγραμμα 5.1-2.

Clockticks %		Clockticks Events	
	median(%)		median
trid	7,72	trid	61051400
rhs	14,476	rhs	114904600
amat	7,786	amat	61584600
ZdivZ1	15,522	ZdivZ1	122902600
ZsubZmulZ1	3,854	ZsubZmulZ1	30659000
Zarg1	7,471	Zarg1	59185200
Znorm1	4,689	Znorm1	37057400
	61,518		487344800

Πίνακας 5.1-1. Οι μετρήσεις και οι μέσοι όροι για το ποσοστό και ακριβή αριθμό των κύκλων που απαιτούνται.



Διάγραμμα 5.1-2. Διάγραμμα απαιτούμενων κύκλων ρολογιού.

Θεωρώντας ότι η συνολική εκτέλεση των συναρτήσεων απαιτεί περίπου $500 \cdot 10^6$ κύκλους ρολογιού (δεδομένα από VTune Performance Analyser) και ο H/Y δουλεύει σε συχνότητα 2,66 MHz προκύπτει ότι ο συνολικός χρόνος εκτέλεσης του αλγορίθμου είναι

$$\frac{500 \cdot 10^6 \text{ cycles}}{2,66 \cdot 10^9 \frac{\text{cycles}}{\text{sec}}} = 0,19 \text{ sec}$$

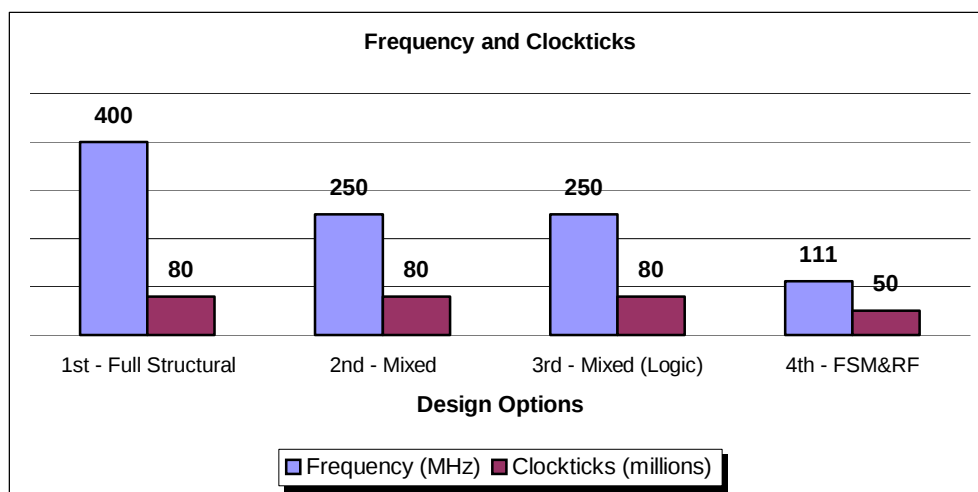
Δηλαδή μία εκτέλεση του αλγορίθμου (των συναρτήσεων rhs, amat και trid) προκύπτει σε 0,19 δευτερόλεπτα. Ο χρόνος αυτός είναι πρακτικά περισσότερος επειδή ο H/Y τρέχει ενδιάμεσα στην εκτέλεση και εντολές από άλλα προγράμματα (π.χ. λειτουργικό).

5.2 Σύγκριση συχνοτήτων και κύκλων ρολογιού

Στην επιλογή της βέλτιστης υλοποίησης από όσες παρουσιάστηκαν παίζει σημαντικό ρόλο η συχνότητα λειτουργίας του κάθε συστήματος καθώς και ο απαιτούμενος αριθμός κύκλων ρολογιού για την παρουσίαση μιας λύσης. Ο Πίνακας 5.2-1 και το Διάγραμμα 5.2-2 παρουσιάζουν αναλυτικά τα παραπάνω στοιχεία.

Implementations	Frequency (MHz)	Clockticks
1 st - Full Structural	400	$80 \cdot 10^6$
2 nd - Mixed	251	$80 \cdot 10^6$
3 rd - Mixed (Logic)	251	$80 \cdot 10^6$
4 th - FSM&RF	111	$50 \cdot 10^6$

Πίνακας 5.2-1. Διαφορετικές υλοποιήσεις με τις συχνότητες λειτουργίας τους.



Διάγραμμα 5.2-2. Διαγραμματική αναπαράσταση Πίνακα 5.2-1.

