



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΦΥΣΙΚΗΣ
ΤΟΜΕΑΣ Ε΄

ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ
ΚΑΙ ΟΡΓΑΝΩΣΗ ΕΠΕΞΕΡΓΑΣΤΩΝ
ΓΙΑ ΑΠΟΔΟΤΙΚΟ ΠΑΡΑΛΛΗΛΙΣΜΟ
ΕΦΑΡΜΟΓΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΚΩΝΣΤΑΝΤΙΝΟΥ Ε. ΜΑΝΩΛΟΠΟΥΛΟΥ

ΑΘΗΝΑ
2013

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΤΟΜΕΑΣ Ε΄

Αρχιτεκτονικές και οργάνωση
επεξεργαστών
για αποδοτικό παραλληλισμό εφαρμογών

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΚΩΝΣΤΑΝΤΙΝΟΥ Ε. ΜΑΝΩΛΟΠΟΥΛΟΥ

Επταμελής Επιτροπή:

Δ. ΡΕΪΣΗΣ,

Α. ΠΟΛΥΔΩΡΟΣ,

Δ. ΦΡΑΝΤΖΕΣΚΑΚΗΣ,

Α. ΚΑΤΣΑΓΓΕΛΟΣ,

Γ. ΣΤΑΣΙΝΟΠΟΥΛΟΣ,

Δ. ΣΟΥΝΤΡΗΣ,

Ε. ΤΣΙΛΗΣ,

Επικ. Καθηγητής Ε.Κ.Π.Α. (επιβλέπων - τριμελής)

Καθηγητής Ε.Κ.Π.Α. (τριμελής)

Καθηγητής Ε.Κ.Π.Α. (τριμελής)

Καθηγητής Ε.Κ.Π.Α.

Καθηγητής Ε.Μ.Π.

Επικ. Καθηγητής Ε.Μ.Π.

Επικ. Καθηγητής Ε.Κ.Π.Α.

ΑΘΗΝΑ

2013

Ευχαριστίες

Παρόλο που οι γραμμές που ακολουθούν βρίσκονται στην αρχή της αυτής διατριβής, στην πραγματικότητα είναι το τελευταίο τμήμα που γράφτηκε. Όχι γιατί είναι το πιο απλό και το πιο εύκολο, όπως ίσως να σκεφτεί κανείς, αλλά ίσως γιατί το θεώρησα σαν την "τελευταία πινελιά" (ας μου επιτραπεί η έκφραση), ή καλύτερα σαν τον επίλογο μιας μεγάλης προσπάθειας. Ποιός καλύτερος τρόπος εξάλλου για να κλείσει κανείς έναν τέτοιο κύκλο, από το να ευχαριστήσει όλους εκείνους που τον βοήθησαν στην ολοκλήρωση του.

Αρχικά λοιπόν χρωστάω ένα μεγάλο ευχαριστώ στην οικογένεια μου και κυρίως στους γονείς μου. Χωρίς τη δικιά τους βοήθεια και υποστήριξη δεν θα μπορούσα να βρίσκομαι στην ευχάριστη θέση του να γράφω αυτές εδώ τις λέξεις.

Στη συνέχεια θα ήθελα να ευχαριστήσω τα μέλη της συμβουλευτικής μου επιτροπής για την καθοδήγηση τους αυτά τα χρόνια. Ιδιαίτερα θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Ρεΐση, για τη βοήθεια του, για όσα μου δίδαξε, αλλά και για όλες τις ευκαιρίες—δυνατότητες που έδωσε, όχι μόνο κατά τη διάρκεια εκπόνησης της διατριβής αυτής, αλλά από όταν πρωτοξεκίνησα να συνεργάζομαι μαζί του σα διπλωματικός φοιτητής του τμήματος Φυσικής.

Φυσικά δεν γίνεται να παραλείψω να αναφερθώ στη βοήθεια που είχα από τους πολύ καλούς μου φίλους, ενώ χρωστάω και ένα ευχαριστώ στη Νικόλ για τη πολύτιμη συμπαράσταση της.

Τέλος, θα ήθελα να ευχαριστήσω όλα τα μέλη της ομάδας Μελέτης και Ανάπτυξης Ψηφιακών Συστημάτων, τωρινά και μη, για τη βοήθεια τους και την πολύ καλή συνεργασία που είχαμε.

Περίληψη

Η παρούσα διατριβή έχει ως βασικό αντικείμενο μελέτης τις ψηφιακές αρχιτεκτονικές και την οργάνωση επεξεργαστών που οδηγούν σε αποδοτική εκτέλεση εντολών και διευκολύνουν τον παραλληλισμό εφαρμογών. Η διατριβή προσφέρει λύσεις σε τρεις κατηγορίες εφαρμογών: την κατασκευή και οργάνωση επεξεργαστών για την επιτάχυνση της κωδικοποίησης εικονορροών, τη μελέτη και ανάπτυξη αρχιτεκτονικών υλοποίησης Ταχύ Μετασχηματισμού Fourier και τη μελέτη και ανάπτυξη αρχιτεκτονικών για πράξεις κινητής υποδιαστολής. Οι αρχιτεκτονικές που παρουσιάζονται αποδίδουν το μέγιστο ρυθμό διαμεταγωγής, ελαχιστοποιούν το κόστος υλοποίησης σε πολύ μεγάλης κλίμακας ολοκλήρωσης (VLSI) και απαιτούν μικρή συχνότητα λειτουργίας.

Η διατριβή συνεισφέρει στη σχετική βιβλιογραφία ξεκινώντας αρχικά με τη μελέτη μιας διάταξης πολυεπεξεργαστή υλοποιημένη σε Σύστημα σε Ολοκλήρωμένο (System on Chip - SoC). Ο κάθε επεξεργαστής της διάταξης αυτής αποτελείται από μια ανοιχτού κώδικα Κεντρική Μονάδα Επεξεργασίας (CPU) που έχει ενισχυθεί με ένα διανυσματικό συνεπεξεργαστή (Vector Co-Processor) και με μια υψηλής ταχύτητας εξωτερική μνήμη. Η διάταξη αυτή εκμεταλλευόμενη τον παραλληλισμό σε επίπεδο νημάτων (Thread Level Parallelism - TLP) οδηγεί στην επιτάχυνση της εκτέλεσης των αλγορίθμων κωδικοποίησης MPEG-2 και MPEG-4. Η βελτίωση αυτή της απόδοσης των κωδικοποιήσεων φαίνεται από τη σημαντική μείωση του δυναμικού αριθμού εντολών που εκτελούνται σε κάθε περίπτωση, ενώ η υλοποίηση του συστήματος πολυεπεξεργαστή είναι εφικτή χάρη σε ένα μηχανισμό φράγματος που διαχειρίζεται αποτελεσματικά το συγχρονισμό των επεξεργαστών του συστήματος.

Στη συνέχεια η διατριβή μελετάει τη σχεδίαση αρχιτεκτονικών υλοποίησης Ταχύ Μετασχηματισμού Fourier για μεγάλα σύνολα δεδομένων εισόδου. Οι αρχιτεκτονικές αυτές στηρίζονται στη χρήση μιας μηχανής βάσης -4^3 (radix- 4^3), που μπορεί να λειτουργεί και σαν ένα αυτόνομο κύκλωμα FFT 64 σημείων. Η διατριβή αναλύει πως η χρήση τέτοιων μηχανών, οδηγεί σε μία αποτελεσματική αρχιτεκτονική υλοποίησης Ταχύ Μετασχηματισμού Fourier για 4096 σημεία, και στη συνέχεια μελετά πώς ο συνδυασμός των κυκλωμάτων υλοποίησης 4K FFT, σε διάφορες διατάξεις, σειριακές και παράλληλες, οδηγεί σε ένα πλήθος αποτελεσματικών αρχιτεκτονικών Ταχύ Μετασχηματισμού Fourier που μπορούν να διαχειριστούν μετασχηματισμούς για μεγάλα σύνολα δεδομένων εισόδου, όπως 16K, 64K και 256K μιγαδικά σημεία. Οι παραπάνω αρχιτεκτονικές

υλοποιήθηκαν τόσο σε διατάξεις πυλών προγραμματιζόμενες στο πεδίο (FPGAs), όσο και σε κυκλώματα πολύ μεγάλης κλίμακας ολοκλήρωσης (VLSI), με τα αποτελέσματα να δείχνουν υψηλές συχνότητες λειτουργίας και μεγάλους ρυθμούς απόδοσης. Η σύγκριση των χαρακτηριστικών και της απόδοσης του κυκλώματος 4K FFT με αντίστοιχες αρχιτεκτονικές της βιβλιογραφίας καταδεικνύει την αποδοτικότητα και τα πλεονεκτήματα της προτεινόμενης μεθόδου υλοποίησης κυκλωμάτων Ταχύ Μετασχηματισμού Fourier.

Τέλος, η διατριβή ολοκληρώνεται προτείνοντας λύσεις στο χώρο των υπολογισμών κινητής υποδιαστολής. Αρχικά γίνεται η μελέτη μιας μονάδας πολλαπλασιαστή κινητής υποδιαστολής και στη συνέχεια μελετώνται τρεις λύσεις για αρχιτεκτονικές Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης (Fused Multiply-Add - MAF). Τα κυκλώματα που παρουσιάζονται υποστηρίζουν - σε διάφορους συνδυασμούς - όλα τα μορφότυπα (format) ακρίβειας που περιγράφονται από την IEEE (μονή, διπλή και τετραπλή ακρίβεια), και εκτελούν εντολές της μορφής $(A \times B + C)$ ή πραγματοποιούν μεμονωμένες πράξεις πολλαπλασιασμού ή πρόσθεσης κινητής υποδιαστολής, εκτελώντας για κάθε μορφή ακρίβειας, παράλληλα πολλαπλές πράξεις από την κάθε κατηγορία. Η υλοποίηση των αρχιτεκτονικών αυτών σε κυκλώματα πολύ μεγάλης κλίμακας ολοκλήρωσης και η σύγκριση των αποτελεσμάτων με αντίστοιχες λύσεις της βιβλιογραφίας φανερώνει ότι οι προτεινόμενες λύσεις είναι ιδανικές για τις περιπτώσεις που απαιτείται αυξημένη ρυθμαπόδοση, με παράλληλη εκτέλεση πολλαπλών εντολών.

Η συμβολή της διατριβής είναι σημαντική γιατί οι προτεινόμενες δομές, και στις τρεις κατηγορίες, μελετούν και βελτιώνουν την απόδοση των αντίστοιχων συστημάτων, προσφέροντας βέλτιστες λύσεις, ως προς τη ρυθμαπόδοση και την πολυπλοκότητα υλοποίησης, με τη χρήση παράλληλων αρχιτεκτονικών για την αποτελεσματική υλοποίηση τους σε υλισμικό (hardware).

Λέξεις κλειδιά: επεξεργαστής, Ταχύς Μετασχηματισμός Fourier, κινητή υποδιαστολή

Abstract

This dissertation focuses on the study of digital architectures and processors organization, leading to the efficient instruction execution, while enhancing the parallelization of targeted applications. It proposes solutions in three different application categories: processor design and organization for speeding up MPEG encoders, design and implementation of Fast Fourier Transform architectures and last, the study and implementation of floating-point architectures.

This thesis contributes to the existing literature first by studying and implementing a configurable System-on-Chip multiprocessor, for speeding up MPEG-2 and MPEG-4 encoding, by using thread-level parallelism. Each processor of this system consists of an open-source CPU, augmented with a vector coprocessor and a high-speed external memory. Improving the performance of the MPEG encoders is shown by reducing the dynamic instruction count at multiple processor contexts. To realize the parallelized algorithms the SoC multi-processor architecture incorporates a novel barrier mechanism, that efficiently synchronizes the parallel sections.

Next, we study the design and implementation of Fast Fourier Transform architectures for large data input sets. These architectures are based on utilizing a novel radix-4³ engine, that operates as stand alone 64-point FFT engine. The outcome of cascading these radix-4³ engines is an improved 4096 complex point, fully systolic FFT architecture. Then, we study and demonstrate how to extend this 4K FFT architecture, by combining these 4K FFT engines, in a cascade or a parallel fashion, to accomplish FFT computations of 16K, 64K and 256K complex points. The proposed FFT architectures have been implemented both on FPGAs and VLSI achieving significantly high operating frequencies and throughput. Comparing the main characteristics and the performance of the 4K FFT architecture, with existing solutions in the literature, we prove the efficiency and the advantages of the proposed solution for implementing FFT architectures.

Finally, the last part of this dissertation presents efficient solutions in the field of floating point calculations. First, a multiple precision floating point multiplier is designed and implemented, and then follows a study of three different solutions for floating point Fused Multiply-Add architectures. The proposed designs can accommodate different combinations of the IEEE

precision formats, and they are able to perform parallel operations of stand alone floating point multiplications and additions, or execute in parallel multiple single instruction equations ($AxB + C$). The implementation of all presented architectures in VLSI, followed by an analysis of their performance and comparing the results with existing solutions in the literature, revealed that the proposed solutions are ideal for applications that demand high throughput operations and parallel execution of multiple instructions.

To conclude, the scientific contribution of this dissertation is established by proposing, in all three sections, efficient solutions that provide a complete study which enhances the performance of the corresponding systems. Furthermore, it provides efficient solutions with respect to throughput and implementation complexity, by utilizing parallel architectures in order to achieve an efficient hardware implementation.

Keywords: processor, FFT, floating-point, FPGA, VLSI

Περιεχόμενα

Ευχαριστίες	4
Περίληψη	5
Abstract	7
1 Εισαγωγή	15
2 Παράλληλισμός νημάτων σε κωδικοποιητές MPEG-2, MPEG-4 για SoC πολυεπεξεργαστές	19
2.1 Εισαγωγή	19
2.2 Μεθοδολογία	22
2.2.1 Μελέτη και κατηγοριοποίηση των κωδικοποιητών	22
2.2.2 Νημάτωση	23
2.2.2.1 MPEG-2 TM5	24
2.2.2.2 MPEG-4 (XViD)	26
2.3 Μικροαρχιτεκτονική	29
2.3.1 Διεπαφή συνεπεξεργαστή	29
2.3.2 Βαθμωτή Κεντρική Μονάδα Επεξεργασίας	31
2.3.3 Διανυσματικός συνεπεξεργαστής	31
2.3.4 Μηχανισμός Φράγματος (Barrier)	32
2.4 Υλοποίηση σε VLSI	33
3 Πλήρως συστολικές αρχιτεκτονικές για εφαρμογές Ταχύ Μετασχηματισμού Fourier	37
3.1 Εισαγωγή	37
3.2 Η πλήρως συστολική αρχιτεκτονική Ταχύ Μετασχηματισμού Fourier 4096 σημείων	39
3.2.1 Ο αλγόριθμος βάσης- 4^3	40
3.2.2 Χαρακτηριστικά του $R4^3$	43
3.2.3 Αρχιτεκτονική του 4K FFT	45
3.2.4 Η πεταλούδα Βάσης-4 ως βάση της μηχανής $R4^3$	46
3.2.5 Η αρχιτεκτονική της μηχανής $R4^3$	48

3.2.6	Χρονοπρογραμματισμός δεδομένων και διευθυνσιοδότηση στον $R4^3$	49
3.2.7	Υλοποίηση του 4K FFT	50
3.3	Αρχιτεκτονικές FFT των 16K, 64K και 256K μιγαδικών σημείων	53
3.3.1	Παράγωντας την αρχιτεκτονική 16K	53
3.3.1.1	Η Σειριακή/Παράλληλη αρχιτεκτονική	54
3.3.1.2	Η Παράλληλη/Παράλληλη αρχιτεκτονική	54
3.3.2	Η 64K FFT αρχιτεκτονική	58
3.3.3	Η 256K FFT αρχιτεκτονική	60
4	Αρχιτεκτονικές Κινητής Υποδιαστολής	62
4.1	Εισαγωγή	62
4.2	Σχετική βιβλιογραφία	64
4.3	Τυπική μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης διπλής ακρίβειας	66
4.4	Πολλαπλασιαστής κινητής υποδιαστολής, τριπλής λειτουργίας	69
4.4.1	Αρχιτεκτονική	69
4.4.1.1	Πολλαπλασιασμός τετραπλής ακρίβειας	72
4.4.1.2	Πολλαπλασιασμός διπλής ακρίβειας	74
4.4.1.3	Πολλαπλασιασμός μονής ακρίβειας	75
4.4.2	Υλοποίηση της αρχιτεκτονικής και ανάλυση των αποτελεσμάτων	76
4.5	Μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης μεταβλητού μήκους σωλήνωσης	78
4.5.1	Η αρχιτεκτονική MAF μεταβλητού μήκους σωλήνωσης	78
4.5.1.1	Πολλαπλασιασμός και ευθυγράμμιση	81
4.5.1.2	Πρόσθεση	81
4.5.1.3	Πρόβλεψη Προπορευόμενων Μηδενικών (LZA)	82
4.5.1.4	Κανονικοποίηση και στρογγυλοποίηση	83
4.5.2	Αποτελέσματα Υλοποίησης	83
4.6	Μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης διπλής λειτουργίας	85
4.6.1	Η αρχιτεκτονική της μονάδας MAF διπλής λειτουργίας	85
4.6.1.1	Πρώτο Στάδιο: Επεξεργασία εκθετών και Πολλαπλασιασμός	88
4.6.1.2	Δεύτερο Στάδιο: Οργάνωση Διπλού Μονοπατιού	89
4.6.1.3	Τρίτο Στάδιο: Συνδυασμός τελικής πρόσθεσης με τη στρογγυλοποίηση	90
4.6.2	Υλοποίηση	90
4.7	Μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης τριπλής λειτουργίας	92
4.7.1	Αρχιτεκτονική	92

4.7.1.1	Λειτουργία τετραπλής ακρίβειας	95
4.7.1.2	Λειτουργία διπλής και μονής ακρίβειας	98
4.7.1.3	Υλοποίηση και ανάλυση των αποτελεσμάτων	100
5	Σύνοψη και Μελλοντική εργασία	103
	Ορολογία	105
	Βιβλιογραφία	107
	Δημοσιεύσεις	114

Κατάλογος Σχημάτων

2.1	Κατανομή του δυναμικού αριθμού εντολών ανά συνάρτηση στον κωδικοποιητή MPEG-2 TM5.	23
2.2	Κατανομή του δυναμικού αριθμού εντολών ανά συνάρτηση στον κωδικοποιητή MPEG-4 XviD.	24
2.3	Θεωρητική απόδοση του MPEG-2 TM5 κωδικοποιητή για τις ακολουθίες εικονορροών (α) 'Snowfall' και (β) 'Rotating City'.	26
2.4	Θεωρητική απόδοση του MPEG-4 XviD κωδικοποιητή για τις ακολουθίες εικονορροών (α) 'Coastguard' και (β) 'Garden'.	27
2.5	Σχηματική αναπαράσταση της μικροαρχιτεκτονικής που περιλαμβάνει μια βαθμωτή ΚΜΕ και ένα συνεπεξεργαστή παράλληλων δεδομένων.	28
2.6	Σχηματική αναπαράσταση του συστήματος επεξεργαστή - συνεπεξεργαστή σε υψηλό επίπεδο.	29
2.7	Παράδειγμα λειτουργίας της διεπαφής επεξεργαστή - συνεπεξεργαστή.	30
2.8	Βαθμωτό μονοπάτι δεδομένων.	33
2.9	Σχηματικό διάγραμμα του μηχανισμού φράγματος.	34
2.10	Χρονικό διάγραμμα λειτουργίας του μηχανισμού φράγματος.	35
2.11	Σχηματική αναπαράσταση του πολυεπεξεργαστή SoC.	35
2.12	Υλοποίηση σε VLSI του πολυεπεξεργαστή SoC για N=2 και N=4 επεξεργαστές.	36
3.1	Διάγραμμα ροής πεταλούδας <i>Radix</i> - 4^3	44
3.2	Μπλοκ διάγραμμα της αρχιτεκτονικής 4K FFT.	45
3.3	Η μηχανή Βάσης-4.	47
3.4	Το δέντρο αθροιστών του μιγαδικού συσσωρευτή.	47
3.5	Μπλοκ διάγραμμα της αρχιτεκτονικής πεταλούδας <i>Radix</i> - 4^3	49
3.6	Υλοποίηση της μηχανής $R4^3$ σε VLSI.	52
3.7	Υλοποίηση του επεξεργαστή 4K FFT σε VLSI α)Σύνθεση 16×4096 β)Σύνθεση 64×4096	52
3.8	Μπλοκ διάγραμμα της Σειριακής/Παράλληλης αρχιτεκτονικής 16K FFT.	56
3.9	Μπλοκ διάγραμμα της Παράλληλης/Παράλληλης αρχιτεκτονικής 16K FFT.	56

3.10	Υλοποίηση σε VLSI της 16K Παράλληλης/Παράλληλης FFT αρχιτεκτονικής.	57
3.11	Μπλοκ διάγραμμα της αρχιτεκτονικής 64K FFT.	59
3.12	Υλοποίηση σε VLSI του 64K FFT.	59
3.13	Μπλοκ διάγραμμα της αρχιτεκτονικής 256K FFT.	60
3.14	Υλοποίηση σε VLSI του 256K FFT.	61
4.1	(α) Μονή ακρίβεια, (β) Διπλή ακρίβεια (γ) Τετραπλή ακρίβεια. .	66
4.2	Μπλοκ διάγραμμα μιας τυπικής αρχιτεκτονικής MAF.	68
4.3	Παρουσίαση της τεχνικής <i>διαίρει και βασίλευε</i> για λέξεις των 8-bit.	70
4.4	Καταχώρηση τελεστών τετραπλής, διπλής και μονής ακρίβειας σε καταχωρητή 128-bit.	70
4.5	Μπλοκ διάγραμμα του πολλαπλασιαστή κινητής υποδιαστολής, τριπλής λειτουργίας.	71
4.6	Η τεχνική <i>Διαίρει και Βασίλευε</i> για τον πολλαπλασιασμό τετραπλής ακρίβειας.	73
4.7	Η τεχνική <i>διαίρει και βασίλευε</i> για τον πολλαπλασιασμό διπλής ακρίβειας.	74
4.8	Υλοποίηση σε VLSI του τριπλής ακρίβειας, κινητής υποδιαστολής πολλαπλασιαστή.	77
4.9	Το μονοπάτι δεδομένων κινητής υποδιαστολής του επεξεργαστή LE1 VLIW	79
4.10	Μπλοκ διάγραμμα της αρχιτεκτονικής MAF.	80
4.11	Μπλοκ διάγραμμα του πολλαπλασιαστή που υλοποιείται στη μονάδα MAF μεταβλητού μήκους σωλήνωσης.	82
4.12	Αποτελέσματα της διαδικασίας σύνθεσης.	84
4.13	Υλοποίηση της μονάδας MAF μεταβλητού μήκους σωλήνωσης σε VLSI.	84
4.14	(α) Αποθήκευση ενός τελεστού διπλής ακρίβειας, (β) Αποθήκευση δύο τελεστών μονής ακρίβειας.	85
4.15	Μπλοκ διάγραμμα της αρχιτεκτονικής MAF διπλής ακρίβειας. .	87
4.16	Πολλαπλασιαστής διπλής λειτουργίας.	89
4.17	Υλοποίηση σε VLSI της μονάδας MAF διπλής λειτουργίας. . .	91
4.18	Υλοποίηση σε VLSI του επεξεργαστή LE1 VLIW, που περιέχει δύο μονάδες MAF διπλής λειτουργίας.	91
4.19	Μπλοκ διάγραμμα της πολλαπλής ακρίβειας MAF αρχιτεκτονικής. .	94
4.20	Ευθυγράμμιση στο <i>Κοντινό</i> και το <i>Μακρινό</i> μονοπάτι.	97
4.21	Παράλληλος πολλαπλασιασμός υπο-λέξεων.	99
4.22	Υλοποίηση σε VLSI της πολλαπλής ακρίβειας αρχιτεκτονικής MAF.	102

Κατάλογος Πινάκων

2.1	Αποτελέσματα της υλοποίησης σε VLSI του SoC πολυεπεξεργαστή για N=2 και N=4 επεξεργαστές.	34
3.1	Σύγκριση πολυπλοκότητας υλισμικού σε αρχιτεκτονικές FFT.	43
3.2	Αποτελέσματα υλοποίησης του 4K FFT σε VLSI.	51
3.3	Αποτελέσματα της υλοποίησης σε VLSI της 16K Παράλληλης/Παράλληλης FFT αρχιτεκτονικής.	55
3.4	Αποτελέσματα υλοποίησης σε VLSI των αρχιτεκτονικών 64K και 256K FFT.	61
4.1	Αποτελέσματα υλοποίησης σε VLSI της μονάδας MAF τριπλής λειτουργίας	101

Κεφάλαιο 1

Εισαγωγή

Η πρώτη δεκαετία του 21ου αιώνα χαρακτηρίστηκε από μια τεράστια ανάπτυξη στο χώρο της τεχνολογίας και των υπολογιστών, με τις εμπορικές συσκευές που εμφανίζονται στην αγορά να χαρακτηρίζονται από τη δυνατότητα τους να ικανοποιούν τις ολοένα και πιο αυξημένες απαιτήσεις των χρηστών, ενώ τα επόμενα χρόνια ο χώρος θα κυριαρχηθεί από ενσωματωμένα συστήματα, πληροφοριακές συσκευές και υπολογιστές κατασκευασμένους ειδικά για συγκεκριμένες λειτουργίες. Υπάρχουν όμως και περιπτώσεις όπου η απόδοση, το κόστος ή οι απαιτήσεις μιας εφαρμογής δεν μπορούν να ικανοποιηθούν από έτοιμες λύσεις που κυκλοφορούν στο εμπόριο. Σε αυτές τις περιπτώσεις οι μηχανικοί και οι επιστήμονες στρέφονται σε "κατά παραγγελία λύσεις" υλισμικού. Ανάμεσα στα κύρια χαρακτηριστικά γνωρίσματα των λύσεων αυτών είναι η υψηλή τους απόδοση, η ελαστικότητα και η διαμορφωσιμότητα τους, η επεκτασιμότητα τους, καθώς η δυνατότητα αύξησης του παραλληλισμού τους, η ικανότητα τους δηλαδή είτε να διαχειριστούν και να πραγματοποιήσουν παράλληλες εργασίες, είτε να μπορούν να λειτουργούν παράλληλα πολλαπλές τέτοιες οντότητες, αυξάνοντας έτσι τη συνολική ρυθμαπόδοση ενός συστήματος.

Η παρούσα εργασία θα ασχοληθεί με την ανάπτυξη αρχιτεκτονικών και την οργάνωση επεξεργαστών, με στόχο τον αποδοτικό παραλληλισμό εφαρμογών. Συγκεκριμένα θα προσπαθήσει να προσφέρει λύσεις σε τρεις πολύ βασικές κατηγορίες εφαρμογών, την κωδικοποίηση εικονορροών, την επεξεργασία σήματος σε πραγματικό χρόνο και τις πράξεις κινητής υποδιαστολής. Στην πρώτη περίπτωση, θα μελετηθεί η βελτίωση της απόδοσης των κωδικοποιητών MPEG-2 και MPEG-4 με την εφαρμογή παράλληλης νημάτωσης στους αλγόριθμους αυτούς και την εκτέλεση τους σε μια διάταξη πολυεπεξεργαστή. Η διάταξη αυτή βασίζεται σε έναν διαμορφώσιμο επεξεργαστή RISC 32-bit, με επεκτάσιμες εντολές διανυσμάτων και εντολές τύπου SIMD που οδηγεί σε ένα σύστημα πολυεπεξεργαστή κοινής μνήμης, ιδανικό για την εκτέλεση αλγορίθμων κωδικοποίησης εικονορροών. Στην δεύτερη κατηγορία της επεξεργασίας σήματος θα παρουσιαστούν αρχιτεκτονικές για την υλοποίηση του Ταχύ Μετασχηματισμού Fourier. Οι αρχιτεκτονικές αυτές στηρίζονται στη χρήση μιας

πρότυπης μηχανής βάσης-4³ και θα μελετηθεί πώς ο συνδυασμός τέτοιων μηχανών σε διάφορες διατάξεις, σειριακές και παράλληλες, οδηγεί σε ένα πλήθος αποτελεσματικών αρχιτεκτονικών που πραγματοποιούν Ταχύ Μετασχηματισμό Fourier για μεγάλα σύνολα δεδομένων εισόδου. Τέλος, στην περίπτωση των υπολογισμών κινητής υποδιαστολής θα παρουσιαστεί μια μονάδα πολλαπλασιαστική κινητής υποδιαστολής, πολλαπλής ακρίβειας, και θα προταθούν τρεις λύσεις για αρχιτεκτονικές Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης. Οι αρχιτεκτονικές αυτές εκτός από της εκτέλεση εντολών της μορφής $(A \times B) + C$, μπορούν επίσης να πραγματοποιήσουν μεμονομένες πράξεις πολλαπλασιασμού ή πρόσθεσης κινητής υποδιαστολής εκτελώντας παράλληλα πολλαπλές εντολές από την κάθε κατηγορία. Τα κυκλώματα που θα παρουσιαστούν προσφέρουν αποτελεσματικές λύσεις υψηλής απόδοσης για όλα τα μορφότυπα ακρίβειας που υποστηρίζει η IEEE.

Πιο αναλυτικά, στο πρώτο μέρος της εργασίας παρουσιάζεται η επιτάχυνση των κωδικοποιήσεων MPEG-2 και MPEG-4 εφαρμόζοντας παραλληλισμό νημάτων για πολυεπεξεργαστές κοινής μνήμης σε Σύστημα σε Ολοκληρωμένο (System-on-Chip - SoC). Η βελτίωση της απόδοσης των κωδικοποιητών MPEG φαίνεται από τη μείωση του δυναμικού αριθμού των εντολών σε πολλαπλά πλαίσια επεξεργαστών τα οποία απεικονίζονται σε έναν πολυεπεξεργαστή SoC. Η μείωση του δυναμικού αριθμού εντολών στον παραλληλισμένο κωδικοποιητή MPEG-2 TM5, για 32 πλαίσια επεξεργαστών, φτάνει σε μέγιστο το 95%, ενώ η μείωση στον κωδικοποιητή MPEG-4 XViD φτάνει το 83% για 16 πλαίσια επεξεργαστών, σε σχέση πάντα με τις αντίστοιχες τυπικές υλοποιήσεις. Για την υλοποίηση των παράλληλων κωδικοποιητών χρησιμοποιείται ένας διαμορφώσιμος, επεκτάσιμος πολυεπεξεργαστής SoC, N-δρόμων, με κοινή αρτηρία και συνεκτική κρυφή μνήμη (cache coherent), ο οποίος έχει επαυξηθεί με συνεπεξεργαστές παράλληλων δεδομένων. Το σύστημα αυτό του πολυεπεξεργαστή βασίζεται στην υλοποίηση πολλαπλών πλαισίων του τροποποιημένου 32-bit, RISC επεξεργαστή *Leon2*. Ένας ειδικός μηχανισμός φράγματος, βασισμένος σε υλισμικό, επιτρέπει τον αποτελεσματικό συγχρονισμό των επεξεργαστών, εξασφαλίζοντας την αποτελεσματική λειτουργία μιας τέτοιας διάταξης πολλαπλής επεξεργασίας.

Στο δεύτερο μέρος της εργασίας παρουσιάζεται μια πρωτότυπη, πλήρως συστολική αρχιτεκτονική για Ταχύ Μετασχηματισμό Fourier (FFT) 4096 σημείων σε VLSI. Η αρχιτεκτονική αυτή βασίζεται στο συνδυασμό τριών συνεχόμενων σταδίων βάσης-4 που συνιστούν μια μηχανή Ταχύ Μετασχηματισμού Fourier (FFT) 64 σημείων. Το αποτέλεσμα της τοποθέτησης δύο τέτοιων διαδοχικών μηχανών 64 σημείων FFT, είναι μια βελτιωμένη αρχιτεκτονική, που μπορεί να διαχειρίζεται αποτελεσματικά μεγάλα σύνολα δεδομένων εισόδου σε πραγματικό χρόνο. Η χρήση των μηχανών αυτών μειώνει την καθυστέρηση και τον απαιτούμενο χώρο για την ενδιάμεση αποθήκευση δεδομένων κατά το ένα τρίτο σε σχέση με την πλήρως ανοιγμένη αρχιτεκτονική βάσης-4, ενώ το ίδιο το σχήμα βάσης-4 απλοποιεί τους υπολογισμούς μέσα σε κάθε μηχανή. Η αρχιτεκτονική των 4096 σημείων υλοποιήθηκε σε FPGA επιτυγχάνοντας συχνότητα ρολο-

γιού 200 MHz με ρυθμαπόδοση 4096 σημεία/20.48usec. Η υλοποίηση της σε VLSI, σε τεχνολογία 0.13um πέτυχε στην χειρότερη περίπτωση (0.9V, 125C) συχνότητα ρολογιού 604.5 MHz και ρυθμαπόδοση 4096 σημεία/3.89usec. Η αρχιτεκτονική επεκτάθηκε επίσης για την επίτευξη υπολογισμών Ταχύ Μετασχηματισμού Fourier για 16K, 64K και 256K μιγαδικά σημεία με αντίστοιχες συχνότητες τα 352, 256 και 188 MHz.

Τέλος, όπως αναφέρθηκε και παραπάνω το τρίτο μέρος της εργασίας θα ασχοληθεί με αρχιτεκτονικές κινητής υποδιαστολής. Αρχικά παρουσιάζεται ένας πολλαπλής λειτουργίας πολλαπλασιαστής κινητής υποδιαστολής, που λειτουργεί αποτελεσματικά σε όλα τα μορφότυπα ακρίβειας που καθορίζονται από το πρότυπο IEEE 754-2008. Το σχέδιο πραγματοποιεί έναν (1) πολλαπλασιασμό τετραπλής ακρίβειας, ή δύο (2) παράλληλους πολλαπλασιασμούς διπλής ακρίβειας, ή τέλος τέσσερις (4) παράλληλους πολλαπλασιασμούς μονής ακρίβειας. Ο πολλαπλασιαστής είναι σωληνομένος έτσι ώστε να επιτρέπει την εκτέλεση ενός τετραπλής ακρίβειας πολλαπλασιασμού σε τρεις (3) κύκλους και των υπολοίπων περιπτώσεων σε δύο (2) μόνο κύκλους. Το σχέδιο βελτιώνει τη ρυθμαπόδοση κατά έναν παράγοντα 2, σε σχέση με έναν διπλής ακρίβειας πολλαπλασιαστή και κατά έναν παράγοντα 4 σε σχέση με έναν πολλαπλασιαστή μονής ακρίβειας. Η συχνότητα λειτουργίας του σε VLSI υλοποίηση φτάνει τα 505 MHz.

Στη συνέχεια της ενότητας αυτής αναλύονται τρεις (3) αρχιτεκτονικές μονάδας Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης (Multiply-Add Fused - MAF). Η αποτελεσματική λειτουργία της μονάδας Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης έχει ρόλο κλειδί στην απόδοση ενός επεξεργαστή, για ένα πλήθος από εφαρμογές. Έτσι, αρχικά παρουσιάζεται ένα σχέδιο που διατηρεί τα πλεονεκτήματα μιας τέτοιας μονάδας σε σχέση με την καθυστέρηση και τη χρησιμοποίηση του υλισμικού, αλλά ταυτόχρονα βελτιώνει την ακρίβεια των αποτελεσμάτων τόσο σε κανονικοποιημένους όσο και σε μη κανονικοποιημένους αριθμούς. Η μονάδα αυτή Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης έχει ενσωματωθεί σε έναν επεξεργαστή VLIW και έχει διαμορφώσει καθυστέρηση, μεταβάλλοντας το μήκος σωλήνωσης της είτε σε τρεις (3) κύκλους, για να επιταχυνθεί η παραγωγή των εξαιρέσεων, είτε σε πέντε (5) κύκλους για να συγχρονιστεί με τις άλλες μονάδες του επεξεργαστή. Η υψηλή ακρίβεια στα αποτελέσματα εξασφαλίζεται με τη χρήση ενός υψηλής ακρίβειας πολλαπλασιαστή και με την υλοποίηση όλων των μεθόδων στρογγυλοποίησης. Η υλοποίηση σε VLSI έγινε σε τεχνολογία 0.13um πετυχαίνοντας συχνότητα λειτουργίας 232.6 MHz.

Για την περαιτέρω βελτίωση της λειτουργικότητας της μονάδας Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης, παρουσιάζεται μια μονάδα διπλής λειτουργίας που πραγματοποιεί μία πράξη διπλής ακρίβειας ή δύο παράλληλες πράξεις μονής ακρίβειας. Το σχέδιο παρουσιάζει χαμηλή καθυστέρηση ακολουθώντας την προσέγγιση της αρχιτεκτονικής διπλού μονοπατιού και συνδυάζοντας την τελική πρόσθεση με τη στρογγυλοποίηση. Η αρχιτεκτονική επιτρέπει την εκτέλεση της εντολής MAF σε τρεις κύκλους και της εντολής πρόσθεσης κινητής υποδιαστολής σε δύο κύκλους.

Τέλος παρουσιάζεται μία μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης πολλαπλών λειτουργιών και πολλαπλής ακρίβειας. Το σχέδιο εκτελεί μία εντολή Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης τετραπλής ακρίβειας, ή δύο (2) εντολές διπλής ακρίβειας παράλληλα ή τέλος τέσσερις (4) παράλληλες εντολές μονής ακρίβειας. Η αρχιτεκτονική βασίζεται σε μια παραλλαγή της προσέγγισης του διπλού μονοπατιού που εξασφαλίζει τη χαμηλή καθυστέρηση των εντολών της πρόσθεσης κινητής υποδιαστολής, αλλά ταυτόχρονα περιορίζει την αύξηση στο χώρο που καταλαμβάνει το κύκλωμα. Στοχεύοντας στην περαιτέρω βελτίωση του χώρου κάλυψης, η αρχιτεκτονική χρησιμοποιεί τον ελάχιστο δυνατό αριθμό από δομοστοιχεία, ο σχεδιασμός των οποίων επιτρέπει τη εκτέλεση πολλαπλών διαφορετικών λειτουργιών.

Η παρούσα διατριβή χωρίζεται σε πέντε κεφάλαια. Στο κεφάλαιο 2 περιγράφεται η νημάτωση των MPEG κωδικοποιητών και αναλύεται η διάταξη του SoC πολυεπεξεργαστή που χρησιμοποιείται για την εκτέλεση των αλγορίθμων κωδικοποίησης των εικονορροών. Στο κεφάλαιο 3 παρουσιάζεται η διάταξη της μηχανής βάσης-4³ και περιγράφεται η δομή και η λειτουργικότητα των αρχιτεκτονικών Ταχύ Μετασχηματισμού Fourier, που βασίζονται στη χρήση της μηχανής αυτής. Το κεφάλαιο 4 παρουσιάζει τις αρχιτεκτονικές κινητής υποδιαστολής, ενώ τέλος, το 5ο κεφάλαιο συνοψίζει τη διατριβή και αναφέρει τα μελλοντικά βήματα για τη συνέχιση της ερευνητικής δραστηριότητας.

Κεφάλαιο 2

Παραλληλισμός νημάτων σε κωδικοποιητές MPEG-2, MPEG-4 για SoC πολυεπεξεργαστές

2.1 Εισαγωγή

Η κωδικοποίηση εικονορροών (video) έχει γίνει τα τελευταία δέκα χρόνια επιτακτική ανάγκη, τόσο για τις ψηφιακές επικοινωνίες μέσω video, όσο και για την εξέλιξη συσκευών αναπαραγωγής/εγγραφής όπως πχ. τα DVD, καθώς και για εφαρμογές όπως τηλεόραση υψηλής ευκρίνειας (HDTV). Η προηγμένη κωδικοποίηση εικονορροών επιτρέπει τη μείωση κατά τάξεις μεγέθους του δυφιακού ρυθμού (bit-rate) και αποτελεί καθοριστικό παράγοντα στην προώθηση των προσωπικών επικοινωνιών σε ένα υψηλότερο επίπεδο διαδραστικότητας. Έτσι, η τεχνολογία αυτή εφαρμόζεται με πολύ μεγάλη επιτυχία και σε άλλα ενσωματωμένα συστήματα για προσωπική χρήση όπως ψηφιακές κάμερες, κινητά τηλέφωνα, υπολογιστές τύπου tablet ή φορητές κονσόλες παιχνιδιών video.

Από τα πιο διαδεδομένα πρότυπα συμπίεσης είναι το MPEG-2 [1] και το MPEG-4 [2], τα οποία αναπτύχθηκαν με στόχο την υλοποίηση εμπορικών συσκευών ικανών να παρέχουν αποτελεσματική επικοινωνία. Τα πρότυπα αυτά χαρακτηρίζονται από τις σημαντικές απαιτήσεις τους σε επεξεργασία, λόγω της χρήσης προηγμένων αλγορίθμων, ειδικά για εφαρμογές πραγματικού χρόνου. Σαν λύση στο πρόβλημα αυτό, τόσο η βιομηχανία όσο και η πανεπιστημιακή κοινότητα, πρότειναν τη χρήση ολοκληρωμένων ειδικών εφαρμογών (Application Specific Integrated Circuits - ASIC) και/ή τη χρήση πολυεπεξεργαστών [3], ώστε να μπορεί να πραγματοποιηθεί η λειτουργία σε πραγματικό χρόνο και να βελτιωθεί το επικοινωνιακό εύρος ζώνης και η ποιότητα της κωδικοποιημένης εικόνας.

Η υλοποίηση των πολυεπεξεργαστών γίνεται συνήθως σε συσκευές Σύστη-

μα σε Ολοκληρωμένο (System-on-Chip ή SoC), οι οποίες έχουν την δυνατότητα εκμετάλλευσής του άπλετου παραλληλισμού που μπορεί να βρεθεί στους πολύπλοκους αλγόριθμους της κωδικοποίησης εικονορροών σε πραγματικό χρόνο. Οι δύο επικρατέστερες μορφές παραλληλισμού σε αυτές τις περιπτώσεις είναι ο *Παραλληλισμός σε Επίπεδο Δεδομένων* (Data Level Parallelism ή DLP) και ο *Παραλληλισμός σε Επίπεδο Νημάτων* (Thread Level Parallelism ή TLP). Ο *Παραλληλισμός σε Επίπεδο Δεδομένων*, που είναι και ο τρόπος του παραλληλισμού που εφαρμόζεται πιο συχνά, υλοποιείται μέσα από αρχιτεκτονικές τύπου διανυσμάτων ή SIMD. Τα πλεονεκτήματα του *Παραλληλισμού σε Επίπεδο Νημάτων* προκύπτουν από τη διανομή του φόρτου εργασίας ενός υψηλής απόδοσης επεξεργαστή σε έναν αριθμό από πιο απλούς και πιο αργούς πυρήνες επεξεργαστών [4].

Στο κεφάλαιο αυτό περιγράφεται πως μπορεί να πραγματοποιηθεί επιτάχυνση των προτύπων MPEG, μέσω του παραλληλισμού των κωδικοποιητών σε επίπεδο νημάτων. Η εκτίμηση της απόδοσης των παράλληλων κωδικοποιητών πραγματοποιείται σε έναν προσομοιωτή κατασκευασμένο ειδικά για το σκοπό αυτό. Ο βασικός λόγος της χρήσης του προσομοιωτή αυτού, είναι για να μετρηθεί η απόδοση των πολλαπλών επεξεργαστών στα διάφορα πακέτα εργασίας (workloads) και η λειτουργικότητα του είναι παρόμοια με αυτήν του μοντέλου Exclusive-Read, Exclusive-Write Parallel Random Access Machine (EREW-PRAM). Για την υλοποίηση των παράλληλων αλγορίθμων χρησιμοποιήθηκε μια διαμορφώσιμη, επεκτάσιμη αρχιτεκτονική πολυεπεξεργαστή SoC. Η αρχιτεκτονική αυτή στηρίζεται σε μια πλατφόρμα που περιλαμβάνει έναν ανοιχτού κώδικα διαμορφώσιμο επεξεργαστή τύπου RISC [5], συμβατό με τον 32 bit επεξεργαστή SPARC V8 [6]. Ο επεξεργαστής αυτός έχει ενισχυθεί, προσαρμόζοντας τον μαζί με έναν παραμετρικό επιταχυντή διανυσμάτων (vector accelerator) και με μια υψηλής ταχύτητας εξωτερική μνήμη. Πολλαπλές οντότητες των επεξεργαστών αυτών δημιουργούν το σύστημα του πολυεπεξεργαστή. Το σύστημα αυτό έχει επαυξηθεί ώστε να περιλαμβάνει έναν καινοτόμο μηχανισμό φράγματος και συνεπεξεργαστές παράλληλων δεδομένων, στους οποίους μπορούν να απεικονιστούν αποτελεσματικά κωδικοποιητές παράλληλων νημάτων, παρέχοντας πλήρη υποστήριξη στα κομμάτια του φόρτου εργασίας με παράλληλα δεδομένα [7]. Η μελέτη που περιγράφεται στη συνέχεια πραγματοποιήθηκε σε συνεργασία με το Πανεπιστήμιο του Loughborough. Η συνεισφορά της παρούσας εργασίας αφορά κυρίως τη μελέτη και υλοποίηση του μηχανισμού φράγματος και του συστήματος διεπαφής ανάμεσα στον κεντρικό επεξεργαστή και το συνεπεξεργαστή της πλατφόρμας.

