ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ ΚΑΙ ΥΛΙΚΟΥ

«ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΕ ΑΝΑΔΙΑΤΑΣΣΟΜΕΝΗ ΛΟΓΙΚΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ MAFFT ΓΙΑ ΣΥΝΕΝΩΣΗ ΚΟΜΜΑΤΙΩΝ DNA»

ΔΕΣΑΡΤΗ ΑΘΗΝΑ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:
Δόλλας Απόστολος **(**ΕΠΙΒΛΕΠΩΝ**)**
ΚΑΘΗΓΗΤΗΣ ΠΟΛΥΤΕΧΝΕΙΟΥ ΚΡΗΤΗΣ
Παπαευσταθίου Ιωάννης
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ ΠΟΛΥΤΕΧΝΕΙΟΥ ΚΡΗΤΗΣ
Ζερβάκης Μιχάλης
ΚΑΘΗΓΗΤΗΣ ΠΟΛΥΤΕΧΝΕΙΟΥ ΚΡΗΤΗΣ

ΧΑΝΙΑ 2009

Περίληψη

Οι μοριακές ακολουθίες συντίθενται από έναν αριθμό από στοιχεία τα οποία ονομάζονται βάσεις. Για παράδειγμα οι ακολουθίες DNA αποτελούνται από τέσσερα είδη νουκλεϊκών οξέων ενώ οι ακολουθίες πρωτεϊνών από είκοσι διαφορετικά είδη αμινοξέων. Στην εξελικτική διαδικασία οι μοριακές αυτές ακολουθίες υφίσταται μεταλλάξεις όπως αντιγραφή, διαγραφή και αντικατάσταση. Αυτά τα γεγονότα μετάλλαξης έχουν ως αποτέλεσμα την ποικιλόμορφη διαφοροποίηση ανάμεσα στις ακολουθίες. Προκειμένου να βρεθούν ποιά μέρη της ακολουθίας είναι όμοια και ποιά διαφέρουν, γίνεται πολλαπλή ευθυγράμμιση ακολουθιών.

Ένας αλγόριθμος που εκτελεί με ταχύτητα και ακρίβεια την πολλαπλή ευθυγράμμιση ακολουθιών είναι ο MAFFT. Για τους λόγους αυτούς επιλέχθηκε ο συγκεκριμένος αλγόριθμος, μελετήθηκε και στην συνέχεια σχεδιάστηκε και υλοποιήθηκε σε αναδιατασσόμενη λογική ένα βασικό μέρος του αλγορίθμου επιδιώκοντας κάποιο κέρδος όσο αφορά το χρόνο εκτέλεσης του.

Ευχαριστίες

Η εργασία αυτή εκπονήθηκε στο Εργαστήριο Μικροεπεξεργαστών και Υλικού του Πολυτεχνείου Κρήτης υπό την επίβλεψη του καθηγητή κ. Απόστολου Δόλλα κατά τη διάρκεια του ακαδημαϊκού έτους 2008-2009.

Θα ήθελα να ευχαριστήσω πρώτα απ' όλα τον επιβλέποντα καθηγητή μου κ. Απόστολο Δόλλα για την υποστήριξή του, την καθοδήγηση και τις συμβουλές του αλλά και την εμπιστοσύνη που έδειξε στις δυνατότητές μου. Επίσης, θα ήθελα να ευχαριστήσω τον Επίκουρο Καθηγητή κ. Ιωάννη Παπαευσταθίου και τον Καθηγητή κ. Μιχάλη Ζερβάκη για τη συμμετοχή τους ως μέλη της εξεταστικής επιτροπής.

Ένα μεγάλο ευχαριστώ επίσης, στο Διδακτορικό φοιτητή Ευριπίδη Σωτηριάδη, για την πολύ μεγάλη βοήθεια του και την καθοδήγηση του σε κάθε βήμα της πορείας από την αρχή μέχρι και την ολοκλήρωση αυτής της εργασίας.

Τις ευχαριστίες μου θα ήθελα να εκφράσω και στον υπεύθυνο του εργαστηρίου Μικροεπεξεργαστών και Υλικού κ. Μάρκο Κιμιωνή, καθώς και σε όλα τα μέλη του εργαστηρίου για την βοήθεια που μου παρείχαν και για το άψογο κλίμα που υπήρχε στις μεταξύ μας σχέσεις .

Για τη φιλία και τη στήριξη τους θα ήθελα να ευχαριστήσω και όλους τους φίλους και συναδέλφους που ήταν δίπλα μου σε όλες τις φάσεις της ζωής μου.

Τέλος θα ήθελα να πω ένα μεγάλο ευχαριστώ στους γονείς μου και τον αδελφό μου, που πάντα ήταν δίπλα μου και με στηρίζουν στα όνειρα και τις αποφάσεις μου.

Περιεχόμενα

1. Εισαγωγή

Στο πρώτο κεφάλαιο της παρούσας διπλωματικής εργασίας γίνεται μια σύντομη αναφορά στην έννοια και την χρησιμότητα της πολλαπλής ευθυγράμμισης ακολουθιών. Επίσης γίνεται μια εισαγωγή στον αλγόριθμο MAFFT και το πρόβλημα της συγκεκριμένης μεθόδου. Τέλος παρατίθεται η δομή του κειμένου που θα ακολουθήσει.

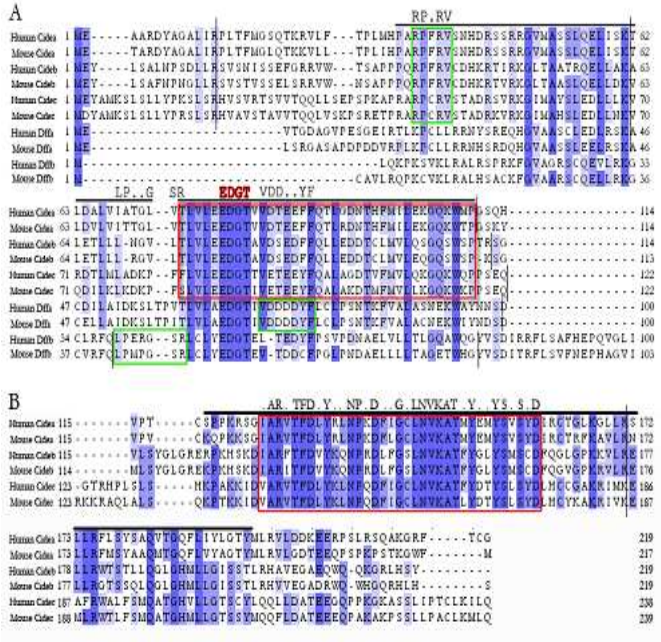
1.1 Γενικά για την πολλαπλή ευθυγράμμιση ακολουθιών (MSA)

Ένα από τα πιο ελκυστικά υπολογιστικά προβλήματα της σύγχρονης Βιοπληροφορικής είναι η τεχνική της πολλαπλής ευθυγράμμισης ακολουθιών, όπου γίνεται ταυτόχρονη σύγκριση της δομικής σχέσης ανάμεσα σε πολλές ακολουθίες διοξυριβονουκλεϊκών οξέων (DNA) ή πρωτεϊνών.

Οι μοριακές ακολουθίες συντίθενται από έναν αριθμό από στοιχεία τα οποία ονομάζονται βάσεις. Για παράδειγμα οι ακολουθίες DNA αποτελούνται από τέσσερα είδη νουκλεϊκών οξέων ενώ οι ακολουθίες πρωτεϊνών από είκοσι διαφορετικά είδη αμινοξέων. Στην εξελικτική διαδικασία οι μοριακές αυτές ακολουθίες υφίσταται μεταλλάξεις όπως αντιγραφή, διαγραφή και αντικατάσταση.

Αυτά τα γεγονότα μετάλλαξης έχουν ως αποτέλεσμα την ποικιλόμορφη διαφοροποίηση ανάμεσα στις ακολουθίες. Προκειμένου να βρεθούν ποιά μέρη της ακολουθίας είναι όμοια και ποιά διαφέρουν, γίνεται πολλαπλή ευθυγράμμιση ακολουθιών.

Αυτή η μέθοδος ευθυγραμμίζει κάθε ακολουθία έτσι ώστε μια βάση σε μία ακολουθία να αντιστοιχεί στις βάσεις των άλλων ακολουθιών, με σκοπό την εύρεση ομοιότητας μεταξύ τους . Όσο μεγαλύτερη είναι η ομοιότητα ανάμεσα στις ακολουθίες τόσο μεγαλύτερη είναι η πιθανότητα να έχουν κοινό πρόγονο , να είναι δηλαδή ομόλογες. Γενικά, οι ακολουθίες προς ευθυγράμμιση έχουν διαφορετικά μήκη. Για καλύτερη σύγκριση εισάγονται μέσα στις ακολουθίες κενά, τα οποία συμβολίζονται με μία παύλα “-” και έτσι οι ακολουθίες που προκύπτουν από την ευθυγράμμιση, έχουν ίσα μήκη (Σχήμα1.1). Τέλος η απόδοση της πολλαπλής ευθυγράμμισης μετράται με ένα σκορ. Λιγότερες μεταλλάξεις αντιστοιχούν σε χαμηλότερο σκορ. Η καλύτερη ευθυγράμμιση είναι αυτή με το υψηλότερο σκορ.



[1]Εικόνα 1.1 Ευθυγράμμιση μοριακών ακολουθιών

[1] Πηγή : Anand et al.,www.scienceexpress.org//10.1126/science.1085658

1.2 Εισαγωγή στο πρόβλημα

Το πρόβλημα της πολλαπλής ευθυγράμμισης ακολουθιών είναι ένα πρόβλημα βελτιστοποίησης ψάχνοντας για την καλύτερη ευθυγράμμιση από ένα μεγάλο πολύπλοκο χώρο αναζήτησης[1].

Τα τελευταία χρόνια ο αριθμός των διαθέσιμων ακολουθιών έχει αυξηθεί εκθετικά. Εξαιτίας αυτής της αύξησης είναι αναγκαίος ο υπολογισμός της MSA να γίνεται σε όσο το δυνατό μικρότερο χρόνο. Σήμερα υπάρχουν αρκετοί αλγόριθμοι που χρησιμοποιούν διάφορες ευριστικές μεθόδους με σκοπό την δημιουργία ταχύτερων και ακριβέστερων MSA.

Η παρούσα διπλωματική εργασία καλύπτει τον σχεδιασμό και την υλοποίηση συγκεκριμένων κομματιών του αλγορίθμου **MAFFT** σε αναδιατασσόμενη λογική με κύριο στόχο την επίτευξη σημαντικής επιτάχυνσης του χρόνου εκτέλεσης σε σχέση με έναν συμβατικό ηλεκτρονικό υπολογιστή. Η μελέτη της εργασίας επικεντρώνεται στην σύγκριση ακολουθιών νουκλεϊκών οξέων και στην λεγόμενη progressive μέθοδο πολλαπλής ευθυγράμμισης, καθώς η διαδικασία είναι σχεδόν ίδια στην περίπτωση σύγκρισης ακολουθιών πρωτεϊνών, με μικρές μόνο αλλαγές.

1.3 Δομή της διπλωματικής εργασίας

Στο **Κεφάλαιο 2,** αναλύονται βασικοί όροι σχετικά με την δομή του DNA και των ακολουθιών που σχηματίζουν, οι οποίοι είναι χρήσιμοι για την κατανόηση

του αλγορίθμου και περιγράφονται συνοπτικά μερικοί από τους υπάρχοντες αλγόριθμους για MSA.

Στο **Κεφάλαιο 3,** γίνεται εκτενής αναφορά στα βήματα του αλγορίθμου MAFFT.

Στο **Κεφάλαιο 4,** παρουσιάζεται η σχεδίαση και υλοποίηση της αρχιτεκτονικής του αλγορίθμου.

Στο **Κεφάλαιο 5,** παρουσιάζεται η απόδοση της αρχιτεκτονικής που υλοποιήθηκε.

Τέλος στο **Κεφάλαιο 6,** παρουσιάζονται οι παρατηρήσεις και τα συμπεράσματα καθώς και μελλοντικές επεκτάσεις και αλλαγές.

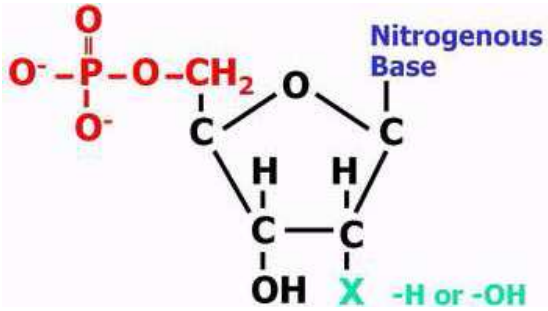
2. Νουκλεϊκά οξέα, ακολουθίες και διάφορες μέθοδοι σύγκρισης τους

Το κεφάλαιο αυτό περιλαμβάνει το απαραίτητο βιολογικό υπόβαθρο το οποίο θα είναι χρήσιμο για την κατανόηση του επόμενου κεφαλαίου. Γίνεται μία εκτενής αναφορά στην έννοια της πολλαπλής ευθυγράμμισης ακολουθιών και επιπλέον παρουσιάζονται διάφορες μέθοδοι σχετικά με αυτή[2].

2.1 Νουκλεϊκά οξέα

Το **DNA** και το **RNA** είναι τα δύο είδη νουκλεονικών οξέων. Σχετίζονται πολύ στενά με την γενετική πληροφορία, δηλαδή είναι τα στοιχεία εκείνα που έχουν να κάνουν με την κληρονομικότητα και τα ιδιαίτερα χαρακτηριστικά κάθε ατόμου. Η δομική μονάδα των νουκλεονικών οξέων είναι τα νουκλεοτίδια. Αποτελούνται από μια πεντόζη η οποία είναι ενωμένη με μία φωσφορική ομάδα και μία αζωτούχο βάση (Εικόνα 2.1). Στο μόνο που διαφέρει το ένα νουκλεοτίδιο με το άλλο είναι η

αζωτούχα βάση με την οποία είναι συνδεδεμένο. Στα νουκλεοτίδια του DNA η αζωτούχος βάση μπορεί να είναι: A - Αδενίνη (Adenine),G - Γουανίνη (Guanine) T - Θυμίνη (Thymine),C - Κυτοσίνη (Cytosine). Στο RNA μπορεί να συναντήσει κανείς όλες τις προηγούμενες έκτος από τη Θυμίνη αντί της οποίας υπάρχει η Ουρακίλη(Uracil) .

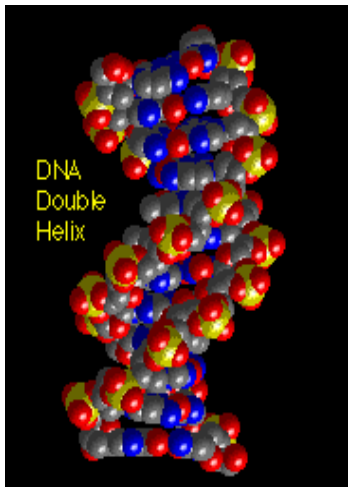


Εικόνα 2.1 Η δομή των νουκλεϊκών οξέων[2]

Τα νουκλεοτίδια συνδέονται μεταξύ τους με ομοιοπολικό δεσμό για να σχηματίσουν το μακρομόριο και μπορούν να εμφανίζονται στην αλυσίδα του DNA με οποιαδήποτε σειρά. Το σύμπλεγμα του σακχάρου και των φωσφορικών οξέων αποτελεί τη ραχοκοκαλιά του μορίου του DNA. Επειδή μία μόνο αλυσίδα του μορίου του DNA είναι αρκετά εύθραυστη συνήθως το DNA εμφανίζεται να σχηματίζει διπλές αλυσίδες. Ο κανόνας για τον σχηματισμό των αλυσίδων αυτών είναι η τήρηση της συμπληρωματικότητας των βάσεων. Τα επιτρεπτά ζεύγη είναι A-T και C-G. Στην περίπτωση σχηματισμού συμπληρωματικής αλυσίδας RNA, οι συνδυασμοί αυτοί γίνονται A-U και C-G, καθώς όπου υπάρχει θυμίνη στο DNA υπάρχει ουρακίλη στο RNA. Η συμπληρωματικότητα είναι το χαρακτηριστικό εκείνο του DNA που του παρέχει τη δυνατότητα του αυτοδιπλασιασμού

[2] Πηγή : http://graphics.csie.ntu.edu.tw/~zick/bio/di®/nucleotide.jpg

(αντιγραφή του DNA). Μια ακόμα διαφορά στη δομή των DNA και RNA είναι ότι ενώ το RNA αποτελείται από μόνο μια αλυσίδα νουκλεοτιδίων, το DNA είναι δίκλωνο(Εικόνα 2.2). Οι σημαντικότερες διαδικασίες που παρατηρούνται στο DNA είναι η αντιγραφή, η μεταγραφή, η μετάφραση και η μετάλλαξη.



[3]Εικόνα 2.2 Διπλή έλικα DNA

Το DNA αποτελεί το γενετικό υλικό όλων των κυττάρων και των περισσοτέρων ιών. Το γενετικό υλικό ενός κυττάρου αποτελεί το γονιδίωμα του. Μια σειρά νουκλεοτιδίων του DNA αποτελούν ένα γονίδιο. Τα γονίδια περιέχουν τον κωδικό για την παραγωγή των πρωτεϊνών, που είναι τα μακρομόρια εκείνα που διεκπεραιώνουν τις περισσότερες εργασίες στο κύτταρο.

[3] Πηγή : http://www.accessexcellence.org/RC/VL/GG/dna2.html

2.2 Ακολουθίες και ευθυγράμμιση

Το φαινόμενο της ζωής μπορεί να εξερευνηθεί διαμέσου του κωδικοποιημένου γενετικού υλικού, δεοξυριβονουλεϊκό οξύ (DNA)[3]. Αναφερόμενοι σε ακολουθίες, εννοούμε ακολουθίες νουκλεοτιδίων (που περιέχονται στο DNA ή RNA). Από έρευνες προκύπτει ότι δεν είναι όλα τα τμήματα μίας ακολουθίας ύψιστης σημασίας, αλλά μόνο τμήματα αυτής κουβαλούν γενετική πληροφορία, η οποία μπορεί να μεταφραστεί σε αμινοξέα πρωτεϊνικών ακολουθιών κάτω από ορισμένες συνθήκες. Από την άλλη για την ανακάλυψη του μυστηρίου της κυριαρχίας της ζωής οι άνθρωποι δεν διαβάζουν μόνο τις ακολουθίες DNA αλλά επιπλέον τις αναλύουν με σκοπό να αποσπάσουν την λειτουργικότητα και την περίπτωση της ενεργοποίησης κάθε γονιδίου. Τότε τα αποτελέσματα της ανάλυσης αυτής μπορούν να χρησιμοποιηθούν για να βοηθήσουν τους ανθρώπους να προλαμβάνουν τις γενετικές αρρώστιες, την εκ γενετής εγκληματικότητα κτλ.

Από τότε που ο J. D. Watson και ο F. H. C. Crick[4] ανακάλυψαν την χημική δομή του DNA το 1953, η μοριακή βιολογία έχει σημειώσει σημαντική πρόοδο. Μεγάλος αριθμός δεδομένων μοριακών ακολουθιών εξάγονται από τα εργαστήρια και αποθηκεύονται σε βάσεις δεδομένων[5]. Επιπλέον μεγάλες ανθρώπινες ακολουθίες DNA έχουν παραχθέι κάτω από τις προσπάθειες του Human Genome Project[6-7] το οποίο έχει ως σκοπό την παραγωγή ολόκληρης της ανθρώπινης ακολουθίας DNA η οποία υπολογίζεται περίπου στις τρία εκατομμύρια βάσεις. Ο ρυθμός αύξησης της ποσότητας των βιολογικών ακολουθιών είναι ταχύς. Γι αυτό, υπάρχει μεγάλη ανάγκη για επαρκεί και αποδοτικούς αλγόριθμους που θα αποσπούν και θα αναλύουν την υπάρχουσα πληροφορία από τις μοριακές ακολουθίες.

Στην εξελικτική διαδικασία οι μοριακές ακολουθίες υφίστανται μεταλλάξεις όπως διαγραφή, εισαγωγή, αντικατάσταση. Για παράδειγμα μία ακολουθία ATTG

μπορεί να αλλάξει σε ATG, ATTG ή ATCG αυτά τα γεγονότα μετάλλαξης έχουν ως αποτέλεσμα ποικίλες διαφορές ανάμεσα στις ακολουθίες. Το κύριο πρόβλημα λοιπόν που τίθεται είναι η αναζήτηση δομικών ή λειτουργικών ομοιοτήτων ανάμεσα στα γονιδιώματα και τις πρωτεΐνες[2].

2.2.1 Ευθυγράμμιση ακολουθιών

Στην ανάλυση μοριακών ακολουθιών[8], μια από τις πιο συχνά χρησιμοποιήσιμες τεχνικές είναι αυτή της ευθυγράμμισης ακολουθιών και κυρίως της πολλαπλής ευθυγράμμισης ακολουθιών. Με αυτή τη μέθοδο οι ακολουθίες στοιχίζονται κατακόρυφα, με στόχο την επίτευξη μέγιστου επιπέδου ταυτοσημότητας. Αναλύοντας τις ομοιότητες και διαφορές ανάμεσα στις ακολουθίες νουκλεϊκών οξέων είναι πιθανό να εισάγουμε δομικές, λειτουργικές ή εξελικτικές σχέσεις των ακολουθιών που μελετώνται (Baxevanis & Ouellette)[2].

Το πρόβλημα ευθυγράμμισης δύο ακολουθιών περιγράφεται ως εξής: Δοθέντων δύο ακολουθιών και ενός συστήματος σκορ που αποδίδει μία τιμή σε κάθε ταίριασμα-αστοχία και μια ποινή για τα κενά ανάμεσα σε δύο χαρακτήρες σκοπός είναι η παραγωγή ενός ζευγαριού από χαρακτήρες των δύο ακολουθιών έτσι ώστε το συνολικό σκορ να είναι το βέλτιστο. Στα ζευγάρια αυτά των χαρακτήρων μπορούν να εισαχθούν κενά σε οποιαδήποτε θέση των ακολουθιών. Παρ' όλα αυτά η σειρά των χαρακτήρων πρέπει να παραμείνει ίδια[2].

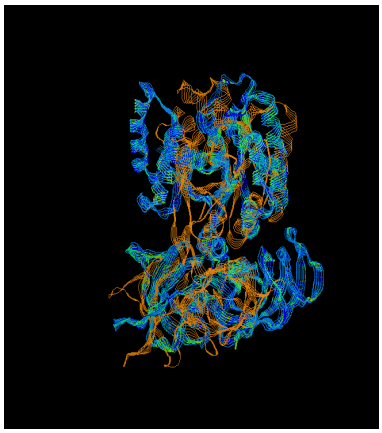
Η «pairwise» ευθυγράμμιση αφορά δύο ακολουθίες ενώ αν έχουμε παραπάνω από δύο το πρόβλημα καλείται πολλαπλή ευθυγράμμιση ακολουθιών (MSA). Στην ευθυγράμμιση ακολουθιών υπάρχουν δύο τύποι ευθυγράμμισης η τοπική(local) και η ολική(global). Η ολική βρίσκει τη βέλτιστης ομοιότητας ευθυγράμμιση δύο ακολουθιών καθ' όλο το μήκος τους ενώ η τοπική βρίσκει μικρού μήκους περιοχές ομοιότητας μεταξύ δύο ακολουθιών. Ένα παράδειγμα τοπικής και ολικής ευθυγράμισης φαίνεται στον πίνακα 2.1.

Global Alignment	Local Alignment
ACTGATTA ||| ||| ACT--TTA	ACTGATTA |||| --TGAT--

Πίνακας 2.1Ολική και τοπική ευθυγράμμιση[4]

Η πολλαπλή ευθυγράμμιση ακολουθιών είναι ολική ευθυγράμμιση, όμως υπάρχουν και αλγόριθμοι που χρησιμοποιούν τοπική ευθυγράμμιση για να αποδώσουν την ολική.

Τέλος αξίζει να σημειωθεί ότι όταν δύο ακολουθίες κρίνονται ως προς την ομοιότητα (similarity) τους , παρατηρούνται ή μετρώνται στοιχεία που είναι ίδια σε αυτές, ανεξάρτητα από την πηγή προέλευσης. Από την άλλη όταν δύο ακολουθίες είναι ομόλογες(homologous), αυτό σημαίνει ότι ανήκουν σε οργανισμούς που έχουν κάποιο κοινό πρόγονο(Εικόνα 2.3). Οι ομόλογες ακολουθίες εξάγονται από παρατηρήσεις σε όμοιες ακολουθίες[2].



[5]Εικόνα 2.3 Ομόλογη δομή

[4] Πηγή: http://www.isrec.isb-sib.ch/pro¯le/isrec96/uba alnbig.gif.