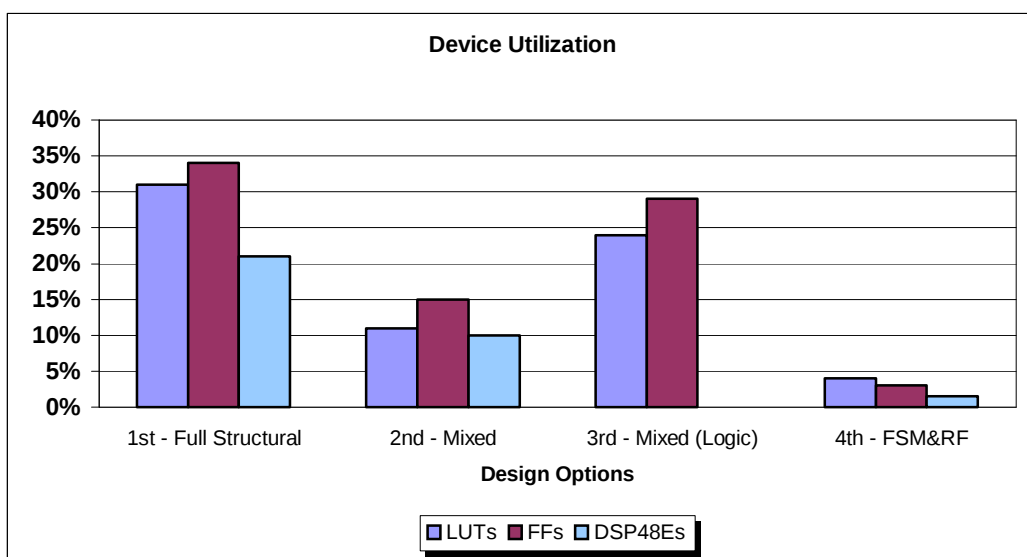
Όπως φαίνεται στο Διάγραμμα 5.2-2, η υλοποίηση με χρήση FSM και αρχείου καταχωρητών παρουσιάζει τη μικρότερη συχνότητα λειτουργίας αλλά απαιτεί και πολύ μικρότερο αριθμό κύκλων για να δώσει αποτέλεσμα.

5.3 Σύγκριση δέσμευσης πόρων

Σημαντικό στοιχείο για την επιλογή της βέλτιστης υλοποίησης από όσες αναφέρθηκαν στο Κεφάλαιο 4.4 είναι η σύγκριση των απαιτούμενων πόρων που απαιτεί η κάθε μια. Η μικρή δέσμευση πόρων της FPGA σημαίνει δυνατότητα να τοποθετηθούν παράλληλα περισσότεροι επιλύτες και κατά συνέπεια επιτάχυνση του όλου συστήματος. Οι βασικοί πόροι συστήματος είναι οι LUTs, FFs, και DSP48Es. Η πρώτη σχεδίαση απορρίπτεται καθώς δεν παρουσιάζει ενδιαφέρον επειδή οι απαιτήσεις σε FPC είναι τόσο μεγάλες που μόνο δύο επιλύτες δεσμεύουν όλους τους πόρους της FPGA. Ο Πίνακας 5.3-1 και το Διάγραμμα 5.3-2 παρουσιάζουν αναλυτικά τα παραπάνω στοιχεία.

	Single Solver FPGA Utilization			
Implementations	LUTs	FFs	DSP48Es	Max Solvers
1 st – Full Structural	31%	34%	21%	2
2 nd - Mixed	11%	15%	10%	7
3 rd - Mixed (Logic)	24%	29%	0	3
4 th - FSM&RF	4%	3%	1,5%	22

Πίνακας 5.3-1. Σύγκριση απαιτούμενων πόρων για κάθε σχεδίαση καθώς και αριθμός παράλληλων επιλυτών.



Διάγραμμα 5.3-2. Αναπαράσταση χρήσης πόρων FPGA για έναν επιλύτη.

Ενδιαφέρον παρουσιάζει το γεγονός ότι η σχεδίαση με χρήση FSM και αρχείου καταχωρητών έχει την ελάχιστη κατανάλωση σε πόρους και κατά συνέπεια επιτρέπει την χρήση 22 παράλληλων επιλυτών στην ίδια FPGA. Στην άλλη πλευρά είναι η συνδυασμένη σχεδίαση με χρήση μόνο λογικής η οποία επιδέχεται παραλληλία 3 επιλυτών.