Σχετική δουλειά που παρουσιάζεται στο [8] έχει διερευνήσει τις διαφορές πλευρές της αρχιτεκτονικής και της μικροαρχιτεκτονικής που ασχολούνται με την εκμετάλλευση του πιο κοινά αποδεκτού τρόπου παραλληλισμού, τον *παραλληλισμό σε επίπεδο δεδομένων*. Επίσης στο [9] οι συγγραφείς ερευνούν τον παραλληλισμό σε επίπεδο νημάτων σε πακέτα εικονορροών χρησιμοποιώντας σαν υπολογιστική μηχανή ένα κατανεμημένο δίκτυο από επεξεργαστές (workstations), αντί για ένα σύστημα πολυεπεξεργαστών με κοινή μνήμη. Στην περι-

πτωση αυτή ο φόρτος εργασίας διανέμεται με πολύ μικρότερο βαθμό ανάλυσης (granularity level) [10], [11], [12], [13], απ'ότι η τεχνική που παρουσιάζεται εδώ. Για τέτοιες περιπτώσεις, που αφορούν νημάτωση με μικρό βαθμό ανάλυσης σε επίπεδο συνόλου εικόνων (Group-of-Pictures), μια αρχιτεκτονική πολλαπλών υπολογιστών είναι πιο κατάλληλη, λόγω των χαμηλών απαιτήσεων του νηματωμένου αλγορίθμου για ενδοεπικοινωνία των επεξεργαστών.

Από την άλλη πλευρά, μια τέτοια αρχιτεκτονική πολλαπλών υπολογιστών είναι λιγότερο ικανή από τα συστήματα κοινής μνήμης που υλοποιούνται σε ένα System-on-Chip, λόγω του μη συνεχόμενου εύρους διευθύνσεων, με αποτέλεσμα να απαιτείται περισσότερος χώρος για τη δημιουργία ξεχωριστής κρυφής μνήμης (cache memory) για κάθε Κεντρική Μονάδα Επεξεργασίας (CPU ή αλλιώς ΚΜΕ). Αντίθετα, τα συστήματα κοινής μνήμης χρειάζονται λιγότερο χώρο και μπορούν να πάρουν τη μορφή πολυεπεξεργαστών πλακιδίου (Chip-Multiprocessors ή CMP), επεξεργαστών πολλαπλών νημάτων (Multi-Threaded processors ή MT), ή πολυεπεξεργαστών πλακιδίου-πολλαπλών νημάτων (CMP-MT). Και στις τρεις αυτές περιπτώσεις ο διαμοιρασμός των πόρων μπορεί να γίνει με πολύ πιο βελτιωμένο τρόπο, και για το λόγο αυτό η μικροαρχιτεκτονική που χρησιμοποιείται είναι στοχευμένη σε τέτοια συστήματα κοινής μνήμης, λόγω της μεγάλης τους ευελιξίας και της δυνατότητας διεύρυνσης της απόδοσης τους. Επιπλέον, έχει εκμεταλλευθεί ο *παραλληλισμός σε επίπεδο νημάτων* σε πολύ μεγαλύτερο βαθμό απ'ότι σε άλλα υπάρχοντα σχήματα, με τα αποτελέσματα να δείχνουν πως το TLP είναι μια πολύ ικανή μορφή παραλληλισμού, κάθετη στο συνήθη τρόπο παραλληλισμού των δεδομένων που χρησιμοποιείται στις επικοινωνίες και τα πολυμέσα.

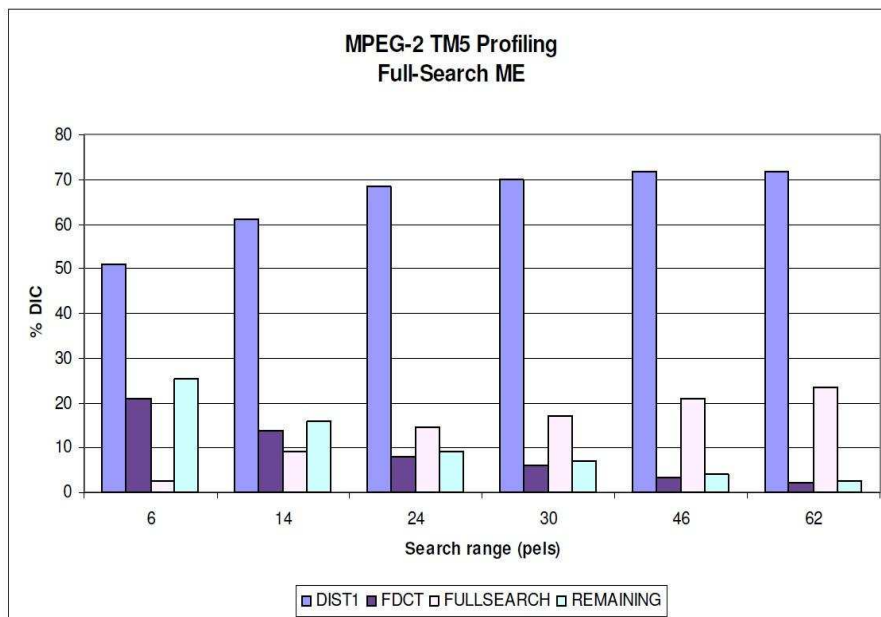
2.2 Μεθοδολογία

2.2.1 Μελέτη και κατηγοριοποίηση των κωδικοποιητών

Στη συνέχεια περιγράφεται εν συντομία η διαδικασία μελέτης των χαρακτηριστικών (profiling) των δύο προτύπων κωδικοποίησης που προηγείται της νημάτωσης τους. Η δουλειά που περιγράφεται εδώ θεωρεί σα μέτρο πολυπλοκότητας τον *Δυναμικό Αριθμό Εντολών* (Dynamic Instruction Count ή DIC) των κωδικοποιητών, τον αριθμό δηλαδή των εντολών που εκτελούνται χωρίς χρονικές επιδράσεις, μεταγλωττισμένες για μια ΚΜΕ τύπου MIPS II. Οι νηματωμένοι κώδικες μεταγλωττίστηκαν σε ένα Προσομοιωτή Συνόλου Εντολών (Instruction-Set-Simulator ή ISS) γνωστό και ως *sim-system* ο οποίος βασίζεται στην υποδομή που περιγράφεται στο [14]. Η κάθε εντολή μοντελοποιείται σαν να εκτελείται σε μια μονάδα χρόνου (1 κύκλο), ανεξάρτητα με τον είδος της εντολής. Εγγραφές και διαβάσματα, από και προς την ίδια τοποθεσία, στην ίδια μονάδα χρόνου (στον ίδιο κύκλο) εκτελούνται αλληλοδιαδοχικά μέσω ενός βρόγχου επανάληψης, και για το λόγο αυτό το σύστημα προσομοίωσης μπορεί να θεωρηθεί και σαν ένα μοντέλο EREW PRAM.

Ο κωδικοποιητής MPEG-2 TM5 κατηγοριοποιήθηκε για έναν μεγάλο αριθμό από ακολουθίες εικονορροών και είδη αναζήτησεων, για να αναγνωριστούν τα πιο απαιτητικά τμήματα του κωδικοποιητή ως προς τους υπολογισμούς σε επίπεδο συναρτήσεων. Η εικόνα 2.1 συνοψίζει τα αποτελέσματα με βάση το επί της εκατό ποσοστό του δυναμικού αριθμού εντολών (DIC) για την Εκτίμηση Κίνησης Πλήρους—Αναζήτησης (Full-Search ME), κατά την κωδικοποίηση δώδεκα (12) πλαίσια. Η νημάτωση του κωδικοποιητή σε αυτό το επίπεδο επιτρέπει επίσης την εύκολη εισαγωγή ευρετικών (γρήγορων) μεθόδων ME, όπως περιγράφεται στο [15]. Οι μέθοδοι αυτές είναι λιγότερο πολύπλοκες απ'ότι οι εξαντλητικές FS-ME, και έχουν οδηγήσει σε υλοποιήσεις MPEG-2 πραγματικού χρόνου (25 πλαίσια/δευτερόλεπτο), εντός των ενεργειακών ορίων κατανάλωσης που έχουν οι φορητές συσκευές. Αυτό αποτελεί ένα σημαντικό στοιχείο της νημάτωσης του MPEG-2 με τον προτεινόμενο τρόπο.

Ο κωδικοποιητής MPEG-4 [16] μελετήθηκε για τρία διαφορετικά επίπεδα ποιότητας, τα οποία επηρεάζουν κυρίως τις παραμέτρους που χρησιμοποιούνται στην εκτίμηση κίνησης και στην επανόρθωση (compensation) και συνεπώς επηρεάζουν και την απόδοση τους. Για να βρεθεί το μέτρο πολυπλοκότητας (DIC) του μη-νηματομένου κωδικοποιητή, τέσσερις παραπεμπτικές ακολουθίες εικονορροών κωδικοποιήθηκαν, με τα αποτελέσματα να παρουσιάζονται στην εικόνα 2.2. Όπως είναι προφανές, οι συναρτήσεις ME, MC, DCT χρήζουν παραλληλισμού τόσο σε επίπεδο δεδομένων όσο και σε επίπεδο νημάτων, ώστε να διευκολυνθεί η υλοποίησή τους σε πραγματικό χρόνο από φορητές συσκευές. Ο κωδικοποιητής MPEG-4 βασίζεται σε μια τμηματική δομή (block based) και περιλαμβάνει επαναληπτικά, ανεξάρτητα βήματα. Έτσι, η ίδια η φύση του κωδικοποιητή οδηγεί στον παραλληλισμό της διαδικασίας σε επίπεδο νημάτων, το οποίο είναι και το αντικείμενο αυτής της μελέτης.

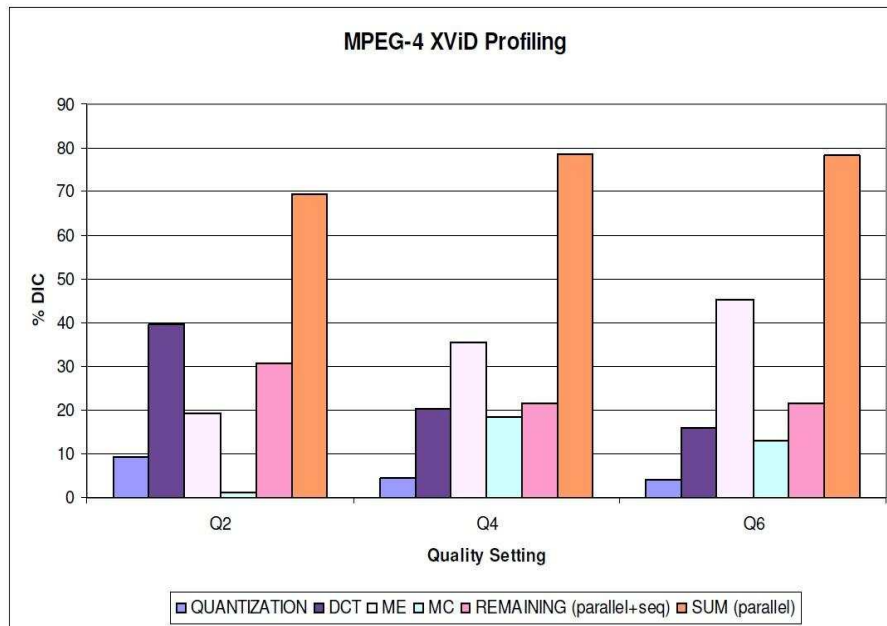


Σχήμα 2.1: Κατανομή του δυναμικού αριθμού εντολών ανά συνάρτηση στον κωδικοποιητή MPEG-2 TM5.

2.2.2 Νημάτωση

Οι χαρακτηριστικές διαφορές ανάμεσα στα δυναμικά και τα στατικά πολυήματα περιγράφονται στο [17]. Στα δυναμικά πολυήματα, το υλισμικό είναι αυτό που αποφασίζει πότε να γεννήσει επιπλέον νήματα, ενώ εκτελείται μια μη νηματωμένη εφαρμογή. Στα στατικά πολυήματα, η εφαρμογή έχει μελετηθεί στατικά (off-line) και έχει παραλληλιστεί από τον προγραμματιστή ή από έναν μεταγλωττιστή παραλληλισμού. Στη δικιά μας περίπτωση ακολουθείται η στατική προσέγγιση σαν πιο κατάλληλη για μια υπολογιστική πλατφόρμα SoC και για τη φύση των δύο προτύπων κωδικοποίησης που είναι βασισμένη σε μπλοκ κομμάτια. Για την αποτελεσματική εφαρμογή της προσέγγισης αυτής θα πρέπει να αναγνωριστούν οι πιο κατάλληλες αναθέσεις των νημάτων σε επεξεργαστές, εξασφαλίζοντας ότι οι προσβάσεις στα δεδομένα δεν δημιουργούν προβλήματα από υπέρμετρη επικοινωνία μεταξύ των επεξεργαστών.

Η μελέτη των δύο προτύπων MPEG έδειξε ότι οι συναρτήσεις με τις υψηλότερες υπολογιστικές απαιτήσεις είναι οι DIST1, FULLSEARCH, FDCT (MPEG-2) και ME/MC, DCT (MPEG-4). Για τον αποτελεσματικό παραλληλισμό έγινε νημάτωση των συναρτήσεων αυτών για κάθε κωδικοποιητή. Για να μειωθεί το κόστος επικοινωνίας ανάμεσα στα παράλληλα νήματα, τα παράλληλα τμήματα συγχρονίστηκαν χρησιμοποιώντας έναν μηχανισμό φράγματος, ο οποίος δημιουργεί ένα ειδικό σήμα καθυστέρησης (stall signal), καθυστερώντας



Σχήμα 2.2: Κατανομή του δυναμικού αριθμού εντολών ανά συνάρτηση στον κωδικοποιητή MPEG-4 XviD.

τον επεξεργαστή που έφτασε το σημείο συγχρονισμού, μέχρι να συγχρονιστούν όλοι οι επεξεργαστές. Με τον τρόπο αυτόν αποφεύγεται η χρήση των τυπικών μηχανισμών συγχρονισμού κοινής μνήμης, τύπου spinlocks [18]. Ο απλός αυτό μηχανισμός φράγματος έχει εξαιρετική σημασία στην αποτελεσματική υλοποίηση της στρατηγικής συγχρονισμού που έχει επιλεγεί.

2.2.2.1 MPEG-2 TM5

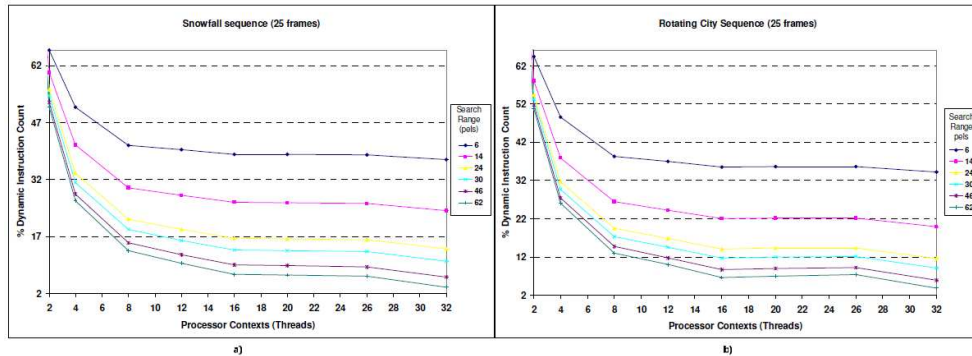
Η νημάτωση του κωδικοποιητή MPEG-2 έγινε αναθέτοντας νήματα λογισμικού σε ξεχωριστά πλαίσια ΚΜΕ, για τις συναρτήσεις που διαπιστώθηκε ότι έχουν έντονες υπολογιστικές απαιτήσεις. Το ME νηματώθηκε σε επίπεδο μάκρομπλοκ, με το κάθε νήμα να είναι υπεύθυνο να πραγματοποιεί τόσο αναζητήσεις όσο και υπολογισμούς αναθέσεων διανυσμάτων κίνησης. Η νημάτωση του βρόχου ME έγινε στη συνάρτηση *motion_estimation*. Ο βρόχος επεξεργάζεται τα μάκρομπλοκ κατά σειρά καλώντας τη συνάρτηση *frame_ME* ή τη συνάρτηση *field_ME* ανάλογα με το εάν η εικόνα είναι διαπλεκόμενη (interlaced) ή όχι αντίστοιχα. Σε κάθε επανάληψη ο δείκτης δομής πληροφορίας του μάκρομπλοκ *mbi* ανανεώνεται ώστε να περιέχει πληροφορία που σχετίζεται με το μάκρομπλοκ που επεξεργάζεται. Για τη σωστή παράλληλη λειτουργία δημιουργήθηκε ένας πίνακας από τέτοιους δείκτες, με μήκος ίσο με τον αριθμό των πλαισίων επεξεργαστών, με τον κάθε δείκτη να περιέχει τις διευθύνσεις του συνόλου των

μάκρομπλοκ που επεξεργάζονται από ένα συγκεκριμένο νήμα.

Η νηματώση του FDCT έγινε με βάση την εκάστοτε συνιστώσα φωτεινότητας/χρώματος, μέσα στη συνάρτηση *transform*, με το κάθε νήμα να επεξεργάζεται ένα μέρος του πλαισίου εισόδου. Οι λειτουργίες στο νηματωμένο βρόχο αφορούν την αρχικοποίηση των παραμέτρων που χρησιμοποιεί η συνάρτηση *sub_pred* για να υπολογίσει το σφάλμα πρόβλεψης και η αφαίρεση του από το αθροιστικό σύνολο, πριν κληθεί ο FDCT. Το αθροιστικό όμως σύνολο δεν είναι αποκλειστικό για κάθε νήμα και έτσι δεν μπορεί να επιτραπεί ο παράλληλος υπολογισμός του. Για να μπορέσουν λοιπόν τα νήματα να καλέσουν τη συνάρτηση *sub_pred* χρησιμοποιώντας τα δεδομένα άλλων πλαισίων δημιουργήθηκαν κοινόι πίνακες στους οποίους το κάθε νήμα αποθηκεύει τα δεδομένα του. Για να αποφευχθούν φαινόμενα ταχύτητας (race effects) δημιουργήθηκαν σημεία συγχρονισμού (φράγματα). Το φράγμα (*barrier*) είναι μια ειδική εντολή στον επεξεργαστή Leon-2. Όταν η ροή εκτέλεσης φτάσει σε ένα φράγμα, ο αντίστοιχος επεξεργαστής εισέρχεται σε μια κατάσταση αναστολής. Στην κατάσταση αυτή δεν εκτελούνται άλλες εντολές μέχρις ότου ένα εξωτερικό σήμα ελέγχου να ακυρώσει την κατάσταση αυτή. Η ακύρωση συμβαίνει μόνον όταν όλοι οι επεξεργαστές έχουν εισέλθει στην κατάσταση αναστολής, επιτυγχάνοντας έτσι συγχρονισμό όλων των παράλληλων πλαισίων.

Αντίστοιχη προσέγγιση στη νηματώση, υιοθετήθηκε και για τη συνάρτηση *itransform* στον υπολογισμό του IDCT. Σε αυτήν την περίπτωση η κλήση του IDCT γίνεται πριν από τη συνάρτηση *add_pred* που προσθέτει το σφάλμα πρόβλεψης στο αθροιστικό σύνολο. Ο νηματωμένος κώδικας επιτρέπει την παράλληλη εκτέλεση των πράξεων του IDCT, όμως και πάλι είναι απαραίτητη η χρήση ενός φράγματος πριν από τις σειριακές πράξεις της συνάρτησης *add_pred*.

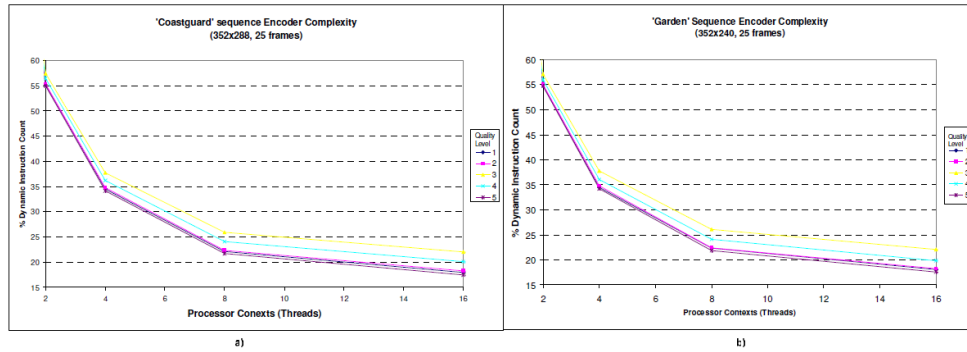
Η απόδοση του νηματωμένου κωδικοποιητή MPEG-2 εκτιμήθηκε με μια σειρά από ακολουθίες εινορρών. Οι ακολουθίες αυτές είναι 25 πλαισίων, με αναλύσεις 325×288 και 512×380 και δυφιακό ρυθμό $4096 Kbps$, για τις οποίες χρησιμοποιήθηκε ο αλγόριθμος Full-Search ME. Η εικόνα 2.3 δείχνει τη μείωση του δυναμικού αριθμού εντολών για το κύριο πλαίσιο επεξεργαστή (νήμα 0), για 2 ακολουθίες ('snowfall', 'rotating city'). Το πλαίσιο 0 είναι το βασικό νήμα στον παράλληλο κωδικοποιητή αφού διαχειρίζεται την είσοδο/έξοδο και είναι αυτό που ενεργοποιεί τα υπόλοιπα νήματα, και επομένως δέχεται τη μεγαλύτερη επιβάρυνση. Τα γραφήματα δείχνουν μια σημαντική μείωση του δυναμικού αριθμού εντολών όταν αυξάνεται ο αριθμός των επεξεργαστών και όταν αυξάνει το παράθυρο του εύρους αναζήτησης, με τη μέγιστη μείωση να φτάνει στο 95% για 32 πλαίσια επεξεργαστών.



Σχήμα 2.3: Θεωρητική απόδοση του MPEG-2 TM5 κωδικοποιητή για τις ακολουθίες εικονορροών (α) 'Snowfall' και (β) 'Rotating City'.

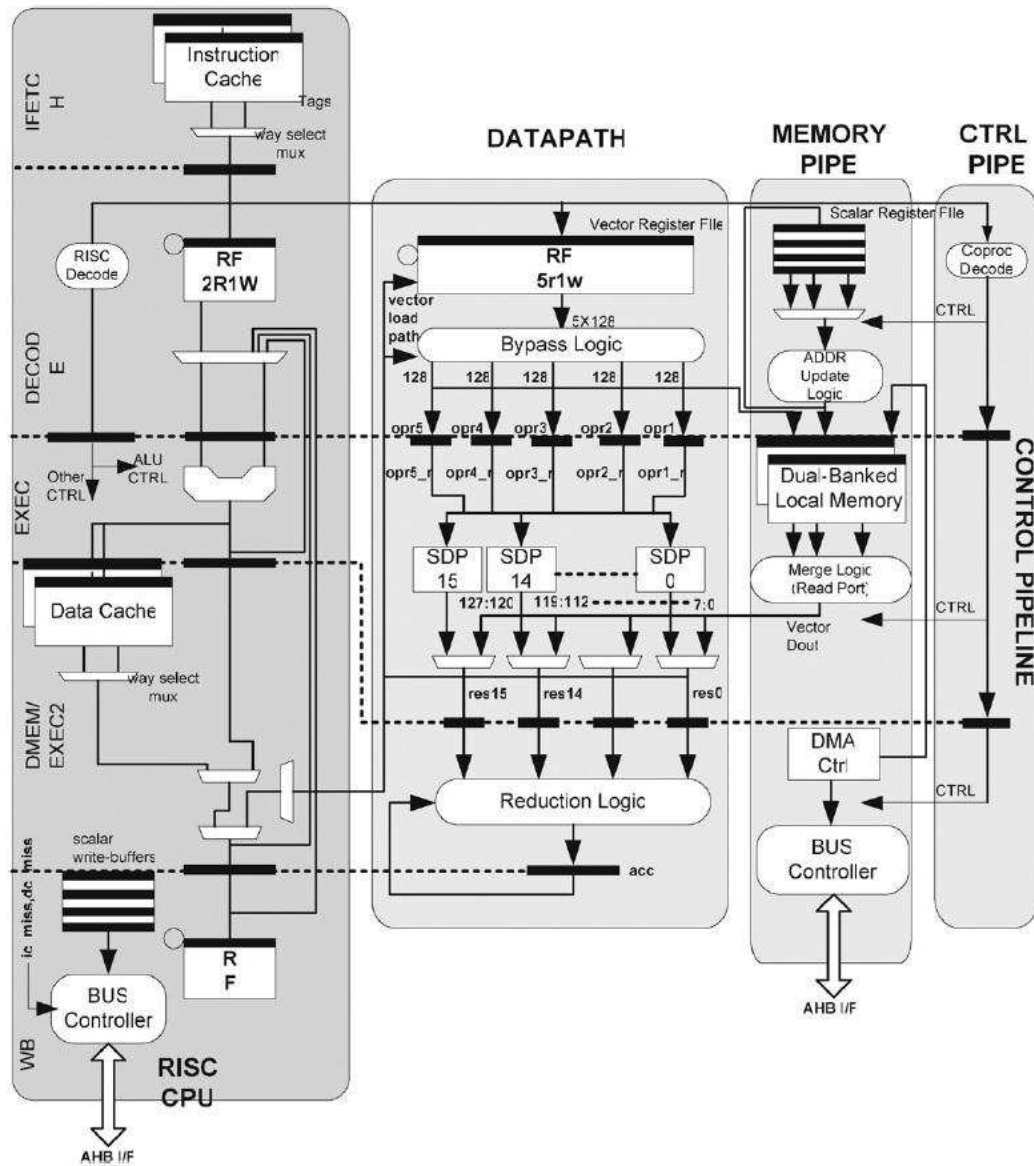
2.2.2.2 MPEG-4 (XViD)

Στον κωδικοποιητή MPEG-4 η πιο απαιτητική συνάρτηση είναι η *MotionEstimation*, που υπολογίζει ανά σειρά τα διανύσματα κίνησης (MVs) του κάθε μάκρομπλοκ (MB). Ο κύριος βρόχος σε αυτήν την συνάρτηση κάνει προβλέψεις για τα διανύσματα κίνησης και στη συνέχεια χρησιμοποιεί τις προβλέψεις αυτές για να βρει τα καλύτερα MVs. Ανάλογα με την επιτυχία της αναζήτησης, αυτή επαναλαμβάνεται το πολύ 4 φορές χρησιμοποιώντας διαφορετικές προβλέψεις. Οι συναρτήσεις αναζήτησης είναι οι *search8* και η *search16*, οι οποίες είναι ανεξάρτητες του μάκρομπλοκ που βρίσκεται στο τρέχον πλαίσιο, αφού χρησιμοποιούν ως αναφορά προηγούμενα πλαίσια. Χρησιμοποιώντας τα τα διανύσματα κίνησης των γειτονικών μάκρομπλοκ που έχουν ήδη επεξεργαστεί, η συνάρτηση πρόβλεψης υπολογίζει τη μέση τιμή τους. Αυτό όμως το βήμα της κωδικοποίησης δημιουργεί μια εξάρτηση δεδομένων που παρεμποδίζει τον παράλληλο υπολογισμό για μια γραμμή μάκρομπλοκ. Για να ξεπεραστεί το πρόβλημα αυτό, εφαρμόστηκε ένας διαφορετικός τρόπος για την αποτελεσματική νημάτωση της συνάρτησης ME του κωδικοποιητή MPEG-4. Η μέθοδος αυτή βασίζεται στην πρόβλεψη των διανυσμάτων κίνησης (MVs) βάσει του τρέχοντος πλαισίου και χρησιμοποιώντας γειτονικά MVs από το προηγούμενο πλαίσιο. Λαμβάνοντας υπ'όψιν τη χρονική διαθεσιμότητα των MVs των γειτονικών μάκρομπλοκ και για να επιτραπεί η παράλληλη κωδικοποίηση των MB μιας γραμμής, η μέθοδος αυτή προβλέπει την κίνηση ενός συγκεκριμένου MB χρησιμοποιώντας τα MB που βρίσκονται πάνω από αυτό. Για την εκτίμηση της απόδοσης της μεθόδου αυτής χρησιμοποιήθηκαν διάφορες ακολουθίες ελέγχου εικονορροών. Τα αποτελέσματα έδειξαν μια μείωση στην ποιότητα κατά 0.1db και αύξηση στο δυφιακό ρυθμό (bit rate) κατά 2% σε σχέση με την κανονική υλοποίηση. Έχοντας όμως εξαφανίσει την εξάρτηση, ο υπολογισμός των διανυσμάτων κίνησης μπορεί πλέον να γίνει από ανεξάρτητα νήματα λογισμικού που τρέχουν σε ξεχωριστά πλαίσια KME.

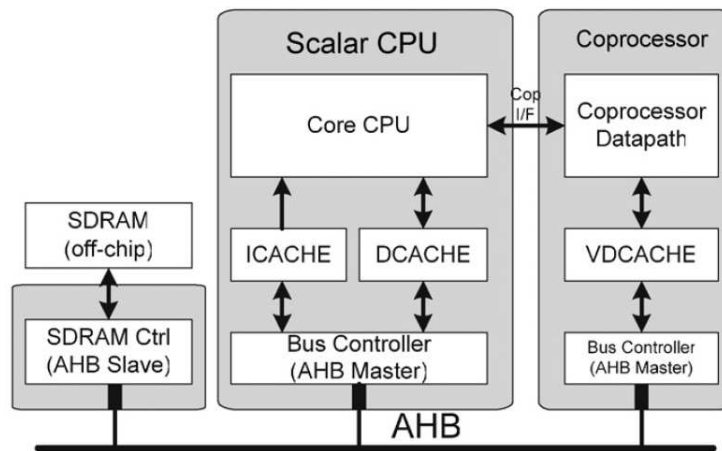


Σχήμα 2.4: Θεωρητική απόδοση του MPEG-4 XViD κωδικοποιητή για τις ακολουθίες εικονορροών (α) 'Coastguard' και (β) 'Garden'.

Η απόδοση του κωδικοποιητή MPEG-4 με παράλληλη νημάτωση εκτιμήθηκε με τρεις ακολουθίες video ('Garden', 'Coastguard', 'Foreman'). Όλες οι ακολουθίες αποτελούνται από 25 πλαίσια και κωδικοποιήθηκαν με ανάλυση $352 \times 240pels$, εκτός της τελευταίας που είχε ανάλυση $352 \times 288pels$. Η εικόνα 2.4 αποτυπώνει τον δυναμικό αριθμό εντολών σαν συνάρτηση του αριθμού των πλαισίων επεξεργαστών και του επιπέδου ποιότητας για δύο από τις ακολουθίες. Τα γραφήματα φανερώνουν μια μέγιστη μείωση της τάξης του 83% του δυναμικού αριθμού των εντολών για 16 πλαίσια επεξεργαστών, στο 5ο επίπεδο ποιότητας, ενώ για διάταξη 2 επεξεργαστών υπάρχει μια σχεδόν γραμμική μείωση της τάξης του 45%. Τα παραπάνω αποτελέσματα, που προκύπτουν από τον προσομοιωτή, περιλαμβάνουν την επιπλέον επιβάρυνση λόγω συγχρονισμού, αλλά δεν μοντελοποιούν επιδράσεις όπως η αύξηση της αναλογίας των κύκλων ανά εντολή (CPI ratio) που αναμένεται να υπάρχει σε έναν SoC πολυεπεξεργαστή κοινής αρτηρίας.



Σχήμα 2.5: Σχηματική αναπαράσταση της μικροαρχιτεκτονικής που περιλαμβάνει μια βαθμωτή ΚΜΕ και ένα συνεπεξεργαστή παράλληλων δεδομένων.



Σχήμα 2.6: Σχηματική αναπαράσταση του συστήματος επεξεργαστή - συνεπεξεργαστή σε υψηλό επίπεδο.

2.3 Μικροαρχιτεκτονική

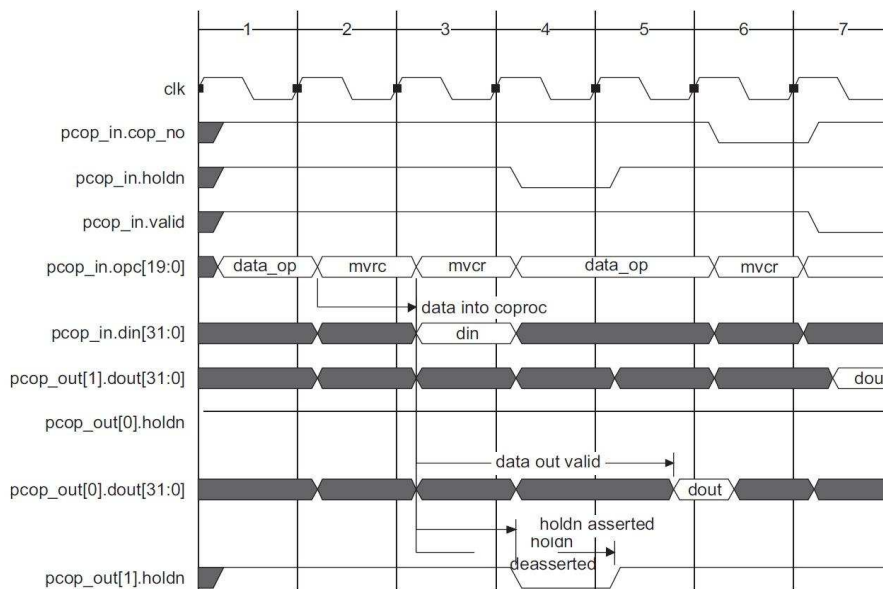
Στην παράγραφο αυτή θα αναλυθεί η υλοποίηση των TLP κωδικοποιητών στον ανοιχτού κώδικα, 32-bit RISC επεξεργαστή, που χρησιμοποιείται σαν βασική πλατφόρμα [5]. Η εικόνα 2.6 δείχνει σε υψηλό επίπεδο τη δομή του συστήματος επεξεργαστή - συνεπεξεργαστή. Τα κύρια κομμάτια του συστήματος αυτού είναι η τροποποιημένη Sparc V8 KME, η διεπαφή του συνεπεξεργαστή και ο επιταχυντής εικονορροών.

Το Σύνολο Εντολών (Instruction Set) έχει επεκταθεί στον ελεύθερο χώρο εντολών, ώστε να συμπεριλάβει επιπλέον εντολές για τα διανύσματα και το συγχρονισμό του φράγματος. Επίσης, το προγραμματιστικό μοντέλο επεκτάθηκε για να συμπεριλάβει ένα ξεχωριστό, μη προγραμματίσιμο, πεδίο ταυτότητας επεξεργαστή (ID), το οποίο χρησιμοποιείται για να αναγνωρίζει το πλαίσιο επεξεργαστή που εκτελεί ένα συγκεκριμένο νήμα υλισμικού. Στην εικόνα 2.5 φαίνεται η μικροαρχιτεκτονική, η οποία περιέχει τον παραμετρικό επιταχυντή διανυσμάτων (vector accelerator) που υλοποιεί τρεις εντολές (custom) για τα τμήματα παράλληλων δεδομένων του κωδικοποιητή MPEG-2, και ο οποίος είναι συνδεδεμένος με μια τυπική βαθμωτή KME 5-σταδίων.

2.3.1 Διεπαφή συνεπεξεργαστή

Η διεπαφή συνεπεξεργαστή που διαθέτει ο ανοιχτού κώδικα, ενσωματωμένος επεξεργαστής RISC δεν μπορεί να λειτουργήσει αποτελεσματικά με λειτουργία σωλήνωσης, επηρεάζοντας έτσι την απόδοση του συνεπεξεργαστή. Για το λόγο αυτό δημιουργήθηκε μια νέα σωληνωμένη διεπαφή συνεπεξεργαστή που μπορεί να διαχειριστεί δύο συνεπεξεργαστές και μπορεί να δίνει μια εντολή ανά κύκλο

σε οποιονδήποτε από τους δύο συνεπεξεργαστές. Οι συνεπεξεργαστές φροντίζουν για τον έλεγχο ροής προς τον κεντρικό επεξεργαστή μέσω ενός σήματος καθυστέρησης (stall signal). Στην εικόνα 2.7 φαίνεται μια πράξη δεδομένων του συνεπεξεργαστή στον κύκλο 1, που ακολουθείται από μια μεταφορά δεδομένων καταχωρητή από τον επεξεργαστή προς τον συνεπεξεργαστή στον κύκλο 2. Στον 3ο κύκλο ο RISC επεξεργαστής ζητάει τα περιεχόμενα ενός καταχωρητή του συνεπεξεργαστή, αλλά λόγω εσωτερικών συνθηκών καθυστέρησης, τα δεδομένα γίνονται διαθέσιμα έναν κύκλο αργότερα (κύκλος 5 αντί για κύκλος 4). Κατά το χρόνο αυτόν ο κεντρικός επεξεργαστής μένει ανενεργός με τη βοήθεια του σήματος *holdn*. Τέλος, μια δεύτερη πράξη διαβάσματος, αυτή τη φορά προς τον άλλο συνεπεξεργαστή ξεκινάει στον 6ο κύκλο, με τα αποτελέσματα να είναι διαθέσιμα έναν κύκλο μετά.



Σχήμα 2.7: Παράδειγμα λειτουργίας της διεπαφής επεξεργαστή - συνεπεξεργαστή.

2.3.2 Βαθμωτή Κεντρική Μονάδα Επεξεργασίας

Η υλοποίηση της ΚΜΕ είναι μια τυπική RISC αρχιτεκτονική με σωλήνωση 5 σταδίων. Οι εντολές προσκομίζονται από την πολλαπλών δρόμων, συνολο-συσχετιστική, κρυφή μνήμη εντολών, και αποθηκεύονται στον καταχωρητή εντολών. Στο στάδιο της αποκωδικοποίησης (decode stage) η καταχωρημένη εντολή αποκωδικοποιείται, ενώ η πρόσβαση στο αρχείο καταχωρητή του RISC γίνεται στην κατερχόμενη ακμή του ρολογιού.

Συνήθως οι υψηλής απόδοσης βαθμωτοί επεξεργαστές, με αντίστοιχο βάθος σωλήνωσης, παρέχουν τις διευθύνσεις των καταχωρητών (τελεστέων) πριν την αποθήκευση της κωδικής εντολής (opcode) σε έναν καταχωρητή προς αποκωδικοποίηση. Έτσι μαζί με την αποκωδικοποίηση της καταχωρημένης εντολής επιτρέπεται και η παράλληλη πρόσβαση στο αρχείο καταχωρητών. Στην περίπτωση όμως του Leon-2 οι διευθύνσεις αυτές δεν μπορούν να είναι διαθέσιμες στον προηγούμενο κύκλο, με αποτέλεσμα να πρέπει η πρόσβαση στο αρχείο καταχωρητή να ξεκινήσει στην πρώτη διαθέσιμη ακμή ρολογιού κατά το στάδιο της αποκωδικοποίησης, που είναι η κατερχόμενη ακμή του ρολογιού.

Η λογική υλοποίησης της παράκαμψης του σταδίου DECODE αποφασίζει αν στους καταχωρητές εισόδου της Αριθμητικής και Λογικής Μονάδας (ALU) θα καταχωρηθούν δεδομένα από το αρχείο καταχωρητών ή από εσωτερικά δεδομένα της σωλήνωσης. Κατά το στάδιο EXEC πραγματοποιείται η λειτουργία της ALU και υπολογίζεται μια εικονική διεύθυνση. Η πρόσβαση στην κρυφή μνήμη δεδομένων πραγματοποιείται κατά το στάδιο DMEM/EXEC2 και τα βαθμωτά αποτελέσματα επιστρέφουν στη σωλήνωση του RISC στον ίδιο κύκλο. Τέλος, τα αποτελέσματα αποθηκεύονται σε έναν ενδιάμεσο καταχωρητή πριν εισέλθουν στο αρχείο καταχωρητή του επεξεργαστή. Ο επεξεργαστής περιλαμβάνει μια διαμορφώσιμη κρυφή μνήμη δεδομένων με διεγγραφή (write-through) και δυνατότητα υποκλοπής (snooping) που διευκολύνει τη δημιουργία ενός μικρής κλίμακας συστήματος SoC πολυεπεξεργαστών.

2.3.3 Διανυσματικός συνεπεξεργαστής

Ο συνεπεξεργαστής εικονορρών συνδέεται με την Κεντρική Μονάδα Επεξεργασίας μέσω της διεπαφής που περιγράφηκε πιο πάνω και αποτελείται από το παραμετρικό, διανυσματικό, μονοπάτι δεδομένων, από τη μνήμη τύπου σωλήνα και το τμήμα ελέγχου.

Το διανυσματικό μονοπάτι δεδομένων αποτελείται από το αρχείο καταχωρητή διανυσμάτων, τη λογική υλοποίησης παράκαμψης (bypass logic), την παράλληλη διάταξη στοιχείων VLMAX και τη λογική μείωσης/συσσώρευσης. Το μονοπάτι δεδομένων είναι σωληνομένο σε τρία στάδια. Το αρχείο καταχωρητή διανυσμάτων είναι από τα πιο κρίσιμα και πιο ακριβά σε υλισμικό τμήματα της μικροαρχιτεκτονικής. Εφοδιάζει το μονοπάτι δεδομένων με τελεστέους και παρέχει ένα μέγιστο εύρος από πέντε διαβάσματα και ένα γράψιμο διανυσμάτων ανά κύκλο για μέγιστη απόδοση.

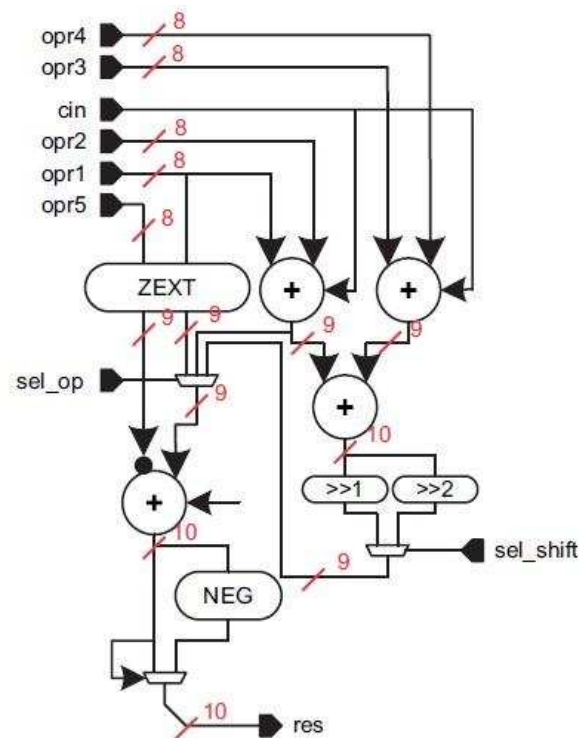
Το βαθμωτό μονοπάτι δεδομένων της μικροαρχιτεκτονικής φαίνεται στην εικόνα 2.8. Υπάρχει ένας παραμετρικός αριθμός (VLMAX) τέτοιων μονοπατιών που έχουν υλοποιηθεί στο στάδιο *exec* και που μπορούν να πραγματοποιήσουν τις πράξεις SAD, AVG2SAD και AVG4SAD.

Η λογική μείωσης απασχολεί το στάδιο *exec2*, που είναι το δεύτερο στάδιο εκτέλεσης, κατά το οποίο το διανυσματικό αποτέλεσμα που έχει παραχθεί από τον προηγούμενο κύκλο προστίθεται/αφαιρείται σε μια τιμή που γράφεται στον καταχωρητή βαθμωτού συσσωρευτή στο τέλος του κύκλου. Η υλοποίηση του τμήματος αυτού έχει τη μορφή ενός δέντρου αθροιστών μαζί με έναν συσσωρευτή.