2.3 Διάφοροι αλγόριθμοι πολλαπλής ευθυγράμμισης (MSA)

Όπως είδη αναφέρθηκε στην υποενότητα 2.2.1 οι μοριακοί βιολόγοι συχνά χρησιμοποιούν την μέθοδο MSA για να εντοπίσουν περιοχές ομοιότητας στις ζητούμενες ακολουθίες. Ο δυναμικός προγραμματισμός και κατ' επέκταση η προοδευτική (progressive) μέθοδος είναι μια ευρέως χρησιμοποιούμενη τεχνική για τον υπολογισμό της MSA. Η μέθοδος «progressive» ξεκίνα την ευθυγράμμιση από τις κοντινότερες σε ομοιότητα ακολουθίες και σταδιακά βάζει τις μακρινότερες.

Παρ' όλα αυτά η επέκταση της τεχνικής αυτής για την ταυτόχρονη ευθυγράμμιση πολλαπλών ακολουθιών είναι μη πρακτική, καθώς η χρονική και χωρική πολυπλοκότητα εξαρτάται από το πλήθος και το μήκος των ακολουθιών. Όμως με την ολοένα και γρηγορότερη αύξηση του όγκου δεδομένων των βιολογικών ακολουθιών η ανάγκη για τον υπολογισμό των MSAs σε όσο το δυνατών λιγότερο χρόνο γίνεται μεγαλύτερη. Για το λόγο αυτό έχουν αναπτυχθεί πολλά προγράμματα υπολογισμού της MSA σε λογικά όρια χρόνου.

Κάποια από τα πιο αντιπροσωπευτικά παρουσιάζονται συνοπτικά παρακάτω:

ClustalW

Ο αλγόριθμος ClustalW [10], ο οποίος βελτιώνει σημαντικά την ευαισθησία της ευθυγράμμισης των διαφορετικών πρωτεϊνικών ακολουθιών, είναι μια από τις πιο δημοφιλείς μεθόδους για «progressive» πολλαπλή ευθυγράμμιση. Αρχικά, στην μέθοδο αυτή σε κάθε ακολουθία αποδίδονται ατομικά βάρη σε μια τμηματική

[5] Πηγή : http://amazingbeauty.org/nature/Fig 8.10-250.jpg

ευθυγράμμιση, με σκοπό την μείωση των βαρών στις ακολουθίες που παρουσιάζουν σημαντικές ομοιότητες και την αύξηση τους σε αυτές που αποκλίνουν σημαντικά. Δεύτερον, οι πίνακες αντικατάστασης αμινοξέων ποικίλουν στα διάφορα στάδια ευθυγράμμισης και αυτή η ποικιλομορφία εξαρτάται από το βαθμό απόκλισης των ακολουθιών που θα ευθυγραμμιστούν. Τρίτον, το κόστος κενών εναλλάσσεται δυναμικά ανάλογα με την δομή των ακολουθιών. Τέλος κατασκευάζεται ένα φυλογενετικό δένδρο (guide-tree) με την μέθοδο Neighbour-Joining.

MUSCLE

Ο αλγόριθμος αυτός[11] είναι μια επαναληπτική μέθοδος που χρησιμοποιεί το λεγόμενο «log-expectation» σύστημα σκορ με πληθώρα βελτιστοποιήσεις. Εξελίσσεται σε τρία στάδια: Το στάδιο «drag progressive» στο οποίο καθορίζεται η ομοιότητα σε ζευγάρια ακολουθιών διαμέσου ενός μετρητή κ-σταδίων, υπολογίζεται ένας πίνακας αποστάσεων(distancematrix), κατασκευάζεται ένα δένδρο με την μέθοδο UPGMA και τέλος γίνεται πολλαπλή ευθυγράμμιση, χρησιμοποιώντας την μέθοδο «drag progressive», με βάση το δένδρο. Το στάδιο βελτίωσης της progressive μεθόδου, όπου υπολογίζεται η ταυτοσημία της MSA, κατασκευάζεται νέο δένδρο και συγκρίνεται με το παλιό για την ύπαρξη βελτιστοποιήσεων. έλος το στάδιο αφαίρεσης όπου γίνεται διαγραφή κάθε κόμβου του δένδρου με profile επανευθυγράμμιση.

ProbCons

Ο ProbCons[12] είναι ένας εκπληκτικά ακριβής αλγόριθμος. Χρησιμοποιεί το μοντέλο Markov(HMM) και εκ των υστέρων πιθανότητας πίνακες, που συγκρίνουν τυχαία ζευγάρια ευθυγραμμισμένων ακολουθιών με τα προσδοκόμενα ζευγάρια ευθυγραμμισμένων ακολουθιών.Επιπλέον χρησιμοποιεί την πιθανότητα ακρίβειας μετατροπής, για την επανεκτίμηση των σκορ και κατασκευάζει ένα φυλογενετικό δένδρο με βάση το οποίο γίνεται η ευθυγράμμιση.

T-Coffee

Ο αλγόριθμος αυτός[13] παρουσιάζει δραματική βελτίωση στην ακρίβεια της MSA με μία μέτριου μεγέθους θυσία στην ταχύτητα, σε σχέση με τους άλλους αλγορίθμους. Και αυτή η μέθοδος βασίζεται στην «progressive» μέθοδο ευθυγράμμισης αλλά αποφεύγει τους πιο σοβαρούς κινδύνους που προκαλούνται από την πλεονάζουσα φύση της μεθόδου αυτής. Ο T-Coffee επανεξετάζει ένα σύνολο δεδομένων από όλες τις ολικές ευθυγραμμίσεις κατά ζεύγη. Αυτό παρέχει μια βιβλιοθήκη με πληροφορίες της ευθυγράμμισης που μπορούν να χρησιμοποιηθούν σαν οδηγός για την «progressive» ευθυγράμμιση. Η βιβλιοθήκη παρέχει συνέπεια και βοηθά στην πρόβλεψη λαθών αφού επιτρέπει να γίνει ορατό με πιο τρόπο η τελική ευθυγράμμιση θα είναι βέλτιστη. Τέλος ο T-Coffee μπορεί να ενώσει μαζί πολλαπλές μεθόδους σαν εξωτερικά τμήματα, φτιάχνοντας έτσι πιο επαρκείς βιβλιοθήκες από τα αποτελέσματα της κάθε μίας, αρκεί οι μέθοδοι αυτοί να είναι εγκατεστημένοι στο σύστημα.

MAFFT

Μια πρωτότυπη μέθοδος πολλαπλής ευθυγράμμισης ακολουθιών είναι ο αλγόριθμος MAFFT[14-15-16]. Ο MAFFT είναι ένας γρήγορος και ακριβής αλγόριθμος βασισμένος στον fast Fourier transform (FFT), ο οποίος επιτρέπει γρήγορη ανίχνευση ομόλογων τμημάτων. Προσφέρει αρκετές μεθόδους ευθυγράμμισης και είναι κατάλληλος τόσο για ακολουθίες πρωτεϊνών όσο και για DNA με σχετικά μεγάλο όγκο δεδομένων.

Παρ’ όλες τις παραπάνω μεθόδους σε αυτήν την εργασία επιλέχθηκε ο αλγόριθμος MAFFT που είναι μια από τις γρηγορότερες και ακριβέστερες μεθόδους, πετυχαίνει μια σημαντική μείωση του χρόνου CPU και η λειτουργία του βασίζεται σε ευριστικές μεθόδους.

3. Μελέτη του αλγορίθμου MAFFT

Στο κεφάλαιο αυτό περιγράφεται ο αλγόριθμος MAFFT και αναφέρεται η μελέτη του αλγορίθμου σε software.

3.1 Εισαγωγή

Η πολλαπλή ευθυγράμμιση ακολουθιών(MSA) είναι ένα σημαντικό βήμα για τις έρευνες πολλών τύπων σύγκρισης των μοριακών ακολουθιών. Η MSA χρησιμοποιείται στην φυλογενετική ανάλυση, στον ακριβή εντοπισμό κάποιων περιοχών, στην πρόβλεψη της δομής των ncRNAs και σε πολλές άλλες καταστάσεις. Παρ' όλα αυτά, είναι δύσκολο να γίνει σωστή ευθυγράμμιση για μακρινά συσχετισμένες ακολουθίες, χωρίς την επέμβαση κάποιου ειδικού. Έχουν γίνει πολλές προσπάθειες στο πρόβλημα αυτό που αφορά την βελτιστοποίηση της πολλαπλής ευθυγράμμισης. Ο Needleman, S.B. και ο Wunsch, C.D.[17] πρώτοι, παρουσίασαν έναν αλγόριθμο για τη σύγκριση ακολουθιών βασισμένο στον δυναμικό προγραμματισμό(DP), με τον οποίο επιτυγχάνεται η βέλτιστη ευθυγράμμιση ανάμεσα σε δύο ακολουθίες. Η γενίκευση όμως του αλγορίθμου στην πολλαπλή ευθυγράμμιση ακολουθιών[18], δεν μπορεί να εφαρμοστεί σε μία πρακτική ευθυγράμμιση της τάξεως των εκατοντάδων ή χιλιάδων ακολουθιών. Αυτό συμβαίνει γιατί απαιτείται τεράστιος χρόνος CPU ανάλογος του N^K, όπου K είναι ο αριθμός ακολουθιών, κάθε μία από τις οποίες έχει μήκος *N*. Για να

ξεπεραστεί αυτή η δυσκολία έχουν προταθεί μέχρι σήμερα πολλές ευριστικές μέθοδοι συμπεριλαμβανομένων των προοδευτικών (progressive)[19] και επαναληπτικών μεθόδων(iterative)[20-22]. Αυτές οι μέθοδοι βασίζονται κυρίως σε διαφορετικούς συνδυασμούς του επιτυχημένου δυναμικού προγραμματισμού δύο διαστάσεων, ο οποίος καταναλώνει χρόνο CPU ανάλογο του N^2.

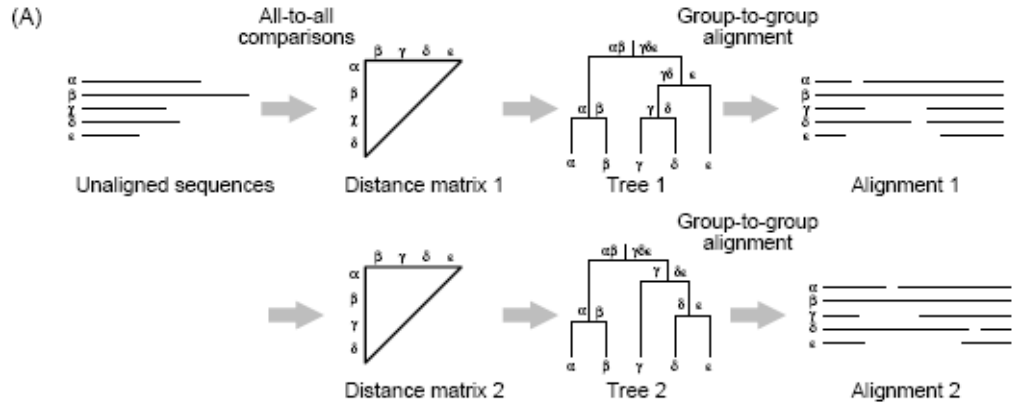
Ακόμα και αν αυτές οι ευριστικές μέθοδοι παρέχουν επιτυχώς την βέλτιστη ευθυγράμμιση, παραμένει ακόμα το πρόβλημα του αν αυτή η βέλτιστη ευθυγράμμιση ανταποκρίνεται στην σωστή, από βιολογικής πλευράς. Η ακρίβεια των ακολουθιών που προκύπτουν επηρεάζεται από το σύστημα σκορ(scoring system). Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο ο Thompson ανέπτυξε ένα πολύπλοκο σύστημα σκορ στο πρόγραμμα του CLUSTALW[10], στο οποίο τα κόστη για την εισαγωγή κενών (gap penalties) και άλλοι παράμετροι προσαρμόζονται προσεκτικά στα χαρακτηριστικά των ακολουθιών εισόδου, όπως στην διαφορετικότητα των ακολουθιών, στο μήκος κτλ. Παρ' όλα, αυτά κανένα από τα υπάρχοντα συστήματα βαθμονόμησης δεν είναι ικανό να παρουσιάσει σωστές ολικές ευθυγραμμίσεις για ποικίλα προβλήματα. Αξίζει να σημειωθεί επιπλέον, ότι έγιναν σημαντικές βελτιώσεις στην ακρίβεια στον CLUSTALW, το πιο λειτουργικό και με εύκολη πρόσβαση πρόγραμμα και στον T-COFFEE[13], ο οποίος παρέχει ευθυγραμμίσεις με πάρα πολύ μεγάλη ακρίβεια. Όμως, λίγες από τις βελτιώσεις πέτυχαν να μειώσουν το χρόνο CPU από τότε που προτάθηκε η προοδευτική μέθοδος από τους Feng και Doolittle[19]. Με την αύξηση όμως του αριθμού των μοριακών ακολουθιών γίνεται ολοένα και μεγαλύτερη η ανάγκη για ένα αρκετά γρήγορο πρόγραμμα εφαρμόσιμο σε μεγάλο εύρος προβλημάτων.

3.2 Ο αλγόριθμος MAFFT

Ο αλγόριθμος MAFFT δημοσιεύτηκε το 2002 από τους Katoh Kazutaka, Kazuharu Misawa, Kei-ichi Kuma και Takashi Miyata [14]. Πρόκειται για μία πρωτότυπη μέθοδο πολλαπλής ευθυγράμμισης βασισμένη στον fast Fourier transform(FFT), ο οποίος επιτρέπει την γρήγορη ανίχνευση ομόλογων τμημάτων. Παρά την μεγάλη αποδοτικότητα του ο FFT έχει χρησιμοποιηθεί ελάχιστα μέχρι τώρα για την εύρεση ομοιότητας στις ακολουθίες[23]. Ο MAFFT περιλαμβάνει δύο καινούργιες τεχνικές. Πρώτον οι ομόλογες περιοχές εντοπίζονται γρήγορα με τον fast Fourier transform(FFT). Δεύτερον προτείνεται ένα απλοποιημένο σύστημα σκορ, το οποίο λειτουργεί αποδοτικά στην μείωση του χρόνου της CPU και αυξάνει την ακρίβεια των ευθυγραμμίσεων ακόμα και για ακολουθίες με μεγάλες εισαγωγές ή επεκτάσεις όπως επίσης και για μακρινά συσχετισμένες ευθυγραμμίσεις όμοιου μήκους. Επιπλέον στον MAFFT εφαρμόζονται δύο διαφορετικές ευριστικές μέθοδοι για τον υπολογισμό της MSA: η προοδευτική μέθοδος (FFT-NS-2) και η βελτιστοποιημένη επαναληπτική μέθοδος (FFT-NS-i).

Στην παρούσα διπλωματική εργασία θα αναλυθεί η πρώτη μέθοδος (progressive) που είναι μια από της πιο συχνά χρησιμοποιούμενες μεθόδους στην πολλαπλή ευθυγράμμιση.

Η διαδικασία της μεθόδου που εφαρμόζεται στον MAFFT αναπαριστάται στην Εικόνα 3.1.



[6]Εικόνα 3.1Διαδικασία υπολογισμού της progressive μεθόδου.

Χρησιμοποιώντας τον αλγόριθμο FFT[23] και ένα κανονικοποιημένο πίνακα αντικατάστασης(similarity matrix), τα οποία θα αναλυθούν παρακάτω, οι ακολουθίες εισόδου ευθυγραμμίζονται προοδευτικά ακολουθώντας την σειρά των κλαδιών ενός δένδρου(guide tree) το οποίο χρησιμοποιείται μόνο για την ευθυγράμμιση. Αυτή η μέθοδος αναφέρεται ως FFT-NS-1.

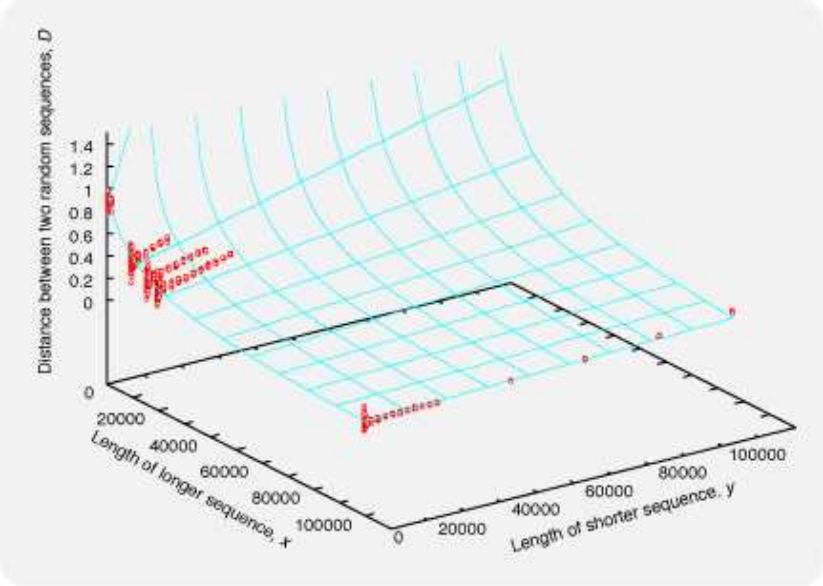
Σε αυτή τη μέθοδο το δένδρο που κατασκευάζεται, βασίζεται στις συγκρίσεις σε όλα τα ζεύγη των ακολουθιών εισόδου, τον πίνακα αποστάσεων δηλαδή (distance matrix), του οποίου ο χρόνος CPU είναι O(K^2), όπου K είναι ο αριθμός των ακολουθιών. Στην περίπτωση ενός μεγάλου K είναι σημαντικό ο υπολογισμός του πίνακα αποστάσεων να γίνεται γρήγορα[24]. Έτσι για την δημιουργία του πίνακα μετράται ο αριθμός από 6-tuples τα οποία μοιράζονται ανάμεσα σε κάθε ζευγάρι ακολουθιών. Αυτός ο αριθμός μετατρέπεται σε μία απόσταση D_{ij} ανάμεσα στην ακολουθία i και την ακολουθία j με την παρακάτω σχέση :

$$D_{ij} = 1 - [T_{ij} / \min(T_{ii}, T_{jj})] \ ,$$

όπου T_{ij} είναι ο αριθμός των 6-tuples τα οποία μοιράζονται ανάμεσα στην ακολουθία i και την ακολουθία j.

[6] Πηγή : A preprint of Katoh and Toh (2008, *Briefings in Bioinformatics* 9:286-298)

Η τιμή της απόστασης D_{ij} σε δύο ασυσχέτιστες ακολουθίες εξαρτάται από το μήκος των ακολουθιών αυτών. Επιπλέον, όταν συγκρίνουμε μια πολύ μικρή σε μήκος ακολουθία και μια πολύ μεγάλη ακολουθία, η τιμή της απόστασης D_{ij} βρίσκεται κατά τύχη κοντά στο μηδέν ακόμα και αν οι ακολουθίες είναι ασυσχέτιστες μεταξύ τους [25]. Παρακάτω φαίνονται οι αποστάσεις κάποιων τυχαίων ακολουθιών με διάφορα μήκη(Εικόνα 3.2) .



[7]Εικόνα 3.2

Το δένδρο(guide tree) κατασκευάζεται από τον πίνακα αποστάσεων με την μέθοδο UPGMA [26]. Τέλος υπολογίζεται η MSA χρησιμοποιώντας έναν group-to-group αλγόριθμο ευθυγράμμισης σε κάθε κόμβο του δένδρου.

Ο αρχικός πίνακας αποστάσεων είναι λιγότερο αξιόπιστος από αυτόν στον οποίο βασίζονται γενικά όλες οι ανά ζεύγη ευθυγραμμίσεις[27]. Για να επιτευχθεί λοιπόν λογική ισορροπία ανάμεσα σε ταχύτητα και ακρίβεια επαναλαμβάνονται τα βήματα της μεθόδου FFT-NS-1. Έτσι η προοδευτική ευθυγράμμιση εκτελείται ξανά, βασισμένη στο καινούργιο δένδρο το οποίο υπολογίζεται με βάση τον καινούργιο πίνακα αποστάσεων. Η μέθοδος αυτή αναφέρεται ως FFT-NS-2.

[7] Πηγή : http://align.bmr.kyushu-u.ac.jp/mafft/software/algorithms/algorithms.html

3.3 Ανάλυση των τεχνικών του MAFFT

3.3.1 Group-to-group ευθυγράμμιση με χρήση του FFT

Η συχνότητα αντικαταστάσεων των νουκλεϊκών οξέων εξαρτάται σημαντικά από την συχνότητα εμφάνισης των διάφορων αζωτούχων βάσεων τους και συγκεκριμένα της A - Αδενίνης (Adenine), G - Γουανίνης (Guanine), T - Θυμίνης (Thymine), C - Κυτοσίνης (Cytosine) και της U-Ουρακίλης(Uracil) αντί της Θυμίνης όταν πρόκειται για το RNA. Οι αντικαταστάσεις ανάμεσα σε όμοια ως προς τις αζωτούχες βάσεις νουκλεϊκά οξέα τείνουν να διατηρούν την δομή των γονιδιωμάτων και τέτοιες ουδέτερες αντικαταστάσεις έχουν συγκεντρωθεί στην μοριακή βιολογία κατά την διάρκεια της εξέλιξης[28]. Έτσι θεωρείται λογικό μία ακολουθία νουκλεϊκών οξέων να μετατραπεί σε ένα διάνυσμα με τέσσερις διαστάσεις του οποίου οι συνιστώσες είναι οι συχνότητες των αζωτούχων βάσεων (A,T(U),G,C).

Υπολογισμός της συσχέτισης ανάμεσα σε δύο ακολουθίες νουκλεϊκών οξέων:

Η συσχέτιση ανάμεσα σε δύο ακολουθίες νουκλεϊκών οξέων δίνεται από την σχέση:

$$c(k) = c_A(k) + c_T(k) + c_G(k) + c_C(k) \quad (1),$$

όπου $c_A(k)$, $c_T(k)$, $c_G(k)$ και $c_C(k)$ είναι οι συσχετίσεις της συχνότητας των αζωτούχων βάσεων ανάμεσα σε δύο ακολουθίες νουκλεϊκών οξέων που πρόκειται να ευθυγραμμιστούν. Η συσχέτιση $\mathrm{c(k)}$ αντιπροσωπεύει τον βαθμό ομοιότητας των δύο ακολουθιών με την αναλογική διαφορά k θέσεων. Η υψηλή τιμή του $\mathrm{c(k)}$ υποδηλώνει ότι οι ακολουθίες μπορεί να έχουν ομόλογες περιοχές.

Η συσχέτιση $\mathrm{c_A(k)}$ της συνιστώσας $\mathrm{f_A}$ (συχνότητα αδενίνης) ανάμεσα στην ακολουθία 1 και την ακολουθία 2 με την αναλογική διαφορά των k θέσεων ορίζεται ως :

$$c_A(k) = \sum_{1 \le n \le N, 1 \le n+k \le M} \hat{a}_1(n)\hat{a}_2(n+k) \quad (2) ,$$

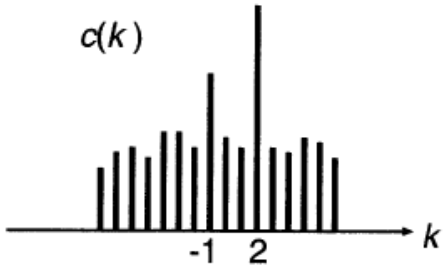
όπου $\hat{a}_1(n)$ και $\hat{a}_2(n)$ είναι οι f_A συνιστώσα της n-οστής θέσης της ακολουθίας 1 με μήκος N και αυτής της ακολουθίας 2 με μήκος M αντίστοιχα. Θεωρώντας ότι $M \simeq N$ σε πολλές περιπτώσεις, η εξίσωση 2 χρειάζεται $O(N^2)$ πράξεις. Ο FFT μειώνει τον χρόνο CPU αυτού του υπολογισμού σε $O(N \log N)$. Αν $A_1(m)$ και $A_2(m)$ είναι οι μετασχηματισμοί Fourier των $\hat{a}_1(n)$ και $\hat{a}_2(n)$ είναι γνωστό ότι η συσχέτιση $c_A(k)$ εκφράζεται ως :

$$c_A(k) \Leftrightarrow A^*_1(m)A_2(m) \quad (3) ,$$

όπου το βέλος αναπαριστά τον μετασχηματισμό Fourier και ο αστερίσκος υποδηλώνει συζυγή μιγαδικό αριθμό. Οι συσχετίσεις των $c_T(k)$, $c_G(k)$ και $c_C(k)$ υπολογίζονται με τον ίδιο τρόπο.

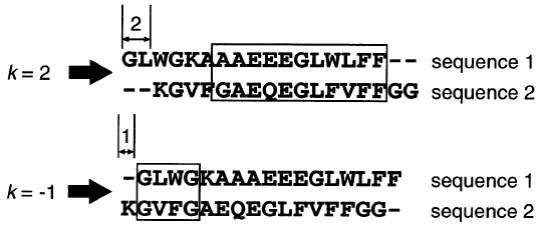
Εύρεση ομόλογων τμημάτων:

Αν δύο συγκρινόμενες ακολουθίες έχουν ομόλογες περιοχές τότε, η συσχέτιση $\mathrm{c(k)}$ έχει κάποιες κορυφές που αντιστοιχούν σε αυτές τις περιοχές (Εικόνα 3.3).