5.4 Τελική σύγκριση

Βασικό κριτήριο για την επιλογή της βέλτιστης υλοποίησης αποτελεί η τελική επιτάχυνση (speedup) που πετυχαίνει το σύστημα σε σχέση με την εκτέλεση του αλγορίθμου σε H/Y.

Ο χρόνος εκτέλεσης του αλγορίθμου μετρήθηκε σε έναν H/Y Intel Pentium 4 με CPU 2,66 GHz και 1024 Mb RAM με λειτουργικό Ubuntu Linux 8.04 και με τη χρήση του προγράμματος Intel Vtune Performance Analyzer. Ο χρόνος που χρειάζεται ο H/Y για τρέξει τις συγκεκριμένες συναρτήσεις είναι 0,19 δευτερόλεπτα.

Έχοντα απορρίψει την τρίτη υλοποίηση (επειδή απαιτεί πολλούς πόρους και δεν προσφέρεται για παραλληλία), η τελική σύγκριση γίνεται ανάμεσα στην δεύτερη και τέταρτη υλοποίηση.

Ο χρόνος εκτέλεσης μιας λύσης δίνεται από τη σχέση

$$\frac{\left\{ \frac{\text{απαιτούμενος αριθμός κύκλων}}{\text{συχνότητα λειτουργίας}} \right\}}{\text{αριθμός παράλληλων επιλυτών}}$$

Για την δεύτερη υλοποίηση ισχύει

$$\frac{\left\{ \frac{80 \cdot 10^6 \text{ cycles}}{251 \cdot 10^9 \frac{\text{cycles}}{\text{sec}}} \right\}}{7 \text{ solvers}}$$

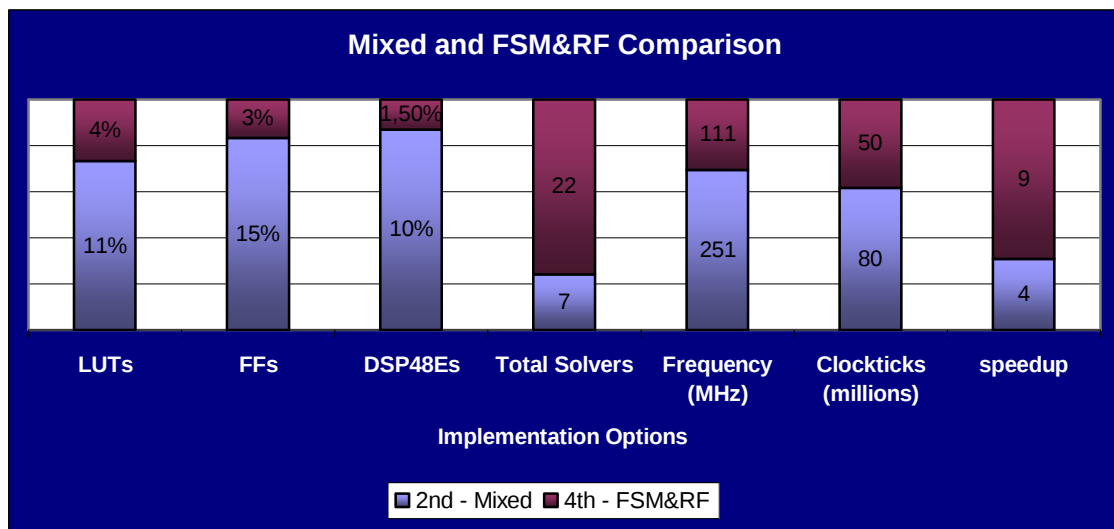
και προκύπτει speedup 4x. Δηλαδή το σύστημα δίνει λύσεις περίπου 4 φορές πιο γρήγορα από έναν H/Y.