Το τελευταίο μεγάλο κομμάτι του συνεπεξεργαστή είναι η μνήμη σωλήνωσης, που είναι υπεύθυνη για να εφοδιάζει με δεδομένα το αρχείο καταχωρητή διανυσμάτων μέσω των πράξεων φόρτωσης/αποθήκευσης διανύσματος. Πρόκειται για ένα κρίσιμο τμήμα του συστήματος της μικροαρχιτεκτονικής αφού υλοποιεί το μονοπάτι δεδομένων από και προς τις σύγχρονες μνήμες SDRAM που βρίσκονται εκτός του πλακιδίου (chip). Αποτελείται από τη μονάδα VLSU (Vector Load Store Unit), τον ελεγκτή μνήμης και τη μονάδα διεπαφής αρτηρίας του πλακιδίου (on-chip-bus). Περιλαμβάνει έναν παραμετρικό αριθμό από βαθμωτούς καταχωρητές διευθύνσεων/δεδομένων, τη λογική για την προσαύξηση των διευθύνσεων, την κρυφή μνήμη των διανυσματικών δεδομένων, τις ενδιάμεσες μνήμες (buffers) γραψίματος διανυσμάτων και τον ελεγκτή αρτηρίας του πλακιδίου (on-chip-bus ή AHB).

2.3.4 Μηχανισμός Φράγματος (Barrier)

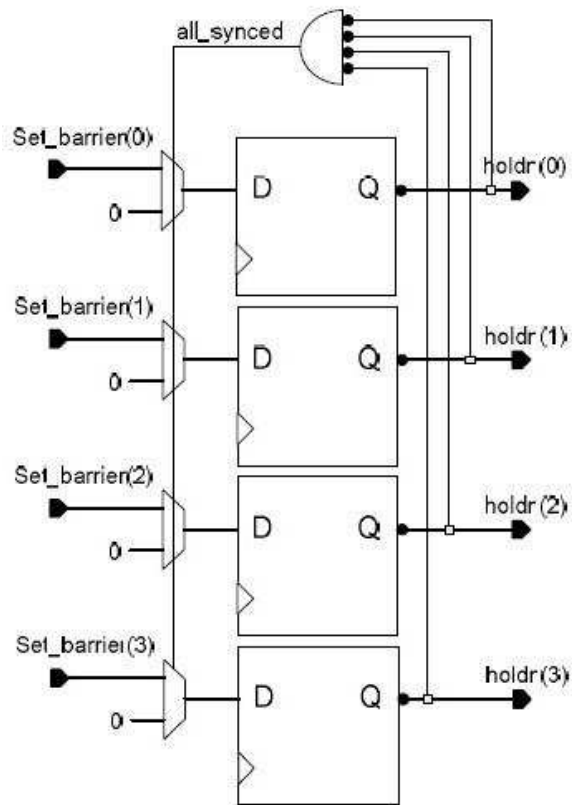
Για την υλοποίηση πολλαπλών τέτοιων πλαισίων KME σε μια διάταξη κοινής μνήμης, δημιουργήσαμε σε υλισμικό έναν πρωτότυπο μηχανισμό *Φράγματος* (Barrier mechanism) ο οποίος αναπαρίσταται στην εικόνα 2.9. Κάθε CPU που εκτελεί την εντολή φράγματος ενεργοποιεί ένα ξεχωριστό flip-flop στη λογική ελέγχου του φράγματος. Εάν υπάρχουν ανενεργά flip-flop στο φράγμα, αυτό σημαίνει πως κάποια ή κάποιες από τις KME δεν έχουν συγχρονιστεί, και η συγκεκριμένη KME που εκτέλεσε την εντολή φράγματος καθυστερείται μέσω ενός καθολικού (ανά KME) σήματος holdn(x), το οποίο προκύπτει από το κύκλωμα ελέγχου του μηχανισμού φράγματος. Όταν τελικά όλα τα flip-flop του φράγματος έχουν ενεργοποιηθεί στον κύκλο N, 'καθαρίζονται' αυτόματα στον κύκλο N+1 και όλα τα σήματα καθυστέρησης μηδενίζονται, επιτρέποντας στον πολυεπεξεργαστή να επανέλθει στη συγχρονισμένη εκτέλεση που πραγματοποιείται μετά το φράγμα. Η εικόνα 2.10 δείχνει το χρονικό διάγραμμα λειτουργίας του μηχανισμού φράγματος. Πολλαπλές τέτοιες τροποποιημένες KME τύπου Leon-2, με την λογική ελέγχου φράγματος και ένα σύνολο από περιφερειακά, σε διάταξη κοινής αρτηρίας (bus) φαίνονται στο σχήμα 2.11.



Σχήμα 2.8: Βαθμωτό μονοπάτι δεδομένων.

2.4 Υλοποίηση σε VLSI

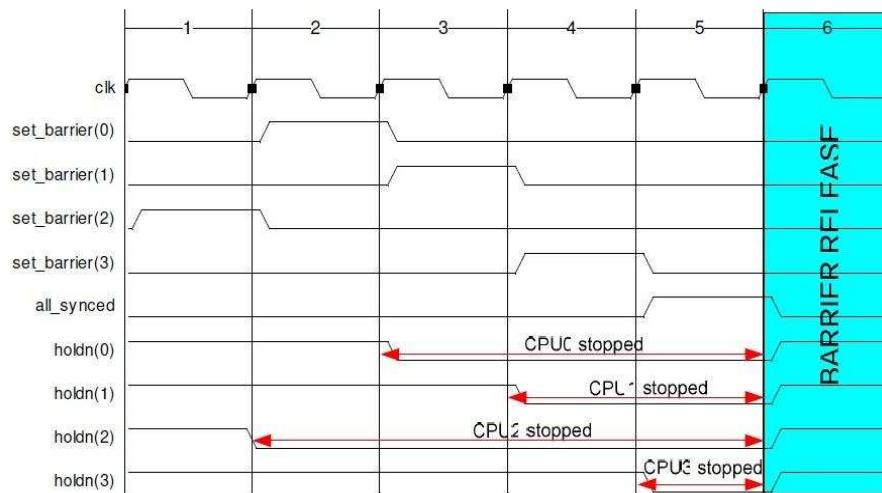
Η αρχιτεκτονική που αναπαρίσταται στο σχήμα 2.11 υλοποιήθηκε σε δύο εκδοχές για $N=2$ και $N=4$ επεξεργαστές. Η υλοποίηση έγινε στη τεχνολογία 0.13 μm CMOS, και το σχέδιο συνθέθηκε αρχικά στο εργαλείο Synopsys Design Compiler και στη συνέχεια οδηγήθηκε στο Cadence SoC Encounter όπου πραγματοποιήθηκε η κατασκευή κάτοψης και η δρομολόγηση του (floorplaning - routing). Στη συνέχεια, οι συστάδες (clusters) που δημιουργήθηκαν οδηγήθηκαν στο Synopsys Physical Compiler για βέλτιστη τοποθέτηση τους και ακολούθησε η τελική, λεπτομερής δρομολόγηση του σχεδίου με τον SoC Encounter. Τα αποτελέσματα της υλοποίησης φαίνονται στην εικόνα 2.12, ενώ ο πίνακας 2.1 παρουσιάζει τα στοιχεία της υλοποίησης.



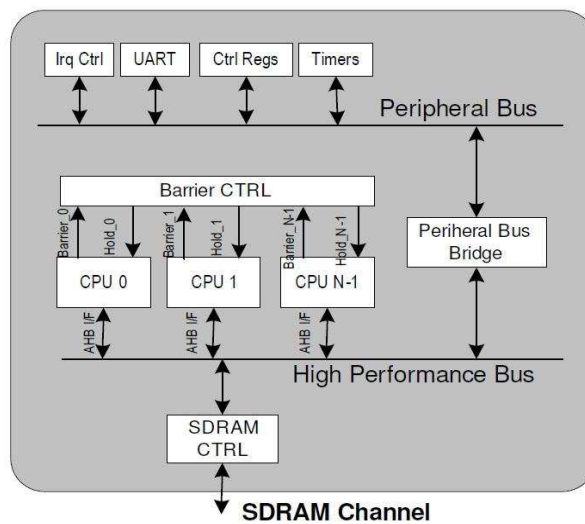
Σχήμα 2.9: Σχηματικό διάγραμμα του μηχανισμού φράγματος.

Πίνακας 2.1: Αποτελέσματα της υλοποίησης σε VLSI του SoC πολυεπεξεργαστή για N=2 και N=4 επεξεργαστές.

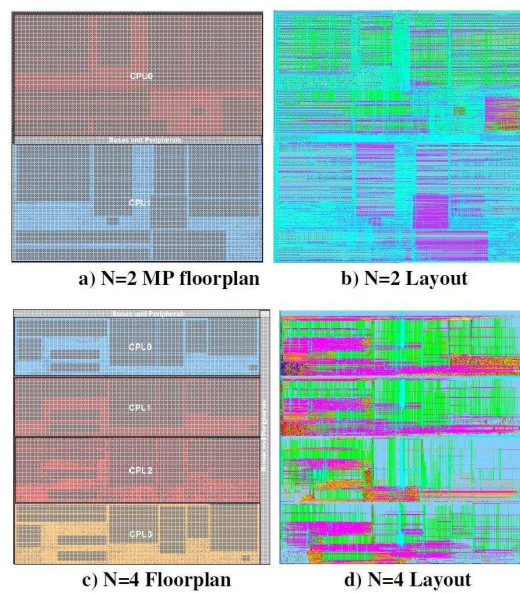
Παράμετροι	1 CPU	2-way config.	4-way config.
Std cells (RAMS)	25068 (26)	110099 (52)	215420 (104)
Fmax (MHz)	219.3	179.5	168.2
Size (μm^2)	5004168	9042033	17270257



Σχήμα 2.10: Χρονικό διάγραμμα λειτουργίας του μηχανισμού φράγματος.



Σχήμα 2.11: Σχηματική αναπαράσταση του πολυεπεξεργαστή SoC.



Σχήμα 2.12: Υλοποίηση σε VLSI του πολυεπεξεργαστή SoC για N=2 και N=4 επεξεργαστές.

Κεφάλαιο 3

Πλήρως συστολικές αρχιτεκτονικές για εφαρμογές Ταχύ Μετασχηματισμού Fourier

3.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται μια τεχνική για την σχεδίαση αρχιτεκτονικών που εκτελούν τον αλγόριθμο του Ταχύ Μετασχηματισμού Fourier (FFT) για μεγάλο αριθμό μιγαδικών σημείων, στοχεύοντας εφαρμογές πραγματικού χρόνου. Η εκτέλεση του Ταχύ Μετασχηματισμού Fourier σε πραγματικό χρόνο απαιτείται σε μια πληθώρα από εφαρμογές στις τηλεπικοινωνίες και στην επεξεργασία σήματος [19]. Οι εφαρμογές αυτές απαιτούν επιπλέον αποτελεσματική κατανάλωση σε χώρο και χαμηλή κατανάλωση ενέργειας. Τέτοιες αυστηρές απαιτήσεις μπορούν να ικανοποιηθούν είτε με χρήση ενσωματωμένων επεξεργαστών υψηλής συχνότητας [20], είτε με τη χρήση ολοκληρωμένων ειδικών εφαρμογών (ASIC - Application Specific Integrated Circuits).

Στην προσπάθεια τους να αντιμετωπίσουν αυτές τις απαιτήσεις, ερευνητές και μηχανικοί ανέπτυξαν διάφορες τεχνικές και αρχιτεκτονικές για FFT [21]-[25], [26]-[34] που ποικίλουν σε σχέση με το επίπεδο παραλληλισμού, τη ρυθμαπόδοση, την καθυστέρηση, το κόστος σε υλισμικό (hardware) και την κατανάλωση ενέργειας. Ανάμεσα στις πιο συνηθισμένες ASIC αρχιτεκτονικές είναι η πλήρως ανοιγμένη (unfolded) υλοποίηση FFT [35] που επιτυγχάνει τη μέγιστη της ρυθμαπόδοση σε χαμηλές συχνότητες ρολογιού, συγκρινόμενη με την οργάνωση μονού επεξεργαστή. Οι ανοιγμένες υλοποιήσεις καταλαμβάνουν περισσότερο χώρο σε VLSI και χρησιμοποιούν μεγαλύτερους πίνακες μνημών ανάμεσα στα διαδοχικά τους στάδια, σε σχέση με την καθυστέρηση και την αποτελεσματική χρήση μνημών που επιτυγχάνουν οι τοπολογίες cascade FFT (αρχιτεκτονικές με αριθμό επεξεργαστών ίσο με τον αριθμό σταδίων του FFT ,

κατά τρόπο που το κάθε στάδιο απεικονίζεται σε έναν επεξεργαστή) [21], [23], [38].

Ο σχεδιασμός των cascade αρχιτεκτονικών, στοχεύοντας υψηλές αποδόσεις, γίνεται πολύπλοκος όταν αυξάνεται το βάθος σωλήνωσης της πεταλούδας. Σε αυτήν την περίπτωση οι αρχιτεκτονικές αυτές πρέπει να χρησιμοποιήσουν μεγαλύτερες μνήμες μέσα σε κάθε πεταλούδα, ενώ ο έλεγχος (control) του κυκλώματος γίνεται πολύπλοκος και θα πρέπει να σχεδιάζεται ειδικά για κάθε διαφορετικό βάθος σωλήνωσης. Τεχνικές που χρησιμοποιούν υψηλότερες βάσεις (radix) ελαττώνουν τον αριθμό των σταδίων του Ταχύ Μετασχηματισμού Fourier, εις βάρος του κόστους σε περιοχή VLSI του κάθε σταδίου. Μια ενδιαφέρουσα εναλλακτική προσέγγιση είναι η χρησιμοποίηση ασύγχρονων αρχιτεκτονικών FFT [36].

Στο κεφάλαιο αυτό περιγράφεται μια τεχνική για το σχεδιασμό αποτελεσματικών αρχιτεκτονικών Ταχύ Μετασχηματισμού Fourier, υψηλής ταχύτητας και σωλήνωσης, που μεγιστοποιούν τη ρυθμαπόδοση, και διατηρούν τον έλεγχο της αρχιτεκτονικής και την οργάνωση των μνημών σε απλά επίπεδα, σε σχέση με τις cascade και με τις πλήρως ανοιγμένες αρχιτεκτονικές FFT. Επιπλέον, αποδεικνύεται ότι τα σχέδια αυτά είναι πιο αποτελεσματικά σε σχέση με τις προαναφερθείσες αρχιτεκτονικές, όσον αφορά την ικανότητα αποδοχής διεύρυνσης (scalability), τη μέγιστη συχνότητα λειτουργίας και εν συνεχεία την κατανάλωση ενέργειας, το βάθος σωλήνωσης και το εύρος σε δυαδία (bit) των δεδομένων και/ή των παραγόντων στροφής (twiddles).

Η τεχνική αυτή βελτιώνει την καθυστέρηση και τις απαιτήσεις σε μνήμη, ειδικά για μεγάλα σύνολα δεδομένων εισόδου, των συστολικών FFT αρχιτεκτονικών συνδυάζοντας τρία (3) κυκλώματα βάσης-4 (radix-4) ώστε να δημιουργηθεί μια μηχανή FFT 64-σημείων. Η αποτελεσματικότητα της οργάνωσης μηχανών FFT 64-σημείων, βασισμένες σε μηχανές βάσης-4, αποδεικνύεται με την υλοποίηση ενός σχεδίου 4096 μιγαδικών σημείων. Η αρχιτεκτονική αυτή χρειάζεται δύο μνήμες διπλής τράπεζας των 4096 λέξεων η κάθε μία, έχοντας σα συχνότητα λειτουργίας τα 200 MHz, με υλοποίηση στο Xilinx Virtex II FPGA, και επιτυγχάνοντας ρυθμαπόδοση 4096 σημείων/20.48 us. Η υλοποίηση του σχεδίου με την υψηλής απόδοσης 0.13 μm , 1P8M CMOS τυπική διαδικασία κελιού της UCM, επιτυγχάνει στην χειρότερη περίπτωση (0.9V, 125C) συχνότητα λειτουργίας τα 604.5 MHz, καταναλώνοντας 4.4 Watt. Θα πρέπει να τονιστεί το γεγονός ότι το σχέδιο ξεπερνά σα συχνότητα λειτουργίας το 1 GHz (ρυθμός 1 Gsample/sec) για τυπικές συνθήκες (1.0V, 25C).

Για το σχεδιασμό αρχιτεκτονικών Ταχύ Μετασχηματισμού Fourier, για μεγάλα σύνολα δεδομένων εισόδου, θεωρούμε την αρχιτεκτονική των 4096 μιγαδικών σημείων FFT σαν έναν πυρήνα. Ο πυρήνας αυτός αποτελεί τη βάση των FFT αρχιτεκτονικών για τον υπολογισμό μετασχηματισμών 16K, 64K και 256K μιγαδικών σημείων. Οι αρχιτεκτονικές αυτές υλοποιήθηκαν στη διαδικασία 0.13 CMOS, έχοντας σα χειρότερη συχνότητα λειτουργίας (0.9V, 125C) τα 352, 256 και 188 MHz αντίστοιχα. Οι αρχιτεκτονικές 16K και 64K έχουν παράλληλες εισόδους/εξόδους τεσσάρων σημείων επιτυγχάνοντας ρυθμαπόδοση 1.4 και 1

Gsample/sec αντίστοιχα.

Το υπόλοιπο κεφάλαιο έχει οργανωθεί σε 2 τεχνικά μέρη. Το πρώτο από αυτά αναλύει την οργάνωση των υπολογισμών βάσης-4 ώστε να προκύψει η βάση-4³ (*radix* - 4³), ισοδύναμη του υπολογισμού βάσης-64 (*radix* - 64), και περιγράφει την πλήρως συστολική αρχιτεκτονική FFT 4096 σημείων. Στο δεύτερο τεχνικό μέρος αναλύεται το πως αυτή η αρχιτεκτονική μπορεί να επεκταθεί ώστε να προκύψουν οι αρχιτεκτονικές 16K, 64K και 256K FFT και περιγράφεται η υλοποίηση τους.

3.2 Η πλήρως συστολική αρχιτεκτονική Ταχύ Μετασχηματισμού Fourier 4096 σημείων

Εφαρμογές που αφορούν μεγάλης κλίμακας υπολογισμούς FFT σε μεγάλες συχνότητες, απαιτούν βελτιστοποίηση των διαφόρων χαρακτηριστικών της αρχιτεκτονικής όπως μέγεθος μνημών, έλεγχος του κυκλώματος, χρήση των πόρων και την αποτελεσματικότητα της αρχιτεκτονικής, που καθορίζεται σαν την απόδοση του σχεδίου σε σχέση με το χώρο που καταλαμβάνει σε VLSI. Η χρήση της μνήμης βελτιστοποιείται σε σχέδια τύπου cascade [21], [38], [39] ή unfolded [21] με στάδια FFT υψηλής βάσης, όπως βάση 16, 32, 64 ή 128. Ενώ οι απαιτήσεις σε μνήμη και καθυστέρηση ευνοούν τις cascade αρχιτεκτονικές, οι unfolded αρχιτεκτονικές συνδυάζουν τον απλούστερο έλεγχο, βάθος σωλήνωσης ανεξάρτητο του κυκλώματος ελέγχου και επομένως μπορούν έτσι να επεκταθούν, αποτελώντας πιο αποτελεσματικά σχέδια για παραλλαγές που σχετίζονται με τις απαιτήσεις σε απόδοση.

Μια ελκυστική λύση που συνδυάζει τα πλεονεκτήματα και των δύο δομών είναι η οργάνωση τύπου unfolded που χρησιμοποιεί στάδια βάσης-64 (*R64*), καθώς το 64 αποτελεί δύναμη είτε του 2 είτε του 4 και είναι επαρκώς μεγάλος αριθμός για να καταλήξει σε μικρό αριθμό σταδίων. Υλοποιώντας τον Ταχύ Μετασχηματισμού Fourier 64 σημείων σε μεγάλη ταχύτητα, με βάσεις μεγαλύτερες του 2 ή του 4 είναι αναποτελεσματικό, καθώς οι υψηλές βάσεις οδηγούν σε πολύπλοκες δομές, μειώνοντας την απόδοση της αρχιτεκτονικής. Οι υπολογισμοί βάσης-4 (*R4*) είναι το ίδιο απλοί στην υλοποίηση τους με αυτούς της βάσης-2 (*R2*), η χρήση τους όμως οδηγεί στην ελάττωση κατά το ήμισυ του αριθμού των μιγαδικών πολλαπλασιαστών, σε σχέση με αυτούς που χρησιμοποιούν τα σχέδια βασισμένα στον *R2*. Επίσης, ο *R4* επεξεργάζεται 4 σημεία, σε έναν υπολογισμό δύο βημάτων συσσώρευσης-πολλαπλασιασμού, με τον πολλαπλασιασμό να απλοποιείται σε πράξεις αντιστροφής προσήμου και ανταλλαγής λέξεων (swap). Το αποτέλεσμα είναι να ελαχιστοποιείται η ύπαρξη πηθανών στενωμάτων (bottlenecks) στη ροή δεδομένων, ενώ ταυτόχρονα επιτρέπεται η μέγιστη δυνατή σωλήνωση ή/και ο παραλληλισμός των τεσσάρων υπολογιστικών βημάτων. Επομένως, η επιλογή του *R4* οδηγεί στη δημιουργία ενός πιο αποτελεσματικού σχεδίου όσον αφορά τη χρήση πολλαπλασιαστών, το βάθος σωλήνωσης και τη δυνατότητα διεύρυνσης της συχνότητας λειτουργίας ή/και

του μήκους λέξεων.

3.2.1 Ο αλγόριθμος βάσης-4³

Για να δημιουργηθεί ο αλγόριθμος βάσης - 4³ (Radix-4³ ή R4³) ξεκινάμε με τον Διακριτό Μετασχηματισμό Fourier (DFT) ενός σήματος $x[n]$ μήκους N ,

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad (3.1)$$

όπου $W_N = e^{-j\frac{2\pi}{N}}$ είναι οι παράγοντες στροφής (twiddle factors) και δηλώνουν τη N -οστή ρίζα της μονάδας [37].

Η αρχιτεκτονική που παρουσιάζεται στη συνέχεια βασίζεται σε μια τετραδιάστατη απεικόνιση δεικτών και στην ανάλυση με βάση-4 των σειρών DFT. Η παραγωγή του αλγόριθμου Radix-4³ πραγματοποιείται σε 3 βήματα διαδοχικής ανάλυσης, όπως παρουσιάζονται στο [41]. Στο πλαίσιο των βημάτων αυτών η γραμμική απεικόνιση μετασχηματίζεται σε μια τετραδιάστατη απεικόνιση δεικτών [23, 38] ως εξής:

$$\begin{aligned} n &= n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 + \frac{N}{4}n_4 \\ k &= 64k_1 + 16k_2 + 4k_3 + k_4 \end{aligned} \quad (3.2)$$

Εφαρμόζοντας την εξίσωση 3.2 στην εξίσωση του Διακριτού Μετασχηματισμού Fourier παίρνουμε:

$$\begin{aligned} X(64k_1 + 16k_2 + 4k_3 + k_4) = \\ \sum_{n_1=0}^{\frac{N}{64}-1} \sum_{n_2=0}^3 \sum_{n_3=0}^3 \sum_{n_4=0}^3 x \left(n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 + \frac{N}{4}n_4 \right) W_N^{nk} \end{aligned} \quad (3.3)$$

Οι παράγοντες στροφής μπορούν να εκφραστούν ως εξής:

$$\begin{aligned} W_N^{kn} &= W_N^{[n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 + \frac{N}{4}n_4][64k_1 + 16k_2 + 4k_3 + k_4]} \Rightarrow \\ W_N^{kn} &= (-j)^{n_2k_2 + n_3k_3 + n_4k_4} W_{16}^{n_3k_4} W_{64}^{n_2(4k_3 + k_4)} \times \\ &\quad \times W_N^{n_1(16k_2 + 4k_3 + k_4)} W_{\frac{N}{64}}^{n_1k_1} \end{aligned} \quad (3.4)$$

Εφαρμόζοντας την εξίσωση 3.4 στην εξίσωση 3.3 και αναλύοντας το άθροισμα με δείκτη n_4 παίρνουμε:

$$\begin{aligned} X(64k_1 + 16k_2 + 4k_3 + k_4) = \\ \sum_{n_1=0}^{\frac{N}{64}-1} \sum_{n_2=0}^3 \sum_{n_3=0}^3 \left[B_{\frac{N}{4}}^{k_4} \left(n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 \right) \right] \times \\ \times (-j)^{n_2k_2 + n_3k_3} W_{16}^{n_3k_4} W_{64}^{n_2(4k_3 + k_4)} \times W_N^{n_1(16k_2 + 4k_3 + k_4)} W_{\frac{N}{64}}^{n_1k_1} \end{aligned} \quad (3.5)$$

όπου ο όρος $B_{\frac{N}{4}}^{k_4} \left(n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 \right)$ δηλώνει την πρώτη μονάδα πεταλούδας (butterfly unit) και μπορεί να γραφτεί ως εξής:

$$\begin{aligned} B_{\frac{N}{4}}^{k_4} \left(n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 \right) &= x \left(n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 \right) \\ &+ (-j)^{k_4} x \left(n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 + \frac{N}{4} \right) + (-1)^{k_4} x \left(n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 + \frac{N}{2} \right) + \\ &+ j^{k_4} x \left(n_1 + \frac{N}{64}n_2 + \frac{N}{16}n_3 + \frac{3N}{4} \right) \end{aligned} \quad (3.6)$$

Αναλύοντας την εξίσωση 3.5 ως προς το επόμενο άθροισμα με δείκτη n_3 παίρνουμε:

$$\begin{aligned} X(64k_1 + 16k_2 + 4k_3 + k_4) &= \\ \sum_{n_1=0}^{\frac{N}{64}-1} \sum_{n_2=0}^3 H_{\frac{N}{16}}^{k_3 k_4} \left(n_1 + \frac{N}{64}n_2 \right) &(-j)^{n_2 k_2} \times \\ \times W_{64}^{n_2(4k_3+k_4)} W_N^{n_1(16k_2+4k_3+k_4)} &W_{\frac{N}{64}}^{n_1 k_1} \end{aligned} \quad (3.7)$$

όπου $H_{\frac{N}{16}}^{k_3 k_4} \left(n_1 + \frac{N}{64}n_2 \right)$ είναι η δεύτερη μονάδα πεταλούδας και η οποία μπορεί να εκφραστεί ως εξής:

$$\begin{aligned} H_{\frac{N}{16}}^{k_3 k_4} \left(n_1 + \frac{N}{64}n_2 \right) &= B_{\frac{N}{4}}^{k_4} \left(n_1 + \frac{N}{64}n_2 \right) + \\ &+ (-j)^{k_3} W_{16}^{k_4} B_{\frac{N}{4}}^{k_4} \left(n_1 + \frac{N}{64}n_2 + \frac{N}{16} \right) + \\ &+ (-1)^{k_3} W_8^{k_4} B_{\frac{N}{4}}^{k_4} \left(n_1 + \frac{N}{64}n_2 + \frac{N}{8} \right) + \\ &+ j^{k_3} W_8^{k_4} W_{16}^{k_4} B_{\frac{N}{4}}^{k_4} \left(n_1 + \frac{N}{64}n_2 + \frac{3N}{16} \right) \end{aligned} \quad (3.8)$$

Τέλος, αναλύοντας το άθροισμα της εξίσωσης 3.7 ως προς το δείκτη n_2 βρίσκουμε ένα σετ των 64 DFTs μήκους $N/64$.

$$\begin{aligned} X(64k_1 + 16k_2 + 4k_3 + k_4) &= \\ \sum_{n_1=0}^{\frac{N}{64}-1} \left[T_{\frac{N}{64}}^{k_2 k_3 k_4} (n_1) W_N^{n_1(6k_2+4k_3+k_4)} \right] &W_{\frac{N}{64}}^{n_1 k_1} \end{aligned} \quad (3.9)$$

όπου $T_{\frac{N}{64}}^{k_2 k_3 k_4} (n_1)$ είναι η τρίτη μονάδα πεταλούδας, που μπορεί να εκφραστεί σύμφωνα με την εξίσωση 3.10

$$\begin{aligned}
T_{\frac{N}{64}}^{k_2 k_3 k_4}(n_1) &= H_{\frac{N}{16}}^{k_3 k_4}(n_1) + \\
&+ (-j)^{k_2} W_{16}^{k_3} W_{64}^{(k_4)} H_{\frac{N}{16}}^{k_3 k_4} \left(n_1 + \frac{N}{64} \right) + \\
&+ (-1)^{k_2} W_8^{k_3} W_{32}^{k_4} H_{\frac{N}{16}}^{k_3 k_4} \left(n_1 + \frac{N}{32} \right) + \\
&+ j^{k_2} W_{64}^{3(4k_3+k_4)} H_{\frac{N}{16}}^{k_3 k_4} \left(n_1 + \frac{3N}{64} \right)
\end{aligned} \tag{3.10}$$

Οι εξισώσεις 3.5 ως 3.10 περιγράφουν έναν Ταχύ Μετασχηματισμό Fourier βάσης-64 (radix-64). Επίσης, οι εξισώσεις 3.6, 3.8 και 3.10 περιγράφουν την εσωτερική δομή της πεταλούδας βάσης-64, η οποία βασίζεται σε 3 πεταλούδες βάσης-4 και γι' αυτόν το λόγο ονομάζεται βάση-4³ (Radix-4³ ή R4³).

3.2.2 Χαρακτηριστικά του $R4^3$

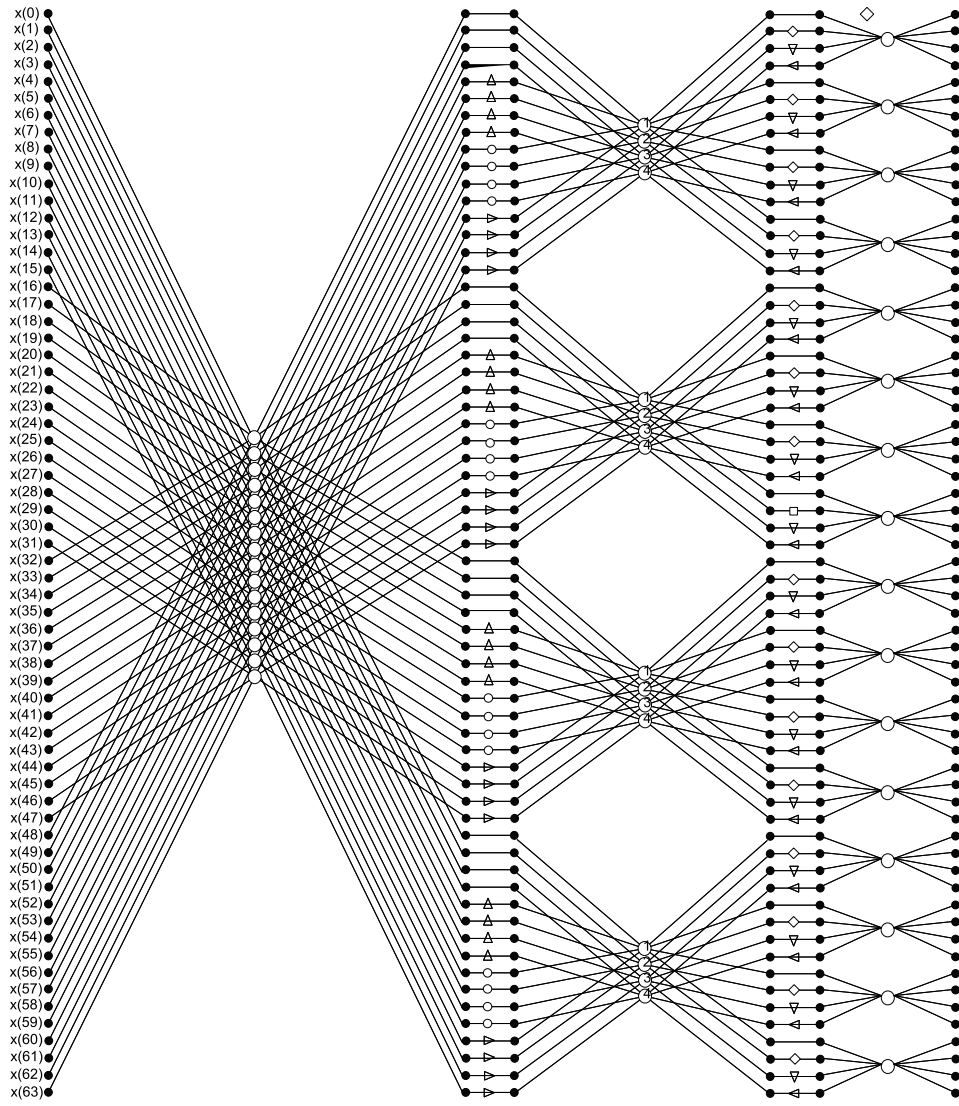
Στην εικόνα 3.1 εμφανίζεται το διάγραμμα ροής του $R4^3$ όπου τα σχήματα Δ , $\circ$ και $\triangleright$ αναπαριστούν πολλαπλασιασμούς με τους όρους $(-j)^{k_3} W_{16}^{k_3}$, $(-1)^{k_3} W_8^{k_4}$ και $j^{k_3} W_8^{k_4} W_{16}^{k_4}$ αντίστοιχα, ενώ τα σχήματα $\diamond$, ∇ , $\triangleleft$ αναπαριστούν πολλαπλασιασμούς με τους όρους $(-j)^{k_3} W_{16}^{k_3} W_{64}^{k_4}$, $(-1)^{k_2} W_8^{k_3} W_{32}^{k_4}$ ανδ $j^{k_2} W_{64}^{3(4k_3+k_4)}$. Σε αυτό το σημείο θα πρέπει να τονιστεί ότι οι παράγοντες στροφής στον $R4^3$ έχουν διαφορετική σειρά εμφάνισης από αυτούς της βάσης-64 ($R64$).

Η αρχιτεκτονική Ταχύ Μετασχηματισμού Fourier που υλοποιείται με στάδια $R4^3$ έχει σαν είσοδο N μιγαδικά σημεία και χρησιμοποιεί $\log_4 N - 1$ μιγαδικούς πολλαπλασιαστές. Επειδή ο κάθε $R4$ χρησιμοποιεί 3 αθροιστές μιγαδικών αριθμών για να παράγει 1 δεδομένο/κύκλο, η αρχιτεκτονική χρησιμοποιεί συνολικά $3\log_4 N$ μιγαδικούς αθροιστές.

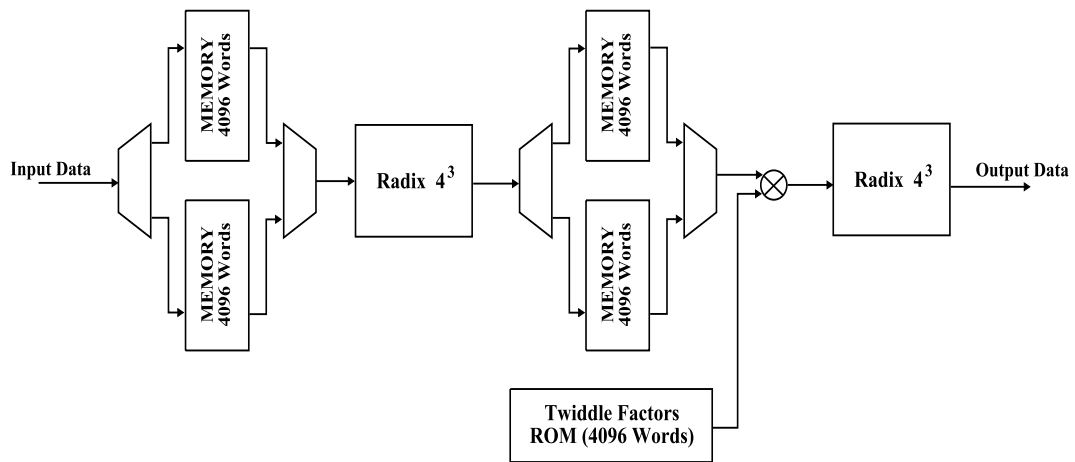
Συγκρίνοντας την πλήρως ανοιγμένη αρχιτεκτονική $R4^3$ με τις αρχιτεκτονικές διαδοχικών σταδίων $R2^2$ [38] και $R2^4$ [39] βρίσκουμε ότι χρησιμοποιούν τον ίδιο αριθμό πολλαπλασιαστών (Πίνακας 3.1). Επίσης, ο $R4^3$ έχει λιγότερους μιγαδικούς αθροιστές ($3\log_4 N$) απ'οτι ο $R2^2$ ($4\log_4 N$), αλλά χρησιμοποιεί περισσότερη μνήμη. Ο $R4^3$ FFT χρησιμοποιεί τον ελάχιστο αριθμό πολλαπλασιαστών και αθροιστών που υπάρχουν στην λογοτεχνία, ίσο με αυτόν του [40], και παρόλο που έχει μεγαλύτερες απαιτήσεις σε μνήμη, χρησιμοποιεί απλό κύκλωμα ελέγχου (control) και μπορεί να επιτύχει υψηλές συχνότητες λειτουργίας, όπως φαίνεται και στην επόμενη ενότητα.

Πίνακας 3.1: Σύγκριση πολυπλοκότητας υλισμικού σε αρχιτεκτονικές FFT.

	Μιγ. Πολλ/στές	Μιγ. Αθροιστές	Μνήμη	Control
R2MDC [38]	$2\log_4 N - 1$	$4\log_4 N$	$3N/2 - 2$	simple
R2SDF [38]	$2\log_4 N - 1$	$4\log_4 N$	$N - 1$	simple
R4SDF [38]	$\log_4 N - 1$	$8\log_4 N$	$N - 1$	medium
R4MDC [38]	$3(\log_4 N - 1)$	$8\log_4 N$	$5N/2 - 4$	simple
R4SDC [38]	$\log_4 N - 1$	$3\log_4 N$	$2N - 2$	complex
R2 ² SDF [39]	$\log_4 N - 1$	$4\log_4 N$	$N - 1$	simple
$R4^3$	$\log_4 N - 1$	$3\log_4 N$	$(N/3)\log_4 N$	simple



Σχήμα 3.1: Διάγραμμα ροής πεταλούδας $Radix - 4^3$.



Σχήμα 3.2: Μπλοκ διάγραμμα της αρχιτεκτονικής 4K FFT.

3.2.3 Αρχιτεκτονική του 4K FFT

Η αρχιτεκτονική του Ταχύ Μετασχηματισμού Fourier 4096 σημείων φαίνεται στην εικόνα 3.2. Η υλοποίηση της αρχιτεκτονικής αυτής, που στηρίζεται στη χρήση πεταλούδων $R4^3$, περιλαμβάνει: τη χρήση μιας δισδιάστατης απεικόνισης δεικτών βασισμένη στην πεταλούδα $R64$, την ανάλυση των 4096 σημείων σε 2 αθροίσματα χρησιμοποιώντας την πεταλούδα $R64$, και τέλος, την αντικατάσταση του κάθε $R64$ με το $R4^3$. Επομένως, η αρχιτεκτονική αποτελείται από 2 μηχανές $R4^3$, 2 δομοστοιχεία μνήμης διπλής τράπεζας 4096 λέξεων (Dual Bank Memory Module), μια μνήμη ROM 4096 λέξεων που έχει αποθηκευμένους τους παράγοντες στροφής W_{4096} , και έναν μιγαδικό πολλαπλασιαστή. Το κάθε δομοστοιχείο χρησιμοποιεί τοπικό κύκλωμα ελέγχου, ενώ ένα καθολικό σήμα *sync* συγχρονίζει τα δομοστοιχεία. Η ρυθμιζόμενη που επιτυγχάνεται είναι 1 μιγαδικό σημείο/κύκλο ρολογιού.

Ο συνολικός σχεδιασμός επιτρέπει τη βελτιστοποίηση της αρχιτεκτονικής είτε καθολικά, είτε τοπικά: Η αρχιτεκτονική μπορεί να βελτιωθεί ως προς τη συχνότητα λειτουργίας, το χώρο που καταλαμβάνει ή και ως προς τα δύο, χωρίς να υπάρξουν αλλαγές στο κύκλωμα ελέγχου, το χρονοπρογραμματισμό των μνημών, στους καταχωρητές και στις μνήμες των μηχανών $R4^3$. Συνεπώς, τοπικές τροποποιήσεις στους σωληνικούς υπολογισμούς ώστε να γίνει εκμετάλλευση των ειδικών χαρακτηριστικών της κάθε τεχνολογίας, οδηγούν σε μεγιστοποίηση της συχνότητας λειτουργίας.

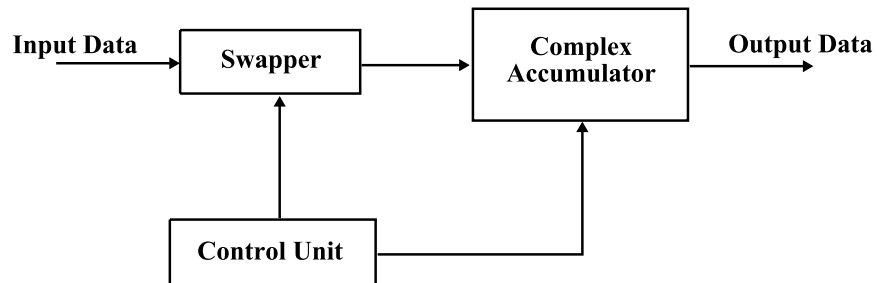
Οι παράγραφοι που ακολουθούν περιγράφουν λεπτομερώς τον επεξεργαστή $R4$, που αποτελεί τη βάση του $R4^3$, την αρχιτεκτονική της μηχανής $R4^3$ καθώς και τη διευθυνσιοδότηση της, ενώ τέλος αναλύουν την απόδοση της συνολικής FFT αρχιτεκτονικής.

3.2.4 Η πεταλούδα Βάσης-4 ως βάση της μηχανής $R4^3$

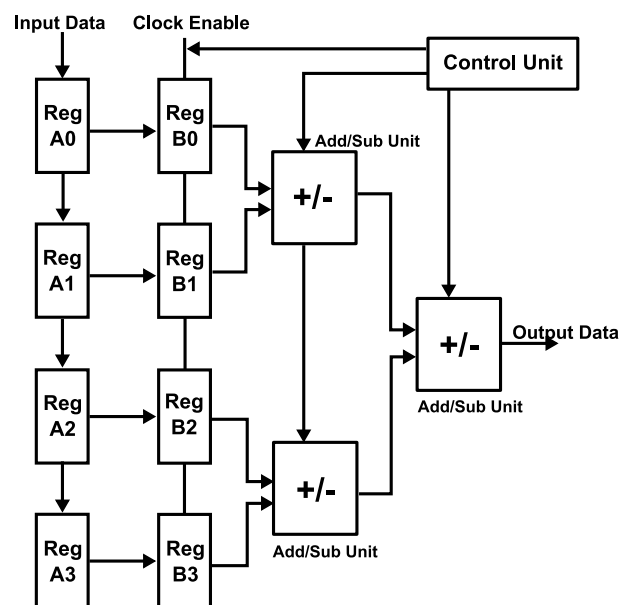
Η εικόνα 3.3 απεικονίζει την αρχιτεκτονική της μηχανής βάσης-4 (Radix-4 engine ή $R4$) που χρησιμοποιείται στην μηχανή $R4^3$. Κάθε μηχανή $R4$ αποτελείται από έναν μιγαδικό συσσωρευτή (Accumulator) και ένα δομοστοιχείο ανταλλαγής (Swap). Ο ρόλος του δομοστοιχείου αυτού είναι να ανταλλάξει το πραγματικό μέρος της κάθε τιμής εισόδου με το αντίστοιχο φανταστικό, και το ανάποδο, ώστε να πραγματοποιηθούν σωστά οι πράξεις πεταλούδας. Στην εικόνα 3.4 φαίνεται η αρχιτεκτονική του δομοστοιχείου συσσωρευτή. Ο συσσωρευτής αποτελείται από 8 καταχωρητές και 3 μονάδες πρόσθεσης/αφαίρεσης. Οι τέσσερις καταχωρητές "A" χρησιμοποιούνται σαν καταχωρητές εισόδου, αποθηκεύοντας τις τετράδες δεδομένων που θα χρησιμοποιηθούν για τις πράξεις της πεταλούδας βάσης-4. Οι τέσσερις καταχωρητές "B" χρησιμοποιούνται ως ένα ενδιάμεσο στάδιο αποθήκευσης της τετράδας (ενώ η επόμενη τετράδα ολισθαίνει στους καταχωρητές "A") και παράλληλης φόρτωσης των δεδομένων στις μονάδες άθροισης/αφαίρεσης. Οι μονάδες άθροισης/αφαίρεσης σχηματίζουν μια αρχιτεκτονική δέντρου αθροιστών (adder-tree) αποφεύγοντας έτσι τη δημιουργία βρόχων ανάδρασης, κάτι που όλοι οι συνήθεις συσσωρευτές χρησιμοποιούν.