[8]Εικόνα 3.3:Αποτέλεσμα της ανάλυσης FFT.Υπάρχουν δύο κορυφές που αντιστοιχούν σε ομόλογες ομάδες.

Παρ' όλα αυτά με την χρήση του FFT μπορεί να γίνει γνωστή μόνο η τοπική διαφορά k μιας ομόλογης περιοχής ανάμεσα σε δύο ακολουθίες αλλά όχι η θέση της περιοχής. Όπως φαίνεται στην εικόνα 3.4, για τον καθορισμό της θέσης της ομόλογης περιοχής σε κάθε ακολουθία, εφαρμόζεται ολίσθηση παραθύρου στην οποία ο βαθμός των τοπικών ομολογιών υπολογίζεται για κάθε μια από τις τέσσερις υψηλότερες κορυφές στην συσχέτιση $c(k)$. Το μέγεθος του παραθύρου είναι 30 θέσεις. Εντοπίζουμε λοιπόν ένα τμήμα τριάντα θέσεων στο οποίο η τιμή του σκορ υπερβαίνει ένα δοσμένο κατώφλι(0.7 ανά θέση) ως ένα ομόλογο τμήμα. Αν εντοπιστούν δύο ή περισσότερα διαδοχικά τμήματα ως ομόλογα, συνδυάζονται σε ένα τμήμα μεγαλύτερου μήκους. Αν το μήκος του συνδυασμένου τμήματος είναι μεγαλύτερο από 150 θέσεις, τότε αυτό χωρίζεται σε επιμέρους τμήματα των 150 θέσεων το καθένα.

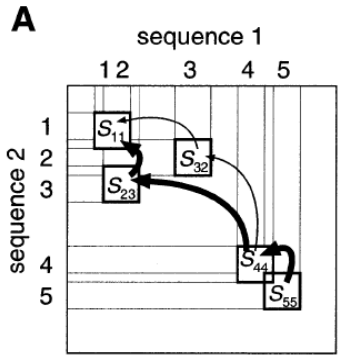


[9]Εικόνα 3.4: Καθορισμός των ομόλογων τμημάτων με ολίσθηση παραθύρου. Χάριν απλότητας το μέγεθος του παραθύρου είναι 4 θέσεις.

[8] Πηγή : MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform Katoh, Misawa, Kuma, Miyata (Nucleic Acids Res. **30**:3059-3066, 2002)

Διαίρεση του ομόλογου πίνακα:

Για να επιτευχθεί μια ευθυγράμμιση μέσα σε δύο ακολουθίες τα ομόλογα τμήματα πρέπει να τοποθετηθούν με συνέπεια και στις δύο ακολουθίες. Ένα πίνακας S_{ij} ($1 \le i$, $j \le n$, n είναι ο αριθμός των ομόλογων τμημάτων) κατασκευάζεται με τον ακόλουθο τρόπο: Αν το i-οστό ομόλογο τμήμα της ακολουθίας 1 αντιστοιχεί στο j-οστό τμήμα της ακολουθίας 2, τότε το S_{ij} παίρνει την τιμή του σκορ του ομόλογου τμήματος που υπολογίστε πιο πάνω αλλιώς το S_{ij} παίρνει την τιμή μηδέν. Εφαρμόζοντας την κλασική διαδικασία του δυναμικού προγραμματισμού στον πίνακα S_{ij} , επιτυγχάνεται το βέλτιστο μονοπάτι, το όποιο αντιστοιχεί στην βέλτιστη τοποθέτηση των ομόλογων τμημάτων. Η εικόνα 3.5 απεικονίζει ένα παράδειγμα στο οποίο υπάρχουν πέντε ομόλογα τμήματα. Η σειρά των τμημάτων στην ακολουθία 1 διαφέρει από αυτήν στην ακολουθία 2. Το βέλτιστο μονοπάτι εξαρτάται από τις τιμές των S_{23} και S_{32}. Αν $S_{23} > S_{32}$, το μονοπάτι με τα έντονα βέλη είναι το βέλτιστο.

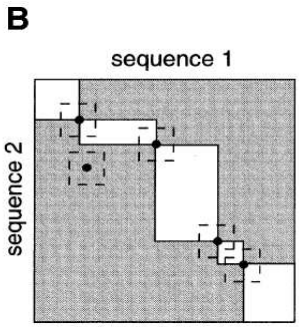


[10]Εικόνα 3.5: Ένα παράδειγμα του τμηματικού-επιπέδου DP

[9] Πηγή : MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform Katoh, Misawa, Kuma, Miyata (Nucleic Acids Res. **30**:3059-3066, 2002)

[10] Πηγή : MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform Katoh, Misawa, Kuma, Miyata (Nucleic Acids Res. **30**:3059-3066, 2002)

Γενικά, ο ομόλογος πίνακας χωρίζεται σε κάποιους υποπίνακες όπως φαίνεται στην εικόνα 3.6.



[11]Εικόνα 3.6: Μείωση της περιοχής του DP σε έναν ομόλογο πίνακα.

Έτσι αποκλείεται η σκιασμένη περιοχή από τον υπολογισμό. Καθώς πολλά ομόλογα τμήματα εντοπίζονται με την χρήση του FFT, ο χρόνος CPU μειώνεται.

Επέκταση σε group-to-group ευθυγραμμίσεις :

Η διαδικασία που περιγράφτηκε παραπάνω μπορεί εύκολα να επεκταθεί σε group-to-group ευθυγραμμίσεις θεωρώντας την εξίσωση (2) ως μία ειδική περίπτωση με μία ακολουθία στην ομάδα. Αυτή η εξίσωση μπορεί να επεκταθεί σε group-to-group ευθυγράμμιση αντικαθιστώντας το $\hat{a}_1(n)$ με το $\hat{a}_{group1}(n)$, το οποίο είναι ο γραμμικός συνδυασμός των συνιστωσών f_A των μελών που ανήκουν στην πρώτη ομάδα:

$$\hat{a}_{group1}(n) = \sum_{i \in group1} w_i \cdot \hat{a}_1(n) \quad (4),$$

[11] Πηγή : (10)

όπου w_i ο παράγοντας βάρους της ακολουθίας i, που υπολογίζεται με τον ίδιο τρόπο όπως στον CLUSTALW[10]. Το ίδιο ισχύει και για τις υπόλοιπες συσχετίσεις.

3.3.2 Σύστημα σκορ

Πριν ακολουθήσει η περιγραφή του συστήματος σκορ, που εφαρμόζεται από τον MAFFT είναι απαραίτητο για την καλύτερη κατανόηση του να καθοριστεί με σαφήνεια η έννοια του σκορ συστήματος.

Όπως αναφέρθηκε στο πρώτο κεφάλαιο, στις διάφορες μεθόδους πολλαπλής ευθυγράμμισης χρησιμοποιούνται κάποιοι αριθμοί-τιμές για να ορίσουμε πόσο βέλτιστη είναι η ευθυγράμμιση. Αυτές είναι τιμές για την ταύτιση κα την μη ταύτιση των νουκελϊκών οξέων και για την εισαγωγή κενών και προέρχονται από πίνακες αντικατάστασης. Το σκορ της ευθυγράμμισης εξαρτάται από αυτές τις τιμές καθώς και από το μήκος κάθε ακολουθίας.

Οι πίνακες αντικατάστασης δημιουργήθηκαν μετά από μελέτη της συχνότητας εμφάνισης των νουκλεϊκών οξέων στις ακολουθίες. Οι πίνακες αυτοί εμφανίζονται σε όλες τις συγκρίσεις που αφορούν ομοιότητες ακολουθιών και η επιλογή τους επηρεάζει σημαντικά το τελικό αποτέλεσμα.

Τέλος η εισαγωγή κενών και το κόστος εισαγωγής κενών είναι απαραίτητα για την καλύτερη δυνατή ευθυγράμμιση των ακολουθιών. Όμως χρησιμοποιώντας πολύ μεγάλο κόστος εισαγωγής κενών σε σχέση με το πεδίο τιμών των σκορ των πινάκων αντικατάστασης(αμινοξέων ή νουκλεϊνικών οξέων) τα κενά δεν θα εμφανίζονται ποτέ σε ευθυγράμμιση, ενώ από την άλλη αν είναι πολύ μικρό θα έχουμε πολλά κενά στην ευθυγράμμιση. Γι αυτό προγράμματα όπως τα BLAST[29] και FASTA[30] προτείνουν default τιμές για το κόστος εισαγωγής κενών.

Μετά την αναλυτική επεξήγηση του συστήματος σκορ ακολουθούν οι τεχνικές που χρησιμοποιούνται στον αλγόριθμο MAFFT.

Πίνακας αντικατάστασης :

Για να αυξηθεί η αποδοτικότητα της ευθυγράμμισης ,έγιναν επίσης μετατροπές στο σύστημα σκορ(πίνακες αντικατάστασης και κόστος εισαγωγής κενών). Ο Vogt et al[31] εισηγήθηκε ότι ο αλγόριθμος Needleman-Wuncsch(NW)[17] λειτουργεί καλά για όλους τους θετικούς πίνακες, στους οποίους όλα τα στοιχεία έχουν θετικές τιμές. Ο Vogt et al όμως εξέτασε μόνο τις περιπτώσεις όπου οι ακολουθίες είναι όμοιες σε μήκος. Έτσι δεν είναι ξεκάθαρο πότε αυτοί οι θετικοί πίνακες είναι κατάλληλοι στα ποικίλα προβλήματα ευθυγράμμισης, κυρίως στις περιπτώσεις που οι ακολουθίες έχουν διαφορετικό μήκος. Συνεπώς, αντίθετα με τις άλλες μεθόδους, υιοθετήθηκε ένας κανονικοποιημένος πίνακας αντικατάστασης M_{ab}(τα a και b είναι νουκλεϊκά οξέα) ο οποίος έχει θετικές και αρνητικές τιμές:

$$\hat{M}_{ab} = \left[\left(M_{ab} - average2 \right) / \left(average1 - average2 \right) \right] + S^a \quad (5),$$

όπου $average1 = \sum_a f_a M_{aa}$, $average2 = \sum_{a,b} f_a f_b M_{ab}$, M_{ab} είναι ο πρότυπος πίνακας αντικατάστασης, f_a είναι η συχνότητα εμφάνισης του νουκλεϊκού οξέος α και S^a είναι μία παράμετρος που λειτουργεί ως το κόστος των επιπρόσθετων κενών. Σε αυτόν τον πίνακα αντικατάστασης $\hat{M}_{ab}$, το σκορ ανά θέση σε δύο τυχαίες ακολουθίες είναι S^a και το σκορ ανά θέση σε δύο πανομοιότυπες ακολουθίες είναι 1.0 + S^a. Αν το S^a είναι πολύ μικρότερο από την μονάδα, τα κενά ουσιαστικά βαθμολογούνται όπως στις τυχαίες ακολουθίες.

Οι εξορισμού παράμετροι του προγράμματος MAFFT για ακολουθίες νουκλεϊκών οξέων είναι οι εξής: ο πίνακας M_{ab} είναι ο «200PAM log-odds» πίνακας που υπολογίστηκε από το μοντέλο δύο παραμέτρων του Kimura[28], με μεταβατική/αντίστροφη αναλογία ίση με 2.0, συχνότητα εμφάνισης f_a 0.25, S^{op} (κόστος κενών που βρίσκονται στην αρχή της ακολουθίας) 2.4 και S^a ίσο με 0.06.

Κόστος κενών(gap penalty):

Ο πίνακας ομοιότητας $H(i, j)$ ανάμεσα σε δύο ακολουθίες νουκλεϊκών οξέων $A(i)$ και $B(j)$ προκύπτει από τον πίνακα αντικατάστασης ως εξής: $H(i, j) = \hat{M}_{A(i)B(j)}$, όπου τα i και j είναι θέσεις στις ακολουθίες. Όταν δύο ομάδες ακολουθιών ευθυγραμμίζονται, ο ομόλογος πίνακας ανάμεσα στην ομάδα 1 και την ομάδα 2 υπολογίζεται ως εξής:

$$H(i, j) = \sum_{n \in group1, m \in group2} w_n w_m \hat{M}_{A(n,i)B(m,j)} \quad (6),$$

όπου το $A(n, i)$ υποδηλώνει την i-οστή θέση της n-οστής ακολουθίας στην ομάδα 1, το $B(m, j)$ είναι η j-οστή θέση της m-οστής ακολουθίας στην ομάδα 2 και w_n είναι ο παράγοντας βάρους που καθορίστηκε προηγουμένως για την n-οστή ακολουθία.

Στον αλγόριθμο NW[17], η βέλτιστη ευθυγράμμιση ανάμεσα σε δύο ομάδες ακολουθιών υπολογίζεται ως εξής :

$$P(i, j) = H(i, j) + \max \begin{cases} P(i-1, j-1) \\ P(x, j-1) - G_1(i, x) \left(1 \le x < i-1\right) \\ P(i-1, y) - G_2(j, y) (1 \le y < j-1) \end{cases}, \quad (7)$$

όπου $P(i, j)$ είναι το συναθροισμένο σκορ για το βέλτιστο μονοπάτι από το (1,1) στο (i,j) και $G_1(i, x)$ και $G_2(j, y)$ είναι τα κόστη κενών που θα οριστούν παρακάτω. Κάθε ομάδα από ακολουθίες μπορεί να περιέχει τα κενά που έχουν ήδη εισαχθεί στα προηγούμενα βήματα. Αν ένα κενό εισέλθει πρόσφατα στην ίδια θέση με κάποιο από τα υπάρχοντα κενά, τότε στο νέο κενό δεν πρέπει να προστεθεί κάποιο κόστος, επειδή το καινούργιο και τα υπόλοιπα κενά είναι πιθανότατα αποτέλεσμα ενός μοναδικού γεγονότος εισαγωγής ή διαγραφής. Τα κόστη των κενών ορίζονται ως εξής:

$$G_1(i, x) = S^{op} \cdot \left\{1 - \left[g_1^{start}(x) + g_1^{end}(i) \right] / 2 \right\} \quad (8),$$

όπου το S^{op} αντιστοιχεί σε ένα κόστος αυτό των εισαγωγικών κενών, το $g_1^{start}(x)$ είναι ο αριθμός των κενών που ξεκινούν στην χ-οστή θέση και το $g_1^{end}(i)$ είναι ο αριθμός των κενών που καταλήγουν στην i-οστή θέση, τα οποία υπολογίζονται από τις σχέσεις:

$$g_1^{start}(x) = \sum_{m \in group1} w_m \cdot a_m(x) \cdot z_m(x+1) \quad (9),$$

$$g_1^{end}(i) = \sum_{m \in group1} w_m \cdot z_m(i-1) \cdot a_m(i) \quad (10),$$

όπου $z_m(i)$=1 και $a_m(i)$=0, αν η i-οστή θέση της ακολουθίας m είναι κενό αλλιώς $z_m(i)$=0 και $a_m(i)$=1. Το w_m είναι ο συντελεστής του βάρους της ακολουθίας m. Το άλλο κόστος $G_2(j, y)$ υπολογίζεται με τον ίδιο τρόπο.

Συνεπώς, επειδή αυτός ο σχηματισμός είναι απλούστερος από αυτούς που υπάρχουν[10,22], ο χρόνος CPU μειώνεται σημαντικά αλλά η ακρίβεια της ευθυγράμμισης είναι συγκρίσιμη με αυτές που παρέχουν τα υπάρχοντα συστήματα σκορ.

3.4 Μελέτη του αλγορίθμου MAFFT

Για την μελέτη του αλγορίθμου χρησιμοποιήθηκε το λογισμικό, σε γλώσσα προγραμματισμού C που είναι διαθέσιμο στο διαδίκτυο[32] για το πρόγραμμα MAFFT. Είναι η έκδοση 6.712 η οποία είναι η επίσημη διανομή του προγράμματος. Στόχος ήταν η κατανόηση του αλγορίθμου αλλά και η χρήση αυτού ως βάση κατά την σχεδιαστική διαδικασία στη φάση της υλοποίησης του σε αναδιατασσόμενη λογική.

Η είσοδος του αλγορίθμου είναι ένα αρχείο txt το οποίο περιέχει έναν αριθμό από ακολουθίες νουκλεϊκών οξέων, συνήθως διαφορετικού μήκους και οι οποίες διαβάζονται από το πρόγραμμα ώστε να ξεκινήσει η διαδικασία της ευθυγράμμισης. Η δομή του αρχείου αυτού περιλαμβάνει για κάθε ακολουθία την ονομασία της και την αλληλουχία των αζωτούχων βάσεων της.

Για να υπάρχει μια πιο πλήρης άποψη όσο αφορά την απόδοση του προγράμματος αυτού, έγιναν μετρήσεις του χρόνου εκτέλεσης με το εργαλείο V-Tune Performance Analyzer for Linux (έκδοση 9.1) της Intel έτσι ώστε να βρεθεί το υπολογιστικά ακριβό κομμάτι του αλγορίθμου. Έτσι προέκυψε ο παρακάτω πίνακας(πίνακας 3.1) ο οποίος περιέχει την υπολογιστική συμπεριφορά του αλγορίθμου και συγκεκριμένα το ποσοστό που καταναλώνει η πιο ακριβή υπολογιστικά συνάρτηση, η οποία βρίσκεται στο $3^{ο}$ βήμα του αλγορίθμου και είναι η A__align.

Ακολουθίες εισόδου(διαστάσεις)			Συνολικός υπολογιστικός φόρτος (PC)	Υπολογιστικός φόρτος(PC) A__align
Όνομα	Πλήθος	Μέγιστο Μήκος		
flydna10x766	10	766	43%	77%
ex10x766	10	766	40%	85%
flydna100x766	100	766	49%	81%
ex100x766	100	766	45%	84%
ex10x1464	10	1464	57%	87%
flydna100x1403	100	1403	49%	79%

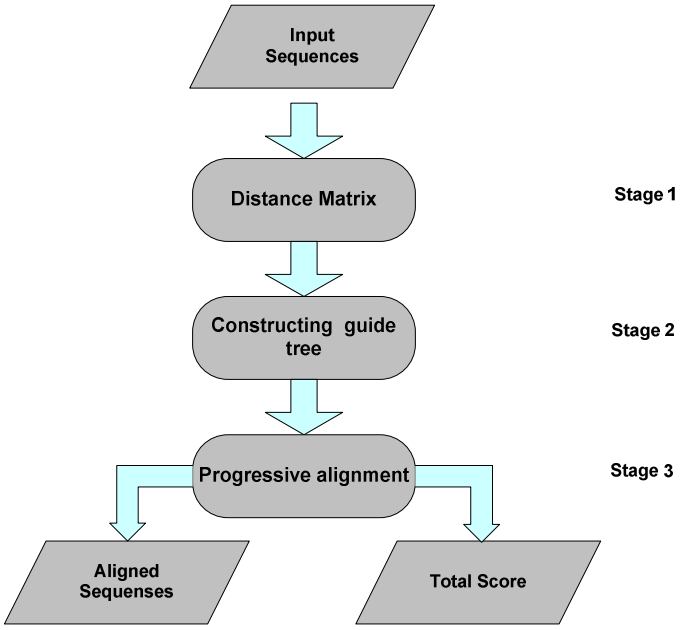
Πίνακας 3-1: Υπολογιστική συμπεριφορά MAFFT

4. Σχεδίαση και Υλοποίηση της Αρχιτεκτονικής για τον Αλγόριθμο MAFFT

Στο κεφάλαιο αυτό παρουσιάζεται η αρχιτεκτονική του αλγορίθμου MAFFT, με ιδιαίτερη έμφαση στην ανάλυση των βασικών υπολογιστικών κομματιών και στις σχεδιαστικές επιλογές που ακολουθήθηκαν.

4.1 Εισαγωγή στην σχεδίαση και την υλοποίηση

Ο αλγόριθμος MAFFT αποτελείται από τρία βασικά βήματα τα οποία φαίνονται στο παρακάτω σχήμα(Σχήμα 4-1) :

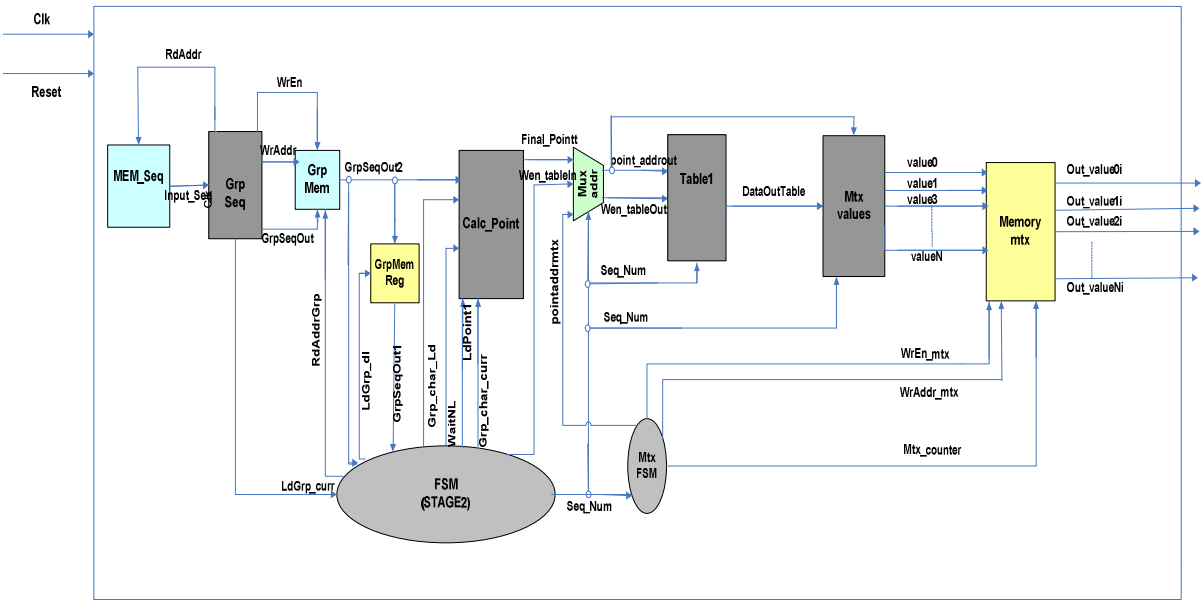


Σχήμα 4-1:Διάγραμμα ροής του αλγορίθμου MAFFT

Στην παρούσα διπλωματική εργασία σχεδιάστηκε και υλοποιήθηκε μια αρχιτεκτονική που υλοποιεί το πρώτο(stage1), το οποίο καταναλώνει ένα μικρό ποσοστό του υπολογιστικού φόρτου και ένα μέρος από το τρίτο(stage3) βήμα του αλγορίθμου, το οποίο είναι και το πιο ακριβό υπολογιστικά κομμάτι(πίνακας 3.1), δηλαδή τον πίνακα αποστάσεων και την διαδικασία κατά την οποία ευθυγραμμίζονται οι ακολουθίες και υπολογίζεται το σκορ της ευθυγράμμισης. Η υλοποίηση έγινε σε γλώσσα VHDL και χρησιμοποιήθηκε το εργαλείο Xilinx Ise 10.1.

4.2 Ο πίνακας αποστάσεων

Η αρχιτεκτονική του τμήματος αυτού δέχεται ως είσοδο έναν αριθμό από ακολουθίες οι οποίες είναι αποθηκευμένες σε μία μνήμη, το σήμα ρολογιού και το σήμα Reset. Η έξοδος αποτελείται από τα δεδομένα του πίνακα αποστάσεων τα οποία είναι σήματα των 9 bit. Η αρχιτεκτονική της σχεδίασης που υπολογίζει τον πίνακα αποστάσεων φαίνεται παρακάτω(Σχήμα 4-2).



Σχήμα 4-2: Αρχιτεκτονική του τμήματος που υπολογίζει τον πίνακα αποστάσεων.

4.2.1 Βασικές δομικές μονάδες της σχεδίασης και της υλοποίησης

Αριθμητικές πράξεις :

Οι αριθμητικές πράξεις που εκτελούνται σε αυτό το στάδιο του αλγορίθμου είναι πράξεις μεταξύ ακεραίων. Οι προσθέσεις και οι αφαιρέσεις εκτελούνται με χρήση των τελεστών «+» και «-» ενώ η πράξη του πολλαπλασιασμού εκτελείται με ολίσθηση προς τα δεξιά, μιας και ο πολλαπλασιασμός συμβαίνει μεταξύ ενός ακεραίου και μιας σταθεράς, που είναι σε δύναμη του δύο.

