Πολυτεχνείο Κρήτης
Τμήμα Ηλεκτρονικών Μηχανικών και Μηχανικών Η/Υ

Εργαστήριο Μικροεπεξεργαστών και Υλικού



"Επιτάχυνση του Αλγορίθμου BLAST με τη χρήση Φίλτρων Bloom και Πολυεπεξεργαστικού Συστήματος"

Γαλανάκης Μ. Κωνσταντίνος

Επιτροπή:
Πνευματικάτος Διονύσιος, Αναπληρωτής Καθηγητής (Επιβλέπων)
Δόλλας Απόστολος, Καθηγητής
Παπαευσταθίου Ιωάννης, Επίκουρος Καθηγητής

Χανιά, Σεπτέμβριος 2008

Περιεχόμενα

Περιεχόμενα 2

Περίληψη 4

Ευχαριστίες 5

1 Υφιστάμενη Τεχνολογία 8

1.1 Αλγόριθμος BLAST 8

1.2 Υλοποιήσεις σε Λογισμικό και Υλικό μέχρι σήμερα 11

1.2.1 Υλοποιήσεις σε Λογισμικό 11

1.2.2 Υλοποιήσεις σε Υλικό 11

1.3 TUC PreBLAST 13

1.3.1 TUC PreBLAST γενικά 13

1.3.2 Η Αρχιτεκτονική του TUC PreBLAST 15

1.4 Φίλτρα Bloom 19

1.4.1 Εισαγωγή 19

1.4.2 Αλγόριθμος Φίλτρου Bloom 19

1.4.3 Πλεονεκτήματα ως προς το χώρο και το χρόνο 20

1.4.4 Πιθανότητα για false positives 21

1.5 Πολυεπεξεργαστικό σύστημα υλοποιημένο σε FPGA (M.PL.EM.) 23

1.5.1 Βασικές Μονάδες του M.PL.EM 23

1.5.2 Υλοποίηση του Πολυεπεξεργαστικού Συστήματος M.PL.EM. 23

1.5.3 Λειτουργία του Πολυεπεξεργαστικού Συστήματος M.PL.EM. 24

1.6 Συνεισφορά της Παρούσας Εργασίας 26

2 Μελέτη 28

2.1 Φίλτρα Bloom στον TUC PreBLAST 28

2.1.1 Προεπεξεργαστής 28

2.1.2 Συναρτήσεις κατακερματισμού 29

2.2 Ολόκληρο Ερώτημα 33

2.2.1 2 συναρτήσεις κατακερματισμού, 2 μνήμες 14 bit διευθύνσεων 33

2.2.2 4 συναρτήσεις κατακερματισμού, 4 μνήμες 14 bit διευθύνσεων 33

2.2.3 8 συναρτήσεις κατακερματισμού, 8 μνήμες 16 bit διευθύνσεων 33

2.2.4 16 συναρτήσεις κατακερματισμού, 16 μνήμες 14 bit διευθύνσεων 34

2.2.5 Συμπεράσματα 34

2.3 Τεμαχισμένο Ερώτημα 35

2.3.1 2 συναρτήσεις κατακερματισμού, 2 μνήμες 14 bit διευθύνσεων 35

2.3.2 4 συναρτήσεις κατακερματισμού, 4 μνήμες 14 bit διευθύνσεων 36
2.3.3 8 συναρτήσεις κατακερματισμού, 8 μνήμες 16 bit διευθύνσεων 38
2.3.4 16 συναρτήσεις κατακερματισμού, 16 μνήμες 14 bit διευθύνσεων 39
2.3.5 Συμπεράσματα 41
2.4 4 Συναρτήσεις Κατακερματισμού, 4 Μνήμες 14 bit Διευθύνσεων 43
2.4.1 Χρωμόσωμα 12 43
2.4.2 Χρωμόσωμα 14 44
2.4.3 Χρωμόσωμα 17 45
2.4.4 Χρωμόσωμα X 46
2.4.5 Συμπεράσματα 48
3 Σχεδίαση και Υλοποίηση 49
3.1 Αρχιτεκτονική TUC PreBLAST με χρήση φίλτρων Bloom 49
3.2 Αρχιτεκτονική πολυεπεξεργαστικού συστήματος M.PL.EM. με ενσωμάτωση TUC PreBLAST με χρήση φίλτρου Bloom 51
4 Μέτρηση Αποδόσεων 55
4.1 TUC PreBLAST με χρήση Φίλτρου BLOOM 55
4.2 Πολυεπεξεργαστικό σύστημα M.PL.EM. με ενσωμάτωση TUC PreBLAST με χρήση Φίλτρου BLOOM 65
5 Συμπεράσματα και Μελλοντική Δουλειά 67
6 Παράρτημα Α 69
Αναφορές 88
Ευρετήριο Σχημάτων 91
Ευρετήριο Πινάκων 92

Περίληψη

Η σύγκριση γενετικών αλληλουχιών, δηλαδή η σύγκριση γονιδίων ή πρωτεϊνών με το γονιδίωμα ενός ή περισσοτέρων οργανισμών αποτελεί ένα από τα πλέον ακριβά υπολογιστικά προβλήματα της υπολογιστικής μοριακής βιολογίας. Το υπολογιστικό κόστος αυξάνει ακόμα περισσότερο με τη συνεχή αύξηση του μεγέθους των βάσεων γενετικών δεδομένων. Από τη δεκαετία του 1970 μέχρι σήμερα έχουν αναπτυχθεί διάφοροι αλγόριθμοι δυναμικού προγραμματισμού που αποσκοπούν στην επίλυση του παραπάνω προβλήματος. Από τις αρχές της δεκαετία του 1990 έχει επικρατήσει ο αλγόριθμος BLAST, ο οποίος εξελίσσεται μέχρι και σήμερα από το National Center for Biotechnology Information (NCBI) με κοινή προσπάθεια της ερευνητικής κοινότητας. Ο υψηλός υπολογιστικός φόρτος της εφαρμογής, οδήγησε στην ιδέα της σχεδίασης συστημάτων βασισμένα σε αναδιατασσόμενη λογική, τα οποία υλοποιούν τον αλγόριθμο BLAST. Οι προτεινόμενες αρχιτεκτονικές έχουν ως σκοπό την ταχύτερη επεξεργασία των δεδομένων.