Η ροή των δεδομένων της μηχανής $R4$ ξεκινά με τα δεδομένα να εισέρχονται σε αυτήν ως σειριακό ρεύμα εισόδου λέξεων, με ρυθμό μιας μιγαδικής λέξης/κύκλο. Σε κάθε περίοδο 4 κύκλων, τέσσερα διαδοχικά δεδομένα εισόδου ολισθαίνουν στους 4 καταχωρητές εισόδου ($RegA$). Στον πέμπτο κύκλο η τετράδα αυτή μεταφέρεται στους 4 καταχωρητές "αποθήκευσης" ($RegB$), ενώ ταυτόχρονα η επόμενη τετράδα αρχίζει την είσοδο της στους καταχωρητές "A". Κατά τον έκτο κύκλο η τετράδα των καταχωρητών "B" εισέρχεται παράλληλα στο δέντρο αθροιστών για την παραγωγή του πρώτου από τα τέσσερα αποτελέσματα της πεταλούδας $R4$. Στους επόμενους 3 κύκλους το δέντρο αθροιστών χρησιμοποιεί σαν είσοδο την τετράδα δεδομένων που είναι αποθηκευμένη στους καταχωρητές "B" και παράγει τα υπόλοιπα 3, από τα τέσσερα, αποτελέσματα της πεταλούδας $R4$ με ρυθμό 1 αποτέλεσμα/κύκλο, ακολουθώντας την υπολογιστική ροή της πεταλούδας βάσης-4. Επομένως, η συνολική καθυστέρηση του κάθε συσσωρευτή είναι $5 + 2k$ κύκλοι, όπου k είναι η καθυστέρηση της κάθε μονάδας πρόσθεσης/αφαίρεσης, με $k = 5$ στην παρούσα υλοποίηση.

Μια μονάδα ελέγχου συγχρονίζει τις λειτουργίες του δομοστοιχείου *Swap* και του συσσωρευτή. Ένας μετρητής 2-bit χρησιμοποιείται για την παραγωγή των αναγκαίων σημάτων ελέγχου των μονάδων πρόσθεσης/αφαίρεσης, ώστε να πραγματοποιηθούν αθροίσεις και/ή αφαιρέσεις σύμφωνα πάντα με το σχήμα βάσης-4.



Σχήμα 3.3: Η μηχανή Βάσης-4.



Σχήμα 3.4: Το δέντρο αθροιστών του μιγαδικού συσσωρευτή.

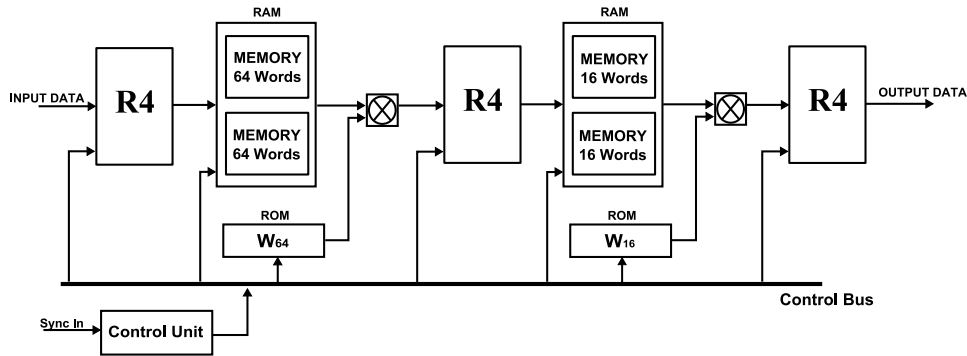
3.2.5 Η αρχιτεκτονική της μηχανής $R4^3$

Η εικόνα 3.5 δείχνει τη μηχανή $R4^3$, που αποτελείται από τρεις (3) πεταλούδες βάσης-4 ($R4$), δύο (2) μιγαδικούς πολλαπλασιαστές, δύο (2) δομοστοιχεία μνήμης διπλής τράπεζας (το ένα αποτελούμενο από 2×64 λέξεις και το άλλο από 2×16 λέξεις), και δύο (2) μνήμες ROM που αποθηκεύουν τους παράγοντες στροφής W_{16} και W_{64} . Οι μιγαδικοί πολλαπλασιαστές αποτελούνται από τέσσερις (4) πολλαπλασιαστές ακεραίων, εισόδου 2×64 bit και σωλήνωσης βάθους πέντε (5) κύκλων, καθώς και από δύο (2) αθροιστές. Το κύκλωμα ελέγχου (control unit) της μηχανής $R4^3$ είναι τοπικό για κάθε στάδιο σωλήνωσης και όλα τα στάδια συγχρονίζονται μεταξύ τους με ένα καθολικό σήμα συγχρονισμού *sync*.

Υλοποιώντας το στάδιο βάσης-64 ($R64$) με τη μηχανή $R4^3$ έχουμε ένα συνδυασμό από πλεονεκτήματα:

1. Την απλότητα των κυκλωμάτων σαν αποτέλεσμα της χρήσης ως βασικής πεταλούδας, μια με μικρή βάση (βάση-4).
2. Την ομαλότητα της αρχιτεκτονικής η οποία χτίζεται ιεραρχικά, με συνεχόμενες μηχανές $R4^3$ σε υψηλό επίπεδο και με κάθε μηχανή να αποτελείται από 3 σωληνομένους υπολογισμούς $R4$ στο βασικό επίπεδο.
3. Επιπλέον, ο σχεδιασμός της μηχανής $R4^3$ είναι αποτελεσματικός και σε σχέση με τη χρησιμοποίηση του καταλαμβανόμενου χώρου VLSI καθώς οι μιγαδικοί πολλαπλασιαστές χρησιμοποιούνται στο 75% των κύκλων ρολογιού, ενώ οι υπόλοιποι πόροι στο 100%.

Η αρχιτεκτονική FFT επιτυγχάνει το δυναμικό εύρος των 84db (πλάτος) χρησιμοποιώντας 14 bit για κάθε ένα από το πραγματικό και φανταστικό μέρος του κάθε δείγματος. Για να διατηρηθεί η μέγιστη ακρίβεια 14 (bit) για κάθε μέρος, ο σχεδιασμός του $R4^3$ χρησιμοποιεί αυξημένη ακρίβεια στους εσωτερικούς υπολογισμούς. Το εύρος του κάθε μέρους είναι 16 και 20 bit στην έξοδο του πρώτου και δεύτερου σταδίου $R4$ αντίστοιχα. Στην έξοδο του τρίτου και τελευταίου σταδίου του $R4$ γίνεται αποκοπή στο πραγματικό και φανταστικό μέρος, ώστε να δημιουργηθεί έξοδος με 14 bit για το κάθε μέρος. Οι παράγοντες στροφής έχουν 18 bit για το πραγματικό και 18 bit για το φανταστικό μέρος. Εφαρμόζοντας την προαναφερθείσα μορφοποίηση στα δεδομένα, η υλοποίηση του FFT 4096 σημείων με υπολογισμούς σταθερής υποδιαστολής είναι σχεδόν το ίδιο ακριβείς με μια υλοποίηση κινητής υποδιαστολής, δεδομένου του δυναμικού εύρους των 84db. Η έξοδος του κυκλώματος FFT σταθερής υποδιαστολής βρίσκεται μεταξύ του ορίου ± 1 , συγκρινόμενη με την έξοδο ενός κυκλώματος FFT κινητής υποδιαστολής του οποίου οι τιμές έχουν κανονικοποιηθεί (διαιρεθεί με $N = 4096$) και αποκοπεί στα 14 bit.



Σχήμα 3.5: Μπλοκ διάγραμμα της αρχιτεκτονικής πεταλούδας $Radix - 4^3$.

3.2.6 Χρονοπρογραμματισμός δεδομένων και διευθυνσιοδότηση στον $R4^3$

Το πρώτο στάδιο του αλγορίθμου του 4K FFT πραγματοποιείται από τον πρώτο $R4^3$ επεξεργαστή, ο οποίος υπολογίζει $\frac{4096}{64}$ FFTs, με τον κάθε μετασχηματισμό να αποτελείται από 64 σημεία. Ο μετασχηματισμός κάθε 64-αδας περιέχει στοιχεία των οποίων οι διευθύνσεις είναι της μορφής $64k+i$, όπου $i = 0, \dots, 63$ και k είναι ο δείκτης πλειάδας που παίρνει τιμές από 0 μέχρι $\frac{4096}{64} - 1$.

Στον πρώτο επεξεργαστή $R4^3$ τα δεδομένα της κάθε 64-αδας επεξεργάζονται σε τετράδες, έτσι ώστε ο πρώτος επεξεργαστής $R4$ να υπολογίζει το Διακριτό Μετασχηματισμό Fourier (DFT) των στοιχείων με διεύθυνση $16i_1 + k_1$, $i_1 = 0, \dots, 3$ και δείκτη πλειάδας $k_1 = 0, \dots, \frac{64}{4} - 1$. Μπορούμε να χρησιμοποιήσουμε το συμβολισμό $[a_{11}, \dots, a_0]$ για τη διεύθυνση εισόδου του κάθε σημείου (τη θέση του δηλαδή ανάμεσα στα 4096 σημεία) και το συμβολισμό $[d_5d_4d_3d_2d_1d_0]$ για τη διεύθυνση της κάθε τράπεζας της πρώτης ενδιάμεσης μνήμης. Τα 64 στοιχεία που προκύπτουν γράφονται σε μια από τις τράπεζες της πρώτης ενδιάμεσης μνήμης σε συνεχόμενες διευθύνσεις π.χ. $[a_{11}a_{10}a_9a_8a_7a_6] \rightarrow [d_5d_4d_3d_2d_1d_0]$.

Ο δεύτερος επεξεργαστής $R4$ διαβάζει τα δεδομένα από την πρώτη ενδιάμεση τράπεζα μνήμης μεγέθους 64 χρησιμοποιώντας τη μετάθεση διευθύνσεων $[a_{11}a_{10}a_9a_8a_7a_6] \rightarrow [d_5d_4d_3d_2d_1d_0]$. Κάθε 4-αδα διαδοχικών αποτελεσμάτων των 4 σημείων DFTs που παράγονται από το δεύτερο $R4$ αποθηκεύονται σε μια από τις δύο τράπεζες της δεύτερης ενδιάμεσης μνήμης μεγέθους 16 λέξεων. Όταν ο τρίτος επεξεργαστής $R4$ ξεκινάει το διάβασμα των δεδομένων αυτών, ο δεύτερος $R4$ αποθηκεύει τα επόμενα 4 αποτελέσματα του DFT στην άλλη τράπεζα της δεύτερης ενδιάμεσης μνήμης.

Εδώ θα πρέπει να σημειωθεί ότι ο δεύτερος επεξεργαστής $R4$ παράγει τα αποτελέσματα έτσι ώστε ο τρίτος $R4$ να μπορεί να χρησιμοποιεί με πλήρη σωλήνωση όλα τα δεδομένα που είναι αποθηκευμένα στην ενδιάμεση μνήμη ώστε να ολοκληρώσει τον υπολογισμό DFT 4 σημείων, κάνοντας 100% αξιοποίηση

των πόρων. Με $[d_3d_2d_1d_0]$ να δηλώνει τη διεύθυνση μέσα σε κάθε ενδιάμεση μνήμη, ο τρίτος επεξεργαστής $R4$ διαβάζει τα δεδομένα χρησιμοποιώντας τη μετάθεση διευθύνσεων $[a_1a_0a_3a_2] \rightarrow [d_3d_2d_1d_0]$.

Κατά το δεύτερο στάδιο του αλγορίθμου 4K FFT, ο δεύτερος επεξεργαστής $R4^3$ υπολογίζει $\frac{4096}{64}$ FFTs με κάθε μετασχηματισμό να αποτελείται από 64 σημεία, έτσι ώστε ο μετασχηματισμός κάθε 64-αδας να περιέχει στοιχεία των οποίων οι διευθύνσεις είναι της μορφής $64k + i$, όπου $i = 0, \dots, 63$ και k ο δείκτης πλειάδας που παίρνει τιμές από 0 μέχρι $\frac{4096}{64} - 1$. Ο πρώτος $R4$ παράγει 64 στοιχεία, τα οποία γράφονται σε μια από τις τράπεζες της πρώτης ενδιάμεσης μνήμης σε συνεχόμενες διευθύνσεις π.χ. $[a_5a_4a_3a_2a_1a_0] \rightarrow [d_5d_4d_3d_2d_1d_0]$. Εκτός αυτής της διαφοράς, η διευσθυνοδότηση και ο χρονοπρογραμματισμός των δεδομένων στο δεύτερο $R4^3$ παραμένουν ακριβώς τα ίδια με όσα περιγράφηκαν παραπάνω για τον πρώτο $R4^3$.

3.2.7 Υλοποίηση του 4K FFT

Η αρχιτεκτονική που παρουσιάστηκε στις παραπάνω παραγράφους έχει υλοποιηθεί σε γλώσσα RTL VHDL χρησιμοποιώντας αριθμητική σταθερής υποδιαστολής. Η υλοποίηση πραγματοποιήθηκε τόσο σε Προγραμματιζόμενες στο Πεδίο Διατάξεις Πύλης (FPGA), της εταιρείας XILINX όσο και σε Πολύ Μεγάλης Κλίμακας Ολοκλήρωσης (VLSI) σε τεχνολογία 0.13 μm . Για την υλοποίηση σε περιβάλλον XILINX χρησιμοποιήθηκαν τα αντίστοιχα εργαλεία της εταιρείας, επιλέγοντας σαν συσκευή το Virtex II 6000. Τα αποτελέσματα της υλοποίησης στο συγκεκριμένο FPGA έδειξαν ότι το σχέδιο κατέλαβε 20% σε χώρο που αφορά κύκλωμα λογικής και 25% σε μπλοκ μνημών τύπου RAM. Η χρήση βέλτιστων στοιχείων της XILINX, όπως πολλαπλασιαστές CoreLib, ελάττωσε τον καταλαμβανόμενο χώρο στο 13% των συνολικών πόρων του Virtex II 6000.

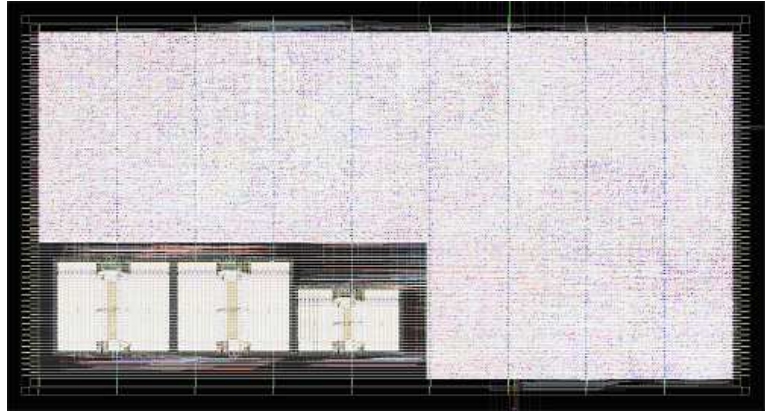
Η υλοποίηση με τη βιβλιοθήκη TSMC 0.13 μm επέφερε ως ελάχιστη συχνότητα λειτουργίας τα 604 MHz. Η υλοποίηση αυτή περιέχει 96K πρότυπα κελιά και 64 μνήμες RAM (1361 πρότυπα κελιά), σύνθεση $\times 64$, στον Πίνακα 3.2, ενώ καταλαμβάνει περιοχή έκτασης $2630 \times 129 \mu\text{m}^2$ ($1.42 \times 10^7 \mu\text{m}^2$), με αξιοποίηση στο 84.2%, και επιτυγχάνει στη χειρότερη περίπτωση (0.9V, 125C) συχνότητα λειτουργίας 604 MHz, καταναλώνοντας ενέργεια 4.4 Watt. Η χρήση τυπικών παραμέτρων (1V, 25C) έχει σαν αποτέλεσμα η συχνότητα λειτουργίας που επιτυγχάνεται να υπερβαίνει το όριο του 1 GHz (ρυθμός δεδομένων 1 Gsample/sec,) καθιστώντας τη συγκεκριμένη αρχιτεκτονική ως ένα από τα πιο γρήγορα κυκλώματα υλοποίησης Ταχύ Μετασχηματισμού Fourier 4096 μιγαδικών σημείων που αναφέρονται στη λογοτεχνία. Επιπλέον, υλοποιήθηκε και μια δεύτερη μηχανή 4096 μιγαδικών σημείων χρησιμοποιώντας την ίδια βιβλιοθήκη, χρησιμοποιώντας όμως αυτήν τη φορά βαθύτερες μνήμες RAM (σύνθεση $\times 16$, στον Πίνακα 3.2). Η κατανάλωση ενέργειας στην περίπτωση αυτή μειώθηκε σημαντικά, πέφτοντας στα 722.8 mW από τα 4.4 W της σύνθεσης $\times 64$. Στην εικόνα 3.6 φαίνεται το πλάνο σε VLSI της μηχανής $R4^3$, ενώ

Πίνακας 3.2: Αποτελέσματα υλοποίησης του 4K FFT σε VLSI.

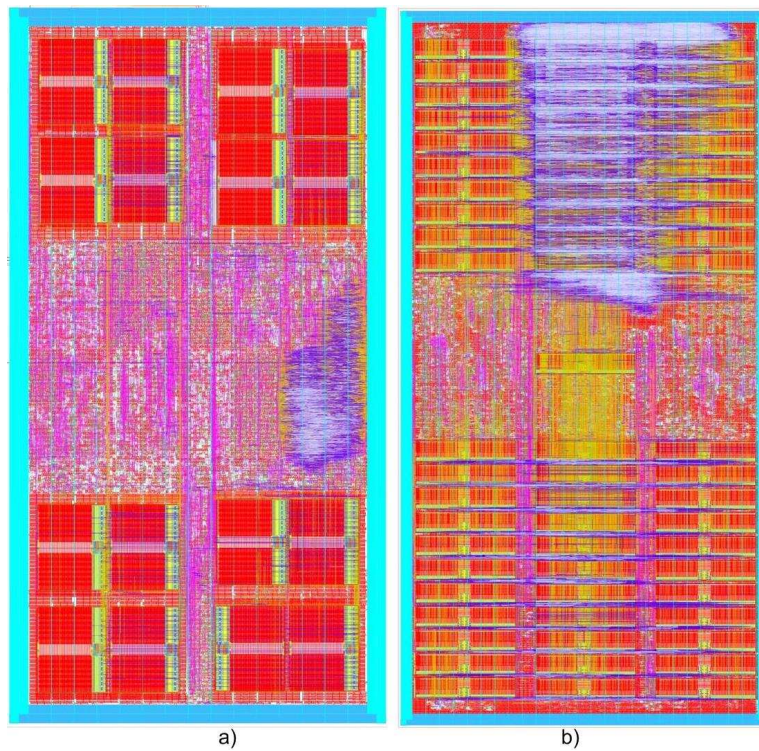
	FFT 4096 Ramsx16	FFT 4096 Ramsx64
Clock (period)	1.6 ns	1.2 ns
Fmax in MHz	386.8	604.5
Std Cells (RAMs)	100347(16)	95897(64)
Std Cells (rows)	805	1361
Chip Size (um sq)	5.34E+06	1.42E+07
Util (%)	80.30%	84.20%
x(um)	1681	2630
y(um)	3165	5129
Power (mW)	722.8	4414.1

η εικόνα 3.7 δείχνει τα τελικά πλάνα του 4K FFT , τόσο για τη σύνθεση $\times 16$ όσο και για την $\times 64$.

Ο 4K FFT σχεδιάστηκε για ένα πείραμα που αφορούσε έναν αναλυτή συχνοτήτων εύρους ζώνης 200 MHz. Η ζώνη διαιρέθηκε διαιρέθηκε σε τέσσερις υποζώνες και κάθε ζώνη χρησιμοποίησε έναν 4K FFT . Το κύκλωμα υλοποιήθηκε σε FPGA και λειτούργησε στα 102.5 MHz σε μια πλακέτα 18 στρωμάτων που διέθετε διασύνδεση PCI με λειτουργία στα 51.25MHz. Στόχος ήταν η πραγματοποίηση ενός κυκλώματος Ταχύ Μετασχηματισμού Fourier και η χρήση ενός φίλτρου κατωφλίου για την αναγνώριση συχνοτήτων μεγαλύτερης ενέργειας μέσα σε κάθε υποζώνη. Η αναμενόμενη έξοδος αποτελούταν από το πολύ 10 συχνότητες ανά υποζώνη, ανά FFT. Μετά την ολοκλήρωση του πρωτότυπου ο 4K FFT παραδόθηκε σαν IP core με προδιαγραφές που πετύχαιναν 5 φορές καλύτερη απόδοση από αυτήν του πρωτότυπου σε FPGA. (Τα κυκλώματα 16K, 64K και 265K FFT, η περιγραφή των οποίων θα γίνει στις επόμενες παραγράφους, υλοποιήθηκαν IP cores σαν αποκλειστικά για ερευνητικούς σκοπούς.)



Σχήμα 3.6: Υλοποίηση της μηχανής $R4^3$ σε VLSI.



Σχήμα 3.7: Υλοποίηση του επεξεργαστή 4K FFT σε VLSI α)Σύνθεση 16×4096 β)Σύνθεση 64×4096 .

3.3 Αρχιτεκτονικές FFT των 16K, 64K και 256K μιγαδικών σημείων

Η χρησιμότητα του σχεδίου 4K FFT που περιγράφηκε στα προηγούμενα γίνεται εμφανής με τη χρησιμοποίηση της για την ανάπτυξη μεγαλύτερων FFT αρχιτεκτονικών για τον υπολογισμό μετασχηματισμών 16K, 64K και 256K σημείων. Η ενότητα αυτή περιγράφει την αξιοποίηση του 4K FFT ως έναν επεξεργαστή "Βάσης-4096" που χρησιμοποιείται για τη δημιουργία αρχιτεκτονικών που μπορούν να ικανοποιήσουν διαφορετικές απαιτήσεις σε σχέση με τον παραλληλισμό, την περιοχή σε VLSI και τη ρυθμ απόδοση. Οι αρχιτεκτονικές αυτές μπορούν να προσαρμοστούν σε μια ευρεία περιοχή εφαρμογών.

3.3.1 Παράγοντας την αρχιτεκτονική 16K

Η εξίσωση Διακριτού Μετασχηματισμού Fourier για έναν 16K σημείων FFT παίρνει τη μορφή:

$$X[k] = \sum_{n=0}^{16383} x[n] W_{16384}^{nk} = \sum_{n=0}^{4 \cdot 4096 - 1} x[n] W_{4 \cdot 4096}^{nk} \quad (3.11)$$

Θέτοντας $n = n_1 + \frac{N}{4096}n_2$ και $k = k_1 4096 + k_2$:

$$\begin{aligned} k \cdot n &= \left(n_1 + \frac{N}{4096}n_2 \right) (4096k_1 + k_2) \Rightarrow \\ k \cdot n &= n_1 k_1 + n_1 k_2 + N n_2 k_1 + \frac{N}{4096} k_2 \Rightarrow \\ W_{4 \cdot 4096}^{kn} &= W_4^{n_1 k_1} W_{4 \cdot 4096}^{n_1 k_2} W_{4096}^{n_2 k_2} \end{aligned}$$

Επομένως, ο μετασχηματισμός γίνεται

$$X[k] = \sum_{n_1=0}^3 W_4^{n_1 k_1} \left(\sum_{n_2=0}^{4095} x[n] W_{4096}^{n_2 k_2} \right) W_{4 \cdot 4096}^{n_1 k_2} \quad (3.12)$$

Η παραπάνω ανάλυση οδηγεί σε δύο ενδιαφέρουσες αρχιτεκτονικές. Σύμφωνα με την εξίσωση 3.12, ο 4K FFT μπορεί να επεκταθεί στα 16K σημεία. Αυτό επιτυγχάνεται πραγματοποιώντας αρχικά τέσσερις συνεχόμενους μετασχηματισμούς FFT 4K σημείων ο καθένας. Στη συνέχεια τα δεδομένα πολλαπλασιάζονται με τους παράγοντες στροφής που αντιστοιχούν στο μετασχηματισμό 16K σημείων FFT, και τέλος, ένα τελευταίο στάδιο βάσης-4 ολοκληρώνει τον μετασχηματισμό 16K FFT. Οι επόμενες παράγραφοι περιγράφουν δύο αρχιτεκτονικές που αποτελούν εφαρμογές της εξίσωσης 3.12. Η πρώτη ονομάζεται *Σειριακή/Παράλληλη* και χρησιμοποιεί έναν επεξεργαστή "βάσης-4096" και ένα στάδιο βάσης-4. Η δεύτερη ονομάζεται *Παράλληλη/Παράλληλη* και χρησιμοποιεί παράλληλα τέσσερις επεξεργαστές "βάσης-4096" και ένα στάδιο βάσης-4.

3.3.1.1 Η Σειριακή/Παράλληλη αρχιτεκτονική

Η Σειριακή/Παράλληλη αρχιτεκτονική που προκύπτει από τη χρήση του 4K FFT, που αναλύθηκε στην προηγούμενη ενότητα, παριστάνεται στην εικόνα 3.8. Ο 4K FFT εκτελεί τέσσερις διαδοχικούς μετασχηματισμούς, με τα παραγόμενα αποτελέσματα να αποθηκεύονται σε τέσσερις μνήμες μεγέθους 4K λέξεων η καθεμία. Στη συνέχεια τα δεδομένα πολλαπλασιάζονται με τους αντίστοιχους παράγοντες στροφής και επεξεργάζονται από την παράλληλη μηχανή βάσης-4. Στην αρχιτεκτονική αυτή ο 4K FFT χρησιμοποιείται στο 100% του χρόνου, ενώ η μηχανή βάσης-4 στο 25%. Ο συνολικός χρόνος που απαιτείται για να ολοκληρώσει η αρχιτεκτονική αυτή έναν μετασχηματισμό 16K είναι περίπου $4 \times T_{4k-FFT}$, όπου T_{4k-FFT} ο απαιτούμενος χρόνος για να ολοκληρωθεί ένας μετασχηματισμός 4K FFT. Το βασικό πλεονέκτημα της αρχιτεκτονικής αυτής είναι η μνήμη που απαιτείται για να ολοκληρωθεί ένας μετασχηματισμός 16K σημείων, αφού πλέον δεν χρειάζεται να υπάρξει ούτε ενδιάμεση αποθήκευση (buffering) κατά το στάδιο εξόδου, ούτε αποθήκευση 16K σημείων στα ενδιάμεσα στάδια του κάθε 4K FFT. Για να αποφευχθεί η χρήση διπλών ενδιάμεσων μνημών στο τελικό στάδιο του αλγορίθμου, χρησιμοποιείται ένα σχήμα διευθυνσιοδότησης το οποίο αντικαθιστά τα ενδιάμεσα στοιχεία στις τράπεζες μνήμης με εκείνα που παράγονται από τους επεξεργαστές $R4$. Οι τράπεζες μνήμης εφαρμόζουν το σχήμα *διάβασμα πριν το γράψιμο* (read-before-write) και η ίδια ακολουθία διευθύνσεων χρησιμοποιείται τόσο για την αποθήκευση της εξόδου ενός επεξεργαστή όσο και για το διάβασμα των αποτελεσμάτων του προηγούμενου σταδίου.

Η τεχνική αυτή περιπλέκει την *αντιστροφή ψηφίου* (digit reversal) στην έξοδο του μετασχηματισμού 16K, αλλά μειώνει το μέγεθος της συνολικής μνήμης που χρησιμοποιείται. Σε μια εναλλακτική λειτουργία, η αποθήκευση των δεδομένων λαμβάνει χώρα στην αντίστοιχη θέση μνήμης που θα οδηγούσε η αντιστροφή ψηφίου και στη συνέχεια πραγματοποιούνται οι υπολογισμοί βάσης-4 με την ίδια σειρά αντεστραμμένων ψηφίων. Εφαρμόζοντας τις τεχνικές αυτές ένας μετασχηματισμός FFT 16K σημείων χρησιμοποιεί συνολικά $2 \times 2 \times 4K + 16K$ λέξεις μνήμης, υπερβαίνοντας μόνο κατά 16K τον μετασχηματισμό 4K FFT.

3.3.1.2 Η Παράλληλη/Παράλληλη αρχιτεκτονική

Η δεύτερη αρχιτεκτονική παριστάνεται στην εικόνα 3.9 και ακολουθεί την ίδια πορεία υλοποίησης με αυτή που περιγράφηκε παραπάνω έχοντας διαφορά μόνο σε ένα σημείο. Στην περίπτωση αυτή, υπάρχουν τέσσερις μηχανές 4K FFT που λειτουργούν παράλληλα. Η αρχιτεκτονική δέχεται σαν είσοδο 4 δεδομένα ανά κύκλο και έτσι, επιτυγχάνει ρυθμαπόδοση τέσσερις φορές μεγαλύτερη απ'ότι το σχήμα Σειριακή/Παράλληλη αρχιτεκτονική. Μια παράλληλη μηχανή βάσης-4 βρίσκεται μετά από τις τέσσερις μηχανές 4K FFT και δέχεται σαν είσοδο τα αποτελέσματα τους. Το κύριο πλεονέκτημα αυτής της προσέγγισης είναι ότι χρειάζεται χρόνο T_{4k-FFT} για να ολοκληρώσει έναν μετασχηματισμό 16K FFT,

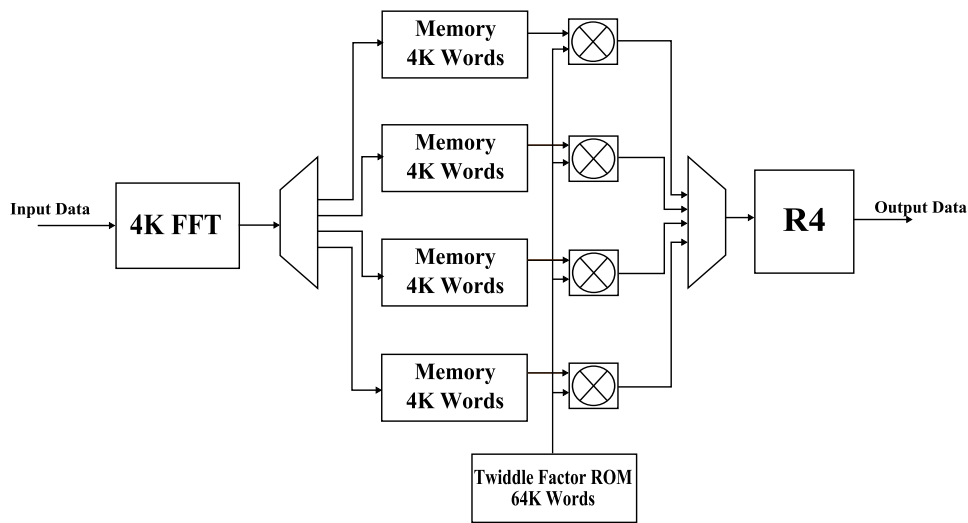
Πίνακας 3.3: Αποτελέσματα της υλοποίησης σε VLSI της 16K Παράλληλης/ Παράλληλης FFT αρχιτεκτονικής.

Parameter	FFT 16K
Core Size	$1.6561e + 07um^2$
Std Cell Rows	1102
Number of Cells	885456
Number of RAMs	64
Statistical Power	3.45 W
Fmax	352 MHz

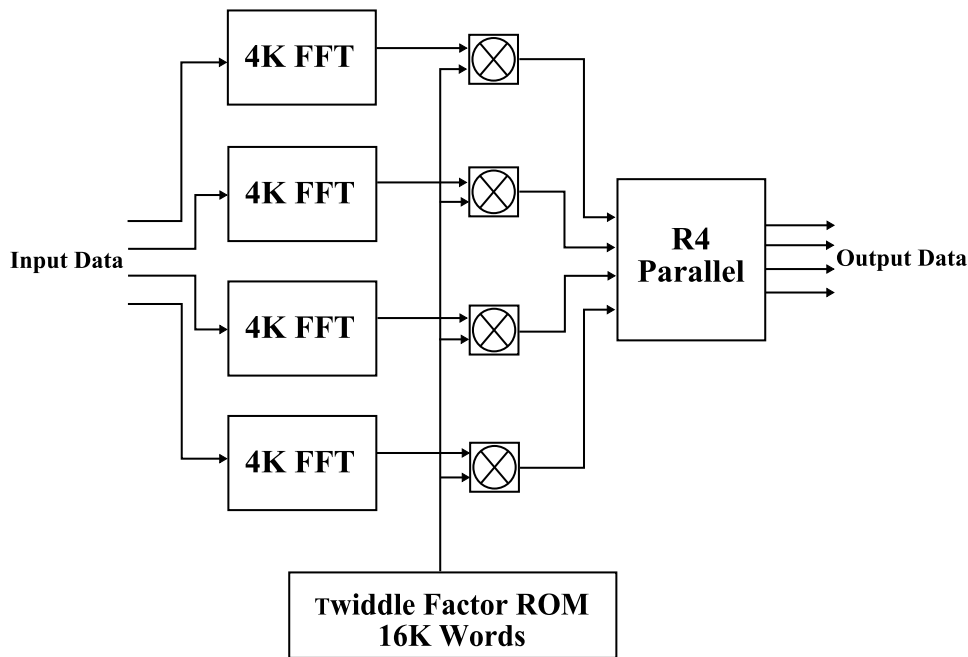
το οποίο όμως επιτυγχάνεται με αύξηση της περιοχής σε VLSI, σε σχέση με τη Σειριακή/Παράλληλη αρχιτεκτονική. Η ίδια αρχή ακολουθείται και από τις δύο 64K FFT αρχιτεκτονικές που χρησιμοποιούν (16) μηχανές των 4K FFT.

Η 16K Παράλληλη/Παράλληλη αρχιτεκτονική υλοποιήθηκε με χρήση της τεχνολογίας TSMC, 1Poly-8Copper, $0.13um$. Το σχέδιο συνθέθηκε αρχικά σε επίπεδο πυλών χρησιμοποιώντας το εργαλείο Synopsys Design Compiler και βελτιστοποιήθηκε για συχνότητα 300 MHz. Στη συνέχεια, τα αποτελέσματα της σύνθεσης οδηγήθηκαν στο εργαλείο Cadence SoC Encounter όπου πραγματοποιήθηκαν οι λειτουργίες ενεργειακής ανάλυσης, σύνθεσης δέντρου ρολογιού (clock tree-synthesis), τοποθέτησης (placement), δρομολόγησης (routing) και χρονικής ανάλυσης (timing analysis). Τέλος, το σχέδιο ξαναγύρισε στο ανώτατο επίπεδο για την τελική τοποθέτηση, δρομολόγηση και χρονική ανάλυση. Ο πίνακας 3.3 παρουσιάζει τα αποτελέσματα της υλοποίησης σε VLSI και η εικόνα 3.10 δείχνει τη διαρρύθμιση σε VLSI της αρχιτεκτονικής 16K FFT.

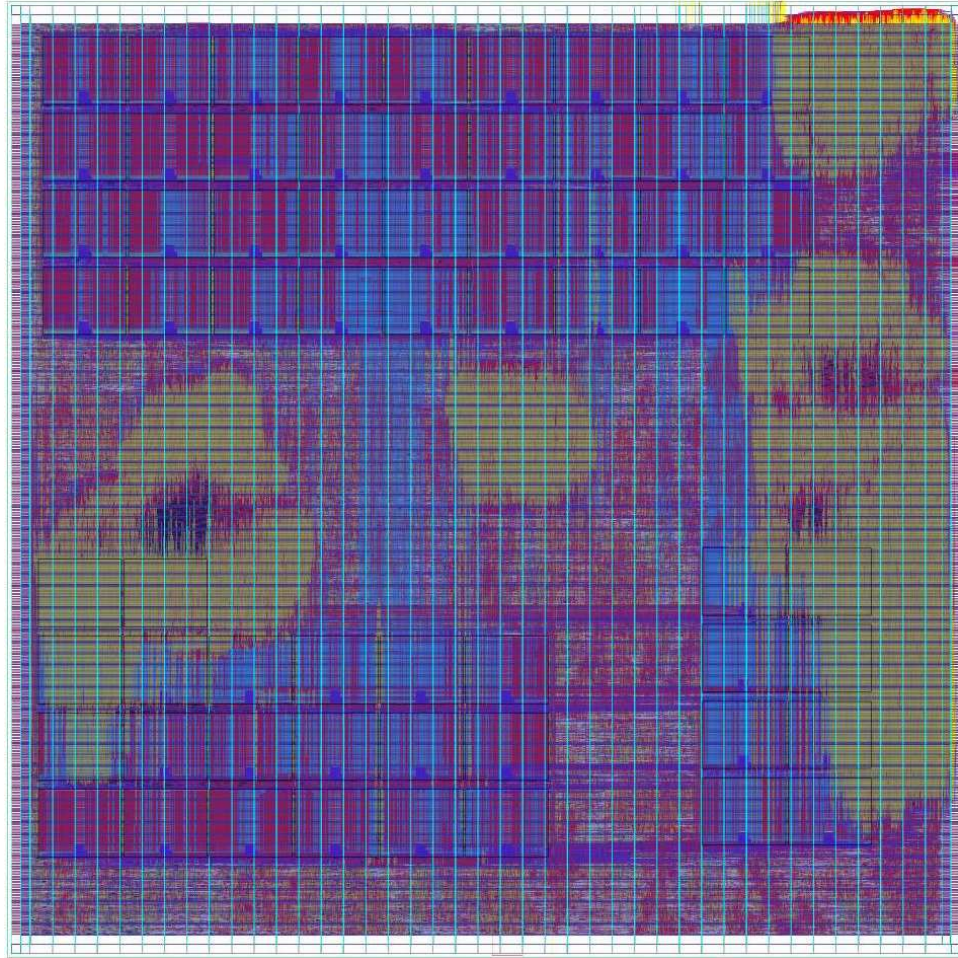
Η 16K Παράλληλη/Παράλληλη αρχιτεκτονική υλοποιήθηκε επίσης στο Xilinx Virtex-5 (-2) επιτυγχάνοντας συχνότητα 250 MHz και καταλαμβάνοντας 12264 slices. Η αρχιτεκτονική έχει τέσσερις παράλληλες εισόδους και εξόδους διατηρώντας ρυθμαπόδοση 1Gsample/sec (28 Gbits/sec).



Σχήμα 3.8: Μπλοκ διάγραμμα της Σειριακής/Παράλληλης αρχιτεκτονικής 16K FFT.



Σχήμα 3.9: Μπλοκ διάγραμμα της Παράλληλης/Παράλληλης αρχιτεκτονικής 16K FFT.



Σχήμα 3.10: Υλοποίηση σε VLSI της 16K Παράλληλης/Παράλληλης FFT αρχιτεκτονικής.

3.3.2 Η 64K FFT αρχιτεκτονική

Η εξίσωση του Διακριτού Μετασχηματισμού Fourier για έναν 64K σημείων FFT παίρνει τη μορφή:

$$X[k] = \sum_{n=0}^{65535} x[n] W_{65536}^{nk} = \sum_{n=0}^{4 \cdot 16384 - 1} x[n] W_{4 \cdot 16384}^{nk} \quad (3.13)$$

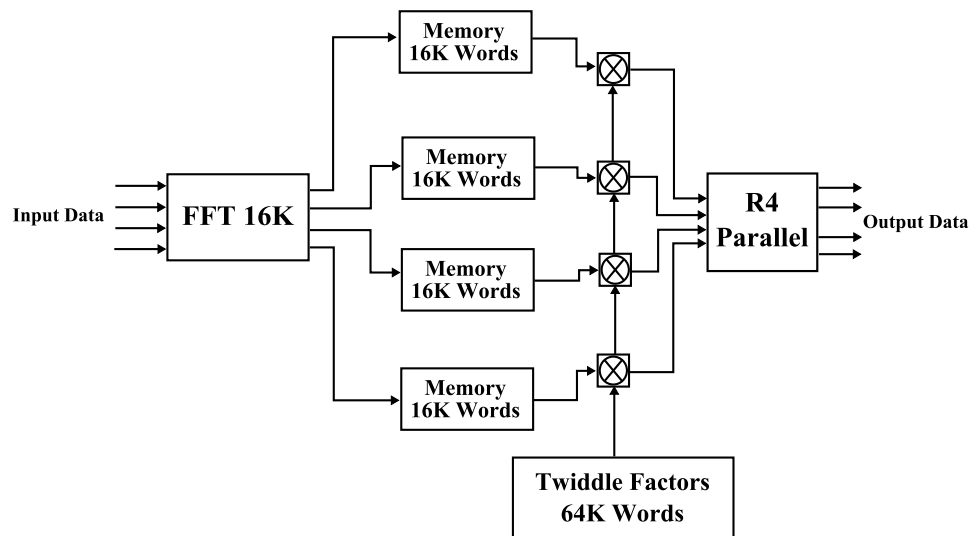
Θέτοντας $n = n_1 + \frac{N}{16384} n_2$ και $k = k_1 16384 + k_2$:

$$\begin{aligned} k \cdot n &= 16384 n_1 k_1 + n_1 k_2 + N n_2 k_1 + \frac{N}{16384} k_2 \rightarrow \\ W_{4 \cdot 16384}^{kn} &= W_4^{n_1 k_1} W_{4 \cdot 16384}^{n_1 k_2} W_{16384}^{n_2 k_2} \end{aligned}$$

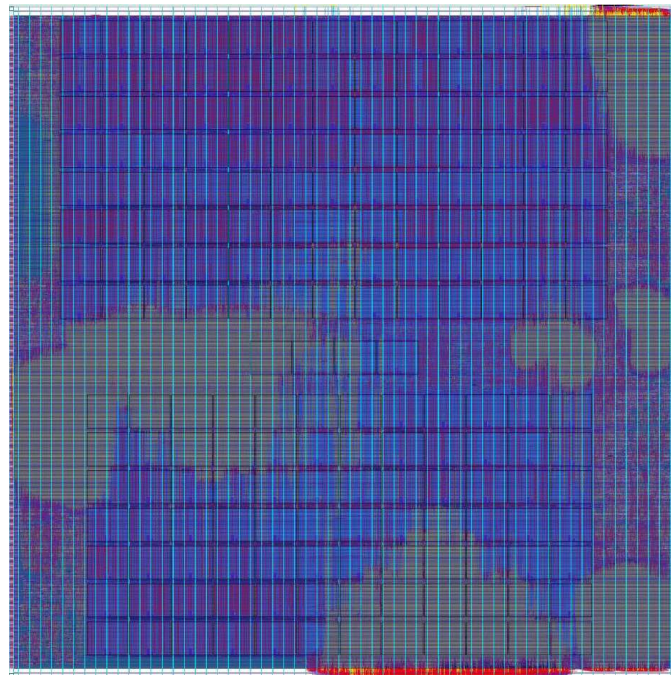
Επομένως, ο μετασχηματισμός γίνεται:

$$X[k] = \sum_{n_1=0}^3 W_4^{n_1 k_1} \left(\sum_{n_2=0}^{16383} x[n] W_{16384}^{n_2 k_2} \right) W_{4 \cdot 16384}^{n_1 k_2} \quad (3.14)$$

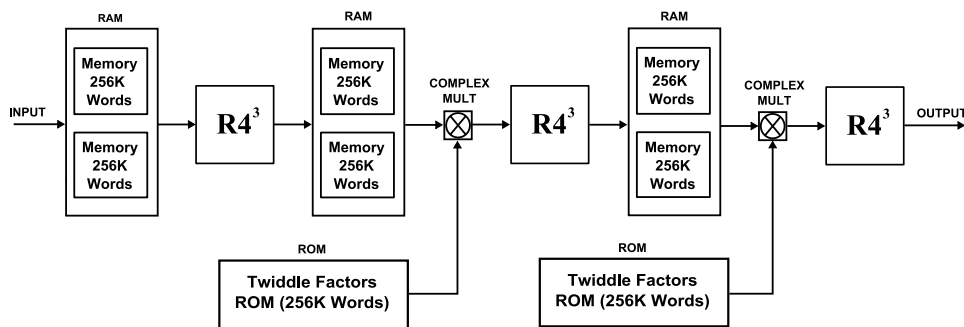
Σύμφωνα με την εξίσωση 3.14, ο 16K σημείων Ταχύς Μετασχηματισμός Fourier μπορεί να επεκταθεί σε 64K σημεία. Αυτό επιτυγχάνεται πραγματοποιώντας αρχικά ένα μετασχηματισμό 16K FFT. Στη συνέχεια, τα δεδομένα πολλαπλασιάζονται με τους παράγοντες στροφής που αντιστοιχούν σε έναν μετασχηματισμό 64K FFT και τέλος, ένα στάδιο βάσης-4 ολοκληρώνει τον υπολογισμό του 64K FFT, όπως φαίνεται και στην εικόνα 3.11. Ο 64K FFT υλοποιήθηκε σε VLSI (και σε FPGA) χρησιμοποιώντας τον 16K παράλληλο/παράλληλο υπολογισμό, μαζί με ένα στάδιο βάσης-4 με 4 παράλληλες εισόδους και εξόδους. Η συχνότητα λειτουργίας του κυκλώματος ήταν τα 256 MHz με ρυθμαπόδοση 1 Gsample/sec (28 Gbits/sec). Η εικόνα 3.12 παρουσιάζει το VLSI κελί του σχεδίου 64K FFT. Ο 64K υλοποιήθηκε επίσης και στο Xilinx Virtex-5 (-2) με συχνότητα λειτουργίας τα 125 MHz, καταλαμβάνοντας 13461 slices. Η αρχιτεκτονική έχει τέσσερις (4) παράλληλες εισόδους και εξόδους διατηρώντας ρυθμαπόδοση 500 Msample/sec (14 Gbits/sec).