Δεδομένα εισόδου:

Η μνήμη με την ονομασία MEM_Seq περιέχει όλες τις προς σύγκριση ακολουθίες νουκλεϊκών οξέων, που αποτελούν την είσοδο του συστήματος. Οι

χαρακτήρες των ακολουθιών αυτών μπορεί να πάρουν τις τιμές a, g, c, t, u, A, G, C,T,U,- ενώ συμπεριλαμβάνονται και οι χαρακτήρες n,N,b,d,h,k,m,n,r,s,v,w,y,x,O για τον έλεγχο παράνομων χαρακτήρων. Οι ακολουθίες βρίσκονται σε ένα αρχείο txt με την μορφή FASTA(ενότητα 3.3.2). Για την ανάγνωση του αρχείου αυτού δημιουργήθηκε μια εφαρμογή σε γλώσσα προγραμματισμού C κατά την οποία επιλέγονται όλες οι χρήσιμες πληροφορίες, μετατρέπονται σε δυαδική μορφή, σύμφωνα με τον πίνακα 4.1 και αποθηκεύονται σε ένα νέο αρχείο με κατάλληλο τρόπο ώστε να έχει την μορφή coe και να χρησιμοποιηθεί για την αρχικοποίηση της μνήμης.

a	**11111**
g	**00001**
c	**00010**
t	**00100**
u	**00101**
A	**00110**
G	**00111**
C	**00011**
T	**01000**
U	**01001**
n	**01010**
N	**01011**
b	**01100**
d	**01101**
h	**01110**
k	**01111**
m	**10000**
r	**10010**
s	**10011**
V	**10100**
w	**10101**
y	**10110**
x	**10111**
-	**11000**
O	**11001**

Πίνακας 4.1: Δυαδική αναπαράσταση στοιχείων ακολουθίας.

Η μνήμη MEM_Seq είναι μια single port ROM και έχει βάθος 640 θέσεων όπου κάθε θέση είναι 200 bits. Θεωρούμε ότι κάθε ακολουθία καταλαμβάνει 10 θέσεις τις μνήμης, όπου κάθε στοιχείο είναι μια ομάδα από 5 bits, ώστε να καλύπτονται οι περισσότερες περιπτώσεις σε σχέση με το μήκος της κάθε ακολουθίας. Το τέλος κάθε ακολουθίας σηματοδοτείται με την δυαδική αναπαράσταση «00000». Έτσι ο αριθμός 640 αντιστοιχεί σε 64 ακολουθίες για τον ίδιο ακριβώς λόγο.

Κωδικοποίηση των χαρακτήρων σε αριθμούς:

Κατά την διαδικασία ανάγνωσης της εισόδου οι χαρακτήρες κάθε ακολουθίας κωδικοποιούνται ανάλογα με την σειρά που δηλώνονται (" a,g,c t,u,A,G,C,T,U,n, N,b,d,h,k,m,n,r,s,v,w,y,x-O") σε δυαδική μορφή. Η δυαδική αναπαράσταση των κωδικοποιημένων χαρακτήρων φαίνεται παρακάτω:

a	**000**
g	**001**
c	**010**
t	**011**
u	**011**
A	**000**
G	**001**
C	**010**
T	**011**
U	**011**

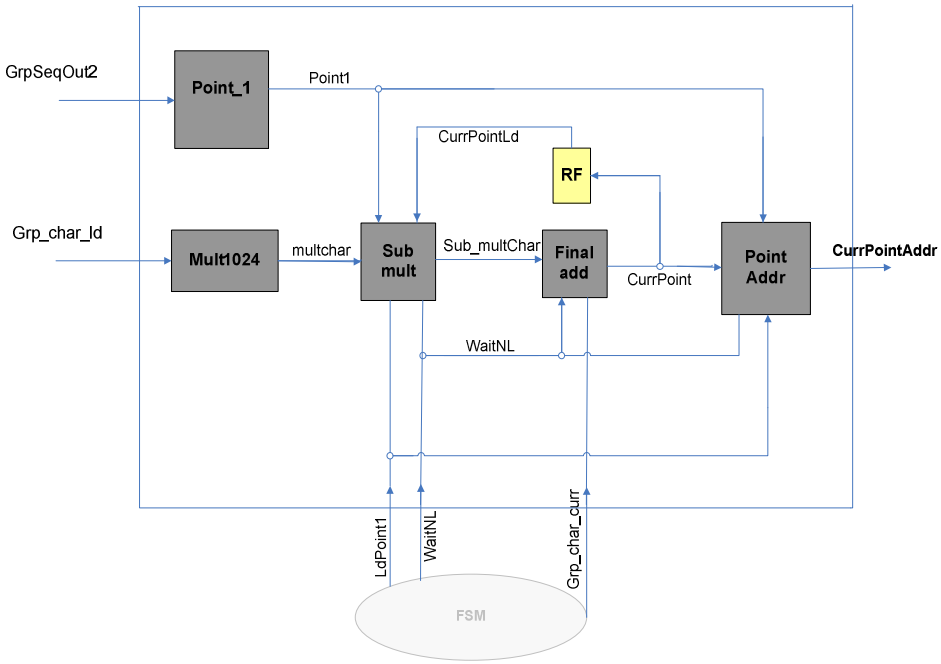
Πίνακας 4.2: Κωδικοποίηση των χαρακτήρων της ακολουθίας

Οι υπόλοιποι χαρακτήρες κωδικοποιούνται σε «101» το οποίο αντιστοιχεί σε λάθος χαρακτήρα ενώ το τέλος κάθε ακολουθίας αντιστοιχίζεται στο «111». Αυτή η κωδικοποίηση υλοποιείται με την μονάδα Grp_Seq κατά την οποία σε κάθε κύκλο διαβάζονται τα δεδομένα της μνήμης MEM_Seq, ελέγχονται ξεχωριστά οι χαρακτήρες κάθε ακολουθίας και μετατρέπονται στην αντίστοιχη κωδικοποίηση.

Οι νέες κωδικοποιημένες ακολουθίες αποθηκεύονται στην μνήμη Grp_mem**.** Η μνήμη αυτή είναι μια dual port RAM και έχει βάθος 640 θέσεων όπου κάθε θέση είναι120 bits. Τα δεδομένα γράφονται με την ίδια λογική όπως στην μνήμη MEM_Seq.

Υπολογισμός της ακριβής θέσης και της συχνότητας των tuples:

Κατά την διαδικασία δημιουργίας του πίνακα αποστάσεων υπολογίζεται η συχνότητα εμφάνισης των tuples, μήκους κ για κάθε ακολουθία. Για τον υπολογισμό αυτό χρησιμοποιείται η μέθοδος ολίσθησης παραθύρου εύρους κ, πάνω στις ακολουθίες και έτσι μετράται πόσες φορές εμφανίζονται τα κ-tuples. Στην περίπτωση του αλγορίθμου MAFFT το μήκος των tuples είναι 6. Η μονάδα Calc_Point υπολογίζει σε κάθε κύκλο μία τιμή για κάθε 6 tuples, η οποία καθορίζει την μοναδικότητα του. Η αρχιτεκτονική της μονάδας αυτής φαίνεται αναλυτικότερα στο σχήμα 4-3



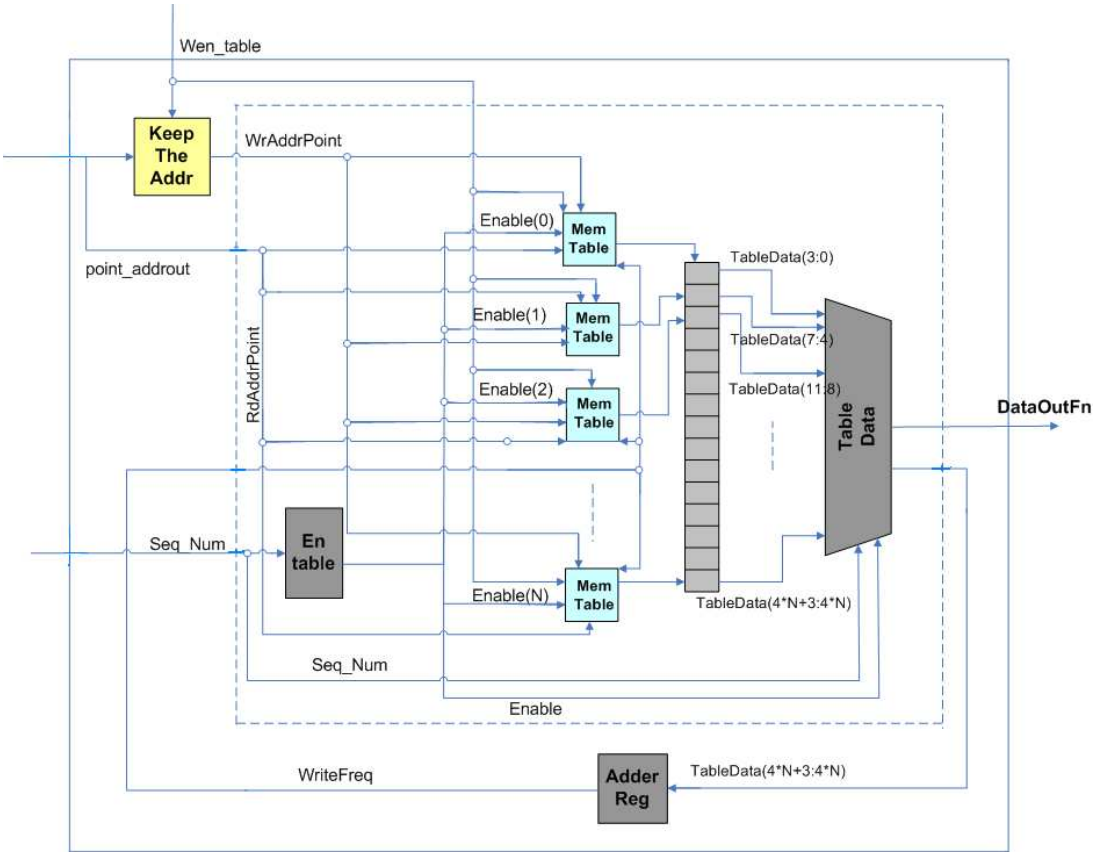
Σχήμα 4-3: Σχεδίαση αρχιτεκτονικής της μονάδας Calc_Point

Αξίζει να σημειωθεί, ότι οι πράξεις του πολλαπλασιασμού έγιναν με ολίσθηση προς τα δεξιά κατά μία δύναμη του δύο, καθώς κάθε κωδικοποιημένος χαρακτήρας κάθε ακολουθίας πολλαπλασιάζεται με μία σταθερά(1024,256,64, 16,4) ανάλογα με την θέση που κατέχει στην ακολουθία.

Υπολογισμός συχνότητας εμφάνισης των tuples:

Για τον υπολογισμό της συχνότητας εμφάνισης όλων των ομάδων των 6-tuples σε μία ακολουθία νουκλεϊκών οξέων, θα πρέπει να υπολογιστεί ο αριθμός εμφάνισης 4096 πιθανών tuples.Έτσι κάθε ακολουθία μπορεί να αποθηκευτεί σε μία μνήμη που περιέχει 4^6 θέσεις, κάθε μια από τις οποίες αντιπροσωπεύει την συχνότητα του αντίστοιχου tuple.

Η μονάδα table (Σχήμα 4-4) δέχεται σε κάθε κύκλο ως είσοδο την τιμή από μία ομάδα από 6-tuples μίας ακολουθίας, την Final_point_addr (έξοδος της μονάδας Calc_Point). Η τιμή αυτή χρησιμεύει ως διεύθυνση εγγραφής της μνήμης mem_table, στην οποία αποθηκεύεται η συχνότητα εμφάνισης της συγκεκριμένης τιμής με την βοήθεια ενός αθροιστή add_reg. Οι συχνότητες εμφάνισης λοιπόν των tuples για κάθε μια από τις ακολουθίες αποθηκεύεται στη μνήμη mem_table τύπου dual port RAM, που έχει βάθος 4096 θέσεων και κάθε θέση είναι 4 bits. Ο αριθμός των μνημών εξαρτάται από το πλήθος των ακολουθιών εισόδου.



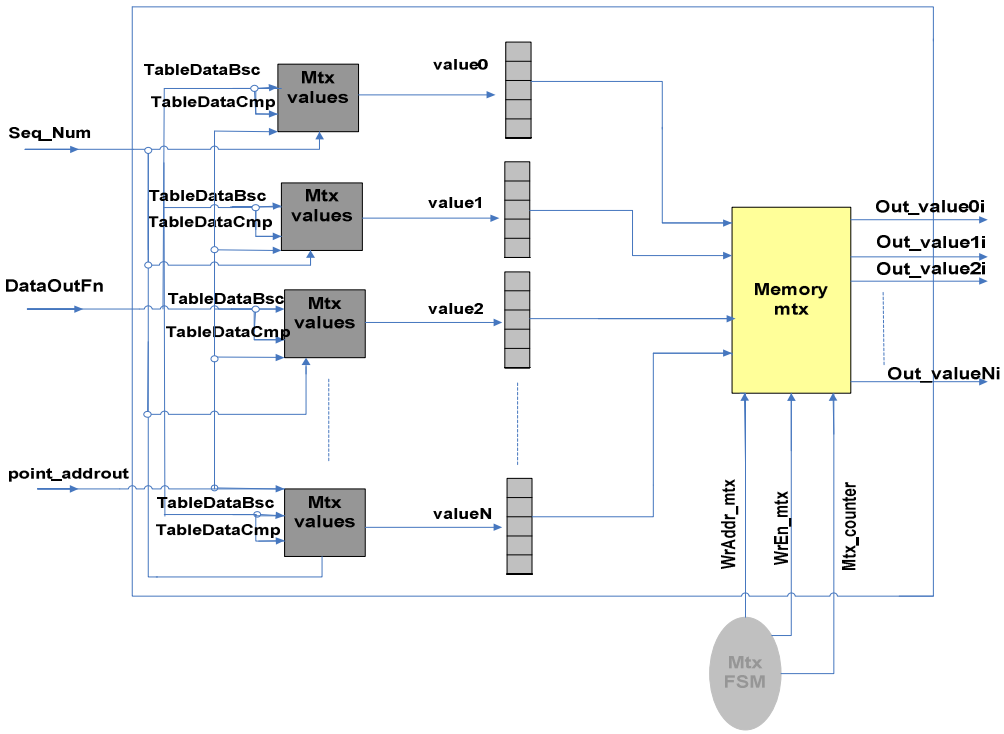
Σχήμα 4-4: Σχεδίαση αρχιτεκτονικής της μονάδας Table

Υπολογισμός πίνακα αποστάσεων:

Τέλος υπολογίζεται ο πίνακας αποστάσεων ο οποίος είναι ένας τριγωνικός πίνακας NxN, οπού N το πλήθος των ακολουθιών, στον οποίο κάθε θέση περιέχει την συνολική συχνότητα εμφάνισης των 6-tuples ανάμεσα στην ακολουθία i και στην ακολουθία j. Αυτό υλοποιείται με την μονάδα Mtx_values η οποία αρχικά, διαβάζει με την σειρά και σε κάθε κύκλο τα δεδομένα της μνήμης mem_table με βάση τις διευθύνσεις που δίνονται από την μονάδα mtx(FSM). Η έξοδος της μνήμης δίνει την συχνότητα των 6-tuples για όλες τις ακολουθίες στην συγκεκριμένη θέση. Στην συνέχεια επιλέγοντας κάθε φορά ως βάση μία ακολουθία γίνεται σύγκρισή της συχνότητα των 6-tuples αυτής της ακολουθίας με

την συχνότητα των 6-tuples των υπόλοιπων ακολουθιών και αν αυτές οι συχνότητες αυτές είναι μικρότερες από αυτήν της βάσης γίνεται πρόσθεση αυτών των τιμών(μονάδα value_matrix).

Για τον λόγο ότι στην VHDL δεν μπορεί να γίνει αναπαράσταση δισδιάστατων πινάκων, τα δεδομένα που υπολογίστηκαν από την μονάδα Mtx_values αποθηκεύονται σε N ξεχωριστές μνήμες, που στην συγκεκριμένη περίπτωσή είναι 64, με διεύθυνση και σήμα ενεργοποίησης που προέρχεται από την μονάδα mtx(FSM) .Οι μνήμες mem_mtx είναι τύπου dual port RAM , έχουν βάθος 64 bits (όσο και το πλήθος των ακολουθιών) και κάθε θέση είναι 9 bits, τα οποία αντιστοιχούν στην συνολική συχνότητα εμφάνισης των 6 tuples ανάμεσα σε δύο ακολουθίες. Κάθε μνήμη περιέχει τα δεδομένα της αντίστοιχης στήλης του πίνακα αποστάσεων. Η αρχιτεκτονική του τμήματος αυτού φαίνεται παρακάτω(Σχήμα 4-5):



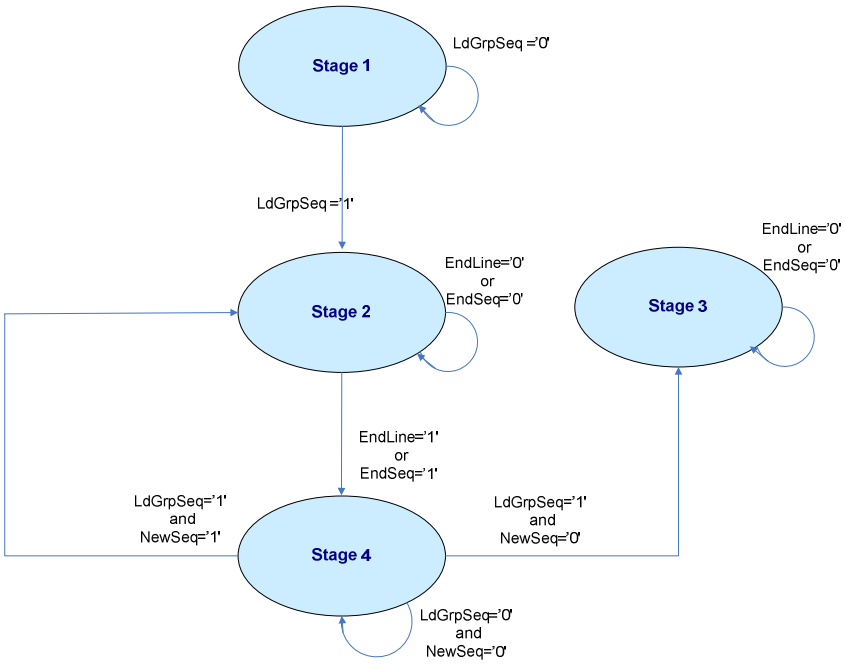
Σχήμα 4-5: Σχεδίαση αρχιτεκτονικής της μονάδας Table Mtx_values.

Μονάδα ελέγχου:

Για να λειτουργήσουν σωστά οι μονάδες που απαρτίζουν την αρχιτεκτονική του τμήματος αυτού ρυθμίζονται από την μονάδα ελέγχου(Σχήμα 4-6). Η μονάδα αυτή υλοποιείται ως μία μηχανή πεπερασμένων καταστάσεων με την οποία καθορίζονται τα σωστά σήματα για την λειτουργία των μνημών, των μονάδων εκτέλεσης αριθμητικών πράξεων, των πολυπλεκτών και γενικά όλων των βασικών υπολογιστικών μονάδων.

Η βασική FSM για το συγκεκριμένο κομμάτι αποτελείται από 4 στάδια:

- Το πρώτο στάδιο είναι η κατάσταση αρχικοποίησης κατά την οποία αρχικοποιούνται όλα τα σήματα ελέγχου στην σωστή τιμή.
- Το δεύτερο και το τρίτο στάδιο είναι ουσιαστικά αυτά που καθορίζουν το βασικό υπολογιστικό κομμάτι όπου γίνεται σε κάθε κύκλο ανάγνωση της μνήμης και διακρίνεται αν πρόκειται για καινούργια ή την τρέχουσα ακολουθία, καθώς κάθε ακολουθία καταλαμβάνει 10 θέσης μνήμης. Το δεύτερο στάδιο αφορά την άφιξη της καινούργιας ακολουθίας ενώ το τρίτο την άφιξη κάθε επόμενης. Και τα δύο στάδια ρυθμίζουν τα απαραίτητα σήματα για τον υπολογισμό του πίνακα αποστάσεως, διαφορετικά για κάθε περίπτωση.
- Το τέταρτο στάδιο είναι η κατάσταση στην οποία γίνεται μετάβαση όταν δεν υπάρχουν άλλα δεδομένα για κάθε ακολουθία, καθώς οι ακολουθίες δεν είναι σταθερού μήκους και δεν είναι δυνατή η δυναμική αναδιάταξη του χώρου. Το στάδιο αυτό ενεργοποιεί και την επιμέρους FSM όταν δεν υπάρχουν άλλες ακολουθίες ώστε να γίνουν οι εγγραφές στις μνήμες mem_mtx.



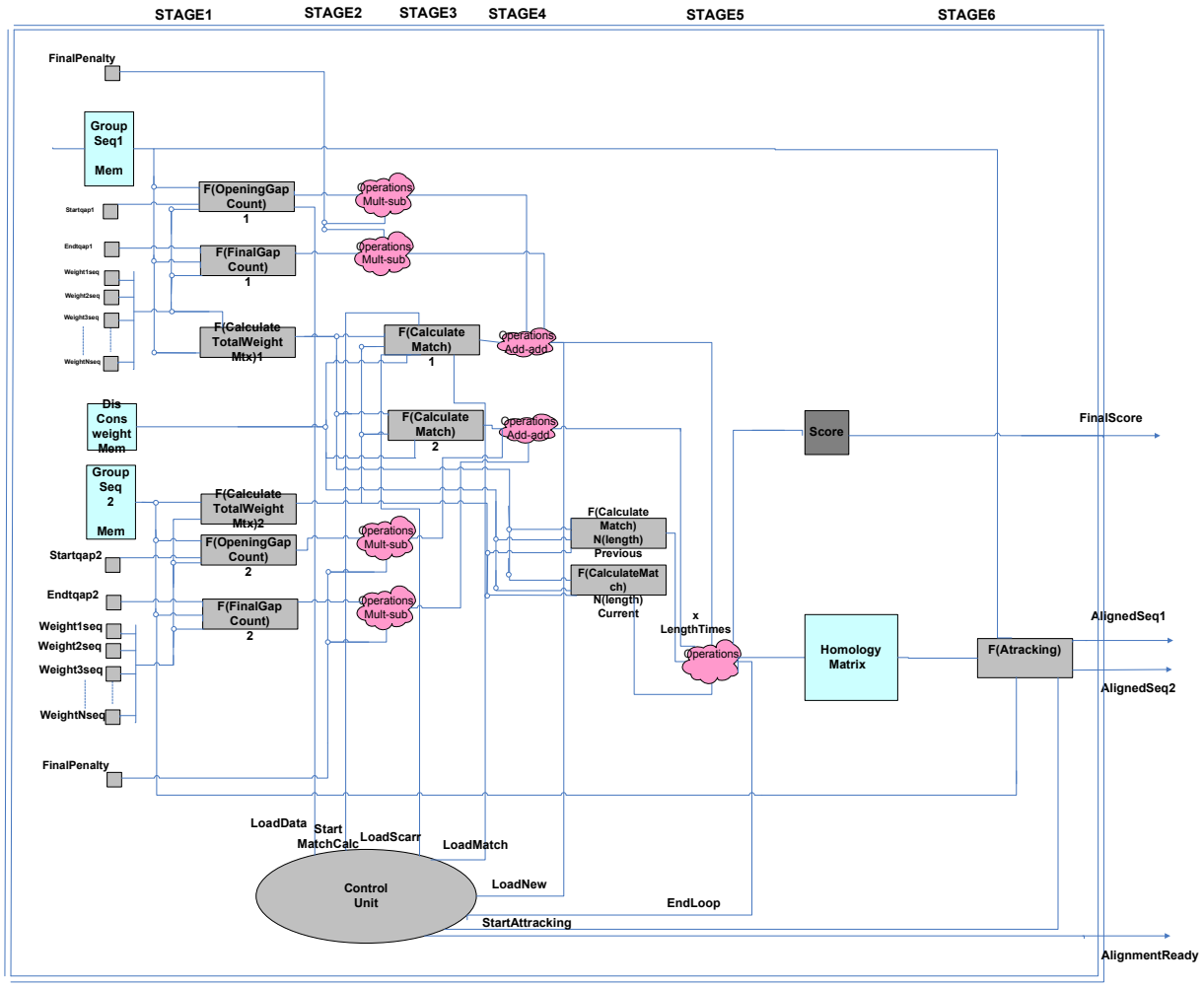
Σχήμα 4-6: Μονάδα ελέγχου του πίνακα αποστάσεων

4.3 Ευθυγράμμιση και υπολογισμός του σκορ

Στο τμήμα αυτό σχεδιάστηκε και υλοποιήθηκε μία αρχιτεκτονική που υλοποιεί την συνάρτηση A__align, η οποία αποτελεί το τελευταίο στάδιο του αλγορίθμου MAFFT. Η συνάρτηση A__align καλείται τόσες φορές όσο το πλήθος των ακολουθιών εισόδου. Κάθε φορά δημιουργούνται 2 ομάδες από τις πιό πιθανές ομόλογες περιοχές των προς σύγκριση ακολουθιών (clusters), με βάση την τοπολογία του δένδρου και με τη χρήση του FFT. Κάθε ομάδα περιέχει τις πιθανολογούμενες ομόλογες περιοχές για ένα πλήθος από τις ακολουθίες. Το πλήθος τμημάτων ακολουθιών κάθε πακέτου εξαρτάται κυρίως από το πλήθος των ακολουθιών εισόδου.