Την προσπάθεια αυτή ωφέλησε και η ανάπτυξη της τεχνολογίας αναδιατασσόμενης λογικής. Η ανάπτυξη αυτή, επέτρεψε τη δημιουργία πολυεπεξεργαστικών ενσωματωμένων συστημάτων, τα οποία μειώνουν την ποσότητα του υλικού ενώ κάνουν τη σχεδίαση πιο ευέλικτη με τη χρήση λογισμικού. Τα πολυεπεξεργαστικά συστήματα είναι δίκτυα προγραμματιζόμενων επεξεργαστών κατασκευασμένων από επεξεργαστικά στοιχεία, στοιχεία λογικής και μνήμες σε μια FPGA. Η επιλογή του αριθμού των επεξεργαστών και η διασύνδεση τους με τα υπόλοιπα στοιχεία, έγκειται στην ευχέρεια του σχεδιαστή, ανάλογα με την επιθυμητή εφαρμογή προς υλοποίηση.

Σ' αυτή την εργασία, παρουσιάζεται μια παραλλαγή ενός προεπεξεργαστή του αλγορίθμου BLAST, ο οποίος υλοποιήθηκε στο Πολυτεχνείο Κρήτης και στο Εργαστήριο Μικροεπεξεργαστών και Υλικού, με το όνομα TUC PreBLAST. Η χρήση του προεπεξεργαστή προσφέρει μείωση του διαστήματος αναζήτησης, επομένως και επιτάχυνση της χρονικής εκτέλεσης του αλγορίθμου. Η παραλλαγή που προτείνει αυτή η εργασία, χρησιμοποιεί φίλτρο BLOOM για τον κατακερματισμό των δεδομένων, αντί για ψευδοτυχαίο κατακερματισμό, τον οποίο χρησιμοποιούσε η αρχική υλοποίηση.

Επίσης, γίνεται ενσωμάτωση του προεπεξεργαστή με τη χρήση φίλτρου BLOOM σε ένα πολυεπεξεργαστικό σύστημα, το οποίο υλοποιήθηκε στο Πολυτεχνείο Κρήτης και στο Εργαστήριο Μικροεπεξεργαστών και Υλικού, με το όνομα M.PL.EM. και το οποίο εκτελεί τον αλγόριθμο BLAST.

Ευχαριστίες

Η εργασία αυτή εκπονήθηκε στο Εργαστήριο Μικροεπεξεργαστών και Υλικού (Microprocessor and Hardware Laboratory – MHL) του Πολυτεχνείου Κρήτης (Technical University of Crete – TUC), υπό την επίβλεψη του Αναπληρωτή Καθηγητή κ. Διονύσιου Πνευματικάτου, κατά τη διάρκεια του ακαδημαϊκού έτους 2007-2008.

Για την εκπόνηση και την ολοκλήρωσης αυτής της εργασίας, θα ήθελα πρωτίστως να ευχαριστήσω τον επιβλέποντα καθηγητή μου, τον κ. Διονύσιο Πνευματικάτο, όχι μόνο για τη συνεισφορά και την υποστήριξη του πάνω στην ολοκλήρωση αυτής της εργασίας, αλλά και στην εμπιστοσύνη που έδειξε από την αρχή στις γνώσεις και τις δυνατότητές μου.

Θα ήθελα να ευχαριστήσω επίσης τον Καθηγητή κ. Απόστολο Δόλλα και τον Αναπληρωτή Καθηγητή κ. Ιωάννη Παπαευσταθίου, οι οποίοι εκτός από τις γνώσεις που μου προσέφεραν κατά τη διάρκεια των προπτυχιακών χρόνων, βοήθησαν και στην ολοκλήρωση αυτής της εργασίας ως μέλη της εξεταστική επιτροπής.

Θα ήθελα να πω ένα μεγάλο ευχαριστώ στο Διδακτορικό Φοιτητή κ. Ευριπίδη Σωτηριάδη, χωρίς τη βοήθεια του οποίου δε θα είχε ολοκληρωθεί αυτή η εργασία. Θέλω να τον ευχαριστήσω για την καθοδήγηση, τη συνεργασία και την αμέριστη βοήθεια του σε θέματα που αφορούσαν την εργασία και μη. Θέλω να τον ευχαριστήσω που ήταν πάντα στην άλλη μεριά του τηλεφώνου, του υπολογιστή, του γραφείου, του τραπεζιού. Ευριπίδη, σ' ευχαριστώ.

Ένα εξίσου μεγάλο ευχαριστώ αξίζει στους φίλους μου Παναγιώτη Αφράτη και Γιώργο Μπλεμένο, οι οποίοι μου μετέδωσαν γνώσεις και τεχνογνωσία πάνω στα αντικείμενα των διπλωματικών τους εργασιών. Να τους ευχαριστήσω για την επιμονή, την υπομονή και την ανοχή που επέδειξαν όλον αυτό τον καιρό, ώστε να βρίσκεται αυτή η εργασία σήμερα στα χέρια σας. Παναγιώτη και Γιώργο, η βοήθειά σας ήταν η καλύτερη στήριξη εντός και εκτός εργαστηρίου.

Ευχαριστώ πολύ και την υπόλοιπη ομάδα του Εργαστηρίου Μικροεπεξεργαστών και Υλικού, τους συνεργάτες μου που στήριξαν την καθημερινότητα της δουλειάς μου με συμβουλές, προτάσεις μα και χαμόγελο. Αυτή η εργασία ανήκει και σ' αυτούς.