Για την τέταρτη υλοποίηση ισχύει

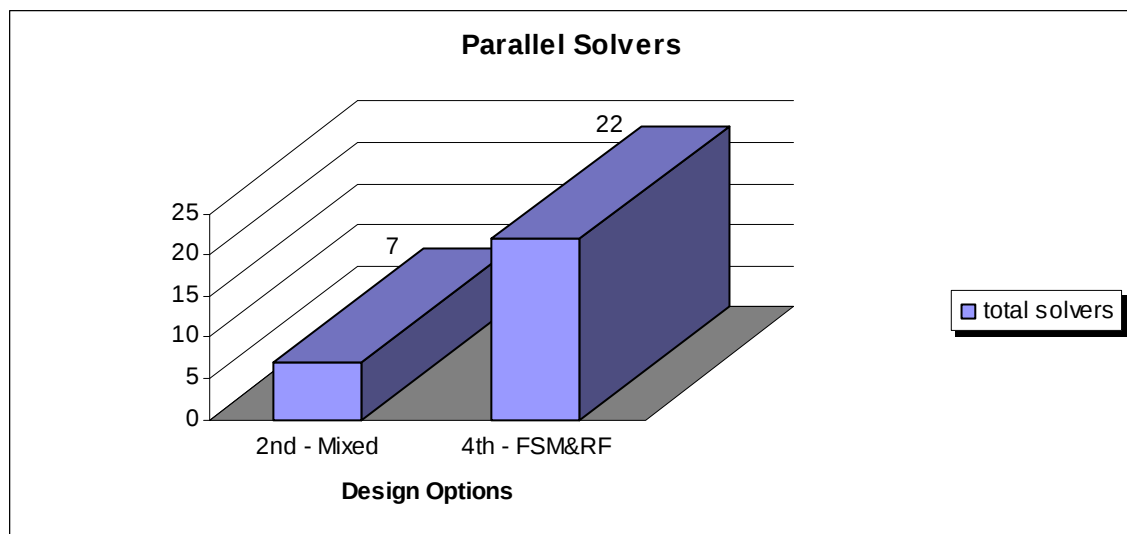
$$\frac{\left\{ \frac{50 \cdot 10^6 \text{ cycles}}{111 \cdot 10^9 \frac{\text{cycles}}{\text{sec}}} \right\}}{22 \text{ solvers}}$$

και προκύπτει speedup 9x. Δηλαδή το σύστημα δίνει λύσεις περίπου 9 φορές πιο γρήγορα από έναν Η/Υ.

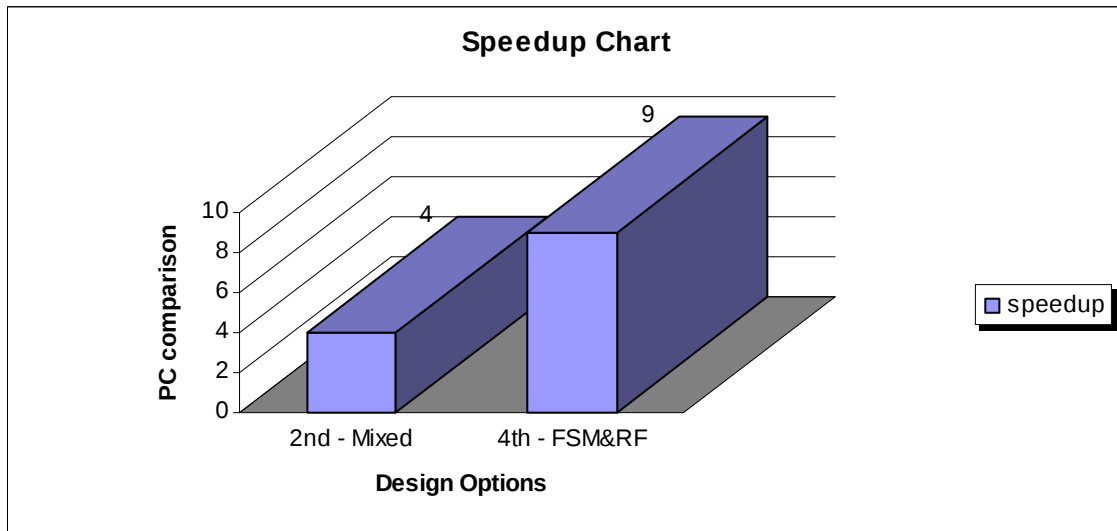
Αναλυτικά όλα τα στοιχεία της κάθε υλοποίησης (πόροι, συχνότητα, παραλληλία και ταχύτητα) παρουσιάζονται στα Διαγράμματα 5.4-1, 5.4-2, 5.4-3.



Διάγραμμα 5.4-1. Σύγκριση των υλοποιήσεων 2 και 4.



Διάγραμμα 5.4-2. Σύγκριση παράλληλων επιλυτών υλοποιήσεων 2 και 4.



Διάγραμμα 5.4-3. Σύγκριση ταχύτητας υλοποιήσεων 2 και 4.