Οι είσοδοι λοιπόν της κύριας σχεδιαστικής μονάδας είναι δύο σύνολα από τμήματα της εκάστοτε ακολουθίας (seq1 και seq2) όπου το πλήθος κάθε συνόλου προσδιορίζεται από τις τιμές cluster1 και cluster2. Επιπλέον το σύστημα δέχεται σαν εισόδους το βάρος κάθε ακολουθίας που έχει υπολογιστεί σε προηγούμενο σημείο του αλγορίθμου, έναν πίνακα αντικατάστασης τον dis_consweight, ο οποίος περιέχει την συχνότητα εμφάνισης των νουκλεονικών βάσεων, μία σταθερή τιμή την Fpenalty που αντιστοιχεί στο σταθερό κόστος κενού και τέλος δύο διανύσματα τα sgap και egap τα οποία χρησιμοποιούνται για τον προσδιορισμό των κενών, που ξεκινούν από ένα σημείο σε μία ακολουθία και το των κενών που καταλήγουν σε ένα σημείο αυτής αντίστοιχα. Οι έξοδοι της συνάρτησης αυτής θα είναι ένα score το οποίο προστίθεται στο συνολικό(total score) και προσδιορίζει πόσο βέλτιστη ήταν η ευθυγράμμιση, τα ευθυγραμμισμένα τμήματα της ακολουθίας καθώς και ένα σήμα που ενεργοποιείται όταν ολοκληρωθεί η διαδικασία.

Η αρχιτεκτονική της σχεδίασης που υπολογίζει αυτό το τμήμα χωρίζεται σε 6 στάδια και φαίνεται παρακάτω(Σχήμα 4-7).



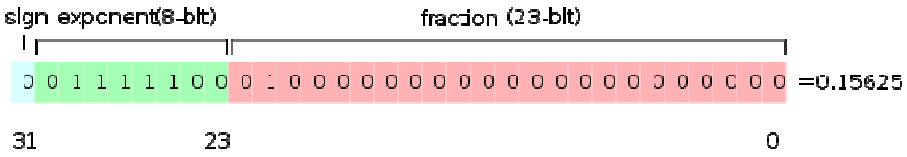
Σχήμα 4-7: Αρχιτεκτονική του τμήματος που υπολογίζει την συνάρτηση A__align

4.3.1 Βασικές δομικές μονάδες της σχεδίασης και της υλοποίησης

Αριθμοί κινητής υποδιαστολής και αριθμητικές πράξεις :

Οι αριθμητικές πράξεις που εκτελούνται σε αυτό το τμήμα της αρχιτεκτονικής είναι πράξεις κινητής υποδιαστολής απλής ακρίβειας και οι αριθμοί

αναπαριστώνται σε δυαδική μορφή με τον τρόπο αναπαράστασης του προτύπου IEEE-745 [33]. Σύμφωνα με αυτό το πρότυπο οι αριθμοί απλής ακρίβειας αναπαριστούνται σε 32-bits, όπου το πρώτο bit είναι το bit προσήμου. Τα επόμενα 8 bits αποτελούν το εκθετικό μέρος (exponent) και τα τελευταία 23 bits το κλασματικό μέρος(mantissa-fraction). Ένα παράδειγμα δυαδικής αναπαράστασης φαίνεται παρακάτω(Εικόνα 4.1):



[12]Εικόνα 4.1: Bit values for the IEEE 754 32bit float 0.15625

Για την εκτέλεση των αριθμητικών πράξεων σε αυτή την σχεδίαση χρησιμοποιήθηκαν αθροιστές, αφαιρέτες και πολλαπλασιαστές κινητής υποδιαστολής απλής ακρίβειας, οι οποίοι δημιουργήθηκαν με το εργαλείο Xilinx Core Generator 10.1.

Συγκεκριμένα χρησιμοποιούνται 6 αθροιστές με latency 12, 4 αθροιστές με latency 9, 4 αθροιστές με latency 5 και 10 αθροιστές με latency 6, όπου κάθε αθροιστής χρησιμοποιεί 2 DSP48Es. Τέλος χρησιμοποιούνται 4 αφαιρέτες με latency 6 και 8 πολλαπλασιαστές με latency 5, όπου κάθε αφαιρέτης χρησιμοποιεί 2 DSP48Es ενώ κάθε πολλαπλασιαστής χρησιμοποιεί 3 DSP48Es.

Οι επιλογές αυτές θα τεκμηριωθούν παρακάτω κατά την ανάλυση της σχεδιαστικής διαδικασίας.

[12] Πηγή: http://en.wikipedia.org/wiki/IEEE_754-1985#Structure_of_a_floating-point_number

Δεδομένα εισόδου:

Οι μνήμες GroupSeq1, GroupSeq2 και dis_consweight:

Οι δύο πρώτες είναι οι μνήμες που περιέχουν τις ομόλογες περιοχές από τα τμήματα κάθε ακολουθίας και αποτελούν τις εισόδους του συστήματος ενώ η τελευταία περιέχει τιμές ομοιότητας και χρησιμοποιείται για την δημιουργία του ομόλογου πίνακα. Οι ομαδοποιήσεις προκύπτουν από την εύρεση των ομόλογων τμημάτων κάθε ακολουθίας, με την χρήση του fast Fourier transform, ενώ παράλληλα επιλέγεται ο πίνακας ομοιότητας για τις συγκεκριμένες ομάδες νουκλεονικών οξέων. Το τμήμα αυτό δεν υλοποιείται από την συγκεκριμένη αρχιτεκτονική γι αυτό τα δεδομένα λαμβάνονται από το πρόγραμμα του αλγορίθμου στο software και αποθηκεύονται σε αρχεία txt. Όπως και στην ενότητα 4.2 έτσι και εδώ για την ανάγνωση των αρχείων αυτών δημιουργήθηκε μια εφαρμογή σε γλώσσα προγραμματισμού C κατά την οποία επιλέγονται όλες οι χρήσιμες πληροφορίες, μετατρέπονται σε δυαδική μορφή και αποθηκεύονται σε αρχεία με κατάλληλο τρόπο ώστε να έχουν την μορφή coe και να χρησιμοποιηθούν για την αρχικοποίηση των μνημών. Οι υπόλοιπες είσοδοι του τμήματος αυτού δηλώνονται ως σήματα(σταθερές) στο εσωτερικό της σχεδίασης.

Οι μνήμες GroupSeq1 και GroupSeq2 είναι τύπου dual port ROM και έχουν βάθος 3210 θέσεις όπου κάθε θέση είναι 5 bits και αντιστοιχεί σε ένα χαρακτήρα σύμφωνα με τον πίνακα 4.. Η διαμόρφωση των μνημών έγινε με τέτοιο τρόπο ώστε να διευκολύνεται η ανάγνωση όλων των τμημάτων κάθε ομάδας.

Τέλος η μνήμη dis_consweight είναι τύπου single port ROM, έχει βάθος 676 θέσεων και κάθε θέση είναι 32 bits η οποία αντιστοιχεί σε μία τιμή ομοιότητας ανάμεσα σε δύο χαρακτήρες.

Υπολογισμός κόστους κενών και συνολικά βάρη :

Για την δημιουργία του ομόλογου πίνακα(εξίσωση 6) από όπου θα προκύψει το σκορ της ευθυγράμμισης(εξίσωση 7), υπολογίζεται αρχικά το συνολικό βάρος των κενών που ξεκινούν από μία θέση(OpenigGapCount) και το συνολικό βάρος των κενών που καταλήγουν σε μία θέση(FinalGapCount) για όλα τα τμήματα κάθε ομάδας(εξισώσεις 9, 10) καθώς και ένας πίνακας που αποθηκεύει τα συνολικά βάρη κάθε χαρακτήρα που εμφανίζεται, στην θέση που εμφανίζεται (TotalWeightMtx).

Στο πρώτο λοιπόν στάδιο αυτής της σχεδίασης υλοποιούνται οι παραπάνω μονάδες οι οποίες εκτελούνται παράλληλα δύο φορές η κάθε μία, μια για την πρώτη ομάδα ακολουθιών και μία για την δεύτερη.

Η μονάδα TotalWeightMtx παίρνει μία, μία τις ακολουθίες του εκάστοτε συνόλου και για κάθε μία από αυτές διαβάζει έναν, έναν τους χαρακτήρες της. Κάθε χαρακτήρας αντιστοιχεί σε μία γραμμή ενός πίνακα και η θέση του χαρακτήρα μέσα στην ακολουθία στην αντίστοιχη στήλη του πίνακα. Στην θέση αυτή προστίθεται κάθε φορά το βάρος της τρέχουσας ακολουθίας, στο ήδη υπάρχον συνολικό βάρος το οποίο αρχικά είναι 0. Τα πιθανά σύμβολα που μπορούν να εμφανιστούν σε μία ακολουθία είναι 26, άρα ο πίνακας θα έχει διαστάσεις 26xN όπου N το μήκος της ακολουθίας. Επειδή όπως ήδη αναφέρθηκε δεν είναι δυνατόν να δημιουργηθεί μία δισδιάστατη μνήμη, τα δεδομένα αποθηκεύονται στην μνήμη Mem_cpmx, τύπου dual port RAM, η οποία έχει βάθος 8346 θέσεων και κάθε θέση της είναι 32 bits. Η μνήμη αυτή χωρίζεται σε 26 τμήματα(γραμμές) ένα για κάθε χαρακτήρα και κάθε τμήμα έχει βάθος όσο το μήκος της ακολουθίας(στήλες) όπου σε κάθε θέση αποθηκεύεται το συνολικό βάρος στο αντίστοιχο τμήμα(ανάλογα με τον χαρακτήρα που εμφανίζεται).

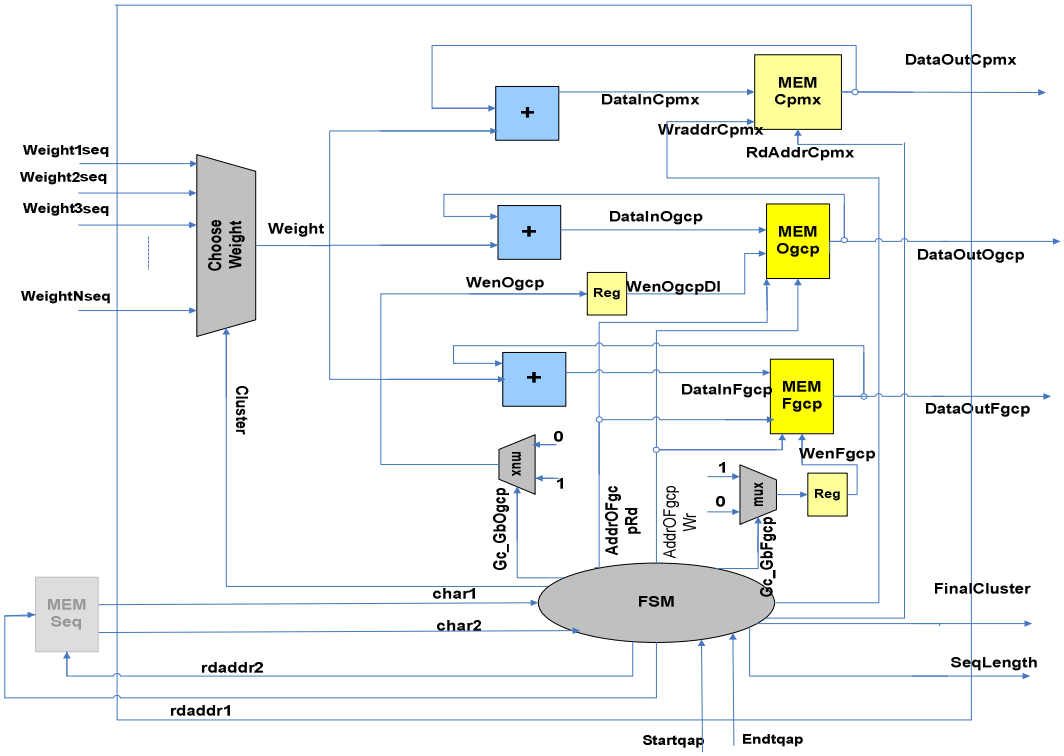
Οι μονάδες OpenigGapCount και FinalGapCount ακολουθούν την ίδια λογική διαβάζουν δηλαδή έναν, έναν τους χαρακτήρες της εκάστοτε ακολουθίας με την διαφορά ότι η πρώτη προσθέτει τα βάρη των ακολουθιών, στην θέση που εμφανίζεται ο πρώτος κενός χαρακτήρας, ενώ η δεύτερη στην θέση που

εμφανίζεται ο τελευταίος κενός χαρακτήρας. Τα δεδομένα αποθηκεύονται στις μνήμες mem_ogcp και mem_Fgcp αντίστοιχα, που είναι τύπου dual port RAM, έχουν βάθος 512 θέσεων και κάθε θέση είναι 32 bits.

Αξίζει να σημειωθεί ότι για τα αθροίσματα επιλέχθηκαν αθροιστές με μέγιστο latency 12, ώστε να διατηρηθεί η συχνότητα του ρολογιού σε ικανοποιητικά επίπεδα.

Για την σωστή λειτουργία αυτόν των μονάδων χρησιμοποιήθηκε μια μηχανή πεπερασμένων καταστάσεων(FSM), η οποία είναι αυτή που ουσιαστικά καθορίζει το βασικό υπολογιστικό κομμάτι. Πιο συγκεκριμένα, αρχικά γίνεται ανάγνωση των μνημών εισόδου και έπειτα των μνημών mem_ogcp, mem_Fgcp, και Mem_cpmx. Στην συνέχεια αποφασίζεται, ύστερα από συγκρίσεις, σε ποία θέση θα προστεθούν τα επιπλέον βάρη και τέλος γίνεται εγγραφή στις μνήμες, με καθυστέρηση 12 κύκλων για το πρώτο αποτέλεσμα, λόγω του latency των αθροιστών. Τέλος η FSM υπολογίζει το πλήθος και το μήκος των ακολουθιών κάθε ομάδας τα οποία είναι χρήσιμα για το υπόλοιπο μέρος της σχεδίασης.

Στο σχήμα 4-8 φαίνεται η αρχιτεκτονική του σταδίου αυτού για την πρώτη ομάδα (για την δεύτερη η σχεδίαση είναι η ίδια).



Σχήμα 4-8: Η σχεδίαση του τμήματος του πρώτου σταδίου που υπολογίζει τις μονάδες OpenigGapCount, FinalGapCount και TotalWeightMtx για μία ομάδα ακολουθιών

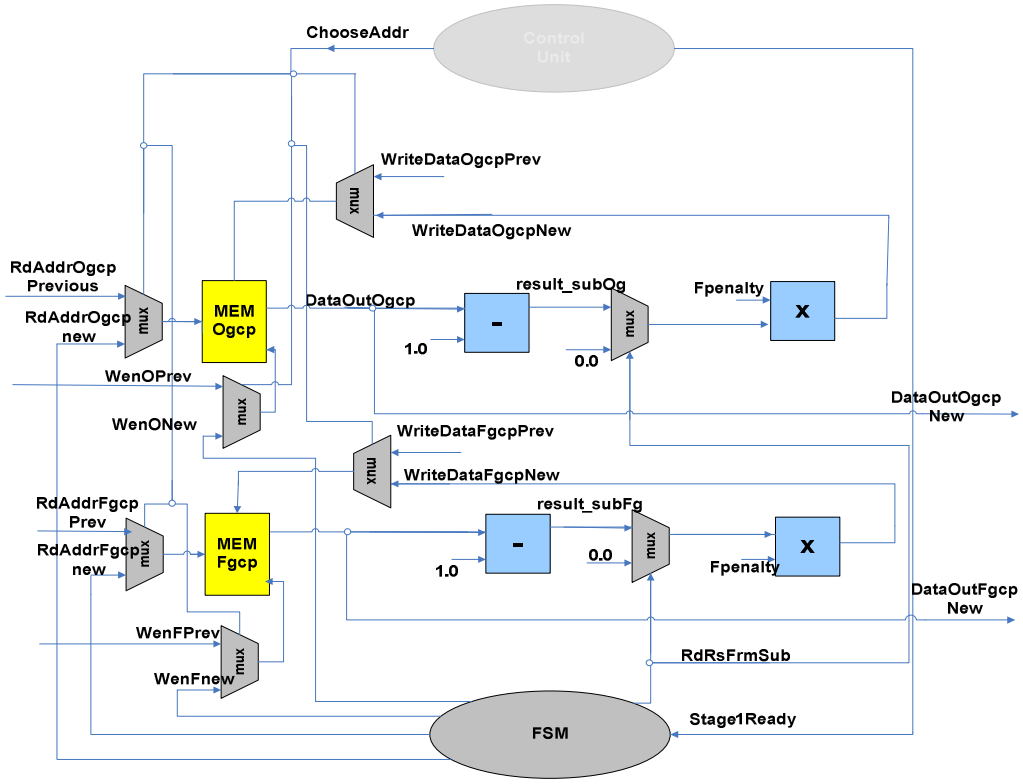
Υπολογισμός ομόλογου πίνακα και τελειοποίηση κόστους κενών:

Στο δεύτερο στάδιο ολοκληρώνεται ουσιαστικά ο υπολογισμός του κόστους κενών σύμφωνα με την εξίσωση 8 (3^{o} κεφάλαιο) και ξεκινάει η δημιουργία του ομόλογου πίνακα.

Για τον υπολογισμό του κόστους των κενών που ξεκινούν σε μία θέση της ακολουθίας και αυτών που καταλήγουν σε μία θέση αυτής, υλοποιήθηκε η μονάδα OpenFinalPenalty η οποία διαθέτει 2 αφαιρέτες με latency 6 και 2 πολλαπλασιαστές με latency 5 για κάθε μια από τις ομάδες των ακολουθιών. Η

επιλογή του latency έγινε με βάση την συχνότητα ρολογιού έτσι ώστε να έχουμε όσο το δυνατόν μικρότερη καθυστέρηση. Τα δεδομένα που διαβάζονται από τις μνήμες mem_ogcp και mem_Fgcp, με την βοήθεια μίας μονάδας ελέγχου, αφαιρούνται από την μονάδα και το αποτέλεσμα πολλαπλασιάζεται με μία σταθερά(Fpenalty). Τα νέα δεδομένα γράφονται πάλι στις ίδιες μνήμες με διεύθυνση και σήμα ενεργοποίησης που προέρχεται από την μονάδα ελέγχου.

Στο σχήμα 4-9 φαίνεται η αρχιτεκτονική του τμήματος αυτού για μία ομάδα (για την άλλη η σχεδίαση είναι η ίδια).



Σχήμα 4-9 Η σχεδίαση του τμήματος που υπολογίζει την μονάδα OpenFinalPenalty για μία ομάδα ακολουθιών(2^o στάδιο)

Παράλληλα με τον υπολογισμό του κόστους κενών ξεκινάει ο υπολογισμός του ομόλογου πίνακα.

Όπως ήδη αναφέρθηκε στο 2^{ο} κεφάλαιο, σκοπός της ευθυγράμμισης είναι η εύρεση των τμημάτων των ακολουθιών που είναι όμοια και αυτών που διαφέρουν(ομόλογος πίνακας). Για να γίνει αυτό, οι ακολουθίες τοποθετούνται η μία κάτω από την άλλη έτσι ώστε μία βάση που βρίσκεται στην θέση i σε μία ακολουθία να αντιστοιχεί σε όλες τις βάσεις των υπόλοιπων ακολουθιών που βρίσκονται σε αυτήν την θέση. Θεωρητικά αυτό συμβαίνει στην μνήμη Mem_cpmx. Αν δηλαδή μια συγκεκριμένη θέση από τα 26 τμήματα είναι μη μηδενική σημαίνει ότι σε αυτή την θέση εμφανίζεται ο χαρακτήρας του αντίστοιχου τμήματος για μία ή περισσότερες ακολουθίες.

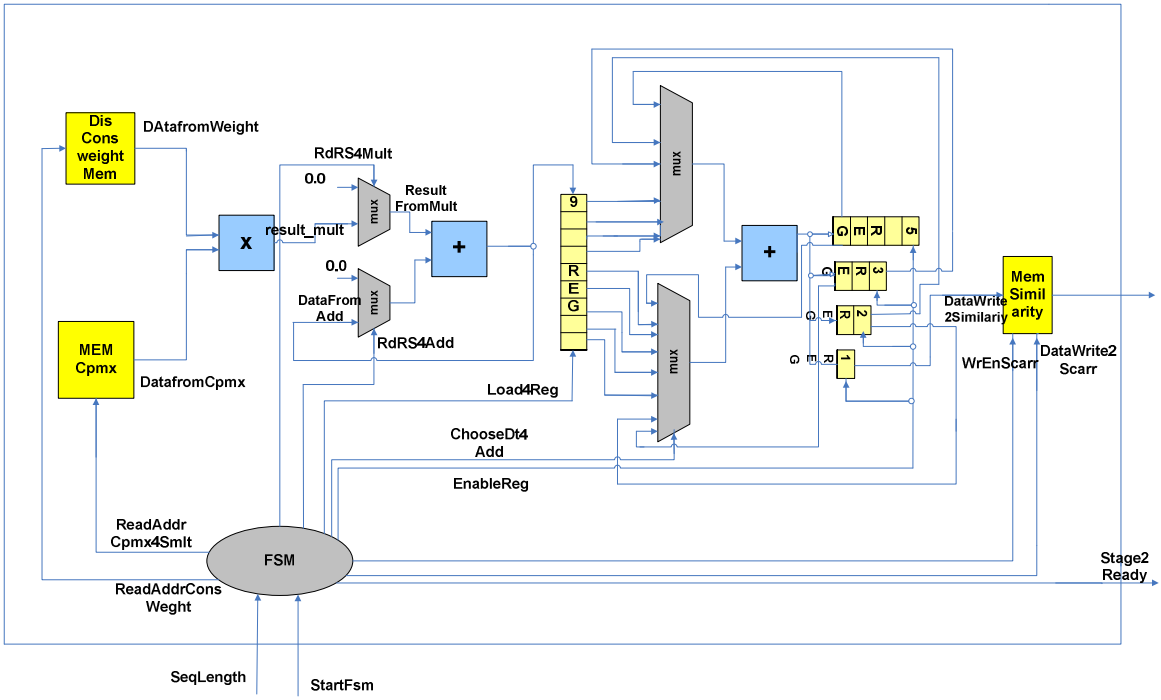
Το πρώτο μέρος για τον υπολογισμό του ομόλογου πίνακα, το οποίο υλοποιείται σε αυτό το στάδιο της σχεδίασης, είναι η δημιουργία ενός διανύσματος(μνήμης Mem_similarity) 26 θέσεων όπου κάθε θέση περιέχει το άθροισμα των γινομένων των στοιχείων της μνήμης dis_consweight με τα στοιχεία της πρώτης στήλης(αντιστοιχούν στον πρώτο χαρακτήρα) της μνήμης Mem_cpmx(πχ για την πρώτη ομάδα) για κάθε ένα από τους 26 χαρακτήρες. Το πιο απαιτητικό μέρος όσο αφορά την σχεδίαση είναι ο συγχρονισμός των δύο αυτών πράξεων. Για κάθε θέση της νέας μνήμης χρειάζονται 26 γινόμενα τα οποία αθροίζονται μεταξύ τους για να αποθηκευτούν στην συνέχεια στη θέση αυτή.

Με την πλήρη εκμετάλλευση των μονάδων για pipelining είναι δυνατόν σε κάθε κύκλο να υπολογίζεται και μία νέα τιμή. Μέχρι στιγμής αυτό είναι εφικτό αφού η οι μνήμες Mem_cpmx και dis_consweight μπορούν εφόσον δέχονται διαφορετική είσοδο σε κάθε κύκλο να δίνουν και διαφορετική έξοδο σε κάθε κύκλο. Επιπλέον όλες οι πράξεις που γίνονται στην συνέχεια, μπορούν και αυτές σε κάθε κύκλο να δίνουν διαφορετικό αποτέλεσμα. Πιο συγκεκριμένα σε κάθε κύκλο φτάνει στην μονάδα άθροισης ένα από τα 26 γινόμενα, το οποίο θα προστεθεί σε αυτά που προηγήθηκαν και στο αποτέλεσμα αυτό θα προστεθεί το γινόμενο που θα φτάσει στον επόμενο κύκλο. Με αυτόν τον τρόπο συνεχίζεται η διαδικασία για τον υπολογισμό του αθροίσματος. Όταν φθάσει και το 26^{ο} γινόμενο θα έρθει το πρώτο νέο γινόμενο που αφορά την επόμενη θέση της μνήμης και η διαδικασία συνεχίζεται με τον ίδιο τρόπο.