Ένα μεγάλο ευχαριστώ αξίζει σε όλους τους συναδέλφους και φίλους μου που με στήριξαν και τους στήριξα σε όλη τη διάρκεια των προπτυχιακών σπουδών και μαζί

ανεβήκαμε τα σκαλιά της ολοκλήρωσης και της επιτυχίας αυτού του δύσκολου δρόμου που διαλέξαμε να ακολουθήσουμε 7 χρόνια πριν.

Ευχαριστώ πολύ τους φίλους και τις φίλες μου, με τους οποίους μεγαλώσαμε, ζήσαμε, διασκεδάσαμε και αγαπήσαμε μαζί. Η στήριξη και η αγάπη που μου έδειξαν από την περίοδο των σχολικών χρόνων μέχρι και σήμερα ήταν αυτή που μου έδωσε δύναμη, κουράγιο μα κυριότερα λύσεις στα «προβλήματά» μας.

Τέλος, θέλω να εκφράσω την ευγνωμοσύνη που νιώθω για τους γονείς και τον αδερφό μου, που αποτέλεσαν, αποτελούν και θα αποτελούν το πιο σταθερό στήριγμα σε όλες τις σκέψεις, ιδέες και προσπάθειες της ζωής μου, αλλά και για τα ψυχικά και οικονομικά εφόδια που μου παρέχουν ώστε να μπορώ να πραγματοποιώ κάθε όνειρο και φιλοδοξία μου. Σας ευχαριστώ πολύ.

στο Μιχάλη, στην Ελένη, στον Ερμαίο

1 Υφιστάμενη Τεχνολογία

1.1 Αλγόριθμος BLAST

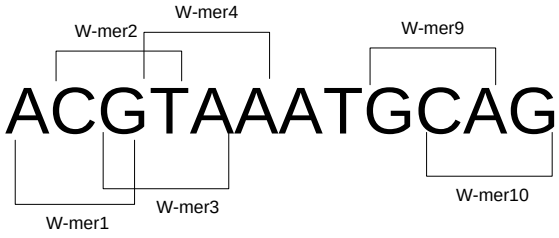
Από τις αρχές τις δεκαετίας του 1970, ξεκίνησαν προσπάθειες για την επιτάχυνση αντιστοίχησης ακολουθιών και την αναζήτηση βάσεων γενετικών δεδομένων, με τη χρήση υπολογιστών. Χρησιμοποιήθηκαν τεχνικές δυναμικού προγραμματισμού για την ανάπτυξη αλγορίθμων καθολική σύγκρισης. Ο αλγόριθμος των Needleman και Wunsch [**1**] παρείχε μια συνολικά βέλτιστη λύση. Στις αρχές της δεκαετίας του 1980, οι Smith και Waterman [**2**] παρουσίασαν το δικό τους αλγόριθμο, ο οποίος επίσης χρησιμοποιούσε δυναμικό προγραμματισμό και παρείχε μια τοπικά βέλτιστη λύση. Μια πιο γρήγορη προσέγγιση της λύσης του προβλήματος παρουσιάστηκε στα τέλη της δεκαετίας του 1980 και στις αρχές της δεκαετίας του 1990 με τον FASTA [**3**] και τον BLAST [**4**]. Οι αλγόριθμοι FASTA και BLAST δεν είναι βέλτιστοι, ωστόσο είναι γρήγοροι και επειδή οι βάσεις γενετικών δεδομένων αυξάνονται εκθετικά και με ρυθμό γρηγορότερο από το νόμο του Moore [**5**] αυτοί οι αλγόριθμοι επικράτησαν στην κοινότητα της Υπολογιστικής Βιολογίας. Από αυτούς τους αλγορίθμους, ο BLAST είναι ο πιο δημοφιλής.

BLAST είναι το ακρωνύμιο των λέξεων Basic Logical Alignment Search Tool. Ο Blast παρουσιάστηκε για πρώτη φορά από τον S. F. Altschul το 1990 [**4**] και θεωρείται ως ο πιο ευρέως χρησιμοποιούμενος ευριστικός αλγόριθμος της Υπολογιστικής Βιολογίας. Ο BLAST χρησιμοποιείται αφενός για την αναζήτηση μεγάλων βάσεων γενετικών δεδομένων προκειμένου να βρεθούν αντιστοιχίες μεταξύ της βάσης και ενός ερωτήματος γενετικών δεδομένων και αφετέρου μπορεί να χρησιμοποιηθεί ως κομμάτι άλλων εφαρμογών, όπως σε αλγορίθμους βιοπληροφορικής.

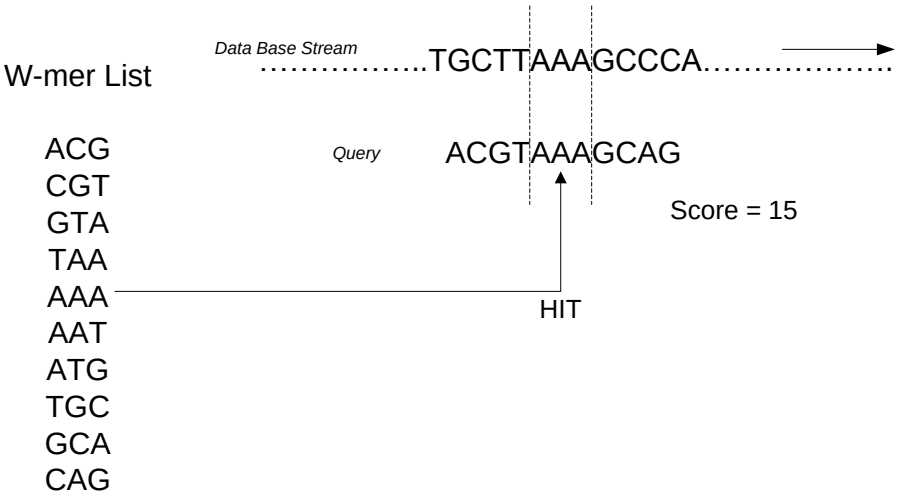
Οι είσοδοι του αλγορίθμου είναι μια βάση γενετικών δεδομένων και ένα ερώτημα. Ο αλγόριθμος ψάχνει να βρει περιοχές υψηλής ομοιότητας του ερωτήματος στη βάση γενετικών δεδομένων. Έξοδοι του αλγορίθμου είναι οι θέσεις των περιοχών όπου βάση και ερώτημα παρουσιάζουν υψηλή ομοιότητα, καθώς επίσης και η βαθμολογία αυτών των ομοιοτήτων. Κάθε ζευγάρι, αποτελούμενο από μια περιοχή της βάσης γενετικών δεδομένων και μιας περιοχής του ερωτήματος, καλείται High Score Pair (HSP). Η βαθμολογία έχει σημαντική αξία για τους βιολόγους, αφού χρησιμοποιείται για να υπολογιστούν συγκεκριμένες τιμές κάποιων μεταβλητών, από τις οποίες, η τιμή του e είναι η πιο σημαντική.