Σχήμα 3.11: Μπλοκ διάγραμμα της αρχιτεκτονικής 64K FFT.



Σχήμα 3.12: Υλοποίηση σε VLSI του 64K FFT.



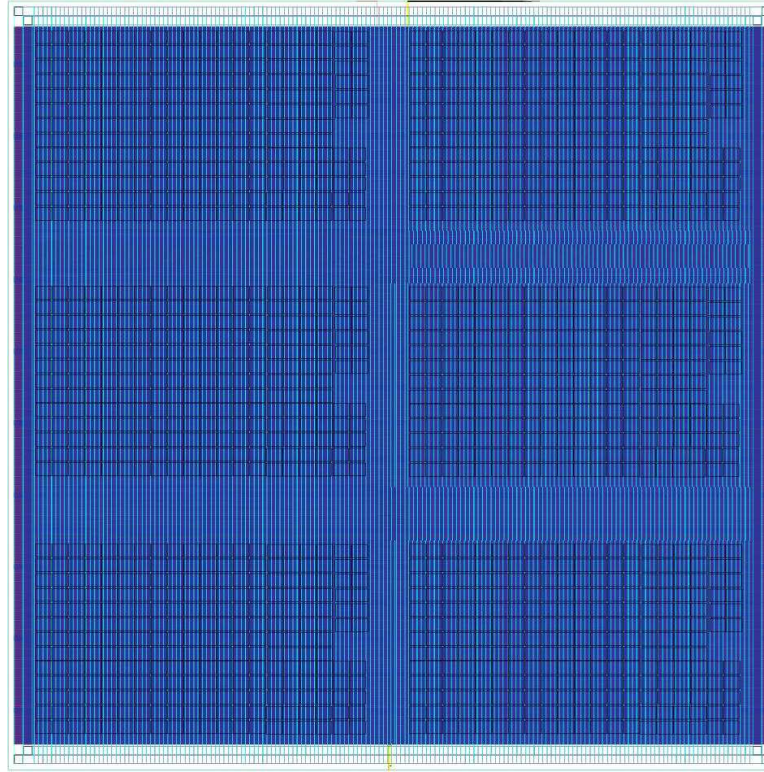
Σχήμα 3.13: Μπλοκ διάγραμμα της αρχιτεκτονικής 256K FFT.

3.3.3 Η 256K FFT αρχιτεκτονική

Στην εικόνα 3.13 φαίνεται η αρχιτεκτονική του 256K FFT. Η αρχιτεκτονική αυτή είναι μια απλή εφαρμογή του αλγορίθμου βάσης -4^3 , με τρεις συνεχόμενες μηχανές $R4^3$ να χρησιμοποιούνται για την υλοποίησης της.

Για την υλοποίηση του, πολύ μεγάλου σε έκταση, 256K FFT ήταν αναγκαία η χρήση της ιεραρχικής ροής (hierarchical flow). Ακολουθώντας την ίδια διαδικασία σύνθεσης (ξανά για 300 MHz) κύκλωμα οδογήθηκε στο Cadence SoC Encounter, όπου έγινε αρχικά τμηματοποίηση του. Η διαδικασία αυτή δημιούργησε 6 τμήματα μνήμης του 256K (1.5 MB on-chip SRAM), τα οποία τοποθετήθηκαν και δρομολογήθηκαν ξεχωριστά. Η ίδια διαδικασία ακολουθήθηκε για τις μηχανές $R4^3$ και για τη μνήμη ROM που περιέχει τους παράγοντες στροφής. Τέλος, το σχέδιο ξαναγύρισε στο ανώτατο επίπεδο για την τελική τοποθέτηση, δρομολόγηση και χρονική ανάλυση.

Θα πρέπει να σημειωθεί, ότι ένας πλήρης υπολογισμός απαιτεί καθυστέρηση για 3 σετ δεδομένων (frames), δηλαδή 786792 κύκλοι, συν την εσωτερική καθυστέρηση των τριών μηχανών $R4^3$ (360 κύκλοι), που ισούται με 4.1 msec. Η αρχιτεκτονική 256K FFT επιτυγχάνει σα συχνότητα λειτουργίας τα 188 MHz. Στην εικόνα 3.14 απεικονίζεται το VLSI κελί του 256K FFT, ενώ ο πίνακας 3.4 παρουσιάζει συνοπτικά τα αποτελέσματα της υλοποίησης των αρχιτεκτονικών 64K και 256K FFT.



Σχήμα 3.14: Υλοποίηση σε VLSI του 256K FFT.

Πίνακας 3.4: Αποτελέσματα υλοποίησης σε VLSI των αρχιτεκτονικών 64K και 256K FFT.

Parameter	FFT 64K	FFT 256K
Core Size	$3.4894e + 08um^2$	$2.7647e + 08um^2$
Std Cell Rows	1600	4500
Number of Cells	896148	735945
Number of RAMs	192	384
Statistical Power	9.8 W	35.75 W
Fmax	256.5 MHz	188 MHz

Κεφάλαιο 4

Αρχιτεκτονικές Κινητής Υποδιαστολής

4.1 Εισαγωγή

Ένα μεγάλο πλήθος εφαρμογών όπως γραφικά, μετασχηματισμοί, ψηφιακά φίλτρα κλπ. επιβάλλει την ανάγκη υπολογισμών κινητής υποδιαστολής, οι οποίοι με τη σειρά τους αποτελούν χαρακτηριστικό στοιχείο όλων των υπολογιστών υψηλής απόδοσης, των επεξεργαστών ψηφιακού σήματος (DSPs) και των επιταχυντών για γραφικά [42], [43], [44], [45]. Επιπλέον, εφαρμογές στο χώρο των τρισδιάστατων γραφικών χρειάζονται την παράλληλη εκτέλεση πράξεων κινητής υποδιαστολής, οι οποίες συνήθως αφορούν τελεστέους μονής ή/και διπλής ακρίβειας. Για να ικανοποιηθούν αυτές οι απαιτήσεις ένας μεγάλος αριθμός από καινοτόμους επεξεργαστές παρέχουν Μονάδες Κινητής Υποδιαστολής (Floating Point Units) ή αλλιώς MKY (FPU) οι οποίες υποστηρίζουν πράξεις όχι μόνο μονής αλλά και διπλής ακρίβειας. Επιτυγχάνοντας το στόχο αυτό όμως, επηρεάζεται συνήθως ο ρυθμός διαμεταγωγής, αφού οι περισσότερες MKY μετατρέπουν τους μονής ακρίβειας τελεστέους σε διπλής και μεταφράζουν ξανά το αποτέλεσμα σε μορφή μονής ακρίβειας. Ένα ακόμα μειονέκτημα για πολλές από τις υπάρχουσες λύσεις είναι ότι εκτελούν πράξεις κινητής υποδιαστολής μόνο σειριακά, ενώ η παράλληλη εκτέλεση τους έχει γίνει αναγκαία, ειδικά στο χώρο των απαιτητικών εφαρμογών σε γραφικά και πολυμέσα [44], [46], [47].

Η μονή και η διπλή ακρίβεια είναι τα δύο πιο διαδεδομένα μορφότυπα για υπολογισμούς κινητής υποδιαστολής. Παρόλα αυτά, οι αυξανόμενες υπολογιστικές απαιτήσεις των επιστημονικών εφαρμογών έχουν δείξει ότι σε πολλές περιπτώσεις υπάρχει η ανάγκη για μεγαλύτερη ακρίβεια στους υπολογισμούς κινητής υποδιαστολής [48]. Τέτοια παραδείγματα εφαρμογών αποτελούν τα πεδία της υπολογιστικής φυσικής [49], [50], υπολογιστικής γεωμετρίας [49], [51], τα κλιματικά μοντέλα [52] κλπ., τα οποία απαιτούν υπολογισμούς υψηλής ακρίβειας και μεγάλης ορθότητας, χαρακτηριστικά τα οποία το μορφότυπο διπλής ακρίβειας δεν μπορεί να παρέχει. Οι περιπτώσεις αυτές επιβάλλουν τη χρήση αριθμητι-

κής τετραπλής ακρίβειας, επειδή παρέχει διπλάσια ακρίβεια απ'ότι το μορφότυπο διπλής ακρίβειας, βελτιώνει την ακρίβεια των υπολογισμών και οδηγεί σε πιο αξιόπιστα αποτελέσματα. Για τους παραπάνω λόγους η IEEE συμπεριέλαβε τις προδιαγραφές της αριθμητικής τετραπλής ακρίβειας στο αναθεωρημένο Πρότυπο για την Αριθμητική Κινητής Υποδιαστολής IEEE 754-2008 [53]. Παρόλο που οι πράξεις με αριθμούς τετραπλής ακρίβειας μπορούν να πραγματοποιηθούν τόσο στο λογισμικό (software), όσο και στο υλισμικό (hardware), το τελευταίο παρέχει πιο αποτελεσματικές λύσεις ως προς το θέμα της απόδοσης.

Οι περισσότερες από τις εφαρμογές που αναφέρθηκαν πιο πάνω συμπεριλαμβάνουν αλγόριθμους που υλοποιούν την εξίσωση εντολής $(A \times B) + C$, η οποία πραγματοποιείται σε γενικού σκοπού επεξεργαστές, σε μια μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης (Multiply-Add Fused Unit ή MAF). Αντί να χρησιμοποιηθεί ένας ξεχωριστός πολλαπλασιαστής και ένας ξεχωριστός αθροιστής κινητής υποδιαστολής, η μονάδα MAF συνδυάζει και τα δύο ώστε να μειωθεί η καθυστέρηση των υπολογισμών. Η χρησιμοποίηση της μονάδας MAF αντί για ξεχωριστές μονάδες πολλαπλασιασμού και πρόσθεσης προσφέρει έναν αριθμό από πλεονεκτήματα. Πρώτον, η μονάδα MAF παρουσιάζει βελτιωμένη ακρίβεια. Από τη στιγμή που απαιτεί μία μόνο πράξη στρογγυλοποίησης, στο τελευταίο στάδιο των υπολογισμών, τόσο ο πολλαπλασιασμός όσο και η πρόσθεση πραγματοποιούνται με πλήρη ακρίβεια, παρέχοντας έτσι και τελικό αποτέλεσμα με βελτιωμένη ακρίβεια. Δεύτερον, η μονάδα MAF πραγματοποιεί κανονικοποίηση και στρογγυλοποίηση μόνο μία φορά κάθε δύο πράξεις κινητής υποδιαστολής. Έτσι, οι απαιτήσεις της σε υλισμικό είναι μειωμένες, ενώ ταυτόχρονα εμφανίζει βελτιωμένη καθυστέρηση, συγκρινόμενη με την τυπική υλοποίηση κατά την οποία πραγματοποιείται πρώτα ένας πλήρης πολλαπλασιασμός και στη συνέχεια πρόσθεση. Τέλος, μία μονάδα MAF εξακολουθεί να μπορεί να πραγματοποιεί κανονικές πράξεις κινητής υποδιαστολής χρησιμοποιώντας σταθερές. Έτσι, θέτοντας $C = 0.0$ ή $B = 1.0$ μπορούμε να πραγματοποιήσουμε αντίστοιχα πολλαπλασιασμό ή πρόσθεση κινητής υποδιαστολής.

Πέραν όμως από τα παραπάνω πλεονεκτήματα, ο συνδυασμός των δύο πράξεων σε μία εντολή παρουσιάζει και κάποια μειονεκτήματα. Η χρήση των σταθερών για την πραγματοποίηση απλού πολλαπλασιασμού ή πρόσθεσης προκαλεί μεγαλύτερες καθυστερήσεις, απ'ότι η εκτέλεση τους σε ανεξάρτητες μονάδες πολλαπλασιασμού ή πρόσθεσης κινητής υποδιαστολής. Επιπλέον, η μονάδα MAF αυξάνει την πολυπλοκότητα του υλισμικού και εμποδίζει την παράλληλη εκτέλεση πράξεων πολλαπλασιασμού ή πρόσθεσης κινητής υποδιαστολής.

Λόγω της μεγάλης σημασίας που έχουν οι εντολές MAF πολλοί επεξεργαστές γενικού σκοπού (GPPs) έχουν ενσωματωμένες μονάδες MAF για να επιταχύνουν την εκτέλεση αυτών των SIMD (Single Instruction Multiple Data) εντολών, ενώ άλλες τις χρησιμοποιούν για να αντικαταστήσουν εντελώς τις μονάδες άθροισης και πολλαπλασιασμού κινητής υποδιαστολής [54], [55], [56].

Στο κεφάλαιο αυτό παρουσιάζονται αποτελεσματικές αρχιτεκτονικές κινητής υποδιαστολής που μπορούν να ικανοποιήσουν τις απαιτήσεις που αναφέρθηκαν πιο πάνω. Οι μονάδες αυτές είναι υψηλής απόδοσης, και είναι σχεδιασμένες έτσι

ώστε να υποστηρίζουν διάφορες ακρίβειες και να επιτυγχάνουν υψηλή ρυθμική απόδοση, χωρίς να αυξάνουν σημαντικά το χώρο που καταλαμβάνουν. Αρχικά παρουσιάζεται ένας πολλαπλασιαστής κινητής υποδιαστολής που υποστηρίζει όλες τις δυνατές ακρίβειες που περιγράφονται από το Πρότυπο IEEE 754-2008, εκτελώντας έναν πολλαπλασιασμό τετραπλής ακρίβειας σε τρεις κύκλους, ή δύο παράλληλους πολλαπλασιασμούς διπλής ακρίβειας σε δύο κύκλους ή τέλος τέσσερις παράλληλους πολλαπλασιασμούς μονής ακρίβειας σε δύο κύκλους. Στη συνέχεια αναλύονται τρεις διαφορετικές αρχιτεκτονικές Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης. Η πρώτη από αυτές μπορεί να διαχειριστεί κανονικοποιημένους και μη κανονικοποιημένους αριθμούς και είναι υλοποιημένη με σωλήνωση μεταβλητού μήκους. Η δεύτερη βασίζεται στην τεχνική διπλού μονοπατιού και μπορεί να εξυπηρετεί αριθμούς μονής και διπλής ακρίβειας. Τέλος, η τελευταία αρχιτεκτονική μπορεί να πραγματοποιεί πράξεις με αριθμούς μονής, διπλής ή τετραπλής ακρίβειας και ο σχεδιασμός της στηρίζεται σε μια τροποποιημένη εκδοχή του διπλού μονοπατιού. Θα πρέπει να σημειωθεί ότι και οι δύο τελευταίες αρχιτεκτονικές Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης, που διαχειρίζονται πάνε από ένα είδος ακρίβειας, χρησιμοποιούν πολλαπλασιαστές που βασίζονται στη λειτουργικότητα του πολλαπλασιαστή που αναφέρθηκε αρχικά.

4.2 Σχετική βιβλιογραφία

Πριν προχωρήσουμε στην παρουσίαση των αρχιτεκτονικών κινητής υποδιαστολής αυτού του κεφαλαίου, γίνεται μια σύντομη αναφορά σε κάποια από τα πιο σημαντικά σχέδια μονάδων Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης που μπορεί να συναντήσει κανείς στη βιβλιογραφία. Η μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης πρωτοπαρουσιάστηκε στον επεξεργαστή IBM POWER1 (γνωστό και ως *RS/6000*) το 1990. Από τότε ένας μεγάλος αριθμός από σχέδια προτάθηκε και υλοποιήθηκε, σε μια προσπάθεια βελτίωσης της αποδοτικότητας των MAF αρχιτεκτονικών. Οι συγγραφείς του [57] συγκρίνουν τη σχεδιαστική πολυπλοκότητα τεχνικών που αφορούν ένα μονό ή ένα διπλό πέρασμα από μια μήτρα πολλαπλασιαστή, για την υλοποίηση μιας Συγχωνευμένης Μονάδας Πολλαπλασιασμού - Πρόσθεσης. Στο [58] παρουσιάζεται μια μονάδα MAF με προσημασμένη πρόσθεση: η προσημασμένη πρόσθεση, μαζί μια μέθοδο κανονικοποίησης σε δύο βήματα ελαττώνουν την καθυστέρηση της πρόσθεσης. Το [59] παρουσιάζει μια μονάδα MAF που πραγματοποιεί την απλή πρόσθεση κινητής υποδιαστολής με χαμηλότερη καθυστέρηση απ'ό,τι πραγματοποιείται ο πολλαπλασιασμός ή η εντολή MAF. Αυτό επιτυγχάνεται παρακάμπτοντας, σε περίπτωση πρόσθεσης, το στάδιο του πολλαπλασιασμού. Το [60] υλοποιεί μια μονάδα κινητής υποδιαστολής Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης με δύο μονοπάτια δεδομένων, και με βάση το αν πραγματοποιείται ή όχι αφαίρεση ενεργοποιεί ένα εκ των δύο, μειώνοντας έτσι τη συνολική καθυστέρηση. Στο [61] παρουσιάζεται μια μονάδα MAF που μπορεί να διαχειρίζεται μη κα-

νονικοποιημένους αριθμούς, επιβαρύνοντας το σχέδιο με μία μικρή αύξηση σε καθυστέρηση και στο απαιτούμενο υλισμικό. Στο [62] μια πλήρως σωληνωμένη μονάδα MAF συνδυάζει την τελική πρόσθεση με τη στρογγυλοποίηση. Στο [63] παρουσιάζεται ένα σχέδιο μονάδας Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης με γέφυρα. Η αρχιτεκτονική αυτή περιλαμβάνει έναν τυπικό πολλαπλασιαστή και έναν τυπικό αθροιστή κινητής υποδιαστολής, καθώς και συγκεκριμένα κομμάτια υλισμικού μεταξύ τους, τα οποία δημιουργούν μια 'γέφυρα' για να στέλνονται τα δεδομένα από τον πολλαπλασιαστή στον αθροιστή, ώστε να εκτελείται η εντολή MAF. Πιο πολύπλοκες αρχιτεκτονικές παρουσιάζονται στο [64] και στο [65]: το [64] προτείνει μια μονάδα κινητής υποδιαστολής MAF διπλής λειτουργίας, που μπορεί να εκτελεί εντολές μονής ή διπλής ακρίβειας και το [65] περιγράφει μια πολλαπλής λειτουργίας μονάδα MAF διπλής και τετραπλής ακρίβειας.

Επιπρόσθετα στα παραπάνω σχέδια MAF, υπάρχουν στη βιβλιογραφία πολλές άλλες αρχιτεκτονικές οι οποίες αφορούν αυτόνομους πολλαπλασιαστές και αθροιστές κινητής υποδιαστολής. Αξιοσημείωτα παραδείγματα σχετικής έρευνας δημοσιεύονται στα [66], [67], [68], [69], [70], [71]. Οι συγγραφείς του [66] παρουσιάζουν έναν πολλαπλασιαστή κινητής υποδιαστολής που εκτελεί την πρόσθεση παράλληλα με τη στρογγυλοποίηση. Στο [67] προτείνεται ένας πολλαπλασιαστής διπλής λειτουργίας: χρησιμοποιώντας πολλαπλασιαστές μήτρας μισού μεγέθους πραγματοποιεί έναν πολλαπλασιασμό διπλής ακρίβειας σε τρεις κύκλους ή έναν πολλαπλασιασμό μονής ακρίβειας σε δύο κύκλους. Το [68] παρουσιάζει δύο σχέδια για πολλαπλασιασμό κινητής υποδιαστολής με διπλής ακρίβεια. Η πρώτη αρχιτεκτονική εκτελεί είτε έναν πολλαπλασιασμό τετραπλής ακρίβειας είτε δύο παράλληλους πολλαπλασιασμούς διπλής ακρίβειας. Το δεύτερο σχέδιο εκτελεί έναν πολλαπλασιασμό διπλής ακρίβειας ή δύο παράλληλους πολλαπλασιασμούς μονής ακρίβειας. Ένας ακόμη πολλαπλασιαστής κινητής υποδιαστολής, διπλής λειτουργίας, που πραγματοποιεί μονής ή διπλής ακρίβειας πράξεις παρουσιάζεται στο [69]. Επιπλέον, στο [70] οι συγγραφείς αναλύουν τον απαιτούμενο χώρο, την καθυστέρηση και το ρυθμό διαμεταγωγής για μονάδες πολλαπλασιαστών και αθροιστών κινητής υποδιαστολής, βάσει του αριθμού των σταδίων σωλήνωσης τους. Τέλος, στο [71] προτείνονται δύο αρχιτεκτονικές που πραγματοποιούν πρόσθεση κινητής υποδιαστολής με διπλή λειτουργία. Η πρώτη βασίζεται στον αλγόριθμο μονού μονοπατιού και υποστηρίζει μια πρόσθεση διπλής ακρίβειας ή δύο παράλληλες προσθέσεις μονής ακρίβειας. Η δεύτερη υλοποιεί τον αλγόριθμο διπλού μονοπατιού και πραγματοποιεί μια τετραπλής ακρίβειας πρόσθεση ή δύο παράλληλες προσθέσεις διπλής ακρίβειας.

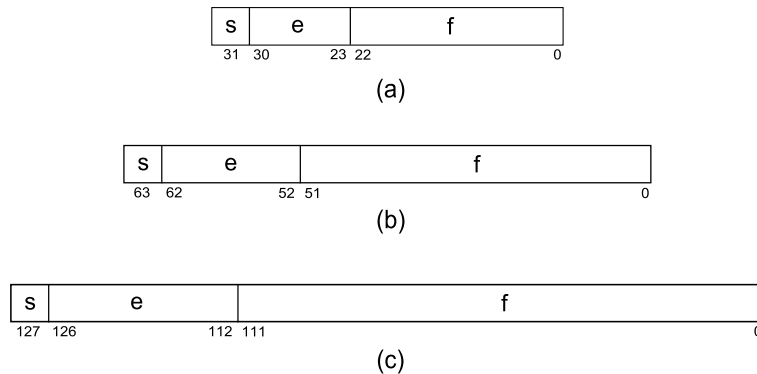
4.3 Τυπική μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης διπλής ακρίβειας

Στην παράγραφο αυτή επισημαίνονται τα κύρια στοιχεία που αναφέρονται στο πρότυπο κινητής υποδιαστολής της IEEE [53] σχετικά με την αναπαράσταση των αριθμών κινητής υποδιαστολής και τις διαφορετικές ακρίβειες που υπάρχουν, και περιγράφεται η δομή μιας τυπικής μονάδας Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης (MAF.)

Το πρότυπο IEEE 754 καθορίζει τρεις βασικές μορφές κινητής υποδιαστολής: την μονή, τη διπλή και την τετραπλή ακρίβεια. Σε κάθε περίπτωση η αναπαράσταση ενός αριθμού αποτελείται από τρία πεδία: το ψηφίο προσήμου S , τον πολωμένο εκθέτη E και τον συντελεστή F . Τα πεδία αυτά αποθηκεύονται σε μία λέξη μήκους 32, 64 και 128 bit για κάθε ακρίβεια αντίστοιχα. Στην εικόνα 4.1 φαίνεται η αναπαράσταση των πεδίων για τις τρεις διαφορετικές ακρίβειες, ενώ η τιμή X ενός κανονικοποιημένου αριθμού κινητής υποδιαστολής δίνεται από την εξίσωση:

$$X = (-1)^S \times 2^{E-bias} \times 1.F$$

Η πόλωση παίρνει την τιμή 127 για την αναπαράσταση μονής ακρίβειας, 1023 για τη διπλή ακρίβεια και 16383 για την τετραπλή ακρίβεια.



Σχήμα 4.1: (α) Μονή ακρίβεια, (β) Διπλή ακρίβεια (γ) Τετραπλή ακρίβεια.

Στη συνέχεια δίνεται μία σύντομη περιγραφή της δομής μιας τυπικής MAF αρχιτεκτονικής διπλής ακρίβειας (εικόνα 4.2), η οποία περιλαμβάνεται σε πολλούς επεξεργαστές γενικού τύπου [57], [58], [61], [62], [64]. Αν και από σχέδιο σε σχέδιο μπορούν να βρεθούν τοπικές βελτιώσεις και τροποποιήσεις σε διάφορα τμήματα, η βασική αρχιτεκτονική παραμένει η ίδια και μπορεί να χωριστεί σε τρία βασικά στάδια.

1. Στάδιο Πολλαπλασιασμού: Στο στάδιο αυτό πολλαπλασιάζονται οι συντελεστές των τελεστών A και B μήκους 53 bit, παράγοντας ένα αποτέλεσμα 106 bit σε μορφή διατήρησης κρατουμένου. Επίσης, πραγματοποιείται η ευθυγράμμιση του τρίτου τελεστέου, με τον C να τοποθετείται αρχικά 55 bit

αριστερά του γινομένου και να ολισθαίνει προς τα δεξιά ώστε να ευθυγραμμιστεί στο σωστό σημείο, σύμφωνα με τη διαφορά των εκθετών. Για το σκοπό αυτό χρησιμοποιείται ένας ολισθητής 161 bit.

Η διαφορά των εκθετών υπολογίζεται με βάση τη σχέση:

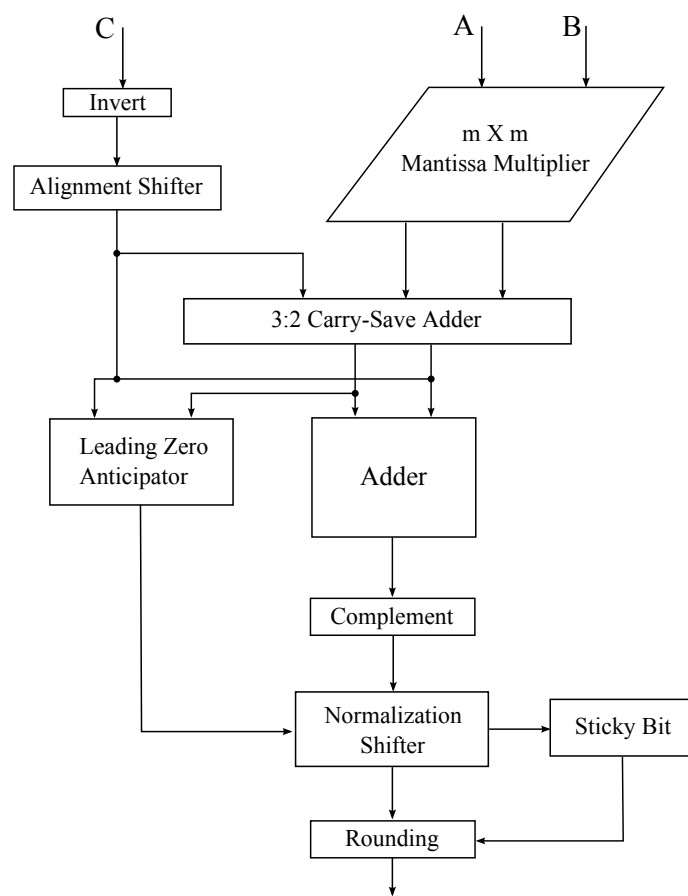
$$diff = expC - (expA + expB - 1023)$$

όπου $expA$, $expB$, $expC$ είναι οι πολωμένοι εκθέτες των τελεστών A , B , C αντίστοιχα και 1023 είναι η πόλωση για τη διπλή ακρίβεια. Έτσι, η απαραίτητη μετατόπιση ισούται με:

$$sh = 55 - diff \Rightarrow sh = expA + expB - expC - 968$$

2. Στάδιο Πρόσθεσης: Στο δεύτερο στάδιο προστίθεται το προϊόν του πολλαπλασιασμού $A \times B$ μήκους 106 bit, με τον μετατοπισμένο τελεστέο C μήκους 161 bit, χρησιμοποιώντας έναν συμπιεστή 3:2 CSA 106-bit και έναν αθροιστή 161 bit. Παράλληλα με την πρόσθεση, το κύκλωμα Πρόβλεψης Προπορευόμενων Μηδενικών (Leading Zero Anticipator ή LZA) επεξεργάζεται τα κάτω 106 bit του συμπιεστή 3:2 CSA και υπολογίζει το αναγκαίο ποσό μετατόπισης για την κανονικοποίηση.

3. Κανονικοποίηση και Στρογγυλοποίηση: Στο τελευταίο στάδιο πραγματοποιείται η κανονικοποίηση και η στρογγυλοποίηση για να παραχθεί το τελικό αποτέλεσμα. Η έξοδος του αθροιστή κανονικοποιείται με βάση την εκτίμηση της μονάδας LZA. Στη συνέχεια γίνεται η στρογγυλοποίηση, και εάν είναι απαραίτητο, μια τελευταία μέτα-κανονικοποίηση. Ταυτόχρονα, υπολογίζονται το πρόσημο και ο εκθέτης του τελικού αποτελέσματος.



Σχήμα 4.2: Μπλοκ διάγραμμα μιας τυπικής αρχιτεκτονικής MAF.

4.4 Πολλαπλασιαστής κινητής υποδιαστολής, τριπλής λειτουργίας

Ο πολλαπλασιασμός κινητής υποδιαστολής παίζει σημαντικό ρόλο σε πάρα πολλές μοντέρνες εφαρμογές όπως 3D επιταχυντές γραφικών, Ψηφιακή Επεξεργασία Σήματος, Υπολογιστικά Συστήματα Υψηλής Απόδοσης κλπ. Στοχεύοντας σε μία αποδοτική αρχιτεκτονική πολλαπλασιασμού κινητής υποδιαστολής, στην ενότητα αυτή παρουσιάζεται ένας πολλαπλασιαστής που μπορεί να διαχειριστεί και τις τρεις διαφορετικές ακρίβειες που υποστηρίζονται από το πρότυπο κινητής υποδιαστολής της IEEE. Το σχέδιο μπορεί να πραγματοποιεί έναν πολλαπλασιασμό κινητής υποδιαστολής τετραπλής ακρίβειας, ή δύο παράλληλους πολλαπλασιασμούς διπλής ακρίβειας, ή τέλος τέσσερις παράλληλους πολλαπλασιασμούς μονής ακρίβειας. Η αρχιτεκτονική βασίζει τη λειτουργικότητα της στην τεχνική *διαίρει και βασίλευε* και χρησιμοποιεί τέσσερα δέντρα πολλαπλασιαστών, μειωμένων διαστάσεων, για να πραγματοποιήσει τους υπολογισμούς κινητής υποδιαστολής στις τρεις διαφορετικές ακρίβειες. Το σχέδιο είναι σωληνομένο έτσι ώστε τέσσερις πολλαπλασιασμοί μονής ακρίβειας ή δύο πολλαπλασιασμοί διπλής ακρίβειας να μπορούν να ξεκινήσουν κάθε δύο κύκλους, ενώ ένας πολλαπλασιασμός τετραπλής ακρίβειας να μπορεί να ξεκινήσει κάθε τρεις κύκλους.

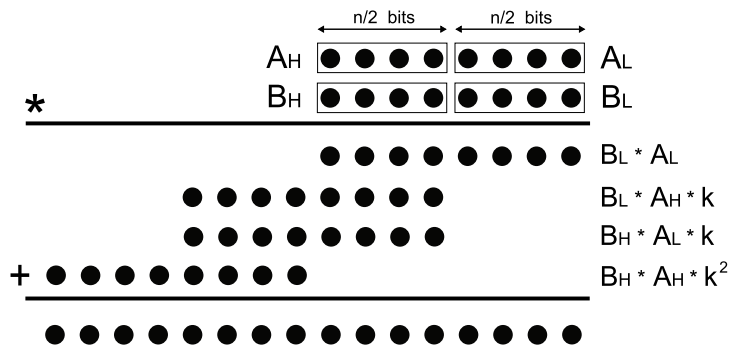
4.4.1 Αρχιτεκτονική

Ο πολλαπλασιαστής που παρουσιάζεται βασίζεται στην τεχνική *διαίρει και βασίλευε* (divide-and-conquer). Για το λόγο αυτό, κρίνεται αναγκαίο να γίνει μια σύντομη αναφορά στη τεχνική αυτή, πριν αρχίσει η ανάλυση της αρχιτεκτονικής του πολλαπλασιαστή. Η τεχνική *διαίρει και βασίλευε* χρησιμοποιείται, μεταξύ άλλων, για την υλοποίηση πολλαπλασιασμών υψηλής ακρίβειας, πραγματοποιώντας μια σειρά από πολλαπλασιασμούς μικρότερου μεγέθους και προσθέτοντας στη συνέχεια τα μερικά γινόμενα των πολλαπλασιασμών. Αυτό επιτυγχάνεται χωρίζοντας αρχικά τους αριθμούς που πολλαπλασιάζονται σε δύο μέρη. Έτσι, για τον πολλαπλασιασμό δύο n -bit αριθμών A και B , ο καθένας τους μπορεί να εκφραστεί ως $A = A_H \cdot k + A_L$ και $B = B_H \cdot k + B_L$, όπου A_H και B_H είναι το πάνω μέρος κάθε αριθμού, και A_L , B_L το κάτω μέρος κάθε αριθμού, ενώ $k = 2^{\frac{n}{2}}$. Με αυτόν τον τρόπο το γινόμενο των A και B μπορεί να εκφραστεί ως εξής:

$$\begin{aligned} A \cdot B &= (A_H \cdot k + A_L) \cdot (B_H \cdot k + B_L) \\ &= A_H \cdot B_H \cdot k^2 + (A_H \cdot B_L + A_L \cdot B_H) \cdot k + A_L \cdot B_L \end{aligned}$$

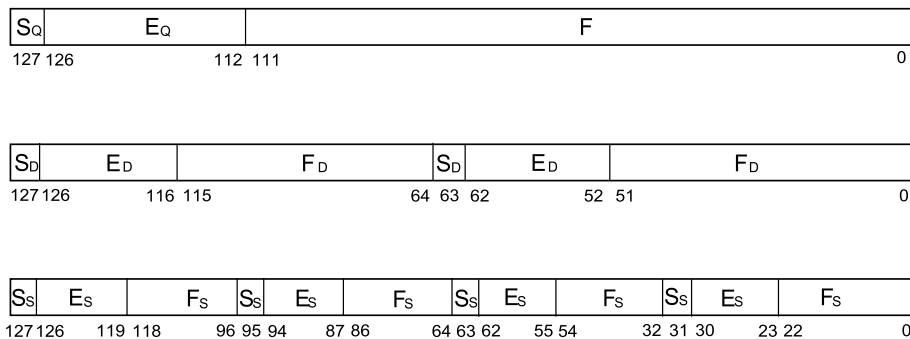
Η εικόνα 4.3 δείχνει τους υπολογισμούς της εξίσωσης 4.1 για $n = 8$ bit.

Όπως αναφέρθηκε και πιο πάνω ο πολλαπλασιαστής κινητής υποδιαστολής, τριπλής λειτουργίας, υποστηρίζει όλες τις μορφές ακρίβειας της IEEE, εκτελώντας έναν πολλαπλασιασμό τετραπλής ακρίβειας, ή δύο παράλληλους πολλαπλασιασμούς διπλής ακρίβειας, ή τέσσερις παράλληλους πολλαπλασιασμούς

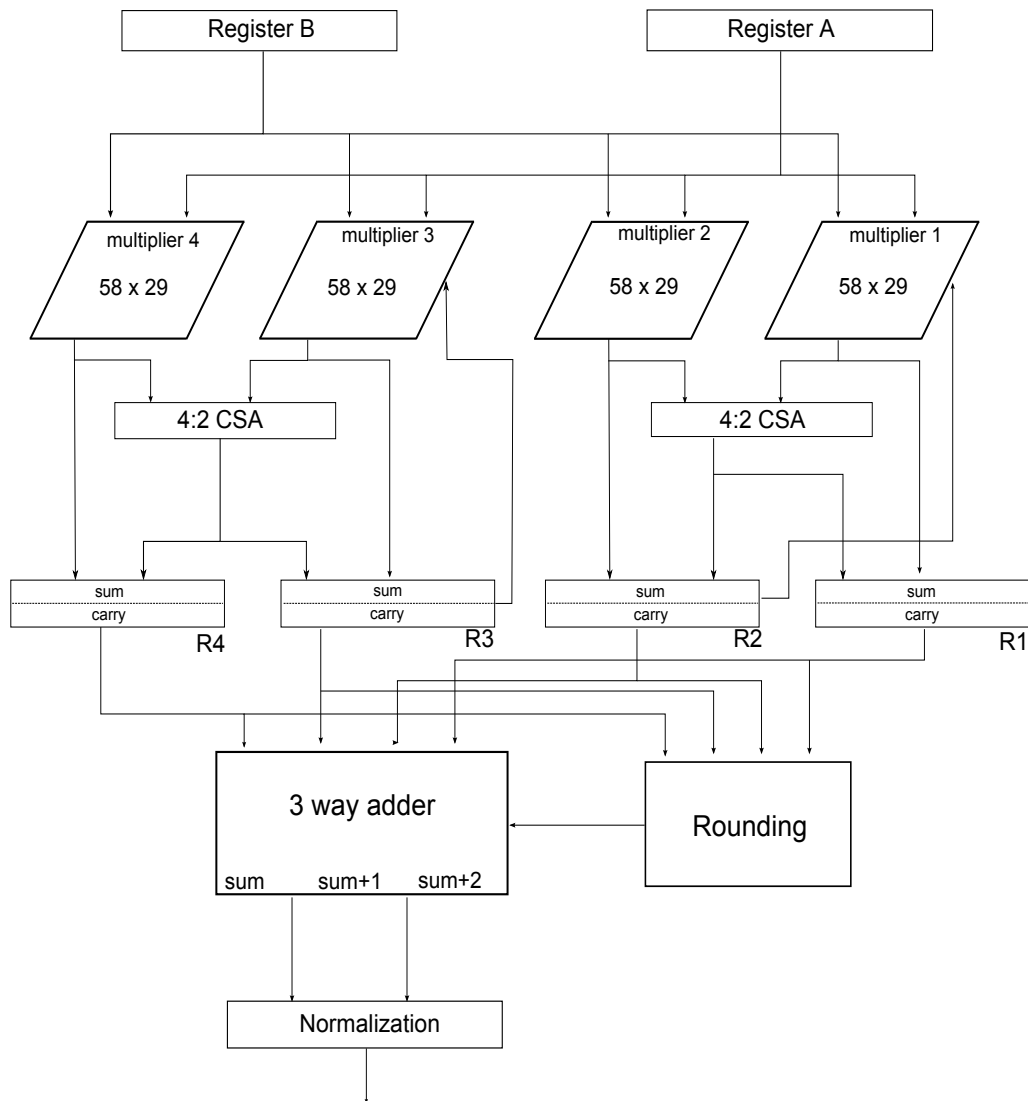


Σχήμα 4.3: Παρουσίαση της τεχνικής διαίρει και βασίλευε για λέξεις των 8-bit.

μονής ακρίβειας. Το σχέδιο χρησιμοποιεί δύο καταχωρητές των 128 bit για να αποθηκεύει τους διαφορετικής ακρίβειας τελεστέους. Στην εικόνα 4.4 φαίνεται πως οι διάφοροι τελεστέοι μπορούν να αποθηκευτούν χρησιμοποιώντας τους καταχωρητές των 128-bit. Ένα σήμα ελέγχου καθορίζει τον τρόπο λειτουργίας και χρησιμοποιείται κατάλληλα για να ελέγχει τους πολυπλέκτες του σχεδίου. Το μπλόκ διάγραμμα της αρχιτεκτονικής του πολλαπλασιαστή κινητής υποδιαστολής παρουσιάζεται στην εικόνα 4.5.



Σχήμα 4.4: Καταχώρηση τελεστέων τετραπλής, διπλής και μονής ακρίβειας σε καταχωρητή 128-bit.



Σχήμα 4.5: Μπλοκ διάγραμμα του πολλαπλασιαστή κινητής υποδιαστολής, τρι-
πλής λειτουργίας.

4.4.1.1 Πολλαπλασιασμός τετραπλής ακρίβειας

Για την πραγματοποίηση του πολλαπλασιασμού τετραπλής ακρίβειας τροποποιούμε τον αλγόριθμο *διαίρει και βασίλευε* που παρουσιάστηκε παραπάνω. Ο ένας από τους δύο πολλαπλασιαστέους διαιρείται σε δύο μέρη και ο άλλος σε τέσσερα. Έτσι, οι n -bit αριθμοί X και Y μπορούν να εκφραστούν σύμφωνα με τις παρακάτω εξισώσεις:

$$Y = Y_1 \cdot 2^{\frac{n}{2}} + Y_0$$

$$X = X_3 \cdot 2^{\frac{3n}{4}} + X_2 \cdot 2^{\frac{n}{2}} + X_1 \cdot 2^{\frac{n}{4}} + X_0$$

Επομένως, το γινόμενο $X \cdot Y$ υπολογίζεται σύμφωνα με τα ακόλουθα:

$$\begin{aligned} X \cdot Y = & Y_1 \cdot X_3 \cdot 2^{n+\frac{n}{4}} + Y_1 \cdot X_2 \cdot 2^n + \\ & + (Y_1 \cdot X_1 + Y_0 \cdot X_3) \cdot 2^{\frac{3n}{4}} + \\ & + (Y_1 \cdot X_0 + Y_0 \cdot X_2) \cdot 2^{\frac{n}{2}} + \\ & + Y_0 \cdot X_1 \cdot 2^{\frac{n}{4}} + Y_0 \cdot X_0 \end{aligned}$$

Εφαρμόζοντας τις εξισώσεις αυτές στην περίπτωση μας, τα X και Y συμβολίζουν τους συντελεστές (mantissas) των δύο αριθμών τετραπλής ακρίβειας. Από τη στιγμή που το κλασματικό μέρος (significant) ενός αριθμού κινητής υποδιαστολής τετραπλής ακρίβειας είναι 112 bit, λαμβάνοντας υπόψη το κρυφό 1, καθένα από τα X και Y είναι 113 bit. Ο Y διαιρείται σε δύο μέρη των 58 bit το καθένα και ο X διαιρείται σε τέσσερα μέρη των 29 bit. Για να επιτευχθεί ίσο μήκος μεταξύ των διαχωρισθέντων μερών των X και Y εφαρμόζεται συμπλήρωση με μηδενικά (zero padding). Οι λέξεις που προκύπτουν έχουν την εξής μορφή:

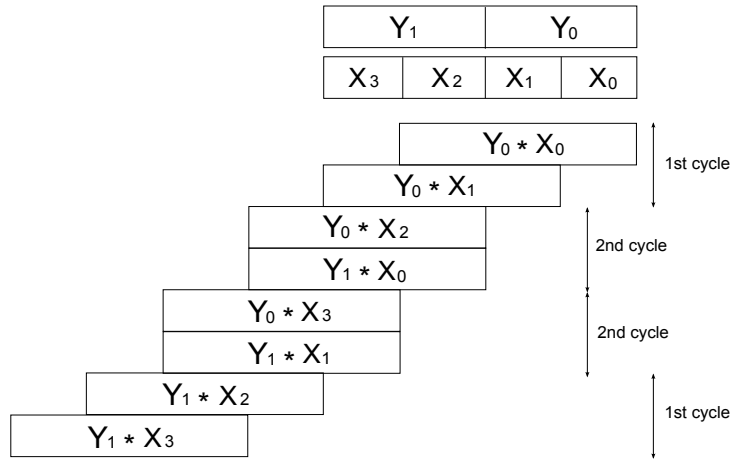
$$Y_1 = '1' \& [111 : 56] \& '0', Y_0 = [55 : 0] \& "00"$$

$$X_3 = '1' \& [111 : 84], X_2 = [83 : 56] \& '0',$$

$$X_1 = [55 : 28] \& '0', X_0 = [27 : 0] \& '0'.$$

Στην εικόνα 4.6 δείχνει ένα διάγραμμα που αναπαριστά τον πολλαπλασιασμό τετραπλής ακρίβειας.