Μια ιδανική περίπτωση θα ήταν να χρησιμοποιήσουμε έναν αθροιστή με latency 1, όπου κάθε έξοδος του να επιστρέφει σε μια από τις εισόδους του , έτσι ώστε η τιμή του νέου γινομένου που φτάνει στην άλλη είσοδο να προστίθεται με το άθροισμα των προηγούμενων. Στην περίπτωση αυτή χρειάζεται και ένας πολυπλέκτης, ο οποίος θα επιλέγει κάθε φορά που θα έρχεται το πρώτο από τα 26 γινόμενα, τα οποία θα προστεθούν μεταξύ τους, να προστίθεται με το 0 ενώ σε όλες τις άλλες περιπτώσεις, θα επιλέγει ως είσοδο την έξοδο του αθροιστή.

Παρά το γεγονός ότι για να υλοποιηθεί η παραπάνω περίπτωση χρειάζονται ελάχιστοι πόροι, παρατηρείται ότι η συχνότητα λειτουργίας του συστήματος θα μειωθεί σημαντικά, λόγω της χρήσης ενός αθροιστή με το ελάχιστο δυνατό latency, σε συνδυασμό βέβαια με τους άλλους πόρους που χρησιμοποιούνται.

Για να λυθεί αυτό το πρόβλημα, έγιναν δοκιμές ώστε να χρησιμοποιηθεί όσο το δυνατόν μεγαλύτερο latency ενώ παράλληλα να παρέχεται η σωστή λειτουργία με την δυνατότητα pipelining. Έτσι αποφασίστηκε το latency του αθροιστή να είναι 9, με βάση το οποίο επιλέχθηκαν και τα latency των υπόλοιπων μονάδων κινητής υποδιαστολής, με απώλεια όμως ενός κύκλου για τον υπολογισμό κάθε θέσης της μνήμης Mem_similarity. Ο πολλαπλασιαστής δεν επηρεάζει την όλη διαδικασία και έτσι κατασκευάστηκε με latency 5 ώστε να μην μεταβάλλει την συχνότητα ρολογιού. Επομένως το πρώτο γινόμενο θα προκύψει μετά από 5 κύκλους ενώ κάθε επόμενο μετά από ένα κύκλο. Στο σχήμα 4-10 φαίνεται η υλοποίηση του τμήματος αυτού.



Σχήμα 4-10 : Σχεδίαση της αρχιτεκτονικής για τον υπολογισμό του 1ου μέρους του ομόλογου πίνακα(2ο στάδιο)

Λόγω του ότι το αποτέλεσμα της πρόσθεσης του πρώτου γινομένου με το 0 δεν θα δοθεί από τον αθροιστή μετά από έναν κύκλο ώστε να προστεθεί με το δεύτερο, θα πρέπει όλα τα υπόλοιπα γινόμενα μέχρι και το 9ο να προστεθούν με το 0, το οποίο επιλέγεται μέσω του 2ου πολυπλέκτη, που βρίσκεται πριν από την είσοδο του αθροιστή. Όταν δοθεί το αποτέλεσμα της πρόσθεσης του 1ου στοιχείου από τον αθροιστή, τότε αλλάζει πλέον το σήμα ελέγχου του πολυπλέκτη και επιλέγεται ως σήμα εισόδου η έξοδος του αθροιστή. Έτσι καθώς φθάνουν τα γινόμενα προστίθενται με τα μερικά αθροίσματα σε ομάδες των 9. Όμως η τελευταία ομάδα που φθάνει για να προστεθεί, περιέχει 8 γινόμενα ενώ τα μερικά αθροίσματα είναι 9. Για το λόγο αυτό η άφιξη των νέων γινομένων καθυστερεί κατά ένα κύκλο και το τελευταίο μερικό άθροισμα προστίθεται με το μηδέν, το οποίο επιλέγεται μέσω του 1ου πολυπλέκτη. Στην συνέχεια η τελευταία

ομάδα των αθροισμάτων μπαίνει σε 9 καταχωρητές, οι οποίοι ενεργοποιούνται με ένα σήμα ελέγχου που προέρχεται από την επιμέρους μονάδα ελέγχου. Οι έξοδοι των καταχωρητών θα πρέπει να προστεθούν έγκαιρα μεταξύ τους πριν αλλοιωθούν από τα επόμενα 9 μερικά αθροίσματα.

Η πρόσθεση αυτή γίνεται μέσο ενός δεύτερου αθροιστή ο οποίος κατασκευάστηκε με μικρότερο latency 5 καθώς δεν επηρεάζεται η συχνότητα ρολογιού. Ο έλεγχος για τις εισόδους του αθροιστή γίνεται με την χρήση δύο πολυπλεκτών 11 προς 1 και 8 προς 1 οι οποίοι ελέγχονται από το ίδιο σήμα. Αναλυτικότερα στην αρχή προστίθεται η πρώτη είσοδος από τον πρώτο πολυπλέκτη με την πρώτη του δεύτερου (1^o + 10^o + 19^o) + (2^o + 11^o + 20^o). Μετά από 2 κύκλους ο αθροιστής δέχεται τις δεύτερες εισόδους των δύο πολυπλεκτών (4^o + 13^o + 22^o)+ (5^o + 14^o + 23^o), μετά τις τρίτες (5^o + 14^o + 23^o)+ (6^o +15^o + 24^o), τις τέταρτες (7^o +16^o + 25^o)+(8^o + 17^o + 26^o) και τέλος, μετά από άλλους δύο κύκλους, σειρά έχουν η πέμπτη είσοδος από τον πρώτο πολυπλέκτη και η πέμπτη είσοδος από τον δεύτερο πολυπλέκτη η οποία είναι και μηδενική (9^o + 18^o + 0) + 0. Σε αυτό το σημείο αξίζει να σημειωθεί ότι επειδή ο δεύτερος αθροιστής έχει μικρότερη καθυστέρηση και το πρώτα νέο άθροισμα βγαίνει μετά από 5 κύκλους, τα νέα αθροίσματα αποθηκεύονται εκ νέου σε 5 καταχωρητές μέχρι να ελευθερωθεί η μονάδα του αθροιστή, στην συνέχεια σε τρείς μετά σε δύο και τέλος σε έναν, ο οποίος κρατάει και το τελικό αποτέλεσμα το οποίο αποθηκεύεται στην αντίστοιχη θέση της μνήμης Mem_similarity.

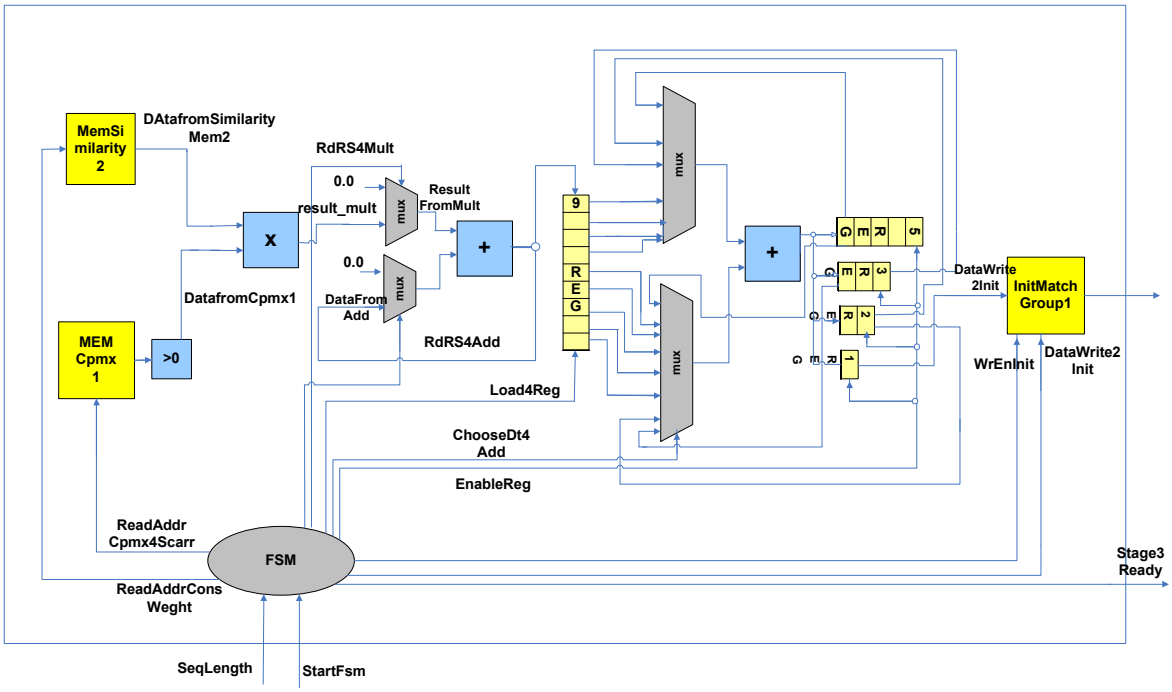
Επομένως αφού περάσουν 9 κύκλοι από την επιλογή των πρώτων εισόδων των πολυπλεκτών, ως είσοδοι του αθροιστή επιλέγονται οι $6^{ες}$ είσοδοι των πολυπλεκτών για να προστεθούν τα νέα μερικά αθροίσματα μεταξύ τους ((1^o + 10^o + 19^o) + (2^o + 11^o + 20^o)) + ((4^o + 13^o + 22^o)+ (5^o + 14^o + 23^o)) κοκ. Τέλος αξίζει να σημειωθεί ότι σε όλες τις περιπτώσεις ο συγχρονισμός των εισόδων στον αθροιστή, λόγω του ότι η μία εξέρχεται από τον αθροιστή έναν κύκλο γρηγορότερα από την άλλη και του ότι πρέπει να ελευθερωθεί ο αθροιστής για να υπολογιστούν τα αθροίσματα των νέων καταχωρητών, γίνεται μέσω των επιμέρους μονάδων ελέγχου που χρησιμοποιούνται στην μονάδα αυτή.

Η μνήμη Mem_similarity είναι τύπου dual port RAM έχει βάθος 26 θέσεων και κάθε θέση είναι 32 bits, κάθε μία από τις οποίες περιέχει μία τιμή που προσδιορίζει την ομοιότητα κάθε χαρακτήρα της μίας ομάδας.

Η διαδικασία εκτελείται παράλληλα ακόμα μία φορά και για την άλλη ομάδα των ακολουθιών.

Στο τρίτο στάδιο συνεχίζεται ο υπολογισμός του ομόλογου πίνακα. Αυτή τη φορά υπολογίζονται δύο διανύσματα(μνήμες) InitMatchGroup1 και CurrMatchGroyp2 με βάση τα δεδομένα των μνημών Mem_similarity1 και Mem_similarity2, τα οποία σε κάθε θέση περιέχουν την τιμή ταιριάσματος των χαρακτήρων της $1^{ης}$ ομάδας με τον πρώτο χαρακτήρα που εμφανίζεται σε όλες τις ακολουθίες της $2^{ης}$ ομάδας και την τιμή ταιριάσματος των χαρακτήρων της $2^{ης}$ ομάδας με αρχικά τον πρώτο χαρακτήρα που εμφανίζεται σε όλες τις ακολουθίες της $1^{ης}$ ομάδας, αντίστοιχα. Οι μνήμες InitMatchGroup1 και CurrMatchGroyp2 έχουν βάθος όσο το μήκος των ακολουθιών της πρώτης και της δεύτερης ομάδας αντίστοιχα και κάθε θέση είναι 32 bits. Κάθε θέση της μίας μνήμης προκύπτει από το άθροισμα των γινομένων των μη μηδενικών τιμών μιας στήλης(αντιστοιχεί στον χαρακτήρα που βρίσκεται στην κ-οστή θέση) τις μίας ομάδας, με την τιμή ομοιότητας του αντίστοιχου χαρακτήρα της άλλης ομάδας που βρίσκεται στην μνήμη Mem_similarity.

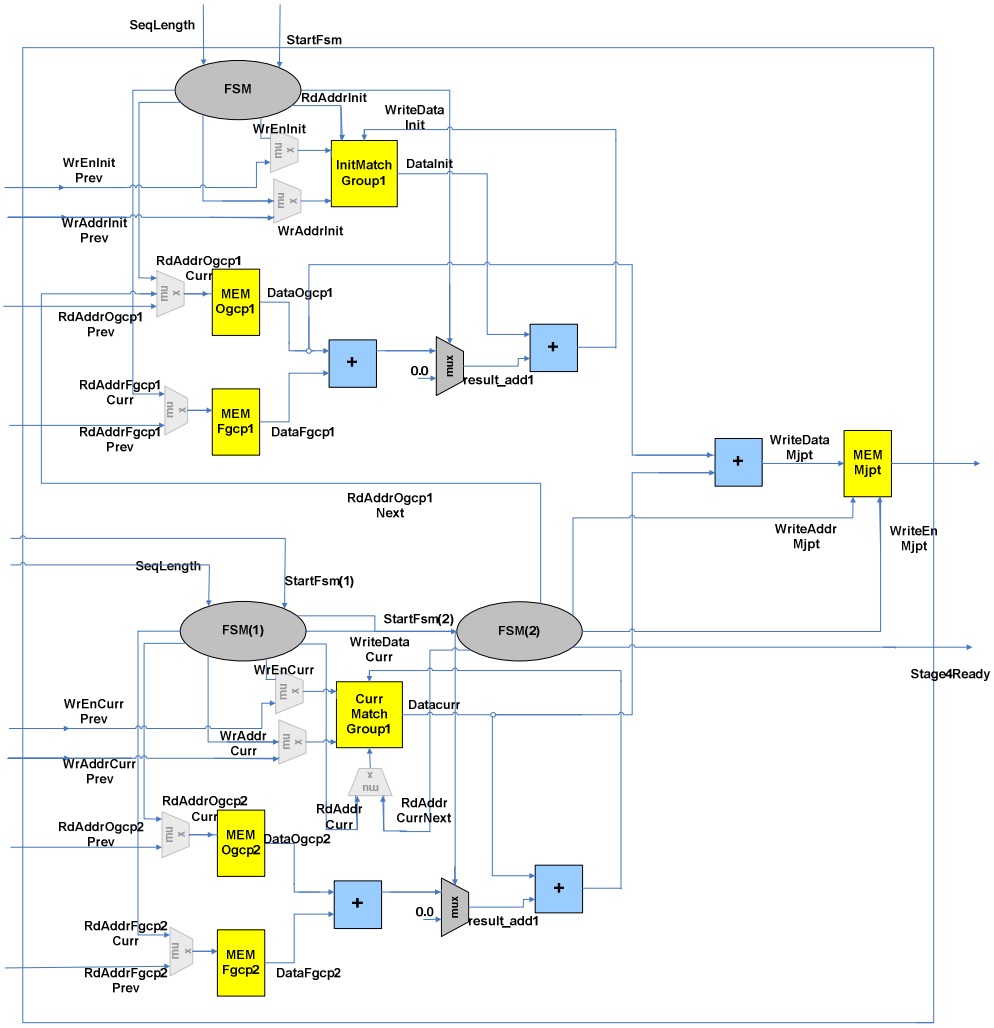
Το πιο απαιτητικό μέρος και σε αυτό το στάδιο είναι ο συγχρονισμός των δύο αυτών πράξεων. Το πρόβλημα αυτό αντιμετωπίζεται με τον ίδιο τρόπο που περιγράφηκε στο $2^{ο}$ στάδιο με την μόνη διαφορά ότι προστίθεται ένας συγκριτής με το μηδέν. Η σχεδίαση της αρχιτεκτονικής για την δημιουργία της πρώτης μνήμης φαίνεται στο σχήμα 4-11. Με τον ίδιο τρόπο υπολογίζεται και η δεύτερη.



Σχήμα 4-11: Σχεδίαση της αρχιτεκτονικής για τον υπολογισμό του ταιριάσματος(match) των ακολουθιών (3^{ο} στάδιο)

Στην συνέχεια, στο τέταρτο στάδιο προστίθεται στα δεδομένα των μνημών InitMatchGroup1 και CurrMatchGroyp2 τα κόστη των αντίστοιχων κενών. Πιο συγκεκριμένα στα περιεχόμενα κάθε θέσης της πρώτης μνήμης προστίθεται το άθροισμα του κόστους OpenigGap του πρώτου χαρακτήρα της πρώτης ομάδας, που βρίσκεται στην μνήμη mem_ogcp, με το κόστος FinalGap του τρέχων χαρακτήρα της πρώτης ομάδας ακολουθιών, που βρίσκονται στην μνήμη mem_Fgcp. Το ίδιο συμβαίνει και για την δεύτερη μνήμη, με την διαφορά ότι τα κόστη αφορούν την δεύτερη ομάδα ακολουθιών. Τα νέα δεδομένα αποθηκεύονται με σήμα ενεργοποίησης και διεύθυνση εγγραφής που προέρχεται από μία επιμέρους μονάδα ελέγχου στις ίδιες μνήμες.

Αφού ολοκληρωθεί ο νέος υπολογισμός όλων των θέσεων της μνήμης CurrMatchGroyp2 στα δεδομένα κάθε θέσης της μνήμης αυτής, προστίθεται το κόστος OpenigGap του δεύτερου χαρακτήρα της πρώτης ομάδας ακολουθιών. Τα νέα αθροίσματα αποθηκεύονται στην μνήμη memMjpt που είναι τύπου dual port RAM έχει βάθος 512 θέσεων και κάθε θέση είναι 32 bits. Η μνήμη αυτή είναι μια ενδιάμεση μνήμη που βοηθάει στην ολοκλήρωση του υπολογισμού του ομόλογου πίνακα. Η σχεδίαση της αρχιτεκτονικής του τέταρτου σταδίου φαίνεται στο σχήμα 4-12.



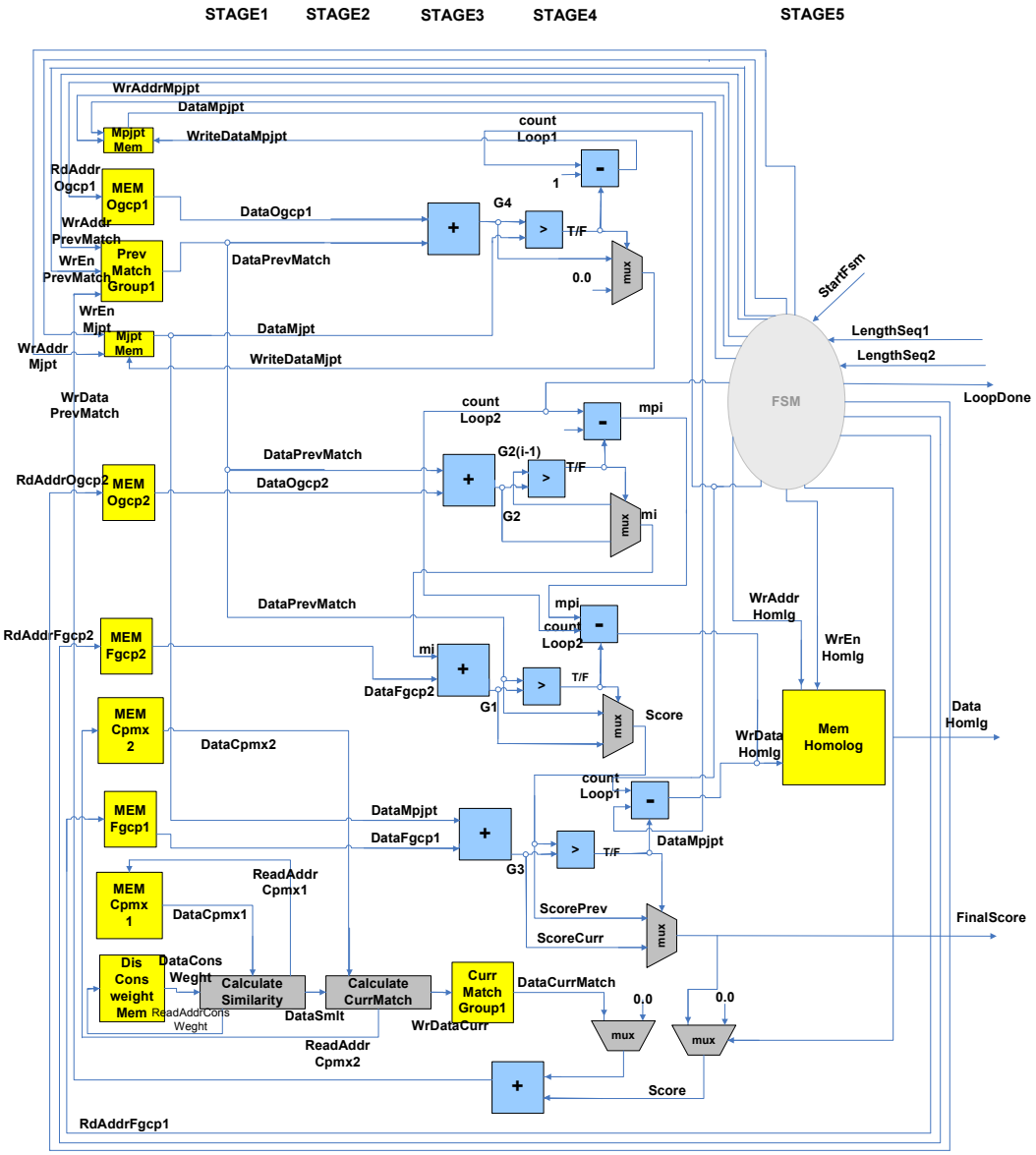
Σχήμα 4-12: Σχεδίαση αρχιτεκτονικής 4^{ου} σταδίου.

Οι πολυπλέκτες που βρίσκονται πριν από τις μνήμες, αποφασίζουν ποια θα είναι η διεύθυνση εγγραφής, η τιμή του σήματος ενεργοποίησης εγγραφής καθώς και η διεύθυνση ανάγνωσης, ανάλογα με το στάδιο στο οποίο βρισκόμαστε. Το σήμα ελέγχου των πολυπλεκτών προέρχεται από την κεντρική μονάδα ελέγχου της ολοκληρωμένης σχεδίασης.

Ολοκλήρωση ομόλογου πίνακα και υπολογισμός του σκορ:

Το πέμπτο στάδιο το οποίο καταναλώνει συνολικά τους περισσότερους κύκλους της αρχιτεκτονικής εκτελεί δύο επαναλήψεις και συγκεκριμένα επαναλαμβάνεται KxN φορές οπού K το μήκος των ακολουθιών της πρώτης ομάδας και N το μήκος ακολουθιών της δεύτερης.

Η υλοποίηση αυτού του σταδίου χωρίζεται σε 5 επιμέρους στάδια και φαίνεται στο σχήμα 4-13.



Σχήμα 4-13: Σχεδίαση αρχιτεκτονικής 4^{ου} σταδίου για μία επανάληψη

Στο πρώτο και στο δεύτερο στάδιο όπου ξεκινάει και η πρώτη επανάληψη (εκτελείται K φορές) επαναλαμβάνεται η διαδικασία του 2^{ου} και 3^{ου} σταδίου της συνολικής αρχιτεκτονικής, που περιγράφηκε παραπάνω και υπολογίζεται ένα νέο διάνυσμα CurrMatch που περιέχει τώρα τα ταιριάσματα των χαρακτήρων της

δεύτερης ομάδας με τον κ-οστό χαρακτήρα της πρώτης ομάδας. Το διάνυσμα αυτό αποθηκεύεται σε μία νέα μνήμη CurrMatch καθώς τα δεδομένα της προηγούμενης μνήμης, η οποία ονομάζεται πλέον PrevMatch, χρησιμεύουν για τον υπολογισμό του ομόλογου πίνακα στην δεύτερη εμφωλευμένη επανάληψη. Την επόμενη φορά που θα ξεκινήσει νέα επανάληψη τα δεδομένα της μνήμης PrevMatch θα περιέχουν τα προηγούμενα δεδομένα της μνήμης CurrMatch συναθροισμένα με το σκορ όπως φαίνεται και στο σχήμα 4-12.

Στο τρίτο στάδιο ξεκινάει η δεύτερη επανάληψη όπου αθροίζονται τα δεδομένα των μνημών όπως φαίνεται στο σχήμα 4-13. Οι αθροιστές κατασκευάστηκαν με latency 6 ώστε να διατηρείται η συχνότητα του συστήματος. Το πρόβλημα που προκύπτει σε αυτό το στάδιο είναι ότι τα αθροίσματα δεν μπορούν να εκτελεστούν ανεξάρτητα αλλά εξαρτώνται από τις συγκρίσεις που ακολουθούν και τις τιμές που υπολογίζονται, με βάση τις συγκρίσεις και το αντίστροφο. Μια ιδανική λύση ώστε να υπάρχει πλήρη εκμετάλλευση της δυνατότητας pipelining είναι να μειωθεί το latency των αθροιστών στο 1 ώστε σε κάθε κύκλο να βγαίνει ένα καινούριο άθροισμα και να γίνονται οι συγκρίσεις, χωρίς να χάνεται κανένας κύκλος. Κάτι τέτοιο όμως θα μείωνε σημαντικά την συχνότητα λειτουργίας του συστήματος. Γι αυτό το λόγο υπάρχει καθυστέρηση 5 κύκλων μέχρι να βγει το κάθε άθροισμα σε κάθε εμφωλευμένη επανάληψη.