Ο αλγόριθμος BLAST χωρίζεται σε 3 βήματα.

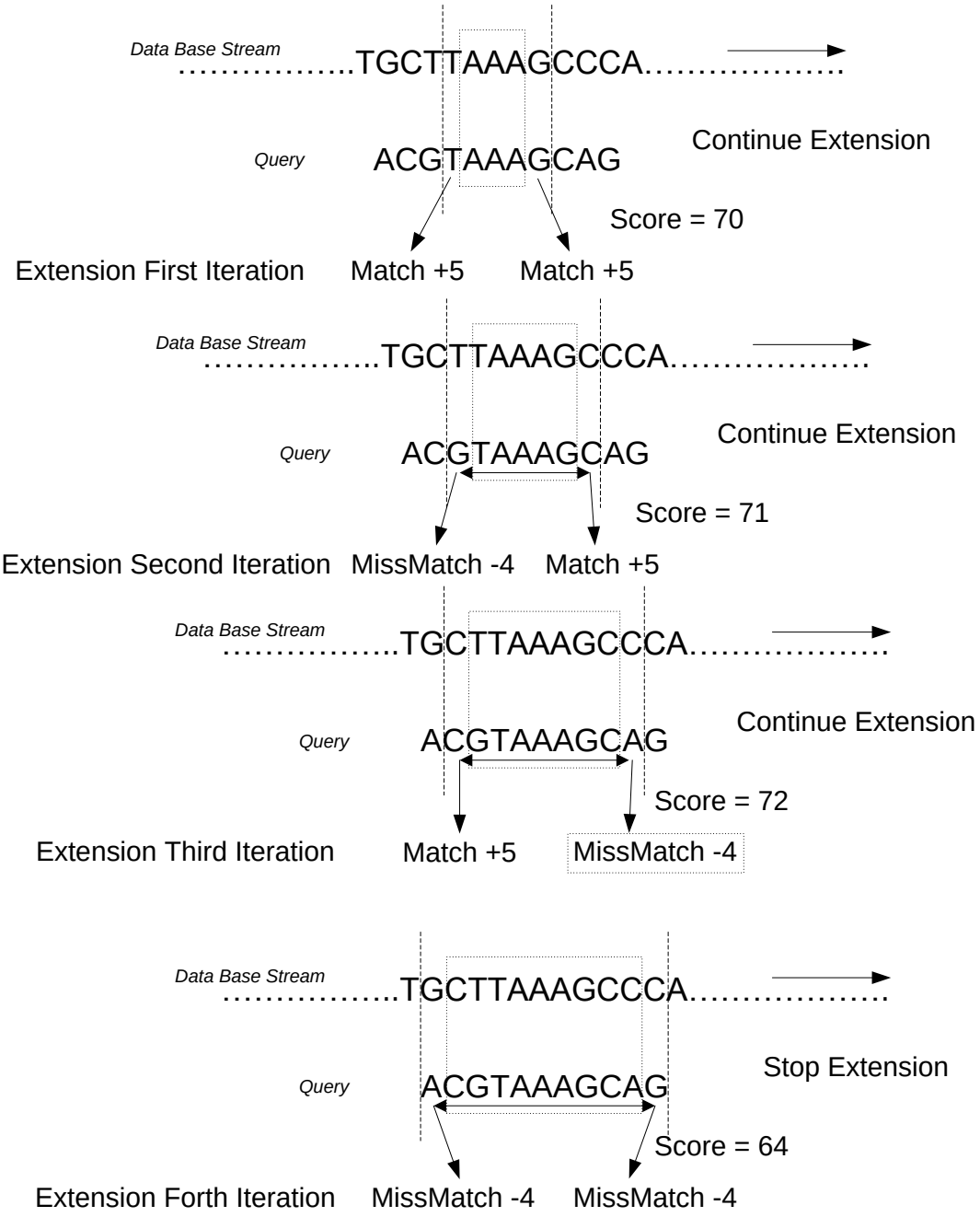
1. Στο πρώτο βήμα, γίνεται επεξεργασία του ερωτήματος, με το να χωρίζεται σε λέξεις μήκους W, οι οποίες ονομάζονται w-mers και αποτελούν συνεχή μέρη του ερωτήματος. Αν, για παράδειγμα, έχουμε το ερώτημα ACGTAAATGCAG και μήκος W=3 τότε θα προκύψουν 10 w-mers όπως φαίνεται στο Σχήμα 1-1.
2. Στο δεύτερο βήμα του αλγορίθμου, γίνεται αναζήτηση αντιστοιχιών στη βάση γενετικών δεδομένων. Αναζητούνται ακριβείς αντιστοιχήσεις ανάμεσα σε οποιοδήποτε w-mer και του εξεταζόμενου μέρους της βάσης. Αναπαράσταση του βήματος αυτού, φαίνεται στο Σχήμα 1-2.
3. Στο τρίτο βήμα, γίνεται επεξεργασία των αντιστοιχιών που προέκυψαν από το δεύτερο βήμα του αλγορίθμου, με σκοπό να γίνει επέκταση και προς τις δύο κατευθύνσεις κατά το μέγιστο δυνατό για όλες αυτές τις αντιστοιχίες. Το μέγεθος της επέκτασης μετριέται με μια μέθοδο βαθμολόγησης της επέκτασης, βασισμένης σε πίνακες. Το τελικό αποτέλεσμα είναι τα κομμάτια της βάσης γενετικών δεδομένων που πέτυχαν τη μεγαλύτερη (τοπικά) βαθμολόγηση. Αναπαράσταση του βήματος αυτού, φαίνεται στο Σχήμα 1-3.