Η αρχιτεκτονική που παρουσιάζεται πραγματοποιεί έναν πολλαπλασιασμό τετραπλής ακρίβειας σε τρεις κύκλους. Κατά τον πρώτο κύκλο, ο πολλαπλασιαστής 1 και ο πολλαπλασιαστής 2 (εικόνα 4.5) υπολογίζουν τα γινόμενα $Y_0 \cdot X_0$ και $Y_0 \cdot X_1$ αντίστοιχα. Παράλληλα, ο πολλαπλασιαστής 3 και ο πολλαπλασιαστής 4 υπολογίζουν τα γινόμενα $Y_1 \cdot X_2$ και $Y_1 \cdot X_3$. Όλοι οι πολλαπλασιαστές παράγουν τα αποτελέσματα τους σε μορφή διατήρησης κρατουμένου (carry save).



Σχήμα 4.6: Η τεχνική *Διαίρει και Βασίλευε* για τον πολλαπλασιασμό τετραπλής ακρίβειας.

Στη συνέχεια, χρησιμοποιώντας έναν συμπιεστή 4:2 CSA συνδυάζονται τα επικαλυπτόμενα μέρη των δύο πρώτων και των δύο επόμενων προϊόντων. Έτσι, στο τέλος του πρώτου κύκλου ο καταχωρητής $R1$ περιέχει τα 58 λιγότερο σημαντικά ψηφία (lsbits) του προϊόντος $Y_0 \cdot X_0$, συνδυασμένα με τα 29 λιγότερο σημαντικά ψηφία του γινομένου $Y_0 \cdot X_1$, ενώ τα υπόλοιπα ψηφία του $Y_0 \cdot X_1$ συνδυάζονται με τα 29 πιο σημαντικά ψηφία (msbits) του $Y_0 \cdot X_0$ και αποθηκεύονται στον καταχωρητή $R2$. Παρόμοια, το πάνω μέρος του καταχωρητή $R3$ περιέχει τα 58 λιγότερο σημαντικά ψηφία του προϊόντος $Y_1 \cdot X_2$ συνδυασμένα με τα 29 λιγότερο σημαντικά ψηφία του γινομένου $Y_1 \cdot X_3$, ενώ τα υπόλοιπα ψηφία του $Y_1 \cdot X_3$ συνδυάζονται με πάνω ψηφία του $Y_1 \cdot X_2$ και αποθηκεύονται στον καταχωρητή $R4$.

Στο δεύτερο κύκλο, ο πολλαπλασιαστής 1 και ο πολλαπλασιαστής 2 υπολογίζουν τα γινόμενα $Y_0 \cdot X_2$ και $Y_1 \cdot X_0$, ενώ ο πολλαπλασιαστής 3 και ο πολλαπλασιαστής 4 υπολογίζουν αντίστοιχα τα προϊόντα $Y_0 \cdot X_3$ και $Y_1 \cdot X_1$. Ταυτόχρονα, τα περιεχόμενα των καταχωρητών $R2$ και $R3$ οδηγούνται πίσω στους πολλαπλασιαστές 1 και 3 αντίστοιχα και αποθηκεύονται σε επιπρόσθετες γραμμές που διαθέτει ο κάθε πολλαπλασιαστής, ενώ οι καταχωρητές 1 και 4 εξακολουθούν να έχουν αποθηκευμένα τα αποτελέσματα που παρήχθησαν κατά τον πρώτο κύκλο. Όταν τα αποτελέσματα και των τεσσάρων πολλαπλασιαστών έχουν παραχθεί, συνδυάζονται ξανά με τους συμπιεστές 4:2 CSA και τα αποτελέσματα αποθηκεύονται ξανά στους καταχωρητές. Αυτήν τη φορά όμως, ο καταχωρητής 2 αποθηκεύει το αποτέλεσμα του συνδυασμού των προϊόντων $Y_0 \cdot X_2$ και $Y_1 \cdot X_0$, ο καταχωρητής 3 αποθηκεύει το αποτέλεσμα του συνδυασμού των $Y_0 \cdot X_3$ με το $Y_1 \cdot X_1$, ενώ οι καταχωρητές 1 και 4 εξακολουθούν να έχουν τα δεδομένα που δημιουργήθηκαν κατά τον πρώτο κύκλο.

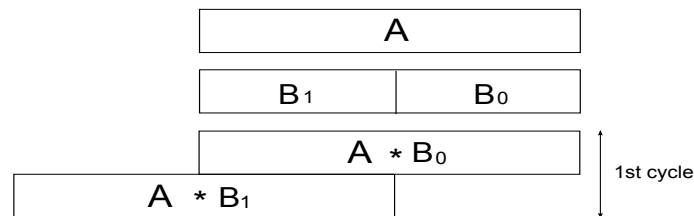
Τέλος, κατά τον τρίτο κύκλο η αρχιτεκτονική πραγματοποιεί την τελική

πρόσθεση, στρογγυλοποίηση και κανονικοποίηση [66], [72], [73]. Τα 114 πιο σημαντικά ψηφία επιλέγονται από τους καταχωρητές και προωθούνται σε έναν αθροιστή 3-δρόμων (3-way adder), ο οποίος υπολογίζει παράλληλα τα διανύσματα sum , $sum + 1$ και $sum + 2$. Τα υπόλοιπα ψηφία χρησιμοποιούνται από το δομοστοιχείο στρογγυλοποίησης για να υπολογιστούν τα ψηφία κρατουμένου, στρογγυλοποίησης και sticky. Το ψηφίο κρατουμένου επιλέγει δύο από τα τρία αποτελέσματα του αθροιστή, ενώ η στρογγυλοποίηση καθορίζει το τελικό αποτέλεσμα. Τέλος, εάν χρειάζεται το σχέδιο πραγματοποιεί κανονικοποίηση 1-bit.

4.4.1.2 Πολλαπλασιασμός διπλής ακρίβειας

Όταν το σχέδιο βρίσκεται σε κατάσταση λειτουργίας διπλής ακρίβειας πραγματοποιεί παράλληλα δύο πολλαπλασιασμούς κινητής υποδιαστολής, διπλής ακρίβειας. Στην περίπτωση αυτή ο κάθε καταχωρητής στην αρχή του πρώτου σταδίου έχει αποθηκευμένους 2 τελεστέους διπλής ακρίβειας.

Για να πραγματοποιηθούν παράλληλα οι δύο αυτοί πολλαπλασιασμοί χρησιμοποιούμε τους πολλαπλασιαστές 1 και 2 για να υπολογίσουμε το γινόμενο $A \cdot B$, ενώ παράλληλα οι πολλαπλασιαστές 3 και 4 υπολογίζουν το δεύτερο πολλαπλασιασμό διπλής ακρίβειας $C \cdot D$. Για κάθε ζευγάρι τελεστών, διατηρούμε τον ένα από τους δύο ανέπαφο και χωρίζουμε το δεύτερο σε δύο μέρη, το πάνω και το κάτω μέρος. Έτσι, ο κάθε πολλαπλασιασμός διπλής ακρίβειας υλοποιείται σύμφωνα με την εικόνα 4.7.



Σχήμα 4.7: Η τεχνική διαίρει και βασίλευε για τον πολλαπλασιασμό διπλής ακρίβειας.

Στη συνέχεια περιγράφονται τα βήματα υπολογισμού του πρώτου πολλαπλασιασμού διπλής ακρίβειας (η ίδια μέθοδος εφαρμόζεται επίσης και για το δεύτερο). Το σημαντικό μέρος του κάθε τελεστέου A και B είναι 52 bit. Προσθέτουμε στο σημαντικό μέρος του πρώτου τελεστέου το χυφό 1 και προεκτείνουμε με μηδενικά στα 58 bit, ενώ το δεύτερο σημαντικό μέρος χωρίζεται σε δύο μέρη των 27 bit, τα οποία συμπληρώνονται στο πάνω μέρος τους με μηδενικά για να φτάσουν τα 29 bit. Οι λέξεις που προκύπτουν έχουν την εξής μορφή:

$$A = "0000"&'1' \&[51 : 0]$$

$$B_1 = "00"&'1' \&[115 : 90], B_2 = "00"&[89 : 64] \&'0'$$

Κατά τον πρώτο κύκλο, ο πολλαπλασιαστής 1 υπολογίζει το γινόμενο $A \cdot B_0$ και ο πολλαπλασιαστής 2 το γινόμενο $A \cdot B_1$. Οι πολλαπλασιαστές παράγουν τα αποτελέσματα τους με διατήρηση κρατουμένου και τα επικαλυπτόμενα μέρη των δύο προϊόντων συνδυάζονται με το συμπιεστή 4:2 CSA. Τα λιγότερο σημαντικά ψηφία του $A \cdot B_0$ αποθηκεύονται στον καταχωρητή $R1$, ενώ ο καταχωρητής $R2$ αποθηκεύει το συνδυασμό του $A \cdot B_0$ με το πάνω μέρος του $A \cdot B_1$. Τέλος, στο δεύτερο κύκλο το σχέδιο πραγματοποιεί την πρόσθεση, στρογγυλοποίηση και κανονικοποίηση. Για την πραγματοποίηση της στρογγυλοποίησης, ο πρώτος πολλαπλασιασμός διπλής ακρίβειας χρησιμοποιεί τη λογική που χρησιμοποιήθηκε για την στρογγυλοποίηση του πολλαπλασιασμού τετραπλής ακρίβειας, ενώ για το δεύτερο πολλαπλασιασμό έχουμε προσθέσει στο κύκλωμα επιπλέον λογική ώστε να πραγματοποιηθεί η δεύτερη στρογγυλοποίηση.

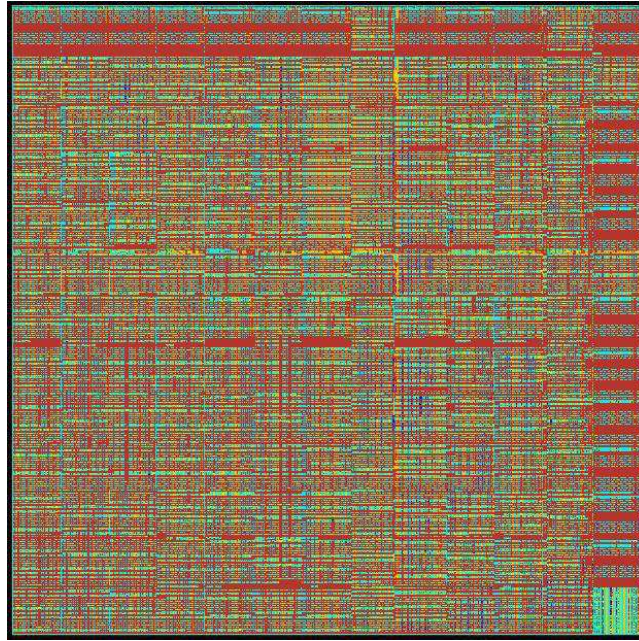
4.4.1.3 Πολλαπλασιασμός μονής ακρίβειας

Όταν το σχέδιο βρίσκεται σε κατάσταση λειτουργίας μονής ακρίβειας ο κάθε καταχωρητής στην αρχή του πρώτου σταδίου έχει αποθηκευμένους τέσσερις τελεστές μονής ακρίβειας. Για την παράλληλη πραγματοποίηση των τεσσάρων πολλαπλασιασμών μονής ακρίβειας, υπολογίζεται ένα γινόμενο σε κάθε πολλαπλασιαστή. Έτσι, οι πολλαπλασιαστές 1 και 2 υπολογίζουν τα γινόμενα των δύο πρώτων ζευγαριών τελεστών, που είναι αποθηκευμένοι στον πρώτο καταχωρητή, ενώ οι πολλαπλασιαστές 3 και 4 τα δύο ζευγάρια τελεστών που βρίσκονται στο δεύτερο καταχωρητή. Τα σταθερά μέρη των τελεστών μονής ακρίβειας είναι 24 bit (περιλαμβανομένου του κρυφού 1), έτσι για κάθε ζευγάρι τελεστών εφαρμόζεται προέκταση με μηδενικά κάνοντας το σταθερό μέρος του πρώτου τελεστή 58 bit και του δεύτερου 29 bit, ώστε να ταιριάζουν με τις διαστάσεις του κάθε πολλαπλασιαστή. Στο τέλος του πρώτου κύκλου τα τέσσερα γινόμενα έχουν παραχθεί σε μορφή διατήρησης κρατουμένου και έχουν αποθηκευτεί στους αντίστοιχους καταχωρητές παρακάμπτοντας τους συμπιεστές 4:2 CSA. Στο δεύτερο κύκλο τα ψηφία των μερικών προϊόντων του κάθε καταχωρητή προστίθενται και πραγματοποιείται η στρογγυλοποίηση. Όπως εξηγήθηκε και στα προηγούμενα, η λογική που εκτελεί τη στρογγυλοποίηση για την τετραπλή και τη διπλή ακρίβεια χρησιμοποιείται για να εξυπηρετήσει τους δύο πολλαπλασιασμούς μονής ακρίβειας, ενώ για τους άλλους δύο εισάγεται επιπλέον κύκλωμα.

4.4.2 Υλοποίηση της αρχιτεκτονικής και ανάλυση των αποτελεσμάτων

Ο τριπλής ακρίβειας, κινητής υποδιαστολής πολλαπλασιαστής σχεδιάστηκε και υλοποιήθηκε σε VHDL. Το σχέδιο συνθέθηκε σε τεχνολογία 45 nm χρησιμοποιώντας τα προγράμματα Synopsis και Cadence Encounter. Η υλοποίηση σε VLSI (εικόνα 4.8) πέτυχε σα μέγιστη συχνότητα λειτουργίας τα 505 MHz, καταλαμβάνοντας χώρο $624100 \mu m^2$ και καταναλώνοντας 458.08 mW.

Η αποτελεσματικότητα του πολλαπλασιαστή πολλαπλής ακρίβειας μπορεί να φανεί εξετάζοντας τις τυπικές λύσεις για τους πολλαπλασιαστές κινητής υποδιαστολής. Ένας τυπικός πολλαπλασιαστής τετραπλής ακρίβειας θα απαιτούσε έναν πολλαπλασιαστή τύπου μήτρας (array multiplier) 113 bit, ενώ η προσέγγισή μας απαιτεί τη μισή σχεδόν περιοχή, λαμβάνοντας υπ'όψιν το επιπλέον κόστος από τη χρήση πολυπλεκτών. Επιπλέον, ο πολλαπλασιαστής που παρουσιάζεται εκτελεί παράλληλα δύο πολλαπλασιασμούς διπλής ακρίβειας, κάτι που υπερτερεί της εναλλακτικής λύσης των δύο τυπικών πολλαπλασιαστών διπλής ακρίβειας, που να λειτουργούν ταυτόχρονα. Επίσης, υπερτερεί των τυπικών λύσεων και σε σχέση με το ρυθμό διαμεταγωγής: για να επιτευχθεί η ρυθμαπόδοση του πολλαπλασιαστή μας σε λειτουργία μονής ακρίβειας, χρειάζονται τέσσερις πολλαπλασιαστές μονής ακρίβειας που να λειτουργούν παράλληλα. Θα πρέπει να σημειωθεί, πως όλες οι παραπάνω τυπικές δομές πολλαπλασιαστών μπορούν να υποστηρίξουν μόνο μια ακρίβεια κινητής υποδιαστολής, σε αντίθεση με τις τρεις που υποστηρίζει ο πολλαπλασιαστής μας. Επομένως, αν και το σχέδιο που παρουσιάστηκε καταλαμβάνει λίγο παραπάνω χώρο σε σχέση με δύο πολλαπλασιαστές διπλής ακρίβειας, ή με τέσσερις πολλαπλασιαστές μονής ακρίβειας, εξακολουθεί να αποτελεί μια αποτελεσματική λύση για τις περιπτώσεις που απαιτείται υψηλή ρυθμαπόδοση για πράξεις πολλαπλασιασμού πολλαπλής ακρίβειας.



Σχήμα 4.8: Υλοποίηση σε VLSI του τριπλής ακρίβειας, κινητής υποδιαστολής πολλαπλασιαστή.

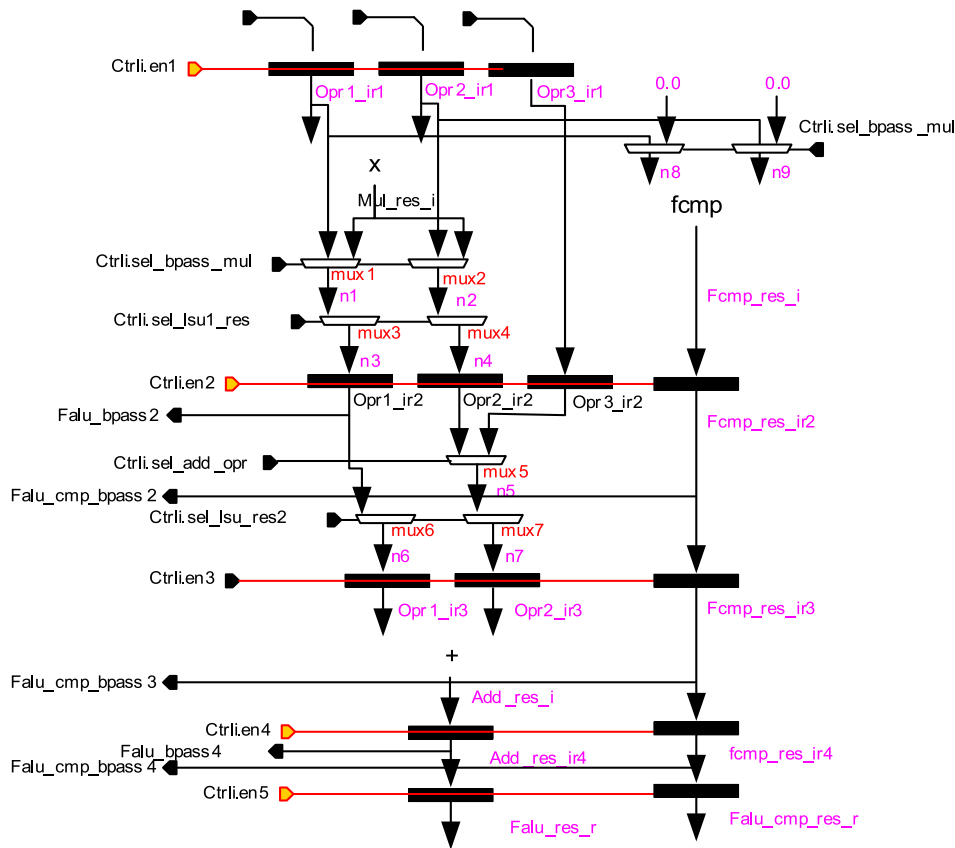
4.5 Μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης μεταβλητού μήκους σωλήνωσης

Η ενότητα αυτή παρουσιάζει μια μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης με μεταβλητό μήκος σωλήνωσης για έναν επεξεργαστή VLIW (Very Long Instruction Word). Η αρχιτεκτονική αυτή έχει το πλεονέκτημα να μπορεί να ολοκληρώνει την εντολή MAF σε τρεις κύκλους, ή να επεκτείνει το βάθος σωλήνωσης από τους τρεις στους πέντε κύκλους. Η δυνατότητα αυτή της διαμόρφωσης του μήκους σωλήνωσης επιτρέπει στη μονάδα MAF είτε να παράγει τις εξαιρέσεις (exceptions) σε τρεις μόνο κύκλους, όταν αυτό απαιτείται από τον προγραμματισμό του VLIW επεξεργαστή, είτε να συγχρονίζεται αποτελεσματικά με άλλες μονάδες του επεξεργαστή. Για να ικανοποιήσει τις απαιτήσεις του επεξεργαστή VLIW για αποτελέσματα υψηλής ακρίβειας, η μονάδα MAF υλοποιεί έναν παράλληλο πολλαπλασιαστή που ολοκληρώνει τους υπολογισμούς σε δύο κύκλους. Η μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης υποστηρίζει υπολογισμούς τόσο για κανονικοποιημένους, όσο και για μη κανονικοποιημένους αριθμούς και παρέχει όλες τις δυνατές μεθόδους στρογγυλοποίησης που περιγράφονται από το Πρότυπο για την Αριθμητική Κινητής Υποδιαστολής IEEE-754 [53].

4.5.1 Η αρχιτεκτονική MAF μεταβλητού μήκους σωλήνωσης

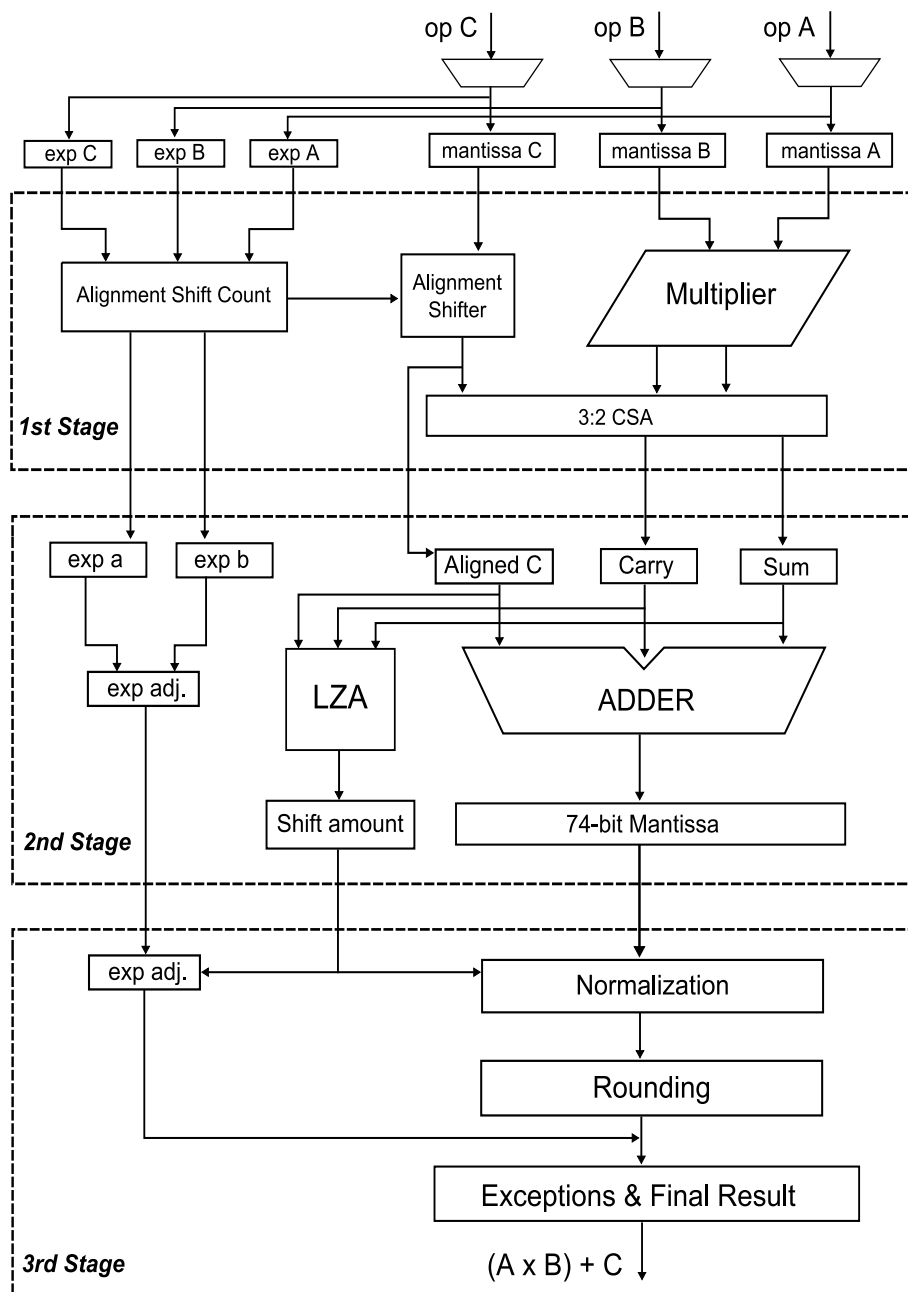
Η μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης που παρουσιάζεται σε αυτήν την ενότητα έχει ενσωματωθεί στον επεξεργαστή LE1 Multi-Cluster VLIW [74], [75], [76]. Ο LE1 είναι ένας διαμορφώσιμος, επεκτάσιμος, πολλαπλής συστάδας (multi-cluster) επεξεργαστής VLIW, φτιαγμένος με τέτοιο τρόπο που να επιτρέπει την πραγματοποίηση μετατροπών στην αρχιτεκτονική του (αρχεία καταχωρητών, μέγεθος, αριθμός, ικανότητα SIMD), και στη μικροαρχιτεκτονική (υποστήριξη εντολών ISA, μνήμες συνεχούς ροής κλπ.). Ο LE1 έχει φτιαχτεί ώστε να περιλαμβάνει έναν πυρήνα κινητής υποδιαστολής και αντίστοιχη κατάσταση στη CPU (αρχείο καταχωρητών κινητής υποδιαστολής, σημαίες κλπ.), στοχεύοντας κυρίως εφαρμογές με έντονες απαιτήσεις στην εκτέλεση των εντολών Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης. Μια σταθερά (που καθορίζεται κατά τη μεταγλώττιση) ορίζει το βάθος της σωλήνωσης της μονάδας κινητής υποδιαστολής σε 3 ή σε 5 στάδια. Μια εποπτική αναπαράσταση του μονοπατιού δεδομένων της μονάδας MAF φαίνεται στην εικόνα 4.9.

Η παραμετροποιήσιμη μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης εκτελεί τις εντολές κινητής υποδιαστολής χρησιμοποιώντας το μορφότυπο μονής ακρίβειας και μπορεί να χωριστεί σε τρία στάδια. Το μπλοκ διάγραμμα της αρχιτεκτονικής φαίνεται στην εικόνα 4.10. Στο στάδιο του Πολλαπλασιασμού



Σχήμα 4.9: Το μονοπάτι δεδομένων κινητής υποδιαστολής του επεξεργαστή LE1 VLIW

(1ο Στάδιο) παράγεται το γινόμενο $A \times B$, ενώ ταυτόχρονα λαμβάνει χώρα η ευθυγράμμιση και η αντιστροφή του τελεστή C . Το στάδιο της Πρόσθεσης (2ο Στάδιο) εκτελεί την πρόσθεση του γινομένου $A \times B$ με το ήδη ευθυγραμμισμένο σημαντικό μέρος του τελεστή C , ενώ παράλληλα, με τη βοήθεια του κυκλώματος Πρόβλεψης Προπορευόμενων Μηδενικών (LZA), καθορίζεται το αναγκαίο ποσό ολίσθησης. Το στάδιο της Κανονικοποίησης και Στρογγυλοποίησης (3ο Στάδιο) πραγματοποιεί τις αντίστοιχες λειτουργίες στο αποτέλεσμα της εξόδου του προηγούμενου σταδίου, ώστε να παραχθεί το τελικό αποτέλεσμα. Η μονάδα MAF μπορεί επίσης να αναγνωρίζει τους μη κανονικοποιημένους αριθμούς και να τους μετατρέπει σε κανονικοποιημένους, αποφεύγοντας έτσι την περαιτέρω αύξηση της πολυπλοκότητας σε υλισμικό. Αν το αποτέλεσμα είναι ένας μη κανονικοποιημένος αριθμός, το ποσό ολίσθησης προσαρμόζεται ανάλογα. Στις επόμενες παραγράφους περιγράφονται τα χαρακτηριστικά και οι λειτουργίες των τριών σταδίων της αρχιτεκτονικής.



Σχήμα 4.10: Μπλοκ διάγραμμα της αρχιτεκτονικής MAF.

4.5.1.1 Πολλαπλασιασμός και ευθυγράμμιση

Στο πρώτο στάδιο της αρχιτεκτονικής τα σημαντικά μέρη των τελεστέων A και B πολλαπλασιάζονται, ενώ παράλληλα πραγματοποιείται η ευθυγράμμιση, μειώνοντας έτσι τη συνολική καθυστέρηση. Η ευθυγράμμιση τοποθετεί το σημαντικό μέρος του τελεστέου C αριστερά από το πιο σημαντικό ψηφίο του γινομένου $A \times B$. Η ευθυγράμμιση πραγματοποιείται σαν μια πράξη ολίσθησης προς τα δεξιά. Το αναγκαίο ποσό ολίσθησης είναι, $shift = 27 - d$ όπου d είναι η διαφορά ανάμεσα στους εκθέτες:

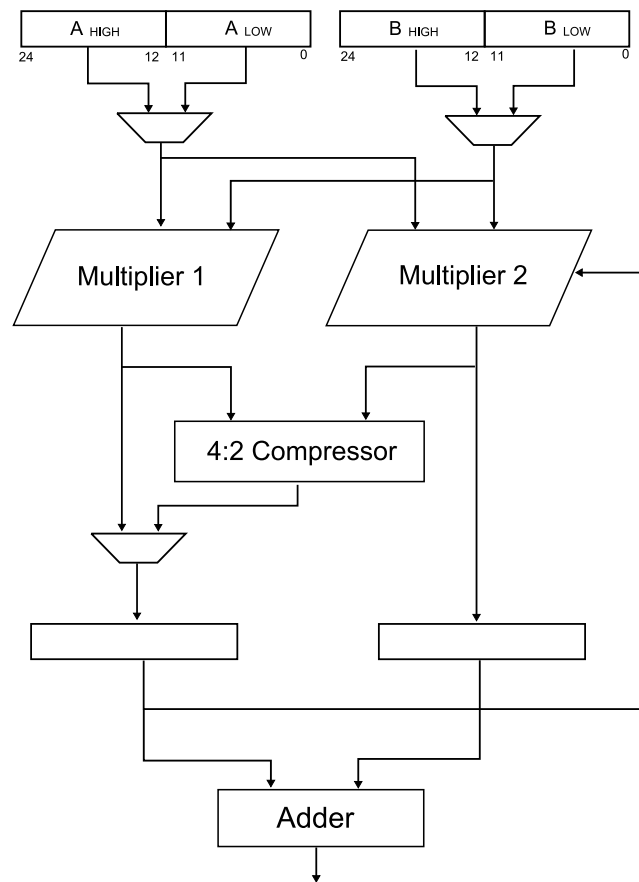
$$d = exp(C) - (exp(A) + exp(B) - 127)$$

και $exp(A)$, $exp(B)$, $exp(C)$ οι εκθέτες των τελεστέων A , B και C αντίστοιχα.

Ο πολλαπλασιαστής που χρησιμοποιεί η μονάδα MAF στηρίζεται στην τεχνική διαίρει και βασίλευε και έχει ανάλογη λειτουργικότητα με αυτήν του πολλαπλασιαστή που παρουσιάστηκε στην προηγούμενη ενότητα. Στη συγκεκριμένη περίπτωση το σημαντικό μέρος των τελεστέων A και B είναι 24 bit (μαζί με το κρυφό 1) και για την πραγματοποίηση του πολλαπλασιασμού χρησιμοποιούνται δύο παράλληλοι πολλαπλασιαστές τύπου μήτρας 12×12 bit. Με τρόπο παρόμοιο με αυτόν που αναλύθηκε στα προηγούμενα, κατά τον πρώτο κύκλο οι πολλαπλασιαστές 1 και 2 υπολογίζουν αντίστοιχα τα γινόμενα $A_L \cdot B_L$ και $A_L \cdot B_H$. Τα επικαλυπτόμενα μέρη των γινομένων αυτών συνδυάζονται με τον συμπίεστή 4:2 και όλα μαζί αποθηκεύονται στους δύο καταχωρητές. Στο δεύτερο κύκλο οι πολλαπλασιαστές 1 και 2 υπολογίζουν τα γινόμενα $A_H \cdot B_H$ και $A_H \cdot B_L$ ενώ τα πάνω 12 bit που είναι αποθηκευμένα στον καταχωρητή 1 επανατροφοδοτούνται στον πολλαπλασιαστή 2. Τα δύο νέα υποπροϊόντα προστίθενται για προκύψει το τελικό αποτέλεσμα. Το διάγραμμα του πολλαπλασιαστή φαίνεται στην εικόνα 4.11.

4.5.1.2 Πρόσθεση

Το δομοστοιχείο άθροισης/αφαίρεσης αποτελείται από έναν αθροιστή 27-bit, που υπολογίζει το άθροισμα του τελεστέου C με το γινόμενο $A \times B$. Ένα εξωτερικό σήμα καθορίζει στην αρχή κάθε εντολής αν θα πραγματοποιηθεί πρόσθεση ή αφαίρεση. Η μονάδα Ελέγχου Προσήμου παράγει το πρόσημο του τελικού αποτελέσματος, βάσει των ψηφίων προσήμων των δύο όρων και του είδους της πράξης που εκτελείται.



Σχήμα 4.11: Μπλοκ διάγραμμα του πολλαπλασιαστή που υλοποιείται στη μονάδα MAF μεταβλητού μήκους σωλήνωσης.

4.5.1.3 Πρόβλεψη Προπορευόμενων Μηδενικών (LZA)

Η μονάδα Πρόβλεψης Προπορευόμενων Μηδενικών *LZA* λειτουργεί παράλληλα με τον αθροιστή και υπολογίζει τον αριθμό των οδηγών μηδενικών. Προβλέπει, κατευθείαν από τους τελεστές, τον αριθμό των μηδενικών ψηφίων που βρίσκονται ανάμεσα στο πιο σημαντικό ψηφίο (την αριστερότερη δηλαδή θέση) και μέχρι να βρεθεί το πρώτο μη μηδενικό ψηφίο. Με βάση αυτήν την πρόβλεψη των οδηγών μηδενικών, καθορίζεται το απαραίτητο ποσό ολίσθησης προς τα αριστερά, για να κανονικοποιηθεί το αποτέλεσμα της πρόσθεσης. Με τον τρόπο αυτόν αυξάνεται η απόδοση της αρχιτεκτονικής, αφού η κανονικοποίηση ξεκινάει αμέσως μετά το αποτέλεσμα του σταδίου της πρόσθεσης, αφού ταυτόχρονα έχει εκτιμηθεί και το κατά πόσο χρειάζεται να γίνει ολίσθηση προς τα αριστερά. Από την άλλη πλευρά, η τεχνική αυτή μπορεί να οδηγήσει σε αβεβαιότητα 1 ψηφίου, στην περίπτωση ύπαρξης κρατουμένου. Για να αποφευχθεί

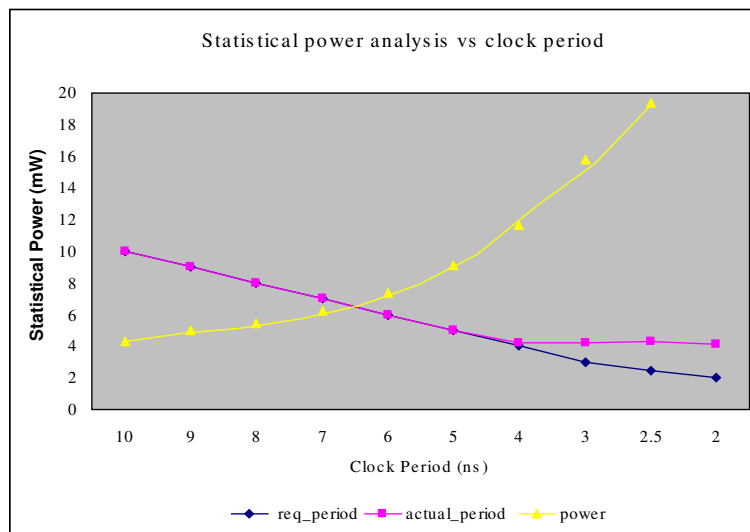
το λάθος αυτό, χωρίς να επιβαρυνθεί η συνολική καθυστέρηση, η μονάδα MAF υλοποιεί την τεχνική ενιαίας διάταξης ψηφίου που περιγράφεται στο [79]. Η τεχνική αυτή προβλέπει τον ακριβή αριθμό των οδηγών μηδενικών χωρίς της ανάγκη περαιτέρω διόρθωσης.

4.5.1.4 Κανονικοποίηση και στρογγυλοποίηση

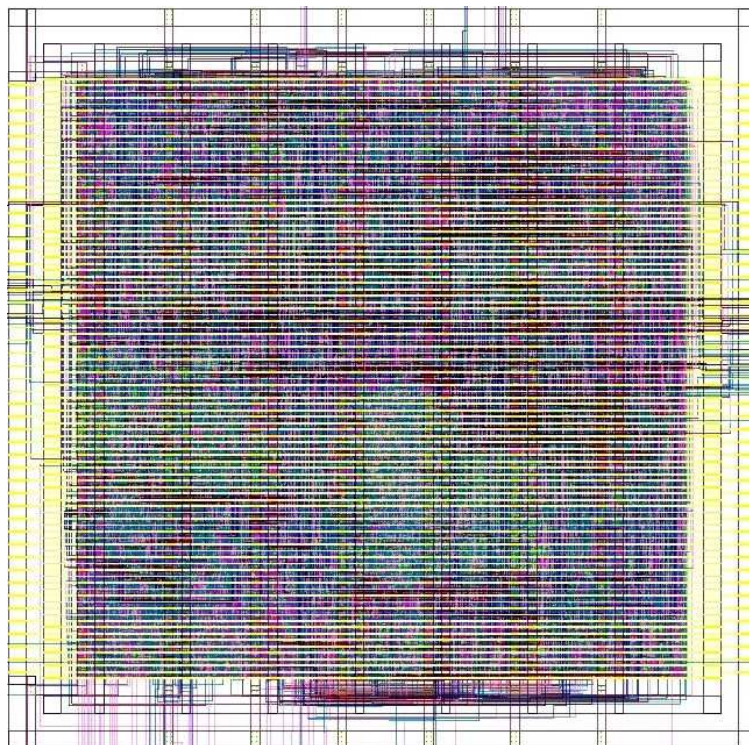
Η κανονικοποίηση ξεκινάει αμέσως αφού έχει παραχθεί το αποτέλεσμα της πρόσθεσης. Χρειάζεται έναν ολισθητή πλήρους μήκους και όταν η πράξη της κανονικοποίησης ολοκληρωθεί πραγματοποιείται η στρογγυλοποίηση. Για να βελτιωθεί η αποτελεσματικότητα της μονάδας MAF, η αρχιτεκτονική υποστηρίζει και τους τέσσερις τρόπους στρογγυλοποίησης που παρουσιάζονται στο Πρότυπο IEEE-754. Ένα σήμα επιλογής επιτρέπει στην αρχή της εκτέλεσης της κάθε εντολής, να καθορίζεται αν θα πραγματοποιηθεί στρογγυλοποίηση προς το $+\infty$, $-\infty$, 0, ή ως προς τον κοντινότερο αριθμό. Για να διασφαλιστεί η ακρίβεια των αποτελεσμάτων έχουν προστεθεί στο δεκαδικό μέρος του αριθμού τρία επιπλέον ψηφία, ως τα λιγότερο σημαντικά. Τα ψηφία αυτά αντιπροσωπεύουν τα ψηφία *guard*, *round* και *sticky* αντίστοιχα. Το πραγματικά λιγότερο σημαντικό ψηφίο του αριθμού χρειάζεται επίσης, για να πραγματοποιηθεί η πράξη της στρογγυλοποίησης. Η δυνατότητα να επιλέξουμε ανάμεσα στις διάφορες μεθόδους στρογγυλοποίησης ενισχύει την ακρίβεια των αποτελεσμάτων και βελτιώνει την απόδοση της μονάδας MAF.

4.5.2 Αποτελέσματα Υλοποίησης

Η μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης μεταβλητού μήκους σωλήνωσης υλοποιήθηκε στην τεχνολογία TSMC 0.13 1P8M. Η σύνθεση έγινε για μεταβλητό κύκλο ρολογιού, από 10 ns μέχρι 2 ns, και καταγράφηκαν τα αποτελέσματα της περιόδου του ρολογιού που επιτεύχθηκε καθώς και της στατιστικής ενεργειακής ανάλυσης. Η εικόνα 4.12 απεικονίζει τα δεδομένα από τη διαδικασία της σύνθεσης. Η αρχιτεκτονική πέτυχε σαν κύκλο ρολογιού τα 4.22 ns (236.9MHz) προ-δρομολόγησης, καταναλώνοντας 11.7 mW. Στη συνέχεια πραγματοποιήθηκε τοποθέτηση (placement) και δρομολόγηση (routing) με τη χρήση του εργαλείου Cadence Encounter για την ίδια βιβλιοθήκη. Η υλοποίηση σε VLSI (εικόνα 4.13) είχε σαν ελάχιστη περίοδο ρολογιού τα 4.43 ns (232.6MHz) και τελικό χώρο (δρομολογημένο) 121900.478 μm^2 .



Σχήμα 4.12: Αποτελέσματα της διαδικασίας σύνθεσης.



Σχήμα 4.13: Υλοποίηση της μονάδας MAF μεταβλητού μήκους σωλήνωσης σε VLSI.

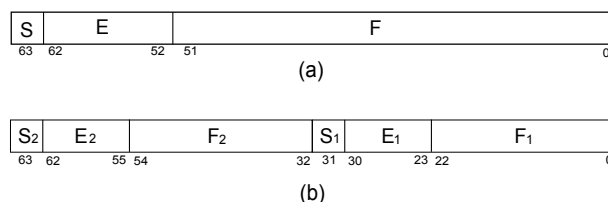
4.6 Μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης διπλής λειτουργίας

Στην ενότητα αυτήν παρουσιάζεται μία μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης διπλής λειτουργίας, που πραγματοποιεί είτε μία πράξη διπλής ακρίβειας, είτε δύο παράλληλες πράξεις μονής ακρίβειας. Το σχέδιο εμφανίζει βελτιωμένη απόδοση, ελαττώνοντας την καθυστέρηση εκτέλεσης της πρόσθεσης κινητής υποδιαστολής. Αυτό πραγματοποιείται παρακάμπτοντας τον πολλαπλασιαστή, κάθε φορά που εκτελείται πρόσθεση κινητής υποδιαστολής. Για να επιτευχθεί αυτό, αφαιρείται από το πρώτο στάδιο ο ολισθητής ευθυγράμμισης και υλοποιείται μια οργάνωση διπλού μονοπατιού. Επιπλέον, η κανονικοποίηση πραγματοποιείται πριν από τη στρογγυλοποίηση, ώστε η τελευταία να συνδυαστεί μαζί με την τελική πρόσθεση, μειώνοντας έτσι περισσότερο την καθυστέρηση.

Στόχος της αρχιτεκτονικής αυτής είναι να παρουσιάσει μια μονάδα MAF ικανή να διαχειρίζεται αριθμούς μονής και διπλής ακρίβειας, να εκτιμηθεί η απόδοση της και να διευκρινιστεί το κόστος της σε υλισμικό.

4.6.1 Η αρχιτεκτονική της μονάδας MAF διπλής λειτουργίας

Η μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης υποστηρίζει δύο διαφορετικές ακρίβειες του προτύπου IEEE-754 και πραγματοποιεί μία πράξη MAF διπλής ακρίβειας ή δύο παράλληλες πράξεις μονής ακρίβειας. Η εικόνα 4.14 δείχνει πώς ένας καταχωρητής 64-bit μπορεί να αποθηκεύσει έναν τελεστέο διπλής ακρίβειας ή δύο τελεστέους μονής ακρίβειας.