Συγκεντρωτικά και για τις τέσσερις υλοποιήσεις:

Device Utilization and Parallel Solvers for XC5VSX240T FPGA				
Implementations	LUTs	FFs	DSP48Es	Parallel Solvers
1st	31%	34%	21%	2
2nd	11%	15%	10%	7
3rd	24%	29%	0%	3
4th	4%	3%	1,50%	22

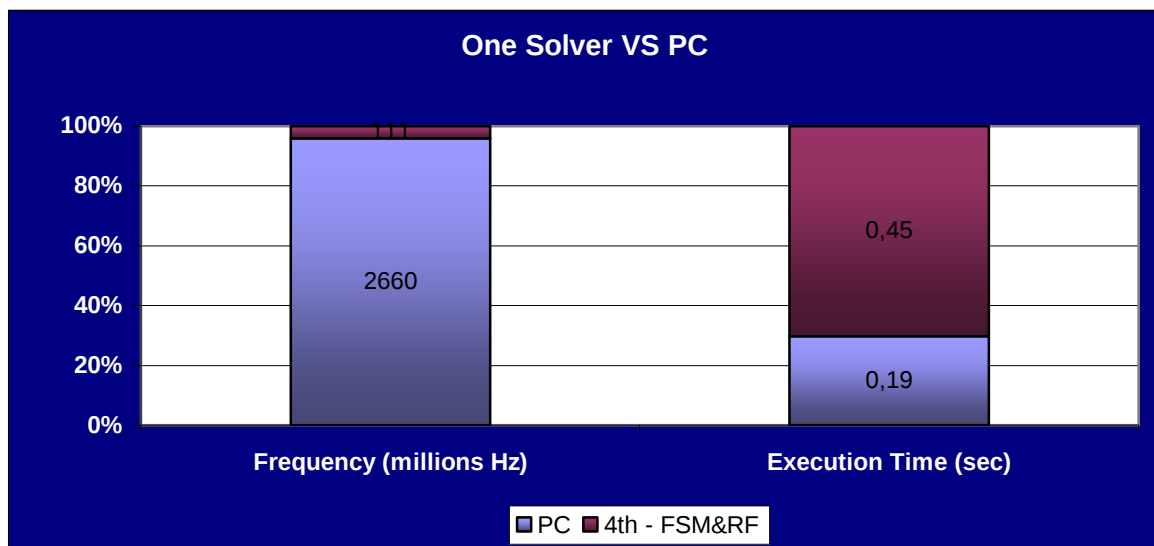
Πίνακας 5.4-4. Πίνακας χρήσης πόρων FPGA για έναν επιλύτη.

Frequency, Clockticks, Execution Time and Algorithm Speedup				
Implementations	Frequency (MHz)	Clockticks (millions)	Execution Time (sec)	Speedup
1st	400	80	0,20	2x
2nd	250	80	0,32	4x
3rd	250	80	0,32	2x
4th	111	50	0,45	9x

Πίνακας 5.4-5. Πίνακας συχνοτήτων, κύκλων, χρόνου εκτέλεσης και speedup.

Η υπεροχή του συστήματος της τέταρτης υλοποίησης που σχεδιάστηκε εξ'ολοκλήρου με χρήση FSM και αρχείου καταχωρητών είναι προφανής. Επιστρέφει αποτελέσματα 9 φορές πιο γρήγορα από ένα Η/Υ. Παρ' όλο που λειτουργεί με τη μικρότερη συχνότητα σε σχέση με τα άλλα συστήματα επιτρέπει τη χρήση πολλαπλών επιλυτών. Έτσι το τέταρτο σύστημα υπερέχει όσον αφορά το speedup το οποίο είναι και το ζητούμενο.

Συγκρίνοντας ένα και μόνο επιλύτη της τέταρτης υλοποίησης με την εκτέλεση του αλγορίθμου σε Η/Υ παρατηρείται ότι ένα σύστημα με 24 φορές μικρότερη συχνότητα λειτουργίας από έναν Η/Υ (111 MHz σε υλικό και 2,66 GHz σε Η/Υ) προκύπτει μόνο 2,4 φορές πιο αργό στην εκτέλεση του ίδιου αλγορίθμου (Διάγραμμα 5.4-4). Κάνοντας παράλληλη χρήση 22 επιλυτών, όπως προαναφέρθηκε, η τέταρτη υλοποίηση τελικά προκύπτει 9 φορές ταχύτερη.



Διάγραμμα 5.4-4. Σύγκριση ενός μόνο επιλύτη με Η/Υ.