Σχήμα 1-1: Το πρώτο βήμα του αλγορίθμου BLAST



Σχήμα 1-2: Το δεύτερο βήμα του αλγορίθμου BLAST



Σχήμα 1-3: Το τρίτο βήμα του αλγορίθμου BLAST

1.2 Υλοποιήσεις σε Λογισμικό και Υλικό μέχρι σήμερα

Διάφορες προσπάθειες έχουν γίνει μέχρι σήμερα για αποτελεσματική αντιστοίχιση ακολουθιών DNA και αναζήτηση βάσεων γενετικών δεδομένων. Η κοινότητα της βιοπληροφορικής έχει διενεργήσει έντονες προσπάθειες για την υλοποίηση εργαλείων Λογισμικού, αλλά εξαιτίας της υπολογιστικής ισχύς που απαιτούν αυτοί οι αλγόριθμοι καθώς και του μεγάλου όγκου δεδομένων που πρέπει να επεξεργαστούν, τα τελευταία χρόνια γίνονται, επίσης, και προσπάθειες ανάπτυξης Υλικού.

1.2.1 Υλοποιήσεις σε Λογισμικό

Συλλογικές προσπάθειες από πολλές ομάδες για αντιστοίχιση ακολουθιών DNA και αναζήτηση βάσεων γενετικών δεδομένων συγκεντρώνονται στο Εθνικό Κέντρο Βιοτεχνολογικής Πληροφορίας [**6**], το οποίο κυριαρχεί στον τομέα παροχής εργαλείων ανοιχτού κώδικα υλοποίησης του αλγορίθμου BLAST. Αυτές οι υλοποιήσεις έχουν χρησιμοποιηθεί επανειλημμένα και σαν προγράμματα αξιολόγησης υπολογιστικών συστημάτων από μεγάλους κατασκευαστές υπολογιστών όπως η IBM [**7**], η DELL [**8**] και η APPLE [**9**].

1.2.2 Υλοποιήσεις σε Υλικό

Η κοινότητα αναδιατασσόμενου υλικού χρησιμοποίησε την αντιστοίχηση ακολουθιών DNA και την αναζήτηση βάσεων γενετικών δεδομένων για να δείξουν την ικανότητα των FPGAs να λύσουν «ακριβά» υπολογιστικά προβλήματα. Η πλατφόρμα Splash 2, η οποία χρησιμοποιήθηκε στη διάρκεια των αρχών της δεκαετίας του 1990 από την ερευνητική ομάδα του Hoang [**10**], [**11**] για να λύσουν το συγκεκριμένο πρόβλημα χρησιμοποιώντας τον αλγόριθμο Smith Waterman. Αργότερα, η ομάδα του Guccione [**12**] χρησιμοποίησε την τεχνολογία Jbits. Στο Virginia Tech Configurable Computing Laboratory [**13**] αλλά και στο Nanyang Technological University [**14**] χρησιμοποίησαν run time reconfiguration για το ίδιο πρόβλημα, υλοποιώντας και αυτοί τον αλγόριθμο Smith Waterman.

Μέχρι σήμερα έχουν γίνει και διάφορες ακαδημαϊκές προσεγγίσεις για την υλοποίηση του αλγορίθμου BLAST. Η πρώτη είναι το project RC-BLAST [**15**], όπου οι σχεδιαστές του υλοποίησαν πλήρως τον αλγόριθμο BLAST του NCBI, αλλά η απόδοση του αλγορίθμου αυτού δεν ήταν τόσο καλή και μάλιστα αποδείχθηκε ότι ήταν ακόμα χειρότερη και από την ανάλογη υλοποίηση σε Λογισμικό. Μια πιο πρόσφατη ακαδημαϊκή προσπάθεια για την αντιστοίχηση ακολουθιών DNA και την αναζήτηση γενετικών βάσεων δεδομένων παρουσιάστηκε το 2005 από το CAAD Lab. στο

πανεπιστήμιο της Βοστώνης [**16**]. Υλοποίησαν τον αλγόριθμο BLAST για μικρά ερωτήματα, μέχρι και 800 χαρακτήρες, προσφέροντας ουσιαστική επιτάχυνση σε σύγκριση με τη υλοποίηση σε Λογισμικό. Το project αυτό ονομάζεται TreeBLAST. Η παράλληλη αρχιτεκτονική Mercury BLAST [**17**] & [**18**] παρουσιάστηκε από το πανεπιστήμιο της Ουάσινγκτον στο St. Louis, υλοποιώντας τον αλγόριθμο BLASTn προσφέροντας μια καλή επιτάχυνση σε σύγκριση με την υλοποίηση σε Λογισμικό η οποία τρέχει σε έναν υπολογιστή γενικού σκοπού. Η πιο πρόσφατη υλοποίηση του αλγορίθμου BLAST με τη χρήση αναδιατασσόμενης λογικής, παρουσιάστηκε από τον IRISA CNRS στη Γαλλία και από το Institute of Computing Technology στο Πεκίνο, Κίνα [**19**]. Είναι μια πλακέτα με αναδιατασσόμενη λογική η οποία λειτουργεί μαζί με μνήμη FLASH, 64 GB πάνω στην οποία υλοποιήθηκαν οι αλγόριθμοι BLASTx/TBLASTn/TBLASTx.