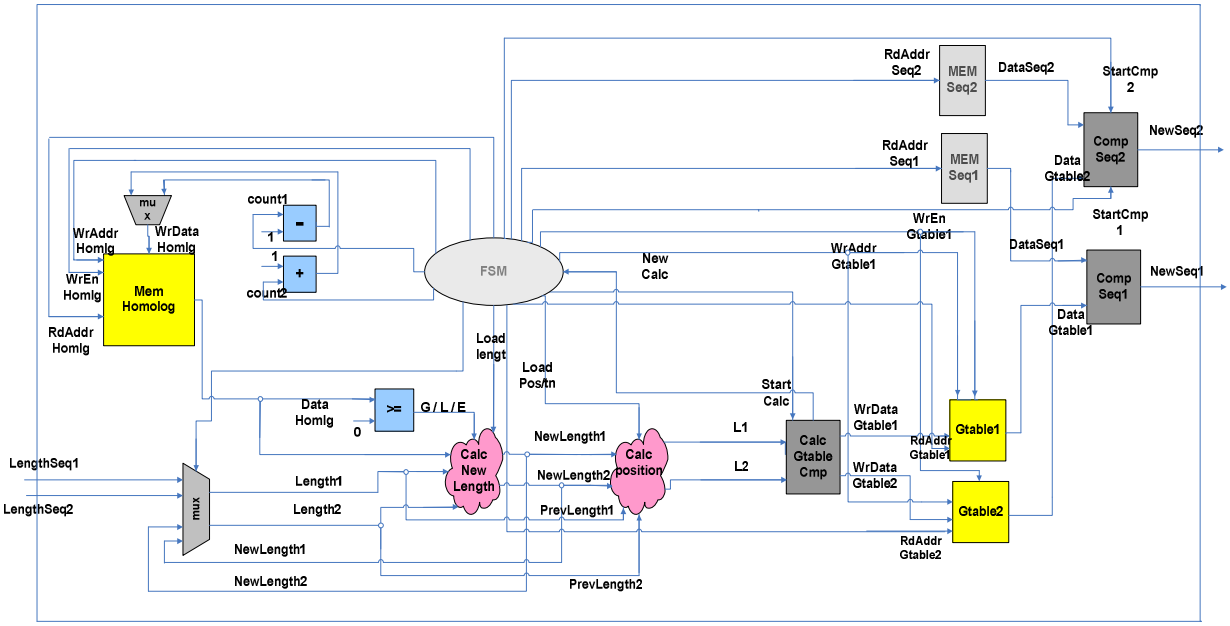
Στο τέταρτο στάδιο γίνονται οι συγκρίσεις των αθροισμάτων είτε με βοηθητικές τιμές που υπολογίζονται σε προηγούμενες επαναλήψεις είτε με τιμές που προέρχονται από την μνήμη. Οι συγκρίσεις G_1, G_2 και G_4 εκτελούνται στον ίδιο κύκλο, ενώ η G_3 μετά από ένα κύκλο, καθώς η τιμή σύγκρισης της συνθήκης (σκορ) εξαρτάται από την G_1. Όταν ισχύουν οι συνθήκες G_1 ή G_3 υπολογίζεται μία τιμή του ομόλογου πίνακα και το σκορ παίρνει την τιμή της συνθήκης που ισχύει σε κάθε κύκλο, ενώ όταν ισχύουν οι συνθήκες G_2 και G_4 υπολογίζονται κάποιες βοηθητικές τιμές, όπως φαίνεται στο σχήμα 4-13. Στην συνέχεια γίνεται μετάβαση στο τρίτο στάδιο όπου ξεκινάει ο υπολογισμός των επόμενων αθροισμάτων και επιπλέον του αθροίσματος της τρέχουσας τιμής της μνήμης CurrMatch με το νέο σκορ, το οποίο θα αποθηκευτεί στην προηγούμενη θέση της μνήμης PrevMatch. Το τρίτο και το τέταρτο στάδιο επαναλαμβάνονται N φορές,

όπου και ολοκληρώνεται ο υπολογισμός μίας σειράς του ομόλογου πίνακα. Στην συνέχεια γίνεται επιστροφή στο $2^{ο}$ στάδιο και η διαδικασία ξεκινάει από την αρχή.

Τέλος αφού ολοκληρωθούν όλες οι επαναλήψεις γίνεται μετάβαση στο πέμπτο στάδιο το οποίο είναι και το τελικό. Πλέον έχουν αποθηκευτεί όλες οι τιμές του ομόλογου πίνακα στην μνήμη Mem_Homolog και έχει υπολογιστεί το τελικό σκορ της ευθυγράμμισης. Οι διευθύνσεις ανάγνωσης, εγγραφής καθώς και τα σήμα ενεργοποίησης εγγραφής στις μνήμες προκύπτουν από μία επιμέρους μονάδα ελέγχου. Η μνήμη Mem_Homolog είναι τύπου dual port RAM έχει βάθος 108900 θέσεων ενώ κάθε θέση περιέχει 10 bits.

Τροποποίηση των ευθυγραμμισμένων ακολουθιών:

Στο τελευταίο στάδιο της αρχιτεκτονικής, το οποίο είναι το έκτο στάδιο, γίνεται εισαγωγή κενών «-» στις ακολουθίες με βάση τις τιμές του ομόλογου πίνακα που υπολογίστηκε στο προηγούμενο στάδιο ώστε να έχουν όλες το ίδιο μήκος. Η σχεδίαση της αρχιτεκτονικής αυτού του σταδίου που υλοποιείται από την μονάδα Atracking φαίνεται στο σχήμα 4-14.



Σχήμα 4-14: Σχεδίαση της αρχιτεκτονικής της μονάδας Atracking

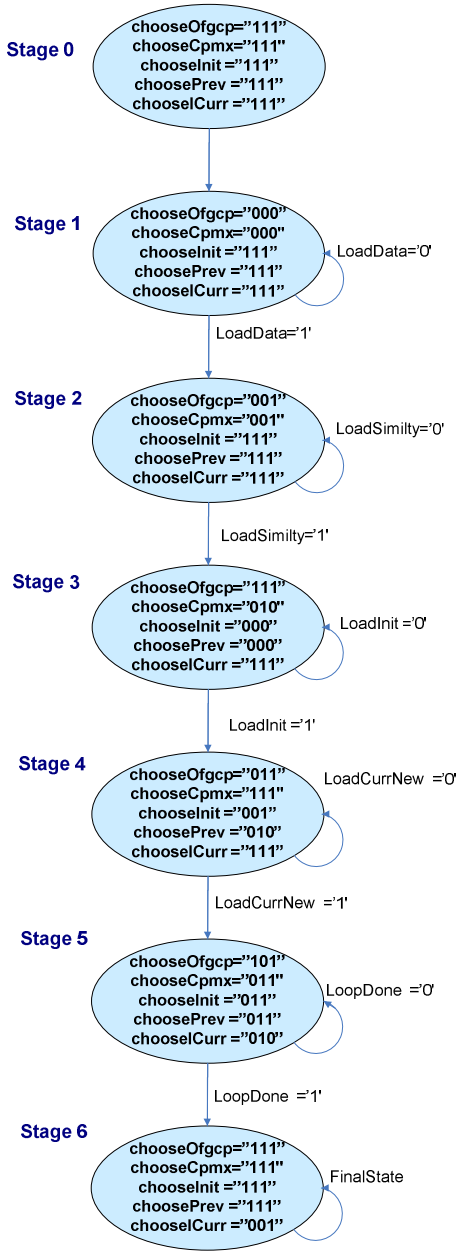
Αρχικά σε αυτή τη μονάδα υπολογίζονται στην πρώτη γραμμή και στην πρώτη στήλη του ομόλογου πίνακα, κάποιες ακέραιες τιμές που αντιπροσωπεύουν την θέση των χαρακτήρων της δεύτερης και της πρώτης ομάδας αντίστοιχα. Στην συνέχεια με βάση τα μήκη των ακολουθιών κάθε ομάδας και τα περιεχόμενα του ομόλογου πίνακα δημιουργούνται δύο βοηθητικοί πίνακες gtable1 και gtable2 που περιέχουν τους χαρακτήρες «ο», «-». Συγκρίνοντας τα περιεχόμενα αυτών των πινάκων με τους χαρακτήρες των ακολουθιών εισόδου, προκύπτουν οι νέες ακολουθίες. Κατά την διαδικασία αυτή υπολογίζονται και κάποιες ενδιάμεσες τιμές οι οποίες βοηθούν στην επιλογή των χαρακτήρων που θα γραφτούν, «ο» ή «-» και σε ποία θέση θα γραφτούν στις μνήμες gtable1 και gtable2.

Οι διευθύνσεις ανάγνωσης, εγγραφής καθώς και τα σήμα ενεργοποίησης εγγραφής στις μνήμες GroupSeq1, GroupSeq2 και Mem_Homolog προκύπτουν από μία επιμέρους μονάδα ελέγχου.

Μονάδα ελέγχου:

Για να λειτουργήσουν σωστά οι μονάδες που απαρτίζουν την αρχιτεκτονική του τμήματος αυτού, ρυθμίζονται από την μονάδα ελέγχου. Η μονάδα αυτή υλοποιείται ως μία μηχανή πεπερασμένων καταστάσεων με την οποία καθορίζονται τα σωστά σήματα για την λειτουργία των μνημών, των μονάδων εκτέλεσης αριθμητικών πράξεων, των πολυπλεκτών, των επιμέρους μονάδων ελέγχου και γενικά όλων των βασικών υπολογιστικών μονάδων.

Πιο συγκεκριμένα η κεντρική μονάδα ελέγχου είναι αυτή που ενεργοποιεί τις επιμέρους μονάδες ελέγχου σε κάθε στάδιο λειτουργίας και δίνει τα σήματα στους πολυπλέκτες που βρίσκονται πριν από τις μνήμες ώστε να παίρνουν τις κατάλληλες διευθύνσεις ανάγνωσης και εγγραφής των μνημών και τα σωστά σήματα ενεργοποίησης εγγραφής στις μνήμες σε κάθε στάδιο. Στο σχήμα 4-15 παρουσιάζονται αναλυτικά όλες οι μεταβάσεις από το ένα στάδιο στο άλλο.



Σχήμα 4-15 : Η κεντρική μονάδα ελέγχου

Το μηδενικό στάδιο είναι ουσιαστικά η κατάσταση Reset όπου τα σήματα παίρνουν τις αρχικές τους τιμές. Τέλος η μετάβαση από το ένα στάδιο στο άλλο γίνεται σύμφωνα με τα σήματα που παίρνει σαν εισόδους από τις επιμέρους μονάδες ελέγχου, τα οποία τίθενται στην τιμή ένα όταν έχει ολοκληρωθεί το εκάστοτε στάδιο.

5. Αποτελέσματα-Μετρήσεις

Στο κεφάλαιο αυτό περιγράφεται η διαδικασία ταυτοποίησης των αποτελεσμάτων της εξόδου του δεύτερου τμήματος της αρχιτεκτονικής που υλοποιήθηκε, το οποίο είναι το βαρύτερο υπολογιστικά κομμάτι του αλγορίθμου και παρουσιάζονται τα αποτελέσματα της απόδοσης σε σύγκριση με το προσωπικό υπολογιστή PC. Η εκτέλεση του πρώτου τμήματος της αρχιτεκτονικής που υλοποιήθηκε στην παρούσα εργασία, καταναλώνει ένα πολύ μικρό ποσοστό του συστήματος στο PC και δεν έχει νόημα να γίνει αντικείμενο σύγκρισης.

5.1 Ταυτοποίηση λειτουργίας

Η λειτουργία της αρχιτεκτονικής προσομοιώθηκε με το πρόγραμμα ISE Simulator 10.1.0.3. Κατά την διαδικασία της υλοποίησης της αρχιτεκτονικής που σχεδιάστηκε, προσομοιώθηκε η λειτουργία των επιμέρους τμημάτων ξεχωριστά και διαπιστώθηκε η ορθή λειτουργία τους. Στην συνέχεια αφού ενώθηκαν όλα τα τμήματα προσομοιώθηκε το συνολικό σύστημα με στόχο την διαπίστωση της σωστής λειτουργίας του.

Η σύγκριση των αποτελεσμάτων της προσομοίωσης έγινε με τα αποτελέσματα του προγράμματος MAFFT (έκδοση 6.712) κατά την μελέτη του software που ήταν διαθέσιμο στο διαδίκτυο[26] όπου και υπήρχε πλήρης συμφωνία.

Λόγω του μεγάλου όγκου δεδομένων και κατά συνέπεια του μεγάλου χρόνου που χρειάζεται η διαδικασία προσομοίωσης της λειτουργίας της αρχιτεκτονικής που υλοποιήθηκε, ενδεικτικά χρησιμοποιήθηκαν είσοδοι για την περίπτωση πλήθους 5 μέχρι 10 ακολουθιών για κάθε ομάδα και μήκους 366 μέχρι 766 στοιχείων για κάθε ακολουθία.

5.2 Σύνολο δεδομένων και Απόδοση συστήματος

5.2.1 Σύνολο Δεδομένων

Σύμφωνα με την εργασία που τον περιγράφει, ο αλγόριθμος MAFFT[14], ο οποίος αναπτύχθηκε για τον υπολογισμό της πολλαπλής ευθυγράμμισης ακολουθιών, είναι η πιο γρήγορος μέθοδος για πολλαπλή ευθυγράμμιση συγκριτικά με τις ήδη υπάρχουσες και σχεδιάστηκε για να χειρίζεται τεράστιο όγκο δεδομένων. Συγκεκριμένα, η progressive μέθοδος πάνω στην οποία στηρίζεται η παρούσα υλοποίηση υποστηρίζει από 10000 ακολουθίες με μικρό μήκος(λίγες εκατοντάδες στοιχεία) έως λίγες ακολουθίες(μερικές δεκάδες) με μέγιστο μήκος κάθε ακολουθίας 5000 στοιχεία.

Με σκοπό την εκτίμηση της απόδοσης του χρόνου εκτέλεσης σε ένα συμβατικό υπολογιστή[14] χρησιμοποιούνται δύο τύποι από σύνολα δεδομένων. Ο ένας

αφορά την σύγκριση ακολουθιών μεγάλου μήκους, της τάξεως των 100 στοιχείων και πάνω(μέχρι 5000) και μικρού σχετικά πλήθους ακολουθιών περίπου 40. Ο άλλος τύπος χρησιμοποιεί σχετικά μικρού μήκους ακολουθίες, περίπου 300 στοιχεία και πλήθος ακολουθιών[14] που κυμαίνεται από 10 μέχρι 10000.

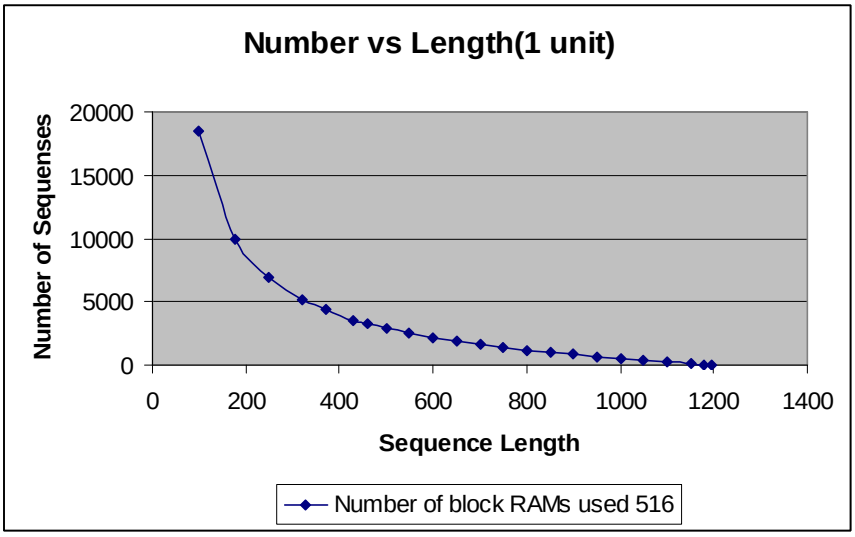
5.2.2 Απόδοση Συστήματος

Η υλοποίηση της αρχιτεκτονικής, η οποία περιγράφηκε αναλυτικά στο προηγούμενο κεφάλαιο, είναι ιδιαίτερα απαιτητική όσο αφορά τον αριθμό των block RAMs. Παρ' όλα αυτά καλύπτει ένα πολύ σημαντικό ποσοστό του όγκου δεδομένων που χρησιμοποιούνται ως είσοδοι στον MAFFT.

Συγκεκριμένα, λόγω της μεγάλης απαίτησης σε μνήμη αλλά και του μεγάλου αριθμού σε DSP4Es slices, τα οποία εκτελούν πράξεις κινητής υποδιαστολής, επιλέχθηκε η FPGA XC5VSX240T της οικογένειας Virtex 5, η οποία έχει τον μεγαλύτερο αριθμό block RAMs(516 blocks των 36 Kb) από όλες τις FPGA's που υπάρχουν στην αγορά σήμερα.

Η σύνθεση πραγματοποιήθηκε με την χρήση του εργαλείου Xilinx 10.1 όπου έγινε η διαδικασία Synthesis και Place & Route με την οποία προέκυψαν η συχνότητα ρολογιού και οι πόροι που καταναλώνει η αρχιτεκτονική που υλοποιήθηκε.

Λαμβάνοντας λοιπόν ως δεδομένο τη χρήση του μέγιστου αριθμού block RAMs που είναι διαθέσιμα και τα οποία περιορίζουν την συγκεκριμένη αρχιτεκτονική, μετρήθηκε ο μέγιστος όγκος των δεδομένων που μπορούν να ληφθούν ως είσοδοι στο σύστημα που υλοποιήθηκε στην παρούσα εργασία, συνδυάζοντας το πλήθος ακολουθιών με το μήκος της κάθε ακολουθίας. Τα στοιχεία αυτά παρουσιάζονται στο παρακάτω διάγραμμα.



Εικόνα 5.1: Διάγραμμα του όγκου δεδομένων εισόδου με χρήση μίας μονάδας του συστήματος

Παρατηρείται ότι καθώς αυξάνεται το πλήθος των ακολουθιών εισόδου, μειώνεται το πλήθος των στοιχείων κάθε ακολουθίας και το αντίστροφο.

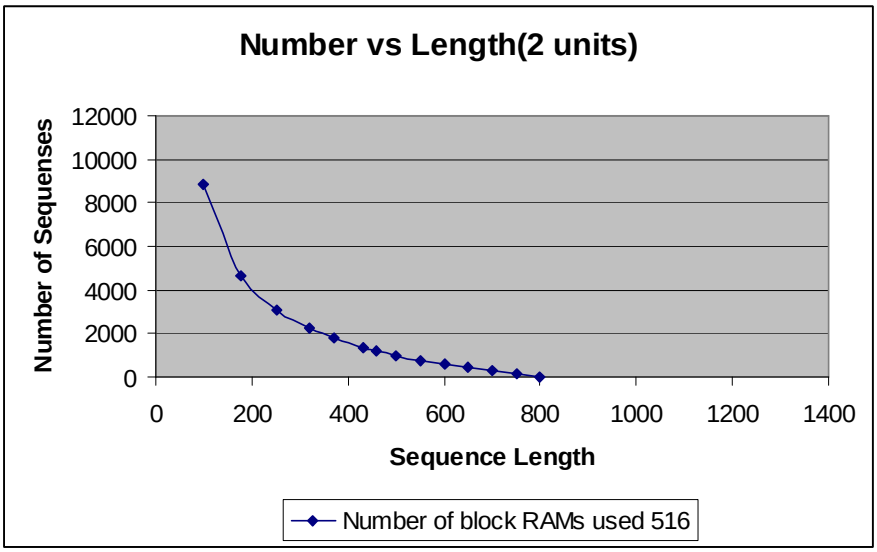
5.2.3 Αξιοποίηση των αχρησιμοποίητων πόρων της FPGA με στόχο την βελτίωση της απόδοσης.

Το τμήμα της αρχιτεκτονικής που υλοποιήθηκε, εκτελεί την σύγκριση ανάμεσα σε δύο ομάδες ακολουθιών. Δεδομένου όμως, ότι για να ολοκληρωθεί η ευθυγράμμιση χρειάζεται η σύγκριση όλων των πιθανών ομάδων ακολουθιών, είναι δυνατή η χρησιμοποίηση περισσότερων από μία μονάδες του συστήματος που υλοποιήθηκε, με στόχο τον ταχύτερο υπολογισμό της ευθυγράμμισης και κατά συνέπεια την βελτίωσης της απόδοσης του συστήματος.

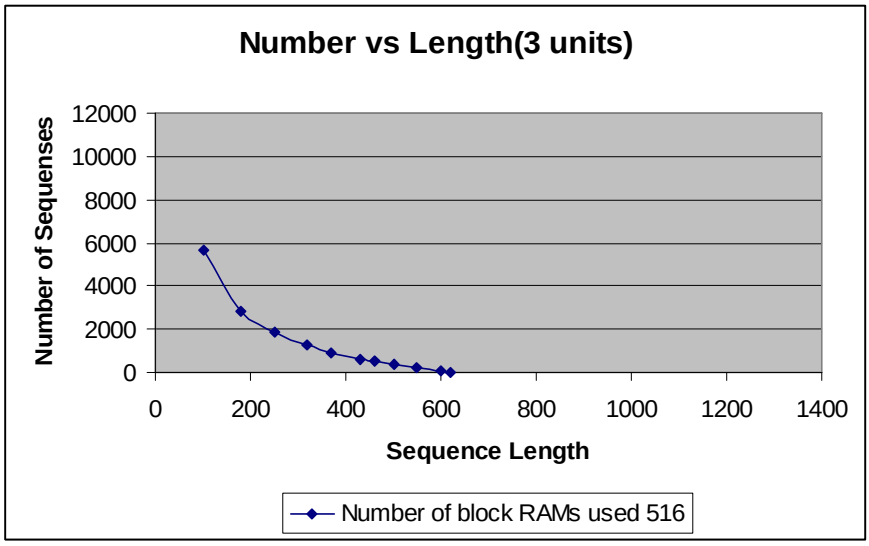
Έτσι με βάση την FPGA που επιλέχθηκε υπάρχει η δυνατότητα να χρησιμοποιηθούν παράλληλα μέχρι και εφτά μονάδες του συστήματος που

υλοποιήθηκε. Καθώς αυξάνεται ο παραλληλισμός όμως, μειώνεται ο όγκος δεδομένων των εισόδων του συστήματος, αφού αυξάνεται η απαίτηση σε μνήμη.

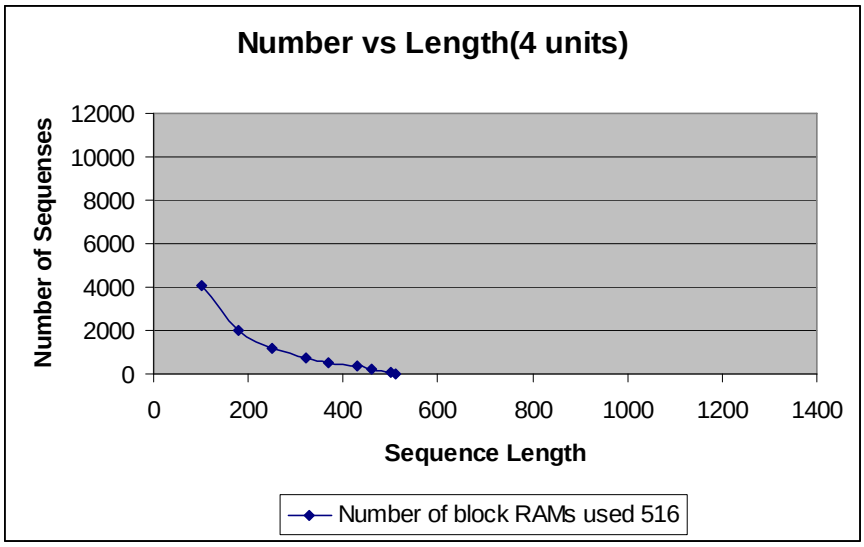
Παρακάτω παρουσιάζονται τα διαγράμματα τα οποία απεικονίζουν το πλήθος των ακολουθιών σε συνδυασμό με το με το μήκος κάθε ακολουθίας που μπορεί να δεχτεί ως εισόδους το σύστημα για όλες τις περιπτώσεις παραλληλίας (από δύο μέχρι 7 μονάδων) και με δεδομένο ότι χρησιμοποιείται ο μέγιστος αριθμός block RAMs.