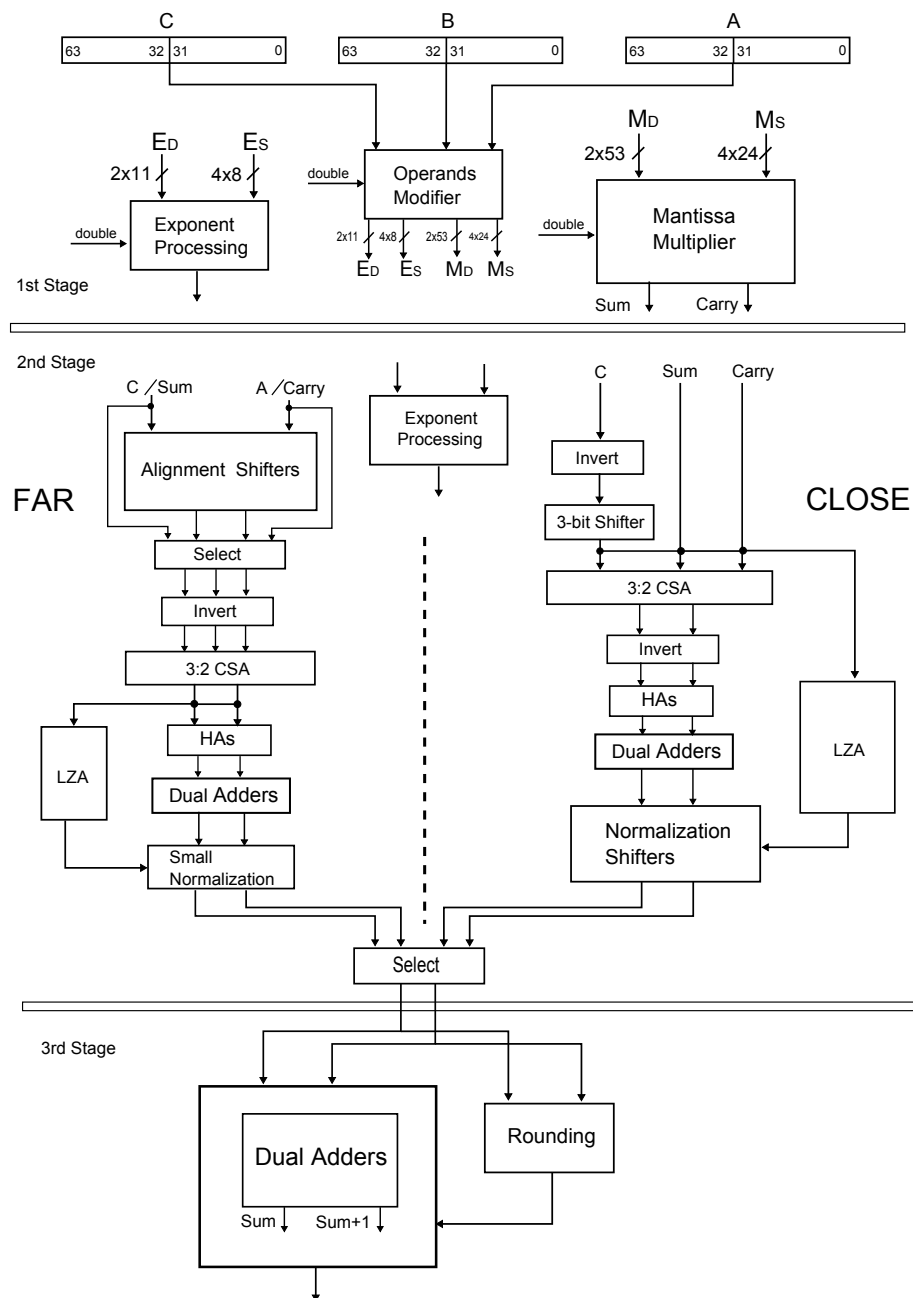


Σχήμα 4.14: (α) Αποθήκευση ενός τελεστέου διπλής ακρίβειας, (β) Αποθήκευση δύο τελεστέων μονής ακρίβειας.

Το σχέδιο της αρχιτεκτονικής MAF διπλής λειτουργίας απεικονίζεται στην εικόνα 4.15. Το σχέδιο είναι σωληνομένο σε τρία στάδια και ακολουθεί μια προσέγγιση διπλού μονοπατιού [59], ελαττώνοντας έτσι την καθυστέρηση των εντολών που αφορούν αυτόνομες πράξεις πρόσθεσης κινητής υποδιαστολής. Η οργάνωση διπλού μονοπατιού, παρουσιάστηκε αρχικά σαν μία αποτελεσματική τεχνική σχεδιασμού αθροιστών κινητής υποδιαστολής, αλλά έχει χρησιμοποιηθεί και στο σχεδιασμό μονάδων MAF [59], [79]. Το σχήμα διπλού μονοπατιού

διαίρει το μονοπάτι δεδομένων κατά το στάδιο της πρόσθεσης σε δύο διαφορετικά μονοπάτια: το *Κοντινό* μονοπάτι (*Close path*) και το *Μακρινό* μονοπάτι (*Far path*). Το *Κοντινό* μονοπάτι διαχειρίζεται τις περιπτώσεις όπου λαμβάνει χώρα μαζική ακύρωση, ενώ το *Μακρινό* μονοπάτι διαχειρίζεται τις υπόλοιπες περιπτώσεις. Η ιδέα είναι να ελαττωθεί η καθυστέρηση της εντολής FP ADD, παρακάμπτοντας τον πολλαπλασιαστή του πρώτου σταδίου, τοποθετώντας τον ολισθητή ευθυγράμμισης στο δεύτερο στάδιο. Η αρχιτεκτονική αποφεύγει τη χρήση δύο ολισθητών μετατόπισης (έναν σε κάθε μονοπάτι) εκμεταλεuόμενη το γεγονός ότι μια πλήρους μήκους ολίσθηση ευθυγράμμισης και μια μερικής μήκους ολίσθηση κανονικοποίησης, είναι δύο αμοιβαία αποκλειόμενες πράξεις και επομένως, κάθε μονοπάτι χρειάζεται μόνο έναν πλήρους μήκους ολισθητή. Έτσι, το *Μακρινό* μονοπάτι απαιτεί μόνο έναν μεγάλο ολίσθηση ευθυγράμμισης (και έναν πολύ μικρότερο ολίσθηση κανονικοποίησης), ενώ το *Κοντινό* μονοπάτι χρειάζεται έναν μεγάλο ολίσθηση κανονικοποίησης (και ένα μικρό ολίσθηση ευθυγράμμισης). Αποφεύγοντας το σχήμα των δύο μεγάλων ολισθητών σε κάθε μονοπάτι οδηγούμαστε σε μείωση της κρίσιμης καθυστέρησης.

Η αρχιτεκτονική διπλής λειτουργίας βασίζεται στη δομή μίας μονάδας MAF διπλού μονοπατιού και διπλής ακρίβειας, η οποία όμως έχει ενισχυθεί για την πραγματοποίηση και δύο παράλληλων εντολών μονής ακρίβειας. Για το σκοπό αυτό διάφορα μέρη του κυκλώματος έχουν επανασχεδιαστεί και επιπλέον υλισμικό έχει προστεθεί. Οι επιπλέον αυτές απαιτήσεις σε υλισμικό αυξάνουν μεν το χώρο, εξακολουθούν όμως δε, να προσφέρουν καλύτερα αποτελέσματα σε σχέση με τον απλό διπλασιασμό των διαφόρων τμημάτων της αρχιτεκτονικής. Ένας αριθμός από πολυπλέκτες μέσα στο σχέδιο διαχειρίζεται τη ροή των διαφορετικής ακρίβειας διανυσμάτων. Ένα σήμα ελέγχου, που μοιράζεται στους πολυπλέκτες και στα διάφορα άλλα δομοστοιχεία του κυκλώματος, καθορίζει το αν θα υπάρχει λειτουργία μονής ή διπλής ακρίβειας.



Σχήμα 4.15: Μπλοκ διάγραμμα της αρχιτεκτονικής MAF διπλής ακρίβειας.

4.6.1.1 Πρώτο Στάδιο: Επεξεργασία εκθετών και Πολλαπλασιασμός

1. *Επεξεργασία Εκθετών* : Η επεξεργασία των εκθετών καθορίζει τις πράξεις ολίσθησης για την ευθυγράμμιση και την κανονικοποίηση. Όταν το κύκλωμα βρίσκεται σε λειτουργία διπλής ακρίβειας, η διαφορά των εκθετών ισούται με

$$diff_{double} = expC - (expA + expB - 1023)$$

και το ποσό ολίσθησης είναι:

$$sh_{double} = 56 - d = expA + expB - expC - 967$$

Αντίστοιχα, στη λειτουργία μονής ακρίβειας έχουμε:

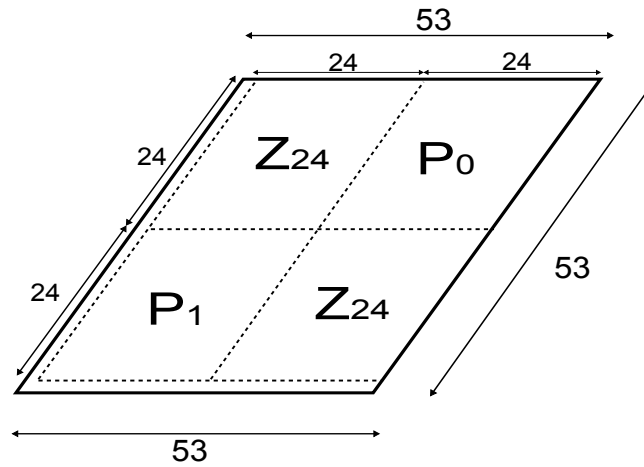
$$diff_{single} = expC - (expA + expB - 127)$$

$$sh_{single} = 27 - d = expA + expB - expC - 100$$

Το υλισμικό που διαχειρίζεται τους εκθέτες διπλής ακρίβειας έχει επεκταθεί ώστε να μπορεί να διαχειρίζεται και τους εκθέτες μονής ακρίβειας.

2. *Πολλαπλασιασμός* : Ο πολλαπλασιαστής μήτρας που χρησιμοποιεί η μονάδα MAF εκτελεί έναν πολλαπλασιασμό διπλής ακρίβειας ή δύο παράλληλους πολλαπλασιασμούς μονής ακρίβειας, υλοποιώντας την τεχνική διαμερισμού σε υπο-λέξεις [78]. Με την τεχνική αυτή η μονάδα MAF μπορεί να πραγματοποιεί και τον κλασσικό πολλαπλασιασμό καθώς και τον παράλληλο πολλαπλασιασμό υπο-λέξεων χρησιμοποιώντας το ίδιο δέντρο μετρητών και τον ίδιο αθροιστή μετάδοσης κρατουμένων.

Όταν πραγματοποιείται πολλαπλασιασμός διπλής ακρίβειας, ο πολλαπλασιαστής λειτουργεί σαν ένας απλός πολλαπλασιαστής μήτρας 53×53 bit. Όταν όμως γίνεται πολλαπλασιασμός μονής ακρίβειας, τα ψηφία των μερικών προϊόντων που βρίσκονται στις περιοχές Z_{24} (εικόνα 4.16) δεν αποτελούν μέρος του πολλαπλασιασμού των υπο-λέξεων, και έτσι γίνονται 0 για να αποφευχθεί αλλοίωση του τελικού αποτελέσματος. Με τον τρόπο αυτό, μπορούν να πραγματοποιηθούν δύο παράλληλοι πολλαπλασιασμοί 24×24 bit.



Σχήμα 4.16: Πολλαπλασιαστής διπλής λειτουργίας.

4.6.1.2 Δεύτερο Στάδιο: Οργάνωση Διπλού Μονοπατιού

Το δεύτερο στάδιο της αρχιτεκτονικής πραγματοποιεί ευθυγράμμιση και κανονικοποίηση. Όπως προαναφέρθηκε το μονοπάτι δεδομένων χωρίζεται σε δύο διαφορετικά μονοπάτια, το *Κοντινό* και το *Μακρινό*. Το *Κοντινό* μονοπάτι διαχειρίζεται τις περιπτώσεις όπου πραγματοποιείται πράξη αφαίρεσης ή πολλαπλασιασμού-αφαίρεσης, με τη διαφορά των εκθετών να βρίσκεται ανάμεσα στο ± 1 ($-1 \leq diff \leq +1$). Το *Μακρινό* μονοπάτι χρησιμοποιείται για όσες εντολές έχουν εκθέτες με διαφορά μεγαλύτερη του 1 ($diff > 1$).

Στο *Κοντινό* μονοπάτι ο τελεστής C αρχικά αντιστρέφεται και τοποθετείται 2 ψηφία αριστερά από το πιο σημαντικό ψηφίο του γινομένου, ώστε να γίνει η ευθυγράμμιση, που πραγματοποιείται με ένα ολισθητή 3 bit. Στη συνέχεια τα τρία διανύσματα εισέρχονται σε έναν συμπιεστή 3:2 CSA, και στην περίπτωση που εκτιμηθεί ότι υπάρχει αρνητικό αποτέλεσμα, οι έξοδοι του συμπιεστή αντιστρέφονται. Μια σειρά από ημιαθροιστές χρησιμοποιείται για να γίνει σωστά η στρογγυλοποίηση [73], ενώ ένα μέρος της τελικής πρόσθεσης γίνεται παράλληλα με τη λειτουργία του κυκλώματος LZA, ώστε να μειωθεί η καθυστέρηση [59]. Τέλος, πραγματοποιείται η κανονικοποίηση, η οποία στο *Κοντινό* μονοπάτι χρειάζεται ολισθητές πλήρους μήκους (ένα για το διάνυσμα άθροισματος και ένα για το διάνυσμα κρατουμένου).

Το *Μακρινό* μονοπάτι χρησιμοποιεί για την ευθυγράμμιση ολισθητές πλήρους μήκους. Για τη διπλή ακρίβεια το μέγιστο ποσό ολίσθησης είναι 106 bit για τον τελεστή C και 53 bit για το γινόμενο $A \times B$, ενώ για τη μονή ακρίβεια έχουμε αντίστοιχα 48 bit και 23 bit. Και στις δύο περιπτώσεις τα ψηφία που ολισθαίνουν εκτός ορίων χρησιμοποιούνται για τον υπολογισμό του ψηφίου sticky. Μετά την ολίσθηση και βάσει της διαφοράς των εκθετών, ένα από τα διανύσματα $A \times B$ ή C αντιστρέφεται ώστε το αποτέλεσμα να είναι πάντα

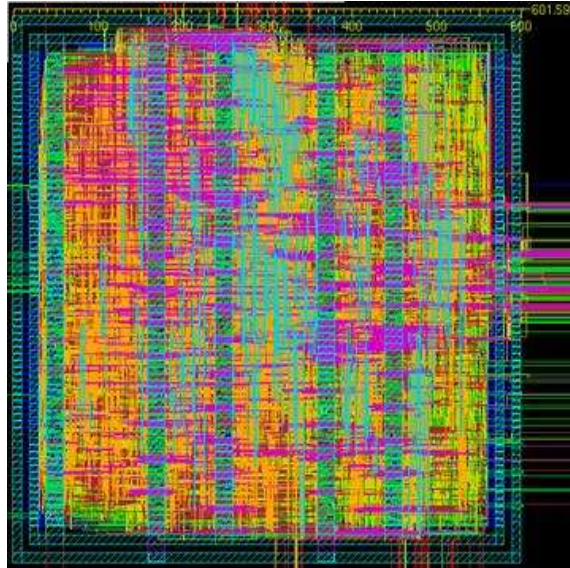
θετικό. Όπως και στο *Κοντινό* μονοπάτι, έτσι και εδώ τα διανύσματα περνάνε από μια σειρά ημιαθροιστών, ενώ το *Μακρινό* μονοπάτι ολοκληρώνεται με την πραγματοποίηση της κανονικοποίησης, η οποία με τη σειρά της χρειάζεται μόνο έναν ολισθητή 3 bit. Και στα δύο μονοπάτια, η αναπαράσταση των διανυσμάτων γίνεται με 2-Συμπλήρωμα. Τα απαραίτητα '1' που χρειάζονται μετά από κάθε αντιστροφή για να ολοκληρωθεί η αναπαράσταση 2-Σ τοποθετούνται σε μια κενή θέση του συμπιεστή, ή της σειράς των ημιαθροιστών.

4.6.1.3 Τρίτο Στάδιο: Συνδυασμός τελικής πρόσθεσης με τη στρογγυλοποίηση

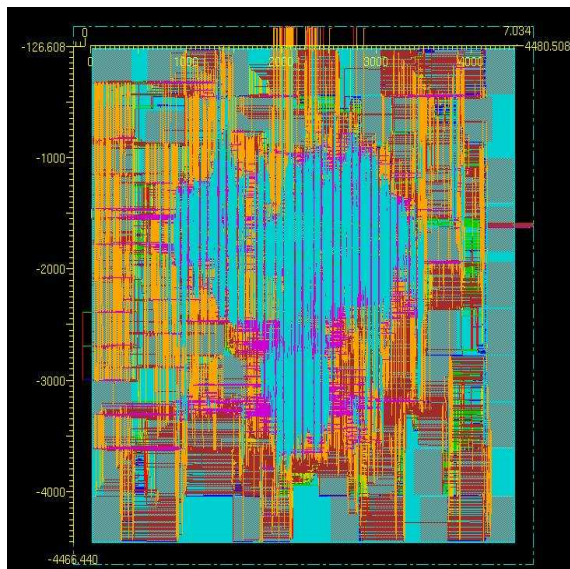
Ο συνδυασμός του πολλαπλασιασμού ή της πρόσθεσης κινητής υποδιαστολής με τη στρογγυλοποίηση είχε αρχικά παρουσιαστεί στο [73]. Η ιδέα της μεθόδου αυτής βασίζεται στο ακόλουθο γεγονός: πραγματοποιώντας πρόσθεση και στρογγυλοποίηση παράλληλα καταργείται η πιθανή επιπλέον πρόσθεση που μπορεί να προκληθεί από τη στρογγυλοποίηση, και έτσι μειώνεται η καθυστέρηση. Αυτό επιτυγχάνεται αλλάζοντας τη σειρά εκτέλεσης των πράξεων της στρογγυλοποίησης και κανονικοποίησης (η κανονικοποίηση λαμβάνει χώρα πριν από τη στρογγυλοποίηση), και συνδυάζοντας την πρόσθεση με τη στρογγυλοποίηση. Το πρόβλημα που προκύπτει όμως, όταν εκτελείται μια εντολή MAF, είναι ότι η θέση στρογγυλοποίησης είναι γνωστή μόνο αφού έχει ολοκληρωθεί η κανονικοποίηση. Για το λόγο αυτόν, η κανονικοποίηση πραγματοποιείται πριν από την πρόσθεση, κατά το δεύτερο στάδιο. Στο τρίτο στάδιο υπολογίζονται τα διανύσματα sum , $sum + 1$ και ανάλογα με το αποτέλεσμα της στρογγυλοποίησης, το σωστό αποτέλεσμα προωθείται στην έξοδο του κυκλώματος.

4.6.2 Υλοποίηση

Η μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης διπλής λειτουργίας σχεδιάστηκε και υλοποιήθηκε σε VHDL. Είναι σωληνομένη έτσι ώστε να έχει καθυστέρηση 3 κύκλων με ρυθμαπόδοση 1 αποτελέσματος ανά κύκλο. Το σχέδιο υλοποιήθηκε στη βιβλιοθήκη TSMC 0.13 1P8M. Η σύνθεση έγινε για μεταβλητό κύκλο ρολογιού, από 10 ns μέχρι 2 ns. Η υλοποίηση σε VLSI πέτυχε σαν τελική περίοδο ρολογιού τα 3.43ns (291 MHz), καταλαμβάνοντας χώρο $286766.316 \mu m^2$ και καταναλώνοντας 35.22 mW. Η υλοποίηση της μονάδας MAF σε VLSI παρουσιάζεται στην εικόνα 4.17. Τέλος, δύο τέτοιες μονάδες ενσωματώθηκαν και στον επεξεργαστή LE1 Multi-Cluster VLIW (εικόνα 4.18).



Σχήμα 4.17: Υλοποίηση σε VLSI της μονάδας MAF διπλής λειτουργίας.



Σχήμα 4.18: Υλοποίηση σε VLSI του επεξεργαστή LE1 VLIW, που περιέχει δύο μονάδες MAF διπλής λειτουργίας.

4.7 Μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης τριπλής λειτουργίας

Στοχεύοντας στην περαιτέρω βελτίωση της απόδοσης της μονάδας MAF που παρουσιάστηκε στα προηγούμενα, υλοποιήθηκε μια αρχιτεκτονική MAF πολλαπλών λειτουργιών και πολλαπλής ακρίβειας, βασισμένη σε μια οργάνωση διπλού μονοπατιού, που λειτουργεί αποτελεσματικά σε τετραπλή, διπλή και μονή ακρίβεια. Το σχέδιο πραγματοποιεί μια εντολή MAF τετραπλής ακρίβειας, ή δύο παράλληλες εντολές MAF διπλής ακρίβειας, ή τέλος τέσσερις παράλληλες εντολές MAF μονής ακρίβειας. Επιπλέον, μπορεί να πραγματοποιήσει για κάθε ακρίβεια πολλαπλές, αυτόνομες πράξεις πολλαπλασιασμού και πρόσθεσης.

Κίνητρο για τη δουλειά αυτή ήταν πρώτον, το να παρουσιαστεί μια MAF αρχιτεκτονική που να υποστηρίζει έναν αριθμό λειτουργιών χωρίς να αυξάνει σημαντικά το χώρο και την καθυστέρηση, δεύτερον, το να διερευνηθεί πως ο αλγόριθμος διπλού μονοπατιού μπορεί να χρησιμοποιηθεί αποτελεσματικά για να δημιουργηθεί ένα σχέδιο υλοποίησης εντολών MAF με πολλαπλές λειτουργίες και τρίτον, να εκτιμηθεί η απόδοση του σχεδίου βρίσκοντας το κόστος σε υλισμικό (hardware) και τις κρίσιμες καθυστερήσεις μονοπατιών και συγκρίνοντας το με τις υπάρχουσες λύσεις.

4.7.1 Αρχιτεκτονική

Η αρχιτεκτονικής Συγχωνευμένου Πολλαπλασιασμού-Πρόσθεσης τριπλής λειτουργίας, μπορεί να λειτουργεί σε όλες τις διαφορετικές ακρίβειες και να πραγματοποιεί πολλαπλές εντολές. Για το σχεδιασμό ακολουθείται μια προσέγγιση που οδηγεί σε μια αρχιτεκτονική υψηλής απόδοσης και χαμηλής καθυστέρησης, ενώ διατηρείται σχετικά χαμηλό το κόστος σε υλισμικό. Η μονάδα MAF είναι σχεδιασμένη έτσι ώστε να υποστηρίζει και τις τρεις διαφορετικές ακρίβειες: την τετραπλή, τη διπλή και τη μονή ακρίβεια. Σε κάθε μια από αυτές τις λειτουργίες το σχέδιο πραγματοποιεί έναν αριθμό από διαφορετικές πράξεις: την εντολή Συγχωνευμένου Πολλαπλασιασμού-Πρόσθεσης (MAF), αυτόνομο πολλαπλασιασμό ή αυτόνομη πρόσθεση.

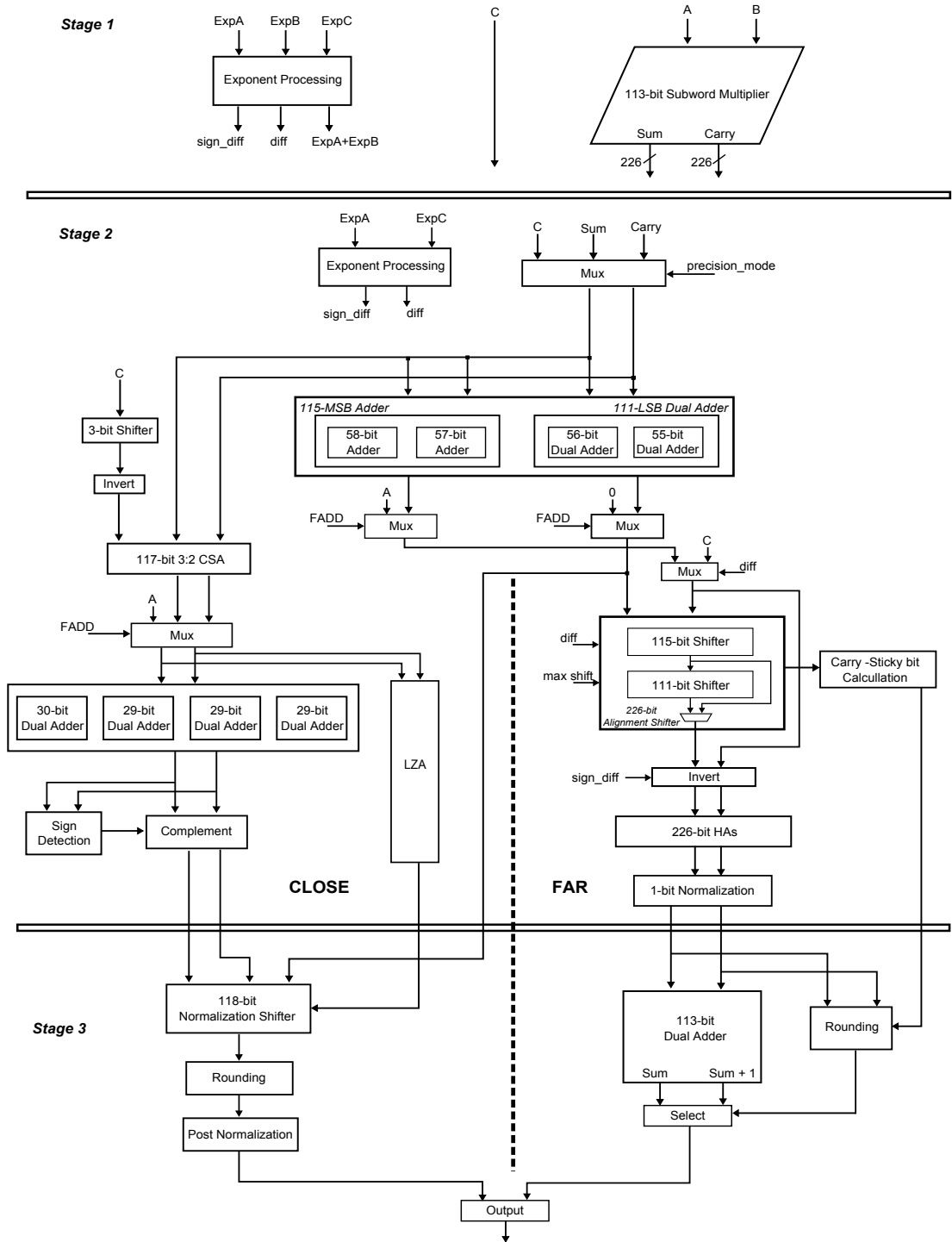
Για να επιτευχθούν οι παραπάνω στόχοι και λειτουργίες η αρχιτεκτονική υιοθετεί αρκετά διαφορετικά σχήματα:

- 1) Το σχέδιο ακολουθεί και πάλι μια προσέγγιση διπλού μονοπατιού [59] ώστε να ελαττωθεί η καθυστέρηση των εντολών που αφορούν την πρόσθεση κινητής υποδιαστολής, όμως σε αυτήν την περίπτωση για να βελτιωθεί ακόμα περισσότερο η αποτελεσματικότητα του σχεδίου, αποφεύγουμε την υλοποίηση της τυπικής αρχιτεκτονικής διπλού μονοπατιού που χρησιμοποιήθηκε προηγουμένως, και ακολουθώντας την προσέγγιση του [80] τα αποτελέσματα του πολλαπλασιαστή προστίθενται στην αρχή του δεύτερου σταδίου. Έτσι, καταφέρνουμε να περιορίσουμε το πλήθος των δομοστοιχείων ενώ το πιο σημαντικό είναι ότι μειώνεται το μέγεθος των ολισθητών ευθυγράμμισης, κάτι που έχει πολύ σημαντικό αντίκτυπο στην κρίσιμη καθυστέρηση. Στη δική μας προσέγγιση όμως

ακολουθούμε την τυπική υλοποίηση των τριών σταδίων (αντί για την υλοποίηση σε τέσσερα στάδια που πραγματοποιείται στο [80]), στην οποία το *Κοντινό* και το *Μακρινό* μονοπάτι επεκτείνονται επίσης και στο τρίτο στάδιο. Το *Μακρινό* μονοπάτι πραγματοποιεί την κανονικοποίηση στο δεύτερο στάδιο, πριν από την τελική πρόσθεση και στρογγυλοποίηση, ενώ το *Κοντινό* μονοπάτι μοιάζει με το σχέδιο της τυπικής MAF, με την κανονικοποίηση και στρογγυλοποίηση να γίνονται στο τελικό στάδιο.

2) Η συγκεκριμένη σχεδίαση διπλού μονοπατιού βασίζεται σε μία μονάδα MAF τετραπλής ακρίβειας και έχει επαυξηθεί για να υποστηρίξει την παράλληλη λειτουργία δύο εντολών διπλής ακρίβειας ή τεσσάρων εντολών μονής ακρίβειας. Για το σκοπό αυτό διάφορα μέρη του κυκλώματος έχουν επανασχεδιαστεί προσεκτικά και επιπλέον υλισμικό έχει προστεθεί. Θα πρέπει εδώ να σημειωθεί, πως αν και χρησιμοποιήθηκε επιπλέον υλισμικό, τα αποτελέσματα του καταλαμβανόμενου χώρου είναι και πάλι βελτιωμένα σε σχέση με την προσέγγιση του απλού διπλασιασμού ή του τετραπλασιασμού των στοιχείων της αρχιτεκτονικής για κάθε ακρίβεια.

3) Η αρχιτεκτονική χρησιμοποιεί τρεις καταχωρητές των 128 bit. Κάθε ένας από τους τρεις αυτούς καταχωρητές - ανάλογα με τον τύπο λειτουργίας - μπορεί να χρησιμοποιηθεί για να αποθηκεύσει έναν τελεστέο τετραπλής ακρίβειας, δύο τελεστέους διπλής ακρίβειας ή τέσσερις τελεστέους μονής ακρίβειας. Το σχέδιο είναι σωληνομένο σε τρία στάδια και έχει ρυθμαπόδοση ενός αποτελέσματος ανά κύκλο. Η εικόνα 4.19 παρουσιάζει το μπλοκ διάγραμμα της πολλαπλής ακρίβειας MAF αρχιτεκτονικής. Στο υπόλοιπο της ενότητας αυτής περιγράφεται η συνολική λειτουργικότητα του σχεδίου και παρέχεται μια λεπτομερής ανάλυση της συμπεριφοράς του σχεδίου σε κάθε λειτουργία.



Σχήμα 4.19: Μπλοκ διάγραμμα της πολλαπλής ακρίβειας MAF αρχιτεκτονικής.

4.7.1.1 Λειτουργία τετραπλής ακρίβειας

Η λειτουργία τετραπλής ακρίβειας (quadruple precision ή QP) αφορά την εκτέλεση μίας εκ των τριών πράξεων: την εντολής της εξίσωσης Συγχωνευμένου Πολλαπλασιασμού-Πρόσθεσης, τετραπλής ακρίβειας πολλαπλασιασμό ή τετραπλής ακρίβειας πρόσθεση.

Πρώτο στάδιο: οι υπολογισμοί σε αυτό το στάδιο ξεκινούν μετατρέποντας το σημαντικό μέρος των τελεστών στη μορφή: $[1', P1(111:0)]$ και $[1', P2(111:0)]$, ώστε να συμπεριληφθεί το κρυφό 1. Στη συνέχεια το δομοστοιχείο επεξεργασίας των εκθετών υπολογίζει τη διαφορά των εκθετών, το απαραίτητο ποσό ολίσθησης και τα σήματα ελέγχου. Στις περιπτώσεις των εντολών MAF ή του πολλαπλασιασμού η επεξεργασία των εκθετών γίνεται παράλληλα με τον πολλαπλασιασμό. Ο πολλαπλασιαστής μήτρας της εικόνας 4.21 πραγματοποιεί τον πολλαπλασιασμό των τροποποιημένων σημαντικών μερών. Στη λειτουργία τετραπλής ακρίβειας ο πολλαπλασιαστής ενεργεί σαν ένας τυπικός πολλαπλασιαστής μήτρας 113×113 bit παράγοντας τα αποτελέσματα σε μορφή διατήρησης ψηφίου.

Το δεύτερο και το τρίτο στάδιο χωρίζονται σε δύο μονοπάτια το *Κοντινό* και το *Μακρινό*. Μόνο ένα από τα δύο μονοπάτια ενεργοποιείται κάθε φορά, ανάλογα με το αποτέλεσμα της αφαίρεσης των εκθετών. Το *Κοντινό* μονοπάτι ενεργοποιείται στις περιπτώσεις ουσιαστικής αφαίρεσης με τη διαφορά των εκθετών να είναι $diff = -1, 0, 1$, ή στην περίπτωση ουσιαστικής αφαίρεσης με τη διαφορά των εκθετών να είναι $diff = 2$ και να υπάρχει υπερχείλιση στον πολλαπλασιασμό. Σε όλες τις υπόλοιπες περιπτώσεις ενεργοποιείται το *Μακρινό* μονοπάτι.

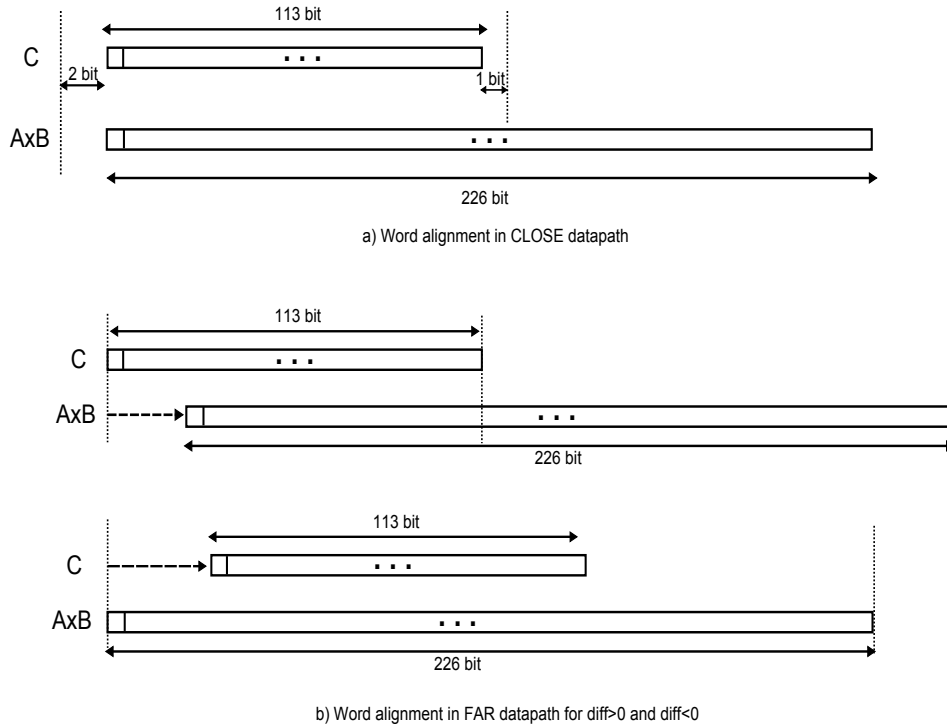
Δεύτερο στάδιο: Για την πρόσθεση τετραπλής ακρίβειας το πρώτο στάδιο παρακάμπτεται και χρησιμοποιείται ένα δεύτερο δομοστοιχείο επεξεργασίας εκθετών που έχει τοποθετηθεί στην αρχή του δεύτερου σταδίου. Σε όλες τις άλλες περιπτώσεις το δεύτερο στάδιο δέχεται σαν είσοδο τα αποτελέσματα του πολλαπλασιαστή, μαζί με τον τρίτο τελεστέο (που αποτελούν την έξοδο του πρώτου σταδίου), τα οποία ανάλογα με το αποτέλεσμα της αφαίρεσης των εκθετών ακολουθούν είτε το *Κοντινό* είτε το *Μακρινό* μονοπάτι. Ανεξάρτητα από το μονοπάτι που επιλέγεται, τα αποτελέσματα του πολλαπλασιαστή προστίθενται χρησιμοποιώντας τέσσερις αθροιστές: δύο διπλούς αθροιστές (dual adders) των 58- και 57-bit, και δύο απλούς αθροιστές των 56- και 55-bit. Οι διπλοί αθροιστές χρησιμοποιούνται για να υπολογιστεί εκ των προτέρων και το συμπλήρωμα της πρόσθεσης, το οποίο θα χρειαστεί εάν το αποτέλεσμα του *Κοντινού* μονοπατιού αποδειχθεί ότι είναι αρνητικό. Στην λειτουργία της τετραπλής ακρίβειας και οι τέσσερις αθροιστές χρησιμοποιούνται μαζί, ενώ στις άλλες ακρίβειες χρησιμοποιούνται ξεχωριστά. Προσθέτοντας τα αποτελέσματα του πολλαπλασιαστή μειώνεται σημαντικά το μέγεθος των ολισθητών ευθυγράμμισης, και επομένως μειώνεται και ο συνολικός χώρος.

Το *Κοντινό* μονοπάτι πραγματοποιεί ευθυγράμμιση του τελεστέου C μετατοπίζοντας τον το πολύ 1-bit δεξιά ή 2-bit αριστερά χρησιμοποιώντας έναν

ολισθητή 3-bit. Έτσι, για την τετραπλή ακρίβεια ο ευθυγραμμισμένος τελεστής είναι 116-bit (εικόνα 4.20). Ακολούθως της πράξης ευθυγράμμισης, ο τελεστής C πάντα αντιστρέφεται, ώστε να εξασφαλιστεί το θετικό αποτέλεσμα και στην περίπτωση που το αποτέλεσμα είναι αρνητικό το πρόσημο αντιστρέφεται ξανά στο τέλος. Η αναπαράσταση που προτιμάται είναι αυτή του 2-Συμπληρώματος, επειδή έτσι αποφεύγεται η διευθέτηση της κυκλικής επαναφοράς του κρατούμενου ψηφίου (end-around carry adjustment). Στη συνέχεια, ο αντεστραμμένος τελεστής C και τα αποτελέσματα του πολλαπλασιαστή εισέρχονται σε έναν συμπιεστή 3:2 CSA 117-bit, ώστε να αποκτηθεί μια πλεονάζουσα αναπαράσταση των ενδιάμεσων διανυσμάτων. Για να ολοκληρωθεί η μετατροπή σε 2-Συμπλήρωμα ένα επιπλέον '1' προστίθεται χρησιμοποιώντας μια άδεια θέση στον 3:2 CSA. Τα δύο διανύσματα που προκύπτουν προστίθενται χρησιμοποιώντας τέσσερις διπλούς αθροιστές. Στη λειτουργία τετραπλής ακρίβειας συνδυάζονται μαζί για να πραγματοποιήσουν την πρόσθεση, ενώ στις άλλες λειτουργίες χρησιμοποιούνται ξεχωριστά ώστε να πραγματοποιήσουν τις παράλληλες προσθέσεις. Για να αποφευχθούν επιπλέον καθυστερήσεις στο *Κοντινό* μονοπάτι, η λειτουργία του δομοστοιχείου LZA λαμβάνει χώρα παράλληλα με τη πρόσθεση. Το δομοστοιχείο LZA σχεδιάστηκε ακολουθώντας τη μέθοδο που παρουσιάζεται στο [77]. Η μονάδα Εντοπισμού Προσήμου (sign detection) προσδιορίζει το σωστό πρόσημο βάσει των αποτελεσμάτων της πρόσθεσης και αν είναι απαραίτητο βρίσκει το συμπλήρωμα.

Το *Μακρινό* μονοπάτι πραγματοποιεί πράξεις ολίσθησης μήκους λέξεων, ώστε να ευθυγραμμίσει είτε το $A \times B$ είτε το C , βάσει του προσήμου της διαφοράς των εκθετών. Έτσι, για $diff > 0$ το γινόμενο $A \times B$ ολισθαίνει δεξιά, το πολύ κατά 115-bit, ενώ για $diff \leq 0$ ο τελεστής C ολισθαίνει δεξιά, το πολύ κατά 226-bit (εικόνα 4). Από τη στιγμή που τα αποτελέσματα του πολλαπλασιαστή έχουν προστεθεί, υπάρχουν μόνο δύο διανύσματα προς διαχείριση, και η ευθυγράμμιση μπορεί να πραγματοποιηθεί χρησιμοποιώντας έναν ολισθητή, αντί για δύο. Ο ολισθητής αυτός είναι κοινός για τα $A \times B$, C και χρησιμοποιείται για να ευθυγραμμίσει ένα από αυτά, ανάλογα πάντα με την περίπτωση. Για το σκοπό αυτό, αντί να χρησιμοποιηθεί ένας ενιαίος ολισθητής 226-bit, υλοποιούνται δύο μικρότεροι ολισθητές των 115- και 111-bit αντίστοιχα. Ο πρώτος από αυτούς χρησιμοποιείται για την ευθυγράμμιση του $A \times B$, ενώ και οι δύο μαζί χρησιμοποιούνται για την ευθυγράμμιση του C . Κατά τη λειτουργία της ευθυγράμμισης τα ψηφία που ολισθαίνουν εκτός ορίου χρησιμοποιούνται για τον υπολογισμό του sticky bit. Στη επόμενη ενότητα θα αναλυθεί περισσότερο η εσωτερική δομή των ολισθητών αυτών και θα εξηγηθεί ο τρόπος σχεδίασης τους ώστε να μπορούν να υποστηρίξουν παράλληλες πράξεις ευθυγράμμισης στις άλλες ακρίβειες. Αφού ολοκληρωθεί η ευθυγράμμιση, το ευθυγραμμισμένο διάνυσμα ($A \times B$ ή C) αντιστρέφεται (άν είναι απαραίτητο), ώστε να είναι τα αποτελέσματα σε θετική μορφή. Στη συνέχεια τα διανύσματα εισάγονται σε μια γραμμή από Μερικούς Αθροιστές (Half Adders ή HAs). Μια κενή θέση στη γραμμή αυτή χρησιμοποιείται για να προστεθεί το απαραίτητο '1' ώστε να ολοκληρωθεί η αναπαράσταση της αντιστροφής σε 2-Συμπλήρωμα. Στο *Μακρινό* μονοπάτι

η κανονικοποίηση πραγματοποιείται πριν από τη πρόσθεση. Ένας ολισθητής κανονικοποίησης 2-bit διαχειρίζεται τις πιθανές πράξεις ολίσθησης 1-bit δεξιά ή 1-bit αριστερά, ώστε να διορθωθεί η πιθανή υπερχειλίση ή η ύπαρξη μη κανονικοποιημένου αποτελέσματος.



Σχήμα 4.20: Ευθυγράμμιση στο *Κοντινό* και το *Μακρινό* μονοπάτι.

Τρίτο στάδιο: Το τρίτο στάδιο είναι σχεδιασμένο έτσι ώστε το *Κοντινό* και το *Μακρινό* μονοπάτι να εξακολουθούν να έχουν διαφορετικές λειτουργίες, σε αντίθεση με την παραδοσιακή προσέγγιση του διπλού μονοπατιού [59], όπου μόνο το δεύτερο στάδιο χωρίζεται σε δύο μονοπάτια. Το *Κοντινό* μονοπάτι στο τρίτο στάδιο εκτελεί την κανονικοποίηση και τη στρογγυλοποίηση. Επειδή η κανονικοποίηση είναι απαραίτητο να γίνει σε όλο το μήκος της λέξης, χρησιμοποιείται ένας πλήρης ολισθητής μήκους 118-bit. Όπως θα εξηγηθεί και στη συνέχεια, ο ολισθητής κανονικοποίησης είναι και αυτός ένας συνδυασμός από μικρότερους ολισθητές, ώστε να μπορούν να πραγματοποιηθούν και οι αντίστοιχες πολλαπλές λειτουργίες στη διπλή και στη μονή ακρίβεια. Η πράξη LZA που πραγματοποιείται στο προηγούμενο στάδιο μπορεί να προκαλέσει αβεβαιότητα 2-bit, γι' αυτό και στο τρίτο στάδιο πραγματοποιείται στρογγυλοποίηση και μετά-κανονικοποίηση ώστε να προσδιοριστεί και να υπολογιστεί το σωστό αποτέλεσμα, το οποίο στη συνέχεια προωθείται στην έξοδο του κυκλώματος.

Το *Μακρινό* μονοπάτι πραγματοποιεί την τελική πρόσθεση και στρογγυ-

λοποίηση. Τα πάνω 113-bit της εξόδου του προηγούμενου σταδίου χρησιμοποιούνται για να γίνει η διπλής πρόσθεση, ενώ τα υπόλοιπα ψηφία προωθούνται στο δομοστοιχείο στρογγυλοποίησης. Δύο διπλοί αθροιστές των 58- και 53-bit συνδυάζονται για τον υπολογισμό των διανυσμάτων sum , $\text{sum}+1$. Ταυτόχρονα, το κύκλωμα στρογγυλοποίησης χρησιμοποιεί τα υπόλοιπα λιγότερο σημαντικά ψηφία, μαζί με το ψηφίο sticky και το ψηφίο στρογγυλοποίησης για να βρεθεί το αποτέλεσμα της στρογγυλοποίησης. Βάσει του τελευταίου, επιλέγεται το σωστό αποτέλεσμα των αθροιστών και προωθείται στην έξοδο του κυκλώματος.

4.7.1.2 Λειτουργία διπλής και μονής ακρίβειας

Τόσο στη λειτουργία διπλής ακρίβειας, όσο και στη μονή, οι εντολές εκτελούνται με τον ίδιο τρόπο που αναλύθηκε παραπάνω και έτσι αυτή η ενότητα θα εστιάσει στον τρόπο λειτουργίας συγκεκριμένων τμημάτων της αρχιτεκτονικής και στο πως μπορούν να υποστηρίξουν τις διάφορες ακρίβειες.