Μαζί με τις ακαδημαϊκές προσπάθειες για την υλοποίηση του αλγορίθμου BLAST σε αναδιατασσόμενη λογική, γίνονται και κάποιες εμπορικές. Εταιρίες όπως η Timelogic Inc. [**20**] και το Πανεπιστήμιο της Καλιφόρνια (BEE engine as a configurable supercomputer – μια από τις εφαρμογές του ήταν και η υλοποίηση του αλγορίθμου BLAST [**21**]) με τα συστήματά τους, όπου κατάφεραν πολύ εντυπωσιακά αποτελέσματα με τις υλοποιήσεις του αλγορίθμου BLAST. Δε δημοσιεύουν, όμως, πληροφορίες για τα δεδομένα που χρησιμοποίησαν, τις τεχνικές μέτρησης των αποτελεσμάτων, ούτε καν τεχνικές πληροφορίες για την υλοποίησή τους. Κατά συνέπεια, τα αποτελέσματά τους δε μπορούν αν συγκριθούν ευθέως με άλλα.

Την ίδια περίοδο με τα project TreeBLAST και Mercury BLAST, μια νέα αρχιτεκτονική για την υλοποίηση του αλγορίθμου BLASTn αναπτύχθηκε από το Πολυτεχνείο Κρήτης (TUC) [**22**], [**23**], [**24**], [**25**] βασισμένη σε αναδιατασσόμενη λογική.

Από το Πολυτεχνείο Κρήτης σχεδιάστηκε και υλοποιήθηκε ακόμα ένα σύστημα, το οποίο ουσιαστικά μειώνει τον υπολογιστικό φόρτο του αλγορίθμου BLAST πριν ακόμα αυτός τρέξει. Υλοποιεί ένα prefiltering της βάσης γενετικών δεδομένων και εξάγει ένα μικρότερο κομμάτι αυτής, το οποίο όμως, θα δώσει τα ίδια αποτελέσματα με την αρχική. Το σύστημα αυτό ονομάζεται TUC PreBLAST [**26**], [**27**], [**28**].

1.3 TUC PreBLAST

1.3.1 TUC PreBLAST γενικά

Ο TUC PreBLAST αποτελεί μια διαφορετική προσέγγιση επίλυσης του προβλήματος αύξησης του υπολογιστικού φόρτου του αλγορίθμου BLAST, η οποία προκύπτει από την ανάγκη επεξεργασίας όλο και μεγαλύτερων βάσεων γενετικών δεδομένων. Δεν επιλύει τον αλγόριθμο, αλλά τον επιταχύνει μειώνοντας τον τελικό υπολογιστικό φόρτο.

Η προσέγγιση της αρχιτεκτονικής του TUC PreBLAST για το prefiltering βασίζεται στην ιδέα ότι ο αλγόριθμος BLAST βρίσκει αντιστοιχίες στις περιοχές υψηλής ομοιότητας ανάμεσα στη βάση γενετικών δεδομένων και στο ερώτημα. Στις περιοχές εκείνες, το τρίτο βήμα του αλγορίθμου BLAST θα προχωρήσει σε μεγάλο αριθμό επεκτάσεων. Με τον όρο «περιοχές υψηλής ομοιότητας» αναφέρονται οι περιοχές εκείνες όπου εμφανίζεται μεγάλος αριθμός ομοιοτήτων ανάμεσα στα w-mers του ερωτήματος και σε εκείνα της βάσης γενετικών δεδομένων.

Ωστόσο, ο TUC PreBLAST δεν αναζητά πραγματικές αντιστοιχίες μεταξύ βάσης γενετικών δεδομένων και ερωτήματος. Αντίθετα, αναζητά τις πιθανές αντιστοιχίες.

Αν σε μια μικρή περιοχή της βάσης (παράθυρο-window) εμφανίζεται μεγάλος αριθμός αντιστοιχιών μεταξύ της βάσης γενετικών δεδομένων και του ερωτήματος, τότε υπάρχει πιθανότητα αυτή η περιοχή να είναι περιοχή υψηλής ομοιότητας. Γίνεται λόγος για πιθανότητα και όχι για βεβαιότητα, αφού πολλές από τις αντιστοιχίες ανάμεσα στη βάση γενετικών δεδομένων και στο ερώτημα μπορεί να έχουν δημιουργηθεί από πολλαπλά w-mers σε ακανόνιστη σειρά, τα οποία όμως δε μπορούν να προκαλέσουν επεκτάσεις κατά το τρίτο βήμα του αλγορίθμου BLAST.

Ο TUC PreBLAST, επιτρέπει τη χρήση μεταβλητού παραθύρου (window), τη μεταβλητή τιμή κατωφλίου (threshold) αλλά και το μεταβλητό πλήθος χαρακτήρων του ερωτήματος. Παράθυρο (window), όπως αναφέρθηκε και προηγουμένως, είναι μια μικρή περιοχή της βάσης την οποία επεξεργάζεται ο TUC PreBLAST την κάθε στιγμή και υπολογίζει το ποσοστό αντιστοιχιών μεταξύ βάσης γενετικών δεδομένων και ερωτήματος. Κατώφλι (threshold), ονομάζεται εκείνη η τιμή την οποία όταν θα φτάσει αλλά και θα ξεπεράσει ο αριθμός των αντιστοιχιών μεταξύ βάσης γενετικών δεδομένων και ερωτήματος για μια δεδομένη περιοχή, τότε αυτή η περιοχή θα θεωρείται περιοχή υψηλής ομοιότητας.

Μεγάλα μεγέθη παραθύρου οδηγούν στην εύρεση περισσότερων αντιστοιχιών μέσα στο παράθυρο αυτό και κατά συνέπεια ανεβάζουν το ποσοστό των αντιστοιχιών. Από έρευνες που έγιναν με διάφορα μεγέθη παραθύρου, προκύπτει ότι όσο μεγαλώνει το μέγεθος του παραθύρου, το μέγεθος του διαστήματος της βάσης γενετικών δεδομένων που εξάγει ο TUC PreBLAST παραμένει είτε ανεπηρέαστο, είτε αυξάνεται. Για το λόγο αυτό, το μικρό μέγεθος του παραθύρου, φαίνεται να είναι πιο αποτελεσματικό και παράλληλα ικανό να διατηρήσει την απαραίτητη πληροφορία.