6. Επίλογος και μελλοντική προοπτική

6.1 Επίλογος

Η CNPE μέθοδος είναι ικανή να δώσει ακριβή αποτελέσματα στο πρόβλημα της συγκεκριμένης διπλωματικής εργασίας. Λόγω όμως της ιδιαιτερότητας των τριδιαγώνιων συστημάτων εξετάστηκαν διάφορες υλοποιήσεις. Λήφθηκαν υπόψιν οι απαιτήσεις σε πόρους του κάθε συστήματος και η τελική επιτάχυνση σε σχέση με την εκτέλεση του κώδικα σε Η/Υ. Η 4^η υλοποίηση με χρήση μηχανής πεπερασμένων καταστάσεων και αρχείου καταχωρητών προέκυψε η αποδοτικότερη καθ' ότι επιταχύνει τον αλγόριθμο εννέα φορές κάνοντας βέλτιστη χρήση του προσφερόμενου υλικού.

6.2 Μελλοντική προοπτική

Σαν μελλοντική εργασία προτείνονται:

- Υλοποίηση κώδικα σε γλώσσα προγραμματισμού ο οποίος θα δέχεται μια αλληλουχία αριθμητικών πράξεων και διαθέσιμων πόρων συστήματος και θα παράγει το βέλτιστο χρονοπρόγραμμα για την υλοποίηση της μεθόδου με χρήση FSM και αρχείου καταχωρητών.
- Σχεδίαση διαφορετικών μεθόδων υλοποίησης π.χ. οι πράξεις θα κωδικοποιούνται σε εντολές 32 bit και θα τοποθετούνται σε μνήμη (instruction memory). Έτσι με χρήση του κατάλληλου αρχείου καταχωρητών (register file) και των πυρήνων κινητής υποδιαστολής θα υλοποιείται ο αλγόριθμος (σχεδίαση παρόμοια με επεξεργαστή MIPS).
- Σχεδίαση και υλοποίηση της GFPE (Green's function Parabolic Equation). Οι πράξεις που απαιτεί η συγκεκριμένη μέθοδος είναι πολύ λιγότερες και απλές. Απαιτείται όμως αριθμητική διπλής ακρίβειας. Μια τέτοια υλοποίηση θα είναι δυνατή όταν υλοποιηθούν και εφαρμοστούν σε FPGA οι πυρήνες κινητής υποδιαστολής διπλής ακρίβειας.

7. Αναφορές

- [1] "Long-Range Noise Propagation and Helicopter path Optimization for Noise Reduction", FORTH Research, <http://www.forth.gr>
- [2] E. T. Flouri, J. A. Ekaterinaris and N. A. Kampanis, "High-Order Accurate Numerical Schemes for the Parabolic Equation", j. Computational Acoustics, vol 13, pp. 613-633, 2005.
- [3] Xilinx, "Floating-Point Operator v4.0, Product Specification", DS335, 2008.
- [4] Xilinx, "Virtex-5 Family Overview, Advance Product Specification", DS100 (v4.4), 2008.
- [5] J. A. Trangenstein, "Numerical Solution of Hyperbolic Partial Differential Equations", ISBN: 978-0521877275, 2009.
- [6] S. Marburg, B. Nolte, "Computational Acoustics of Noise Propagation in Fluids - Finite and Boundary Element Methods", ISBN: 978-3540774471, 2008.
- [7] N. A. Kampanis, V. Dougalis, J. A. Ekaterinaris, "Effective Computational Methods for Wave Propagation (Numerical Insights)", ISBN: 978-1584885689, 2008.
- [8] J. A. Trangenstein, "Numerical Solution of Hyperbolic Partial Differential Equations", ISBN: 978-0521877275, 2009.
- [9] G. Barton, "Elements of Green's Functions and Propagation: Potentials, Diffusion, and Waves", ISBN: 978-0198519980, 1989.
- [10] D. Lee, M. H. Schultz, "Numerical Ocean Acoustic Propagation in Three Dimensions", ISBN: 978-9810223038, 1995.
- [11] D. Lee, S. T. McDaniel, "Ocean Acoustic Propagation by Finite Difference Methods", ISBN: 978-0080348711, 1988.
- [12] M. Gen, R. Cheng, 'Genetic Algorithms and Engineering Optimization', ISBN: 978-0471315315, 1999.
- [13] D. G. Luenberger, "Optimization by Vector Space Methods", ISBN: 978-0471181170, 1997.

- [14] F. W. J. Olver, "Asymptotics and Special Functions", ISBN: 978-1568810690, 1997.
- [15] G. E. Andrews, R. Askey, R. Roy, "Special Functions", ISBN: 978-0521789882, 2001.
- [16] C. D. Cantrell, "Modern Mathematical Methods for Physicists and Engineers", ISBN: 978-0521598279, 2000.