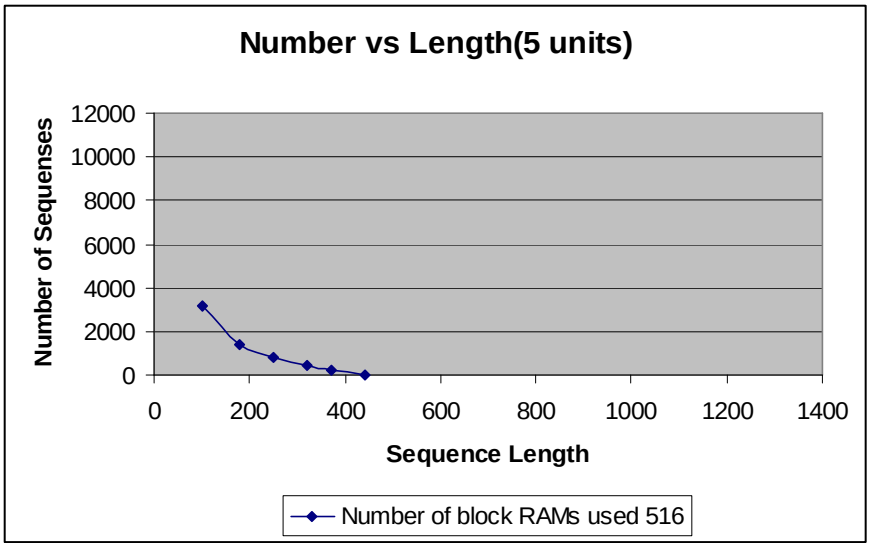
Εικόνα 5.2 Διάγραμμα του όγκου δεδομένων εισόδου με χρήση δύο μονάδων του συστήματος παράλληλα



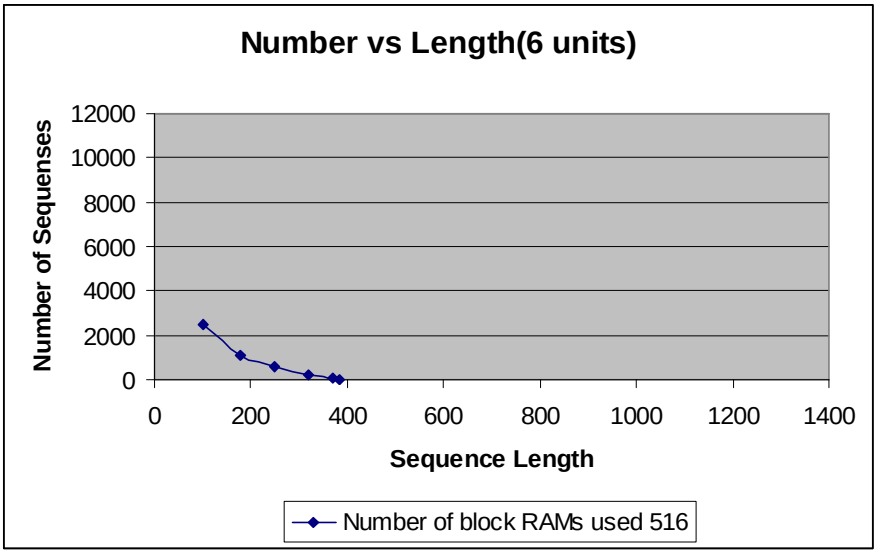
Εικόνα 5.3 Διάγραμμα του όγκου δεδομένων εισόδου με χρήση δύο μονάδων του συστήματος παράλληλα



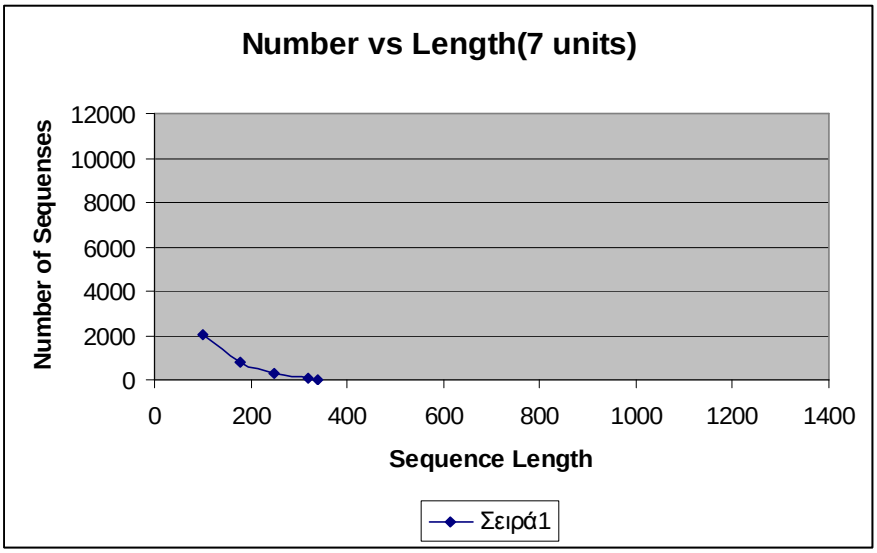
Εικόνα 5.4 Διάγραμμα του όγκου δεδομένων εισόδου με χρήση δύο μονάδων του συστήματος παράλληλα



Εικόνα 5.5 Διάγραμμα του όγκου δεδομένων εισόδου με χρήση δύο μονάδων του συστήματος παράλληλα



Εικόνα 5.6 Διάγραμμα του όγκου δεδομένων εισόδου με χρήση δύο μονάδων του συστήματος παράλληλα



Εικόνα 5.7 Διάγραμμα του όγκου δεδομένων εισόδου με χρήση δύο μονάδων του συστήματος παράλληλα

Από τα παραπάνω διαγράμματα παρατηρείται ότι καθώς αυξάνεται η παραλληλία, μειώνεται το πλήθος των δεδομένων που μπορούν να χρησιμοποιηθούν ως είσοδοι στο σύστημα αλλά ταυτόχρονα μειώνεται ο χρόνος υπολογισμού του συνόλου των συγκρίσεων που απαιτούνται για μία ολοκληρωμένη ευθυγράμμιση.

Όσο αφορά τις εισόδους, το πρόγραμμα MAFFT στο software(version 6.712) όταν φτάνει στο τελευταίο στάδιο της υλοποίησης, το οποίο αντιστοιχεί στην αρχιτεκτονική που υλοποιείται στην παρούσα εργασία,τις ομαδοποιεί παίρνοντας κάθε φορά δύο ομάδες από τμήματα των ακολουθιών με διαφορετικά μήκη ώστε να τα συγκρίνει. Ουσιαστικά λοιπόν, οι είσοδοι της υλοποίησης σε FPGA δεν είναι ολόκληρη η είσοδος αλλά αυτές οι ομάδες ακολουθιών όπου το μέγιστο μήκος τους μπορεί να είναι μέχρι και δύο φορές μεγαλύτερο ή μικρότερο σε σχέση με την αρχική είσοδο, ενώ το μέγιστο πλήθος των ακολουθιών παραμένει σταθερό. Αυτό εξαρτάται από τον βαθμό ταύτισης-αστοχίας μεταξύ των ακολουθιών εισόδου.

Στον πίνακα 5-1 και 5-2 παρουσιάζεται το μέγιστο σύνολο δεδομένων που μπορεί να υποστηρίξει η παρούσα αρχιτεκτονική για όμοιες και μακρινά

συσχετισμένες μεταξύ τους ακολουθίες αντίστοιχα, για όλες τις περιπτώσεις παραλληλίας και η αντιστοιχία κατά προσέγγιση με τις πραγματικές εισόδους του προγράμματος στο software.

Παράλ/λες Μονάδες	Πλήθος ακολουθιών	Μέγιστο μήκος ακολουθίας	Μήκος εισόδου στον MAFFT	Μέγιστο πλήθος ακολουθιών	Μήκος ακολουθίας	Μήκος εισόδου στον MAFFT
1	10	1197	2000	18550	100	168
2	10	800	1340	8820	100	168
3	10	620	1033	5650	100	168
4	10	510	850	4080	100	168
5	10	442	737	3130	100	168
6	10	385	642	2500	100	168
7	10	340	567	2028	100	168

Πίνακας 5-1: Αντιστοιχία μήκους ακολουθιών με το μήκος των εισόδων του MAFFT για όμοιες ακολουθίες

Παράλ/λες Μονάδες	Πλήθος ακολουθιών	Μέγιστο μήκος ακολουθίας	Μήκος εισόδου στον MAFFT	Μέγιστο πλήθος ακολουθιών	Μήκος ακολουθίας	Μήκος εισόδου στον MAFFT
1	10	1197	798	18550	100	168
2	10	800	530	8820	100	168
3	10	620	413	5650	100	168
4	10	510	340	4080	100	168
5	10	442	295	3130	100	168
6	10	385	257	2500	100	168
7	10	340	227	2028	100	168

Πίνακας 5-2: Αντιστοιχία μήκους ακολουθιών με το μήκος των εισόδων του MAFFT για μακρινά συσχετισμένες ακολουθίες

Όπως φαίνεται και από την αντιστοιχία του πίνακα 5-1 και 5-2 η αρχιτεκτονική που υλοποιήθηκε καλύπτει όλες και πολλές παραπάνω από τις περιπτώσεις όσο αφορά τις συγκρίσεις με βάση το πλήθος των ακολουθιών αλλά και αρκετές που αφορούν τις συγκρίσεις με βάση το μήκος των ακολουθιών(ενότητα 5.2.1).

5.3 Αποτίμηση απόδοσης

Το πρόγραμμα που είναι διαθέσιμο στο software(έκδοση 6.712) για τον αλγόριθμο MAFFT εκτελέστηκε σε προσωπικό υπολογιστή με επεξεργαστή Intel Pentium 4 στα 2,66 GHz, με 1 GB RAM και λειτουργικό σύστημα ubudu 8.04.2. Για κάθε είσοδο το πρόγραμμα εκτελέστηκε 5 φορές και χρησιμοποιήθηκε το πρόγραμμα Vtune Performance Analyzer for Linux (έκδοση 9.1) της Intel ώστε να μετρηθούν οι κύκλοι που καταναλώνονται και στη συνέχεια ο χρόνος CPU.

Επειδή οι είσοδοι που ήταν διαθέσιμοι για το πρόγραμμα ήταν μικροί σε διαστάσεις και περιορισμένοι σε αριθμό, έγινε συνδυασμός των εισόδων ώστε να ληφθούν όσο το δυνατόν περισσότερες περιπτώσεις.

Ο μέσος χρόνος εκτέλεσης στον προσωπικό υπολογιστή με επεξεργαστή Intel Pentium 4 στα 2,66 GHz, για τις 5 μετρήσεις σε κάθε μία από τις εισόδους φαίνεται παρακάτω(πίνακας 5-3).

Ακολουθίες εισόδου(διαστάσεις)			Χρόνος εκτέλεσης A__align (sec)
Όνομα	**Πλήθος**	**Μήκος**	
samplerna	5	366	0,007
flydna10x766	10	766	0,101
ex10x766	10	766	0,210
flydna100x766	100	766	0,541
ex100x766	100	766	1,428
ex10x1464	10	1464	0,983
Flydna100x1403	100	1403	1,251

Πίνακας 5-3: Χρόνος εκτέλεσης του συνόλου δεδομένων σε επεξεργαστή Intel Pentium 4 στα 2,66 GHz

Η προσομοίωση της αρχιτεκτονικής που υλοποιήθηκε πραγματοποιήθηκε για μικρού μεγέθους δεδομένα(πλήθος και μήκος ακολουθίας) επειδή όπως αναφέρθηκε προηγουμένως η προσομοίωση με μεγάλες εισόδους είναι αρκετά χρονοβόρα λόγω της μεγάλης απαίτησης σε μνήμη. Έτσι μετρήθηκαν πειραματικά μόνο οι περιπτώσεις μικρής εισόδου για τον υπολογισμό των κύκλων ρολογιού και του χρόνου εκτέλεσης. Παρ' όλα αυτά για να υπάρχει μία πιο πλήρης άποψη για την αποτίμηση της απόδοσης υπολογίστηκε θεωρητικά οι κύκλοι και ο χρόνος εκτέλεσης όλων των πειραμάτων.

Πιο αναλυτικά, το πρώτο και το έκτο στάδιο εξαρτώνται τόσο από το πλήθος των ακολουθιών εισόδου όσο και από το μήκος της κάθε ακολουθίας. Έτσι για τον υπολογισμό των κύκλων ρολογιού λήφθηκαν υπόψη και οι δύο παράμετροι. Τέλος, τα στάδια 2 μέχρι 5 εξαρτώνται μόνο από το μήκος κάθε ακολουθίας, όπου αυτός είναι και ο λόγος που δεν μπορεί η συγκεκριμένη αρχιτεκτονική να υποστηρίξει ακολουθίες με πολύ μεγάλο μήκος.

Παρακάτω παρουσιάζονται οι χρόνοι εκτέλεσης της υλοποίησης σε FPGA των εισόδων για μία μονάδα με συχνότητα ρολογιού **135** MHz, για δύο μονάδες με συχνότητα ρολογιού **115** MHz και για εφτά μονάδες με συχνότητα ρολογιού **104** MHz.

Ακολουθίες εισόδου(διαστάσεις)			Χρόνος εκτέλεσης(sec) 135 MHz (1 unit)	Εκτιμώμενος Χρόνος (sec) 135 MHz (1 unit)
Όνομα	**Πλήθος**	**Μέγιστο Μήκος**		
samplerna	5	366	0,0065	0,0074
flydna10x766	10	766	0,0212	0,0213
ex10x766	10	766	-	0,1719
flydna100x766	100	766	0,108	0,1039
ex100x766	100	766	-	0,2580
ex10x1464	10	1464	-	0,2328
flydna100x1403	100	1403	-	0,2381

Πίνακας 5-4: Χρόνοι εκτέλεσης της αρχιτεκτονικής για μία μονάδα

Ακολουθίες εισόδου(διαστάσεις)			Χρόνος εκτέλεσης(sec) 115 MHz (2 units)	Εκτιμώμενος Χρόνος (sec) 115 MHz (2 units)
Όνομα	**Πλήθος**	**Μέγιστο Μήκος**		
samplerna	5	366	0,0038	0,0043
flydna10x766	10	766	0,0124	0,0125
ex10x766	10	766	-	0,1001
flydna100x766	100	766	0,0633	0,0609
ex100x766	100	766	N/A	N/A
ex10x1464	10	1464	N/A	N/A
flydna100x1403	100	1403	N/A	N/A

Πίνακας 5-5: Χρόνοι εκτέλεσης της αρχιτεκτονικής για 2 μονάδες

Ακολουθίες εισόδου(διαστάσεις)			Χρόνος εκτέλεσης(sec) 104 MHz (7 units)	Εκτιμώμενος Χρόνος (sec) 104 MHz (7 units)
Όνομα	**Πλήθος**	**Μέγιστο Μήκος**		
samplerna	5	366	0,0012	0,0015
flydna10x766	10	766	0,0039	0,0038
ex10x766	10	766	N/A	N/A
flydna100x766	100	766	0,0200	0,0193
ex100x766	100	766	N/A	N/A
ex10x1464	10	1464	N/A	N/A
flydna100x1403	100	1403	N/A	N/A

Πίνακας 5-6: Χρόνοι εκτέλεσης της αρχιτεκτονικής για 7 μονάδες

Από τους παραπάνω πίνακες παρατηρείται ότι ο εκτιμώμενος χρόνος αποκλίνει ελάχιστα από τον πραγματικό χρόνο άρα μπορεί να θεωρηθεί αξιόπιστος για την εκτίμηση της απόδοσης .

Συνδυάζοντας τις πληροφορίες από τους πίνακες 5-3 ,5-4, 5-5 και 5-6 γίνεται σύγκριση του χρόνου εκτέλεσης της FPGA, λαμβάνοντας υπόψη το πραγματικό και τον εκτιμώμενο χρόνο εκτέλεσης τόσο για την μία μονάδα όσο και για δύο και εφτά παράλληλες μονάδες, με το χρόνο εκτέλεσης στο PC. Έτσι προκύπτει ο παρακάτω πίνακας(5-7) που αφορά την επιτάχυνση που πραγματοποιείται από την FPGA αντί για το PC.

Ακολουθίες εισόδου(διαστάσεις)			Αριθμός Μονάδων	Χρόνος εκτέλεσης(sec) FPGA	Χρόνος εκτέλεσης(sec) Pentium 4 στα 2,66 GHz	Speedup FPGA vs. PC
Όνομα	Πλήθος	Μέγιστο Μήκος				
samplerna	5	366	7	0,0012	0,007	5,8
flydna10x766	10	766	7	0,0039	0,101	25,9
ex10x766	10	766	2	0,1001	0,210	2,1
flydna100x766	100	766	7	0,0200	0,541	27,05
ex100x766	100	766	1	0.2580	1,428	5,53
ex10x1464	10	1464	1	0.2328	0,983	4,2
flydna100x1403	100	1403	1	0.2381	1,251	5,25

Πίνακας 5-7 Επιταχύνσεις του χρόνου εκτέλεσης σε FPGA σε σχέση με το PC.

Τέλος αξίζει να παρατηρήσουμε ότι στις περιπτώσεις που έχουμε εισόδους με το ίδιο πλήθος ακολουθιών και με το ίδιο μέγιστο μήκος ακολουθίας, ο χρόνος εκτέλεσης του PC διαφέρει αισθητά. Αυτό οφείλεται στο γεγονός ότι το πρόγραμμα του αλγορίθμου MAFFT στο software είναι αρκετά γρήγορο για όμοιες(ομόλογες) μεταξύ τους ακολουθίες, λόγω της αποδοτικής χρήσης του αλγορίθμου FFT ενώ όσο μειώνεται η ομοιότητα μειώνεται η απόδοση του FFT και συνεπώς η διαφορά του μήκους των ακολουθιών, με αποτέλεσμα να αυξάνεται ο χρόνος CPU. Επειδή όμως δεν υπάρχει σαφή εικόνα της φύσης των ακολουθιών η αλλαγή της επιτάχυνσης διαφέρει ακόμα και σε περιπτώσεις που θεωρείται ότι οι ακολουθίες έχουν την ίδια φύση.

6. Συμπεράσματα μελλοντικές επεκτάσεις και αλλαγές

Σε αυτό το κεφάλαιο αναφέρονται τα συμπεράσματα και κάποιες επεκτάσεις για μελλοντικές αλλαγές.

6.1 Συμπεράσματα

Η παρούσα διπλωματική εργασία ασχολήθηκε με την μελέτη του αλγορίθμου MAFFT, ο οποίος εκτελεί πολλαπλή ευθυγράμμιση ακολουθιών, και την υλοποίησης του κυριότερου μέρους του σε αναδιατασσόμενη λογική. Στόχος ήταν το σύστημα αυτό να είναι αποδοτικότερο σε σύγκριση με έναν προσωπικό υπολογιστή στην προκειμένη περίπτωση έναν Pentium 4 στα 2,66 GHz.

Όπως παρατηρήθηκε από τα αποτελέσματα ο στόχος εκτιμάται ότι ικανοποιείται σε μεγάλο βαθμό στις περιπτώσεις που οι ακολουθίες είναι όμοιες μέταξύ τους και κατά συνέπεια υπάρχει πλήρη εκμετάλευση του παραλληλισμού, ενώ στις περιπτώσεις όπου οι ακολουθίες είναι ανόμοιες μεταξύ τους δεν έχουμε τόσο καλή επιτάχυνση. Όμως ο αλγόριθμος MAFFT και συγκεκριμένα η μέθοδος

που υλοποιείται σε αυτην την εργασία εφαρμόζεται για ακολουθίες που είναι όμοιες μεταξύ τους .

6.2 Μελλοντικές επεκτάσεις

Η αρχιτεκτονική που σχεδιάστηκε, υλοποιήθηκε και παρουσιάστηκε σε αυτή την εργασία, έχει την δυνατότητα να υποστεί αλλαγές που θα βελτιώσουν την απόδοση, αλλά και επεκτάσεις για μία πιο ολοκληρωμένη προσέγγιση του αλγορίθμου MAFFT.

Σημαντικές επεκτάσεις και αλλαγές που μπορούν να πραγματοποιηθούν είναι:

1. Επέκταση της αρχιτεκτονικής για πλήρη υλοποίηση του αλγορίθμου ώστε να μην χρειάζεται κάθε φορά η αρχικοποίηση των μνημών εισόδου, καθώς επίσης και για να υπάρχει μία πιο ξεκάθαρη εικόνα για το μέγεθος της βελτιστοποίησης που επιτυγχάνεται με την υλοποίηση του αλγορίθμου σε αναδιατασσόμενη λογική.
2. Αλλαγή της αρχιτεκτονικής ώστε να γίνει καλύτερη εκμετάλλευση των Block RAMs. Με αυτόν τον τρόπο θα αυξηθεί το εύρος των εισόδων που μπορεί να υποστηρίξει το σύστημα.
3. Χρήση εξωτερικης μνήμης ώστε να χρησιμοποιηθούν όσο το δυνατών περισσότερες παράλληλες μονάδες. Με τον τρόπο αυτό θα αυξηθεί σημαντικά η επιτυτάχυνση το συστήματος.
4. Χρήση συστήματος software και hardware co-design.

7. Βιβλιογραφία

[1] Carrillo H, Lipman D. (1988) The multiple sequence alignment problem in biology. SIAM J. Applied Math.,; 48:1073-1082

[2] Andreas D . Baxevanis and B. F. Francis Ouellette. Bioinformatics. , (2001): A Practical Guide to the Analysis of Genes and Proteins. John Wiley and Sons, Inc.

[3] Erwin SchrÄodinger(1993). What is life. Cambridge University Press.

[4] J.D. Watson and F.H.C. Crick, "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid."

[5] DNA Data Bank of Japan (DDBJ), http://www.ddbj.nig.ac.jp/

[6] J. Venter, et al. The Sequence of the Human Genome. Science, 291, 5507 1304{1351 (2001).

[7] International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. Nature, 409, 860{921.

[8] David W. Mount. Bioinformatics (2001): Sequence and Genome Analysis. Cold Spring Harbor Laboratory University Press.

[9] Ebedes, J., Datta, A. (2004): Multiple sequence alignment in parallel on a workstation cluster, Bioinformatics 20 (7) 1193-1195.

[10] Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W(1994): improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res.; 22:4673–4680.

[11] Edgar RC (2004). MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics; 5:113.

[12] Do, C.B., Mahabhashyam, M.S.P., Brudno, M., and Batzoglou, S. (2005). PROBCONS: Probabilistic Consistency-based Multiple Sequence Alignment. Genome Research 15: 330-340

[13] Notredame, C., Higgins, D., Heringa, J (2000).: T-Coffee: A novel method for multiple sequence alignments., J. Mol. Biol. **302**(1) 205–217,

[14] Katoh, K., Misawa, K., Kuma,K., and Miyata,T. (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acid Res., 30:3059-3066

[15] Katoh, K., Kuma, K., Toh, H., and Miyata,T (2005).,MAFFT version 5: improvement in accuracy of multiple sequence alignment, Nucleic Acids Res. **33**:511-518.

[16] Katoh, Toh , (2008)Recent developments in the MAFFT multiple sequence alignment program , Briefings in Bioinformatics **9**:286-298, Preprint

[17] Needleman SB, Wunsch CD (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol*; 48:443–453.

[18] Sankoff, D., J. B. Kruskal. (1983). Simultaneous comparison of three or more sequences related by a tree. In Sankoff, D. and J. B. Kruskal (eds), *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparisons*. Addison Wesley, London , UK,pp.253-364

[19] Feng DF, Doolittle RF (1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.,*; 25:351–360.

[20] Barton GJ, Sternberg MJ (1987). A strategy for the rapid multiple alignment of protein sequences. Confidence levels from tertiary structure comparisons. *J. Mol. Biol.,*; 198:327–337.

[21] Berger MP, Munson PJ (1991). A novel randomized iterative strategy for aligning multiple protein sequences. *Comput. Appl. Biosci.,*; 7:479–484.

[22] Gotoh O (1993). Optimal alignment between groups of sequences and its application to multiple sequence alignment. *Comput. Appl. Biosci.*; 9:361–370.

[23] S. Rajasekaran, X. Jin, John L. Spouge(2002): The Efficient Computation of Position-Specific Match Scores with the Fast Fourier Transform. Journal of Computational Biology 9(1): 23-33

[24] Jones DT, Taylor WR, Thornton JM (1992). The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci*; 8:275–282.

[25] http://align.bmr.kyushu-u.ac.jp/mafft/software/algorithms/algorithms.html

[26] http://align.bmr.kyushu-u.ac.jp/mafft/software/source.html

[27] Sokal R.R. and Michener C.D. (1958) "A Statistical Method for Evaluating Systematic Relationships". *The University of Kansas Scientific Bulletin* 38: 1409-1438.

[28] Hirosawa M, Totoki Y, Hoshida M, *et al*. Comprehensive study on iterative algorithms of multiple sequence alignment. *Comput. Appl. Biosci.,* 1995; 11:13–18.

[29] Kimura, M. 1983. The Neutral Theory of Molecular Evolution. Cambridge University Press, Cambridge.

[30] Altschul SF, Madden TL, Schaffer AA, *et al*. Gapped BLAST and PSI-BLAST: a new generation of protein database search pro grams. *Nucleic Acids Res.,* 1997; 25:3389–3402.

[31] Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA,* 1988; 85:2444–2448.

[32] Vogt G, Etzold T, and Argos P: An assessment of amino acid exchange matrices in aligning protein sequences: The twilight zone revisited. *J Mol Biol* 1995, 249:816-831.

[33] http://align.bmr.kyushu-u.ac.jp/mafft/software/source.html

[34] IEEE Computer Society (1985), *IEEE Standard for Binary Floating-Point Arithmetic*, IEEE Std 754-1985