Αναφορικά με τη μεταβλητή τιμή κατωφλίου και σύμφωνα με τις έρευνες που έγιναν, προκύπτει ότι όσο η τιμή κατωφλίου αυξάνεται, τόσο μειώνεται το τελικό διάστημα της βάσης γενετικών δεδομένων που εξάγει ο TUC PreBLAST. Παρόλο που κάτι τέτοιο είναι επιθυμητό, η επιλογή της τιμής του κατωφλίου δε μπορεί να γίνει αυθαίρετα. Ορίζοντας μικρή τιμή για το κατώφλι, τότε ο TUC PreBLAST θα εξάγει μεγάλο διάστημα της βάσης γενετικών δεδομένων. Ορίζοντας μεγάλη τιμή για το κατώφλι, υπάρχει η πιθανότητα, στο διάστημα της βάσης γενετικών δεδομένων που θα εξάγει οTUC PreBLAST να αγνοηθούν περιοχές της αρχικής βάσης γενετικών δεδομένων που ενδεχομένως να περιείχαν αντιστοιχίες. Στη σχεδίαση του TUC PreBLAST επιλέχθηκε τιμή κατωφλίου 2. Η επιλογή αυτή βασίστηκε στο συλλογισμό ότι για να ξεκινήσει τη διαδικασία επέκτασης, στο τρίτο βήμα, ο αλγόριθμος BLAST, χρειάζεται τουλάχιστον μια αντιστοιχία. Από τη στιγμή που υπάρχει τουλάχιστον μία επέκταση, τότε θα βρεθεί άλλη μια αντιστοιχία για ένα w-mer, το οποίο επικαλύπτει το πρώτο.

Τέλος, όσον αφορά το μεταβλητό μέγεθος ερωτήματος, από τις έρευνες προκύπτει ότι με το μικρού μεγέθους ερώτημα η διαδικασία του prefiltering είναι πιο αποδοτική. Μια πιθανή εξήγηση γι' αυτό το αποτέλεσμα είναι ότι ένα ερώτημα μεγάλου μεγέθους περιέχει περισσότερα w-mers από ένα μικρό, επομένως η πιθανότητα εύρεσης αντιστοιχιών μεταξύ της βάσης γενετικών δεδομένων και δύο (2 εξαιτίας του μεγέθους κατωφλιού) από αυτών είναι μεγαλύτερη. Προκύπτει, επομένως, το συμπέρασμα ότι μικρά μεγέθη ερωτήματος, καθιστούν τη λειτουργία του TUC PreBLAST πιο αποδοτική. Ο TUC PreBLAST, για να εκμεταλλευτεί την ιδιότητα αυτή, προχωράει σε τεμαχισμό του ερωτήματος σε ερωτήματα μικρότερου μεγέθους και πιο συγκεκριμένα σε τεμαχισμένα ερωτήματα των 1000, 500, 250 χαρακτήρων κ. ο. κ.

Η ιδέα του prefiltering, βασίζεται στην εύρεση των αντιστοιχιών μεταξύ βάσης γενετικών δεδομένων και ερωτήματος. Για να βρεθούν αυτές οι αντιστοιχίες, θα έπρεπε να γίνει σύγκριση μεταξύ όλων των w-mers του ερωτήματος με ολόκληρη τη βάση γενετικών δεδομένων. Οι συγκρίσεις αυτές είναι μεγέθους 24 bits (12 χαρακτήρες x 2

bits/χαρακτήρα) και πλησιάζουν σε ποσότητα το μέγεθος του ερωτήματος. Το υπολογιστικό κόστος για τις συγκρίσεις αυτές είναι απαγορευτικά υψηλό. Η λύση που προτείνει ο TUC PreBLAST για αυτό το πρόβλημα είναι η αναζήτηση πιθανών αντιστοιχιών και όχι πραγματικών. Ως «πιθανή αντιστοιχία» ορίζεται η αντιστοιχία μεταξύ των bits των θέσεων 0 ως 14, 3 ως 17, 6 ως 20 και 9 ως 23 ενός w-mer, με το εκάστοτε κομμάτι της βάσης γενετικών δεδομένων, το οποίο υπόκειται σε επεξεργασία. Η αντιστοιχία που θα προκύψει δεν είναι απαραίτητα πραγματική. Τα αποτελέσματα της μεθόδου αυτής είναι χειρότερα από αυτά που θα προέκυπταν αν αναζητιόντουσαν πραγματικές αντιστοιχίες, δηλαδή με τη μέθοδο του TUC PreBLAST το τελικό διάστημα της βάσης γενετικών δεδομένων που εξάγει η μονάδα του prefiltering συγκριτικά με αυτό που περιέχει μόνο τις πραγματικές αντιστοιχίες μεταξύ βάσης γενετικών δεδομένων και ερωτήματος, δεν είναι πολύ μεγαλύτερο.