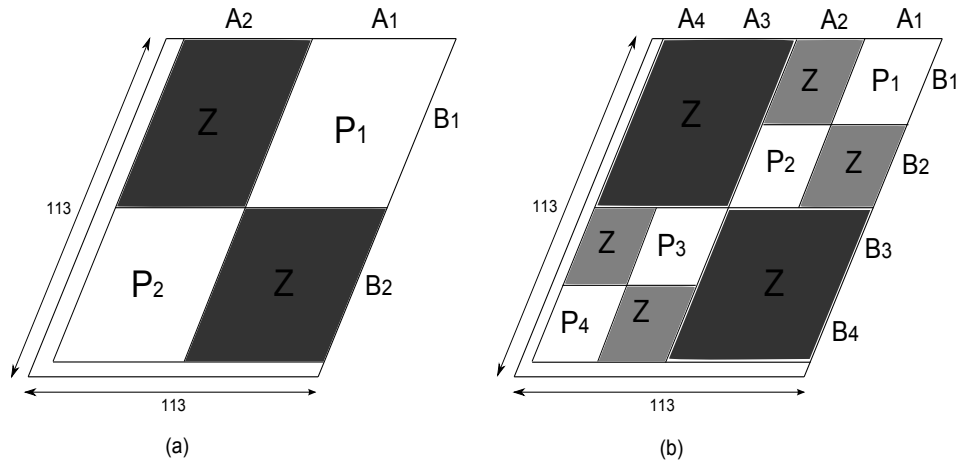
Στη λειτουργία διπλής ακρίβειας (double precision ή DP), η αρχιτεκτονική μπορεί να εκτελέσει δύο εντολές MAF παράλληλα, ή δύο παράλληλους πολλαπλασιασμούς ή δύο παράλληλες προσθέσεις διπλής ακρίβειας. Στη λειτουργία αυτή ο κάθε καταχωρητής στην αρχή της αρχιτεκτονικής περιέχει δύο τελεστέους διπλής ακρίβειας.

Παρόμοια, όταν το σχέδιο λειτουργεί σε μονή ακρίβεια (single precision ή SP), μπορεί να υποστηρίξει την παράλληλη εκτέλεση τεσσάρων εντολών MAF, ή τεσσάρων πολλαπλασιασμών ή τεσσάρων προσθέσεων μονής ακρίβειας.

Επεξεργασία Εκθετών: Για την παράλληλη εκτέλεση εντολών στη διπλή και στη μονή ακρίβεια έχει προστεθεί στο κύκλωμα επιπλέον υλισμικό. Συγκεκριμένα, έχουν προστεθεί μια μονάδα επεξεργασίας εκθετών διπλής ακρίβειας και δύο μονάδες επεξεργασίας εκθετών μονής ακρίβειας. Όταν το σχέδιο εκτελεί δύο παράλληλες εντολές MAF διπλής ακρίβειας, η μονάδα επεξεργασίας εκθετών που χρησιμοποιείται στην τετραπλή ακρίβεια εξυπηρετεί την πρώτη εντολή και η επιπλέον μονάδα διπλής ακρίβειας εξυπηρετεί τη δεύτερη. Με τον ίδιο τρόπο, στη μονή ακρίβεια καθεμία μονάδα επεξεργασίας εκθετών εξυπηρετεί μία εντολή μονής ακρίβειας.

Πολλαπλασιασμός: Στη λειτουργία διπλής ακρίβειας ο πολλαπλασιαστής πραγματοποιεί ταυτόχρονα και τους δύο πολλαπλασιασμούς διπλής ακρίβειας. Στην εικόνα 4.21a φαίνεται πώς οι δύο αυτοί πολλαπλασιασμοί μπορούν να εκτελεστούν παράλληλα. Τα τέσσερα δεκαδικά μέρη, αφού επεκταθούν με το κρυφό '1', εισάγονται στον πολλαπλασιαστή, όπου τα γινόμενα $A_1 \times B_1$, $A_2 \times B_2$ υπολογίζονται μέσα στους δύο υποπίνακες P_1 και P_2 , εφαρμόζοντας παράλληλο πολλαπλασιασμό υπολέξεων [78]. Τα ψηφία των μερικών γινομένων που βρίσκονται στις σκιασμένες με το γράμμα 'Z' περιοχές μηδενίζονται, ώστε να αποφευχθεί αλλοίωση του τελικού αποτελέσματος. Επίσης, όπως φαίνεται και στην εικόνα 4.21b, ο πολλαπλασιαστής μπορεί ακόμη να εκτελέσει παράλληλα και τέσσερις πολλαπλασιασμούς μονής ακρίβειας, εφαρμόζοντας την ίδια τεχνι-

κή με τη διπλή ακρίβεια.



Σχήμα 4.21: Παράλληλος πολλαπλασιασμός υπο-λέξεων.

Ολισθητές ευθυγράμμισης και κανονικοποίησης: Οι ολισθητές ευθυγράμμισης που χρησιμοποιούνται στο *Μακρινό* μονοπάτι θα πρέπει να μπορούν να πραγματοποιούν και παράλληλες λειτουργίες ευθυγράμμισης. Όταν το σχέδιο λειτουργεί σε διπλή ακρίβεια το μέγιστο ποσό ολίσθησης για την ευθυγράμμιση των διανυσμάτων $A \times B$ και C είναι 55- και 106-bit αντίστοιχα, ενώ στη μονή ακρίβεια είναι 26-bit και 48-bit αντίστοιχα. Επιπλέον, στη διπλή ακρίβεια το σχέδιο πραγματοποιεί το πολύ δύο παράλληλες πράξεις ολίσθησης, ενώ στη μονή ακρίβεια πρέπει να υποστηρίζει τέσσερις παράλληλες ολισθήσεις. Όπως αναφέρθηκε και παραπάνω, ο ολισθητής ευθυγράμμισης στο *Μακρινό* μονοπάτι αποτελείται από δύο ολισθητές των 115 και 111-bit. Για να υποστηρίξουν όμως όλες τις διαφορετικές ακρίβειες, οι ολισθητές αυτοί διαιρούνται ακόμα περισσότερο. Ο ολισθητής των 115-bit διαίρεται σε δύο ολισθητές των 60-bit και 55-bit ο καθένας, ενώ η πράξη ολίσθησης των 111-bit επιτυγχάνεται συνδυάζοντας δύο μικρότερους ολισθητές των 65 και 56-bit.

Έτσι, στη λειτουργία διπλής ακρίβειας οι δύο πρώτοι ολισθητές συνδυάζονται για να πραγματοποιήσουν την ευθυγράμμιση της πρώτης εντολής MAF, και οι δύο επόμενοι ολισθητές χρησιμοποιούνται μαζί για τη λειτουργία ολίσθησης της δεύτερης εντολής. Τέλος, στη λειτουργία μονής ακρίβειας καθένας από τους τέσσερις μικρούς ολισθητές χρησιμοποιείται για την ευθυγράμμιση μιας εντολής.

Ο ολισθητής κανονικοποίησης στο *Κοντινό* μονοπάτι είναι σχεδιασμένος με έναν τρόπο παρόμοιο με τον ολισθητή ευθυγράμμισης στο *Μακρινό* μονοπάτι. Η ολίσθηση κανονικοποίησης 118-bit του *Κοντινού* μονοπατιού πραγματοποιείται χρησιμοποιώντας τέσσερις ολισθητές: δύο των 30-bit, και δύο των 29-bit αντίστοιχα. Στη λειτουργία διπλής ακρίβειας, οι ολισθητές αυτοί συνδυάζονται σε ζευγάρια για να πραγματοποιήσουν δύο παράλληλες πράξεις

κανονικοποίησης, ενώ στη λειτουργία μονής ακρίβειας ο καθένας από αυτούς λειτουργεί ανεξάρτητα σαν ολισθητής κανονικοποίησης.

4.7.1.3 Υλοποίηση και ανάλυση των αποτελεσμάτων

Η αρχιτεκτονική που παρουσιάστηκε στα παραπάνω σχεδιάστηκε και υλοποιήθηκε σε γλώσσα VHDL. Το σχέδιο συνθέθηκε για μια διαδικασία των 45nm σιλικόνης χρησιμοποιώντας το εργαλείο Synopsys Design Compiler για το μπροστά μέρος της σύνθεσης και Cadence Encounter για το πίσω μέρος της σύνθεσης και την πραγματοποίηση του Place & Route. Ο πίνακας 4.1 δείχνει τα αποτελέσματα για το χώρο που καταλαμβάνει το κάθε στάδιο και την αντίστοιχη καθυστέρηση του.

Για να εκτιμηθεί η απόδοση του συγκεκριμένου σχεδίου, αρχικά επισημαίνονται τα βασικά χαρακτηριστικά της αρχιτεκτονικής, τα οποία στη συνέχεια συγκρίνονται με άλλες αρχιτεκτονικές διπλού μονοπατιού. Στη συνέχεια εκτιμάται η απόδοση του σχεδίου συγκρίνοντας την με υπάρχουσες MAF αρχιτεκτονικές πολλαπλής ακρίβειας. Τέλος, λαμβάνεται υπόψη και εκτιμάται η εναλλακτική λύση της υλοποίησης πολλαπλών MAF αρχιτεκτονικών διπλής ή μονής ακρίβειας.

Η αρχιτεκτονική που παρουσιάστηκε ακολουθεί την προσέγγιση ενός διαμορφωμένου διπλού μονοπατιού, στην οποία το *Κοντινό* μονοπάτι λειτουργεί σαν μια τυπική MAF αρχιτεκτονική πραγματοποιώντας πρόσθεση στο δεύτερο στάδιο και κανονικοποίηση και στρογγυλοποίηση στο τρίτο στάδιο. Το *Μακρινό* μονοπάτι όμως, προβλέπει την κανονικοποίηση στο δεύτερο στάδιο, πριν από την πρόσθεση, χρησιμοποιώντας μόνον έναν ολισθητή ευθυγράμμισης για να εξυπηρετήσει τις διάφορες ολισθήσεις. Επιπλέον, το σχέδιο συνδυάζει τα αποτελέσματα του πολλαπλασιαστή στο ξεκίνημα του δεύτερου σταδίου, μειώνοντας έτσι το μέγεθος των διανυσμάτων που πρέπει να διαχειριστούν τα διάφορα δομοστοιχεία. Επιπροσθέτως, οι αθροίσεις και οι πράξεις ευθυγράμμισης πραγματοποιούνται συνδυάζοντας μικρής κλίμακας αθροιστές και ολισθητές αντίστοιχα. Με τον τρόπο αυτό επιτυγχάνεται η εξυπηρέτηση των διαφόρων ακριβειών κρατώντας ταυτόχρονα την καθυστέρηση σε χαμηλά επίπεδα. Συνεπώς, συγκρίνοντας την προσέγγισή μας με υπάρχοντες αρχιτεκτονικές βλέπουμε ότι το σχέδιο στο [59] απαιτεί δύο ολισθητές ευθυγράμμισης πλήρους μήκους, αντί για έναν στη δική μας περίπτωση, ενώ ο αλγόριθμος στο [80] χωρίζει το *Μακρινό* μονοπάτι σε δύο υπο-μονοπάτια και παρόλο που μειώνεται η καθυστέρηση σε σχέση με το [59], χρειάζεται τέσσερα στάδια για να ολοκληρωθεί και χρησιμοποιεί περισσότερο υλισμικό για το το *Μακρινό* μονοπάτι συγκρινόμενο με το σχέδιό μας.

Συγκρίνοντας την αρχιτεκτονική με κάποιο σχέδιο πολλαπλής ακρίβειας σαν το [65], το οποίο επίσης μπορεί να λειτουργεί σε όλες τις ακρίβειες, το σχέδιό μας εμφανίζει 9.5% αύξηση στο χώρο, αλλά ταυτόχρονα παρουσιάζει και μια σημαντική βελτίωση στην καθυστέρηση. Επίσης, η προσέγγισή μας επιτυγχάνει μια επιπλέον μείωση στην καθυστέρηση στις περιπτώσεις που αφορούν

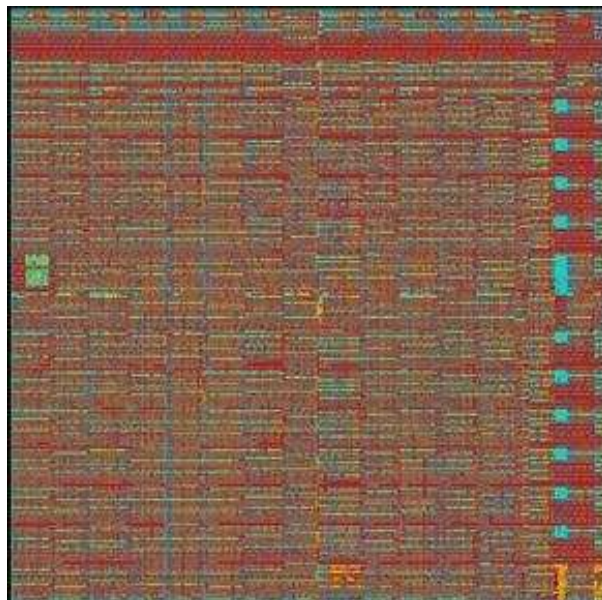
Πίνακας 4.1: Αποτελέσματα υλοποίησης σε VLSI της μονάδας MAF τριπλής λειτουργίας

Pipeline Stage	Stage 1	Stage 2	Stage 3
Area (gates)	110231	29795	21836
Delay (ns)	1.2	1.4	0.9

την εκτέλεση εντολών πρόσθεσης FADD, καθώς παρακάμπτει το πρώτο στάδιο. (Θα πρέπει εδώ να τονιστεί το γεγονός ότι παρόλο που τα σχέδια που παρουσιάζονται στο [65] έχουν συνθεθεί με διαφορετικές βιβλιοθήκες, η σημαντική διαφορά στις καθυστερήσεις ανάμεσα στις δύο μεθόδους, δείχνουν την ανωτερότητα του σχεδίου μας, όσον αφορά την καθυστέρηση του ρολογιού.

Τέλος, συγκρίνοντας το σχέδιο μας με μια τυπική υλοποίηση η οποία πραγματοποιεί μόνο εντολές τετραπλής ακρίβειας (όπως μια από τις αρχιτεκτονικές που παρουσιάζονται στο [65]), παρατηρούμε ότι το σχέδιο μας χρειάζεται 23.3% περισσότερο χώρο, παρουσιάζοντας όμως σημαντική μείωση στην καθυστέρηση, παρέχοντας ταυτόχρονα δύο επιπρόσθετες λειτουργίες. Παρόμοια, υλοποιήσεις που χρησιμοποιούν παράλληλα δύο μονάδες MAF διπλής ακρίβειας ή τέσσερις μονάδες μονής ακρίβειας, καταλαμβάνουν αναλογικά λιγότερο χώρο [65], κάτι που δεν αποτελεί έκπληξη, αφού το σχέδιο μας υποστηρίζει τρεις διαφορετικές ακρίβειες αντί για μια. Επιπλέον, αν και οι υλοποιήσεις αυτές παρουσιάζουν (σε κάθε περίπτωση) την ίδια ρυθμαπόδοση με αυτήν της αρχιτεκτονικής μας, επιδεικνύουν αυξημένη καθυστέρηση και μπορούν να εξυπηρετήσουν μόνο έναν τύπο ακρίβειας κινητής υποδιαστολής.

Με βάση τα παραπάνω, η αρχιτεκτονική κινητής υποδιαστολής πολλαπλής ακρίβειας που παρουσιάστηκε είναι ιδανική για εφαρμογές που απαιτούν εκτέλεση πολλαπλών εντολών με υψηλό ρυθμό απόδοσης και λειτουργία σε διαφορετικές ακρίβειες, ενώ δεν υπάρχει κρίσιμη εξάρτηση τους σε περιορισμούς που αφορούν τον καταλαμβανόμενο χώρο.



Σχήμα 4.22: Υλοποίηση σε VLSI της πολλαπλής ακρίβειας αρχιτεκτονικής MAF.

Κεφάλαιο 5

Σύνοψη και Μελλοντική εργασία

Η παρούσα διατριβή ασχολήθηκε με την ανάπτυξη αρχιτεκτονικών και την οργάνωση επεξεργαστών με στόχο τον αποδοτικό παραλληλισμό εφαρμογών, προσφέροντας λύσεις σε τρεις τομείς: στην κωδικοποίηση εικονορροών, στην υλοποίηση Ταχύ Μετασχηματισμού Fourier και στην πραγματοποίηση πράξεων κινητής υποδιαστολής.

Στην πρώτη περίπτωση αναλύθηκε η υλοποίηση μιας παραμετρικής αρχιτεκτονικής ενός διανυσματικού πολυεπεξεργαστή που χρησιμοποιήθηκε για την επιτάχυνση της εκτέλεσης των κωδικοποιήσεων MPEG-2 και MPEG-4. Το σύστημα αυτό του πολυεπεξεργαστή βασίζεται στην επέκταση μιας παραμετρικής αρχιτεκτονικής ενός διανυσματικού επεξεργαστή-συνεπεξεργαστή, ώστε να συμπεριλάβει και άλλα πλαίσια επεξεργαστών, δημιουργώντας ένα σύστημα SoC, κοινής μνήμης. Η επιτάχυνση της κωδικοποίησης των εικονορροών πραγματοποιήθηκε εκμεταλλευόμενη τον παραλληλισμό σε επίπεδο νημάτων (TLP) που μπορεί να εφαρμοστεί σε αυτές, οδηγώντας σε σημαντική μείωση του δυναμικού αριθμού εντολών που εκτελούνται από τα πλαίσια των επεξεργαστών του συστήματος.

Στη δεύτερη περίπτωση έγινε μελέτη και ανάπτυξη κυκλωμάτων Ταχύ Μετασχηματισμού Fourier για μεγάλα σύνολα δεδομένων εισόδου. Υλοποιήθηκαν σε υλισμικό αρχιτεκτονικές υψηλής ρυθμικόδοσης και χαμηλής συχνότητας, που πραγματοποιούν Ταχύ Μετασχηματισμό Fourier για 4K, 16K, 64K και 256K μιγαδικά σημεία. Τα κυκλώματα αυτά βασίζονται τη λειτουργικότητα τους στη χρήση μιας μηχανής βάσης-4³. Η μηχανή αυτή είναι ένα αυτόνομο κύκλωμα υλοποίησης Ταχύ Μετασχηματισμού Fourier 64 σημείων και αποτελεί δομικό στοιχείο όλων των παραπάνω αρχιτεκτονικών. Η μελέτη που παρουσιάστηκε έδειξε πως η αποτελεσματική χρήση των μηχανών αυτών, τόσο σε επίπεδο παραλληλισμού όσο και σειριακά, μπορεί να προσφέρει αποτελεσματικές λύσεις στο χώρο των αρχιτεκτονικών Ταχύ Μετασχηματισμού Fourier.

Τέλος, στην τρίτη περίπτωση προτάθηκαν λύσεις για την παράλληλη πραγ-

ματοποίηση υπολογισμών κινητής υποδιαστολής. Μελετήθηκε και αναπτύχθηκε ένας πολλαπλασιαστής κινητής υποδιαστολής πολλαπλής ακρίβειας, που υποστηρίζει όλες τις δυνατές λειτουργίες ακρίβειας που περιγράφονται από την IEEE, και πραγματοποιεί έναν πολλαπλασιασμό τετραπλής ακρίβειας, ή δύο παράλληλους πολλαπλασιασμούς διπλής ακρίβειας, ή τέσσερις παράλληλους πολλαπλασιασμούς μονής ακρίβειας. Υλοποιήθηκε μια μονάδα Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης, μεταβλητού μήκους σωλήνωσης, για έναν επεξεργαστή VLIW. Η μονάδα αυτή χρησιμοποιήθηκε στη συνέχεια σαν αφετηρία, για τη δημιουργία δύο ακόμη μονάδων Συγχωνευμένου Πολλαπλασιασμού - Πρόσθεσης, πολλαπλής ακρίβειας που διαχειρίζονται είτε αριθμούς διπλής και μονής ακρίβειας, είτε αριθμούς τετραπλής, διπλής και μονής ακρίβειας, εκτελώντας παράλληλα πολλαπλές εντολές της μορφής MAF, FMUL, ή FADD.

Τα θέματα της μελλοντικής έρευνας που προκύπτουν από την παρούσα εργασία αφορούν τα εξής:

α) Να μελετηθεί η απόδοση του συστήματος πολυεπεξεργαστή και σε άλλες περιπτώσεις, όπου μπορεί να εφαρμοστεί ο παραλληλισμός σε επίπεδο νημάτων.

β) Να χρησιμοποιηθούν τα συμπεράσματα που έχουν προκύψει από την υλοποίηση των αρχιτεκτονικών κινητής υποδιαστολής για να ενισχυθεί η αρχιτεκτονική του πολυεπεξεργαστή με μια Μονάδα Κινητής Υποδιαστολής.

γ) Να εφαρμοστεί η τεχνική της μηχανής βάσης-4³ στο χώρο της υλοποίησης κυκλωμάτων Ταχύ Μετασχηματισμού Fourier για 2 και 3 διαστάσεις.

δ) Μελέτη και ανάπτυξη νέων αρχιτεκτονικών Ταχύ Μετασχηματισμού Fourier με έμφαση στην αύξηση του παραλληλισμού των δεδομένων εισόδου για την επίτευξη ακόμα πιο υψηλών ρυθμαποδόσεων.

ε) Να συνεχιστεί η μελέτη αρχιτεκτονικών κινητής υποδιαστολής με τη δυνατότητα της παράλληλης εκτέλεσης εντολών. Η μελέτη αυτή μπορεί να κινηθεί προς την κατεύθυνση της υλοποίησης μιας μονάδας άθροισης κινητής υποδιαστολής, πολλαπλής ακρίβειας με απώτερο στόχο το συνδυασμό της με τις υπάρχουσες αρχιτεκτονικές για τη δημιουργία μιας ενιαίας Μονάδας Κινητής Υποδιαστολής, που θα μπορεί να διαχειρίζεται όλες τις ακρίβειες και θα εκτελεί παράλληλα πολλαπλές εντολές.

Ορολογία

Η απόδοση των όρων που αναφέρθηκαν στην παρούσα διατριβή έγινε με χρησιμοποιώντας τη Βάση Όρων Πληροφορικής INFOTERM που έχει δημιουργηθεί από την Ομάδα ΕΛΟΤ/ΤΕ48/ΟΕ1 *Ορολογία Πληροφορικής*.
(<http://inforterm.cs.aueb.gr/>)

Ξενόγλωσσος Όρος	Ελληνικός Όρος
alignment shifter	ολισθητής ευθυγράμμισης
Application Specific Integrated Circuits	Ολοκληρωμένα Κυκλώματα Ειδικών Εφαρμογών
Arithmetic Logic Unit (ALU)	αριθμητική λογική μονάδα
bit-rate	διφυσικός ρυθμός
bias	πόλωση
bottleneck	στένωση
buffering	ενδιάμεση αποθήκευση
cache memory	κρυφή μνήμη
cascade architecture	διαδοχική αρχιτεκτονική
Central Processing Unit (CPU)	Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)
configurable	διαμορφώσιμο
coprocessor	συνεπεξεργαστής
Data Level Parallelism (DLP)	Παραλληλισμός σε Επίπεδο Δεδομένων
DFT	Διακριτός Μετασχηματισμός Fourier
Dynamic Instruction Count (DIC)	Δυναμικός Αριθμός Εντολών
extensible	επεκτάσιμο
FFT	Ταχύς Μετασχηματισμός Fourier
Floating Point Unit (FPU)	Μονάδα Κινητής Υποδιαστολής
format	μορφότυπο
FPGA	προγραμματιζόμενοι πίνακες λογικών πυλών
fully unfolded architecture	πλήρως ανοιγμένη αρχιτεκτονική
granularity level	βαθμός ανάλυσης
Group-of-Pictures (GoP)	σύνολο εικονων
hardware	υλισμικό
High Definition TV (HDTV)	τηλεόραση υψηλής ευκρίνειας
Instruction Set Simulator (ISS)	προσομοιωτής συνόλου εντολών

Leading Zero Anticipation (LZA)	Πρόβλεψη Προπορευόμενων Μηδενικών
mantissa	σημαντικό μέρος
module	δομοστοιχείο
multi-cluster	πολλαπλής συστάδας
multi-thread	πολυνημάτωση
Multiply - ADD Fused (MAF) Unit	Μονάδα Συγχωνευμένου Πολλαπλασιασμού-Πρόσθεσης
operand	τελεστής
pipeline	σωλήνωση
processor	επεξεργαστής
radix	βάση
significant	κλασματικό μέρος
software	λογισμικό
scalable	επεκτάσιμο
standard cell process	τυπική διαδικασία κελιού
System-on-Chip (SoC)	Σύστημα σε Ολοκληρωμένο
thread	νήμα
Thread Level Parallelism (TLP)	Παραλληλισμός σε Επίπεδο Νημάτων
throughput	ρυθμαπόδοση
tuple	βάση
twiddle factors	παράγοντες στροφής
video	εικονοροή
VLSI	πολύ μεγάλης κλίμακας ολοκληρωμένα κυκλώματα
workload	φόρτος εργασίας
write through	διεγγραφή

Βιβλιογραφία

- [1] T. Sikora, "MPEG Digital Video-Coding Standards," IEEE Signal Processing Magazine, Vol. 14, No. 5, September 1997, pp. 82-100.
- [2] ISO/IEC JTC 1, "Coding of audio-visual objects - Part 2: Visual," ISO/IEC 14496-2 (MPEG-4 Part 2), Jan.1999.
- [3] LSI Logic Domino 8802 dual encode processor product brief, www.lsillogic.com/files/docs/marketing_docs/consumer/dmn-8802_pb.pdf
- [4] J. F. Martinez, J. Torrellas, "Speculative synchronization: applying thread-level speculation to explicitly parallel application," ACM SIGARCH Computer Architecture News, Vol. 30, No. 5, Dec. 2002.
- [5] 'The Leon-2 processor User's manual, XST edition, ver. 1.0.14', [http : //www.gaisler.com](http://www.gaisler.com)
- [6] The Sparc Architecture Manual Version 8", [http : //www.sparc.org](http://www.sparc.org)
- [7] V. A. Chouliaras, J. L. Nunez-Yanez, S. Agha, "Silicon Implementation of a Parametric Vector Datapath for real-time MPEG2 encoding," Proceedings of the IASTED (SIP) 2004, Honolulu, Hawaii, USA.
- [8] V. A. Chouliaras, E.J Delp, "A multi-standard video coding accelerator based on a vector architecture," Proc. of the IEEE International Conference in Consumer Electronics (ICCE 2005), Las Vegas, Nevada, USA, 9-13 January 2005.
- [9] K. ShenJ. L. Nunez, F. Rovati, D. Alfonso, "A parallel implementation of an MPEG encoder: faster than real-time!", Proc. of the SPIE Conference on Digital Video Compression: Algorithms and Technologies, pp. 407-418, San Jose, California, 5-10 February 1995.
- [10] P. Tiwari, E. Viscito, "A parallel MPEG-2 video encoder with look-ahead rate control," Proceedings of the 1996 IEEE ICASSP-96, pp. 1994 - 1997 vol. 4.

- [11] A. Rodriguez, A. Gonzalez, M. Malumbres, "Performance evaluation of parallel MPEG-4 video coding algorithms on clusters of workstations," International Conference on Parallel Computing in Electrical Engineering 2004, p. 354.
- [12] Y. He, I. Ahmad, M. Liou, "A software based MPEG-4 video encoder using parallel processing," IEEE Transaction on Circuits and Systems for Video Technology 8 (7) (1998) 909-920.
- [13] A. Hamosfakidis, Y. Paker and J. Cosmas, "A study of concurrency in MPEG-4 video encoder," IEEE International Conference on Multimedia Computing and Systems, 1998, pp. 204-207.
- [14] SimpleScalar LLC, [http : //www.simplescalar.com/](http://www.simplescalar.com/)
- [15] V.A. Chouliaras, T.R. Jacobs, S. Agha, V.M. Dwyer, "Quantifying the benefits of thread and data parallelism for fast motion estimation in MPEG-2", IEE Electronic Letters, Volume 42, Issue 13, pp. 747-748, 22 June 2006.
- [16] [http : //www.xvid.com/](http://www.xvid.com/)
- [17] T. Ungerer, B. Robic, J. Silc, "A survey of processors with explicit multithreading," ACM Computing Surveys (CSUR), Volume 35 Issue 1, March 2003.
- [18] Z. Fanga, L. Zhangb, J. B. Cartera, L. Chenga, M. Parkerc, "Fast synchronization on shared-memory multiprocessors: An architectural approach," Journal of Parallel and Distributed Computing, vol. 65, Issue 10, October 2005, pp.1158-1170.
- [19] O.K . Ersoy, *Fourier-Related Transforms, Fast Algorithms and Applications*. Englewood Cliffs, NJ:Prentice Hall, 1997.
- [20] J. Lee, J. Lee, M. H. Sunwoo, S. Moh and S. Oh "A DSP Architecture for High-Speed FFT in OFDM Systems," ETRI Journal, 2002.
- [21] Clark D. Thompson "Fourier Transform in VLSI," IEEE Transactions on Computers, 1983.
- [22] E. H. Wold and A. M. Despain "Pipeline and Parallel FFT Processors for VLSI Implementations," IEEE Transactions on Computers, vol. C-33, 1984.
- [23] S. He and M. Torkelson "A New Approach to Pipeline FFT Processor," Proceedings of the IPPS, 1996.

- [24] S. Choi, G. Govindu, J. W. Jang, V. K. Prasanna "Energy-Efficient and Parameterized Designs of Fast Fourier Transforms on FPGAs," The 28th International Conference on Acoustics, Speech, and Signal Processing (ICASSP), April 2003.
- [25] I. S. Uzun, A. Amira and A. Bouridane, "FPGA implementations of Fast Fourier Transforms for real-time signal and image processing," IEEE Vision, Image and Signal Processing, 2005.
- [26] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A New Radix-2/8 FFT Algorithm for Length- $q \times 2^m$ DFTs," IEEE Trans. Circuits Syst. I, vol. 51, no. 9, September 2004.
- [27] B. G. Jo and M. H. Sunwoo, "New Continuous-Flow Mixed-Radix (CFMR) FFT Processor Using Novel In-Place Strategy," IEEE Trans. Circuits Syst. I, vol. 52, no. 5, May 2005.
- [28] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "New Radix- $(2 \times 2 \times 2)/(4 \times 4 \times 4)$ and Radix- $(2 \times 2 \times 2)/(8 \times 8 \times 8)$ DIF FFT Algorithms for 3-D DFT," IEEE Trans. Circuits Syst. I, vol. 53, no. 2, February 2006.
- [29] W. H. Chang, T. Nguyen, "An OFDM-Specified Lossless FFT Architecture," IEEE Trans. Circuits Syst. I, vol. 53, no. 6, June 2006.
- [30] L. Yang, K. Zhang, H. Liu, J. Huang and S. Huang "An Efficient Locally Pipelined FFT Processor," IEEE Trans. Circuits Syst. II, vol. 53, no. 7, July 2006.
- [31] Y.N. Lin, H.Y. Liu, and C.Y.Lee "A 1-GS/s FFT/IFFT Processor for UWB Applications," IEEE Journ. of SSC, vol. 40, Issue 8, Aug. 2005.
- [32] J. Takala and K. Punkka Scalable FFT Processors and Pipelined Butterfly Units Journal of VLSI Signal Processing 43, 113-123, 2006.
- [33] S.S. Wang and C.S. Li An Area-Efficient Design of Variable-Length Fast Fourier Transform Processor Journal of VLSI Signal Processing, March 2007.
- [34] D. Reisis, N.Vlassopoulos, "Address Generation Techniques for Conflict Free Parallel Memory Accessing in FFT Architectures" ICECS, pp.1188-1191, December 2006.
- [35] L. R. Rabiner and B. Gold "Theory and Application of Digital Signal Processing," Prentice-Hall.
- [36] B. Suter and K. S. Stevens "A Low Power, High Performance approach for Time-Frequency / Time-Scale Computations," Proceedings SPIE98 Conference on Advanced Signal Processing Algorithms, Architectures and Implementations VIII. Vol. 3461, pp. 86-90, July 1998.

- [37] A. Oppenheim, R. Schaffer "Digital Signal Processing". Prentice Hall, 1975.
- [38] S. He and M. Torkelson "Design and Implementation of a 1024-point Pipeline FFT Processor," IEEE 1998 Custom Integrated Circuits.
- [39] J. Y. OH and M. S. Lim, "New Radix-2 to the 4th Power Pipeline FFT Processor," IEICE Trans. Electron., VOL. E88-C, NO. 8, August 2005.
- [40] E. Bidet, D. Castelain, C. Joanblanq and P. Stenn "A fast single-chip implementation of 8192 complex point FFT" IEEE Journ. of SSC, 30(3):300-305, Mar. 1995.
- [41] Ν. Βλασσόπουλος "Διδακτορική Διατριβή: Αποδοτικές αρχιτεκτονικές για παραλληλισμό υπολογισμών, για σχηματισμό Fourier" Εθνικό Καποδιστριακό Πανεπιστήμιο Αθηνών, Τμήμα Φυσικής, Αθήνα 2008.
- [42] C. Hinds, "An Enhanced Floating Point Coprocessor for Embedded Signal Processing and Graphics Applications," IEEE Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers, pp. 147-151, 1999.
- [43] D. Bossen, J. Tendler, K. Reick, "POWER4 system design for high reliability," IEEE Micro 22 (2002) 16-24.
- [44] D.R. Lutz, C.N. Hinds "A new floating-point architecture for wireless 3D graphics," in Proc. of the 38th Asilomar Conference on Signals, Systems and Computers. November 2004, pp. 355-359.
- [45] A. Naini, A. Dhablania, W. James, D. Sarma, "1 GHz HAL SPARC64 dual floatingpoint unit with RAS features," Proceedings of 15th Symposium on Computer Arithmetic, 2001, pp. 173-183.
- [46] Nvidia, GeForce Family, Online 2007. URL: <http://www.nvidia.com>.
- [47] A. Servetti, A. Rinotti, J.D. Martin "Fast implementation of the MPEG-4 AAC main and low complexity decoder," in IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, May 2004, pp. 249-252.
- [48] G. Howell, G.A. Geist, "Necessity of high precision arithmetic for large-scale computations," in Proc. of Neural, Parallel and Scientific Computations, May 1995, pp.219-222.
- [49] D. H. Bailey, "High-Precision Arithmetic and Applications to Physics and Mathematics," International Symposium on Nonlinear Theory and its Applications (NOLTA2007), Sep 2007.
- [50] F. de Dinechin, G. Villard, "High precision numerical accuracy in physics research," Nuclear Inst. and Methods in Physics Research, 2006.

- [51] Y. Hida, X. Li, D. Bailey, "Algorithms for quad-double precision floating point arithmetic," in Proc. of 15th IEEE Symposium on Computer Arithmetic, 2001, pp.155-162.
- [52] Y. He, C. Ding, "Using accurate arithmetic to improve numerical reproducibility and stability in parallel applications," Journal of Supercomputing 18, 2001, pp. 259-277.
- [53] IEEE Computer Society "IEEE Standard for Floating-Point Arithmetic," IEEE Std 754-2008.
- [54] E.N. Linzer, "Implementation of Efficient FFT Algorithms on Fused Multiply-Add Architectures," IEEE Transactions on Signal Processing, Vol. 41, No. 1, pp. 93-107, Jan, 1993.
- [55] H. Sharangpani, K. Arora, "Itanium Processor Microarchitecture," IEEE Micro Mag. Vol. 20, No. 5. pp. 24-43, 2000.
- [56] Y. Voronenko and M. Puschel, "Automatic Generation of Implementations for DSP Transforms on Fused Multiply-Add Architectures," International Conference on Acoustics, Speech and Signal Processing, pp. V- 101-4, 2004.
- [57] R. M. Jessani, M. Putrino, "Comparison of Single- and Dual-Pass Multiply-Add Fused Floating-Point Units," IEEE Transactions on Computers, vol. 47 no. 9, pp. 927-937, 1998.
- [58] C. Chen, L-A. Chen, J-R. Chen, "Architectural Design of a Fast Floating-Point Multiplication-Add Fused Unit Using Signed-Digit Addition," Proceedings of the Euromicro Symposium on Digital Systems Design (DSD'01), pp. 346, 2001.
- [59] J. D. Bruguera, T. Lang, "Floating-Point Fused Multiply-Add: Reduced Latency for Floating-Point Addition," 2002 IEEE International Conference on Computer Design (ICCD'02), pp.145, 2002.
- [60] H. S, M. G, "A novel architecture for floating-point multiply-add-fused operation," ICICS-PCM, vol.3, pp. 1675 - 1679, December 2003.
- [61] H. He, Z. Li, "Multiply-add fused float point unit with on-fly denormalized number processing," 48th IEEE Midwest Symposium on Circuits and Systems, Vol. 2, pp. 1466 - 1468, August 2005.
- [62] G. Li, Z. Li, "Design of A Fully Pipelined Single-Precision Multiply-Add-Fused Unit," 20th International Conference on VLSI Design (VLSI-SID'07), pp.318-323, 2007.

- [63] E. Quinell, E. E. Swartzlander, C. Lemonds, "Bridge Floating-Point Fused Multiply-Add Design," IEEE Transactions on VLSI Systems, Volume 16, Issue 12, pp. 1727-1731, December 2008.
- [64] L. Huang, L. Shen, K. Dai, Z. Wang, "A New Architecture For Multiple-Precision Floating-Point Multiply-Add Fused Unit Design," Proceedings of the 18th IEEE Symposium on Computer Arithmetic, pp. 69-76, 2007.
- [65] M. Gok, M. M. Ozbilen, "Multi-functional floating-point MAF designs with dot product support," Microelectronics Journal, Volume 39, Issue 1, pp. 30-43, January 2008.
- [66] W-C. Park, T-D. Han, S-D. Kim, S-B. Yang, "A floating point multiplier performing IEEE rounding and addition in parallel," Journal of Systems Architecture: the EUROMICRO Journal, Volume 45, Issue 14, pp. 1195-1207, July 1999.
- [67] G. Even, S.M. Mueller, P-M. Seidel, "A dual precision IEEE floating-point multiplier," Integration, the VLSI Journal, Volume 29, Issue 2, pp. 167-180, September 2000.
- [68] A. Akkas, M.J. Schulte, "Dual-mode floating-point multiplier architectures with parallel operations," Journal of Systems Architecture: the EUROMICRO Journal archive, Volume 52, Issue 10, pp. 549 - 562, October 2006.
- [69] X. Zhou, Z. Tang, "A New Architecture of a Fast Floating-Point Multiplier," in Proc. of APPT'2003. pp.23-30.
- [70] G. Govindu, L. Zhuo, S. Choi, V. Prasanna "Analysis of High-performance Floating-point Arithmetic on FPGAs," in Proc. of 18th Intl. Parallel and Distributed Processing Symposium, pp. 149-156, April 2004.
- [71] A. Akkas, "Dual-mode floating-point adder architectures," Journal of Systems Architecture: the EUROMICRO Journal, Volume 54, Issue 12, pp. 1129-1142, December 2008.
- [72] M.R. Santoro, G. Bewick, M.A. Horowitz, "Rounding Algorithms for IEEE Multipliers," Proceedings of 9th Symposium on Computer Arithmetic, pp. 176-183, 1989.
- [73] G. Even, P.M. Seidel, "A Comparison of Three Rounding Algorithms for IEEE Floating-Point Multiplication," IEEE Trans. Comput. Vol. 49, No. 7, pp. 638-650, 2000.
- [74] S. Leibson and J. Kim, "Configurable processors: a new era in chip design," IEEE Computer, Vol.38, No.7, pp. 51-59, July 2005.

- [75] V.-A. Chouliaras, K. Koutsomyti, T. Jacobs, S. Parr, D. Mulvaney and R. Thomson, "SystemC-defined SIMD instructions for high performance SoC architectures," Proceedings of the 13th IEEE International Conference on Electronics, Circuits and Systems, Nice, France, Dec 10 - 13, 2006
- [76] R.-E. Gonzalez, "Xtensa: a configurable and extensible processor," *Micro*, IEEE Vol. 20, Issue 2, pp: 60-70, Mar/Apr 2000.
- [77] M. S. Schmookler, K. J. Nowka, "Leading Zero Anticipation and Detection - A comparison of Methods," Proceedings of the 15th IEEE Symposium on Computer Arithmetic (ARITH15), pp.7-12, 2001.
- [78] S. Krithivasan, M. J. Schulte, "Multiplier Architecture for Media Processing," Proc. 37th Asilomar Conf. Signals, Systems, and Computers, pp.2193-2197, 2003.
- [79] H. Sun and M. Gao, "Unified bit pattern for leading-zero anticipatory logic for high-speed floating-point addition," Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology, Vol. 14, pp. 786 - 789, December 2003.
- [80] Z. Qi, Q. Guo, G. Zhang, X. Li, W. Hu "Design of Low-Cost High-performance Floating-point Fused Multiply-Add with Reduced Power," 23rd International Conference on VLSI Design, 2010.

ΔΗΜΟΣΙΕΥΣΕΙΣ

- Δημοσιεύσεις σε περιοδικά

1. K. Manolopoulos, D. Reisis, V.A. Chouliaras, "An Efficient Multiple Precision Floating-Point Multiply-Add Fused Unit," Elsevier Journal of Microelectronics (submitted).
2. K. Babionitakis, V.A. Chouliaras, K. Manolopoulos, K. Nakos, D. Reisis, N. Vlassopoulos, "Fully Systolic FFT Architecture for Giga-sample Applications," The Journal of Signal Processing Systems, Springer 2009.
3. V.A. Chouliaras, V.M. Dwyer, S. Agha, J.L. Nunez-Yanez, D. Reisis, K. Nakos, K. Manolopoulos, "Customization of an embedded RISC CPU with SIMD extensions for video encoding," The VLSI Journal of Integration, Elsevier, pages:135-152, 2007.
4. V.A. Chouliaras, T. Jacobs, J. Nunez-Yanez, K. Manolopoulos, K. Nakos, D. Reisis, "Thread Parallel MPEG-2 and MPEG-4 Encoders for Shared-Memory System-on-Chip Multiprocessors," International Journal of Computers and Applications, Issue 4, Volume 29, 2007, Acta Press.

- Δημοσιεύσεις σε συνέδρια

1. K. Manolopoulos, A. Belias, G. Georgis, D. Reisis, E.G. Anasontzis, "Signal Processing for Deep-Sea Observatories with Reconfigurable Hardware," IEEE International Conference on Electronics, Circuits and Systems, Seville, December 2012 (in print).
2. K. Manolopoulos, A. Belias, "Readout and Data Acquisition for KM3NeT," ARENA 2012 - Acoustic and Radio EeV Neutrino Detection Activities, Erlangen 2012 (in print).
3. K. Manolopoulos, D. Reisis, V.A. Chouliaras, "An Efficient Multiple Precision Floating Point Multiplier," 17th IEEE International Conference on Electronics, Circuits and Systems, December 2011, pp.153-156.
4. K. Manolopoulos, A. Belias, et. al., "FPGA Shore Station demonstrator for KM3NeT," Nuclear Inst. and Methods in Physics Research A, VLVnT Workshop, Erlangen, October 2011 (in print).
5. K. Manolopoulos, A. Belias, V. Koutsoumpos, C. Kachris, "Reconfigurable hardware applications on NetFPGA for network monitoring in large area sensor networks," Nuclear Inst. and Methods in Physics Research A, VLVnT Workshop, Erlangen, October 2011 (in print).

6. K. Manolopoulos, D. Reisis, V.A. Chouliaras, "An Efficient Dual-Mode Multiply-Add Fused Floating Point Unit," 17th IEEE International Conference on Electronics, Circuits and Systems. Athens, December 2010, pp. 5-8.
7. K. Manolopoulos, D. Reisis, V.A. Chouliaras, "A Configurable Length, Fused Multiply-Add Floating Point Unit for a VLIW Processor," 22nd IEEE International Conference on Systems on Chip, Belfast, 2009.
8. K. Manolopoulos, K. Nakos, D. Reisis, N. Vlassopoulos, V.A. Chouliaras, "High Performance 16K, 64K, 256K VLSI Systolic FFT Architectures," IEEE International Conference on Electronics, Circuits and Systems, Marrakesh, December 2007, pp. 146-149.
9. K. Babionitakis, K. Manolopoulos, K. Nakos, D. Reisis, N. Vlassopoulos, V.A. Chouliaras, "A High Performance VLSI FFT Architecture," IEEE International Conference on Electronics, Circuits and Systems, Nice, December 2006, pp. 810-813.
10. K. Babionitakis, K. Manolopoulos, K. Nakos, D. Reisis, N. Vlassopoulos, "A Memory Efficient, Low Power Fast Fourier Transform Architecture," The IEE, ACM SoC Design, Test and Technology Postgraduate Seminar, Loughborough UK, September 2004.
11. K. Manolopoulos, K. Nakos, D. Reisis, N. Vlassopoulos, "Reconfigurable Fast Fourier Transform Architecture for Orthogonal Frequency Division Multiplexing Systems," in the Proceedings of the International Conference on Computer, Communication and Control Technologies, Florida, June 2003, pp.VI 24-29.