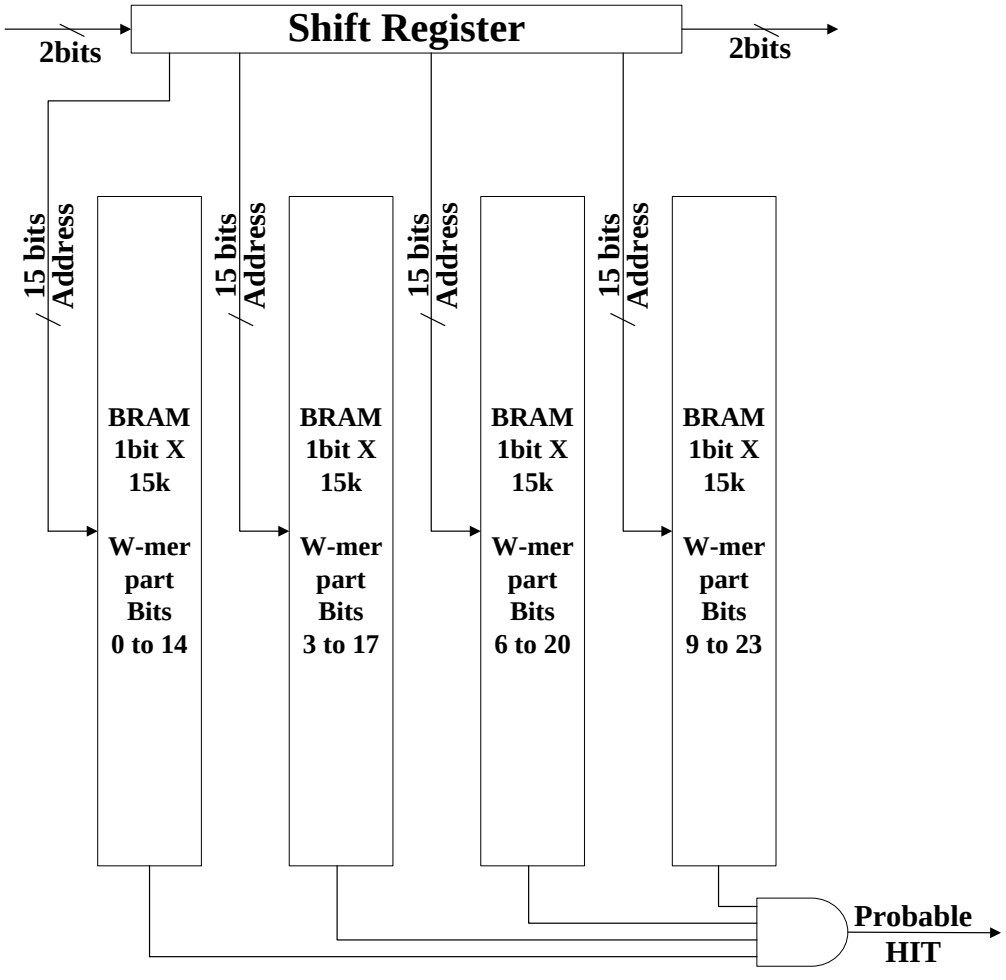
Το μέγεθος των κομματιών στο οποίο τεμαχίζεται το w-mer δεν επιλέχθηκε τυχαία. Η ιδέα ήταν να χρησιμοποιηθούν 4 μνήμες των 32768 θέσεων (2^{15} θέσεις μνήμης, όσες και οι θέσεις μνήμης μιας BRAM της XILINX), η κάθε μια από της οποίες θα διευθυσιοδοτούταν από το αντίστοιχο κομμάτι του w-mer. Η μνήμες, κατά την αρχικοποίηση του συστήματος γεμίζουν με '0' σε όλες τις θέσεις τους. Όταν μπει στον TUC PreBLAST ένα w-mer από το ερώτημα, τότε αυτό το w-mer τεμαχίζεται σε δεκαπεντάδες, και κάθε δεκαπεντάδα γίνεται εκείνη η διεύθυνση της μνήμης, στην οποία το σύστημα θα τοποθετήσει ένα '1'. Στη συνέχεια και αφού προσπελαστεί όλο το ερώτημα, θα αρχίσει να εισάγεται στο σύστημα η βάση γενετικών δεδομένων. Τα w-mers της βάσης γενετικών δεδομένων θα τεμαχίζονται, επίσης, με τον ίδιο τρόπο ώστε να προκύψουν οι ανάλογες δεκαπεντάδες. Αυτές τις δεκαπεντάδες θα πρέπει να χρησιμοποιήσει το σύστημα ώστε να βρει τις διευθύνσεις των μνημών, στις οποίες πρέπει να ψάξει και να εντοπίσει τα '1'. Αν και οι 4 διευθύνσεις στη μνήμη περιέχουν '1', τότε εντοπίστηκε μια πιθανή αντιστοιχία.

1.3.2 Η Αρχιτεκτονική του TUC PreBLAST

Ο TUC PreBLAST αποτελείται από δύο κομμάτια. Το πρώτο κομμάτι είναι αυτό που περιέχει τις μνήμες στις οποίες θα γίνει η αρχικοποίηση από το ερώτημα και ο έλεγχος από τη βάση. Είναι, δηλαδή, το κομμάτι εκείνο, το οποίο θα εντοπίσει τις πιθανές ομοιότητες μεταξύ βάσης γενετικών δεδομένων και ερωτήματος. Το δεύτερο κομμάτι είναι αυτό το οποίο εντοπίζει και καταμετράει τις πιθανές επιτυχείς αντιστοιχίες και αποθηκεύει τις θέσεις των αντιστοιχιών αυτών σε μια μνήμη. Δηλαδή, καταμετρά τις πιθανές ομοιότητες που προκύπτουν από το προηγούμενο κομμάτι σε συγκεκριμένες περιοχές και εφαρμόζει το κατώφλι για τις περιοχές αυτές. Τελικά και

εφόσον οι πιθανές ομοιότητες της κάθε περιοχής είναι ίσες ή και ξεπερνούν την τιμή του κατωφλιού, οι θέσεις των περιοχών αυτών αποθηκεύονται σε μια μνήμη.

Το πρώτο κομμάτι του TUC PreBLAST, θα ονομάζεται κομμάτι των μνημών. Το κομμάτι των μνημών (1 καταχωρητής ολίσθησης, 4 μνήμες των 2^{15} θέσεων και μία πύλη AND 4 εισόδων) έχει την ακόλουθη λειτουργικότητα. Το ερώτημα εισάγεται στο σύστημα σαν ροή δεδομένων από τον καταχωρητή ολίσθησης του κομματιού των μνημών και αρχικοποιεί τις μνήμες με τον τρόπο που περιγράφηκε παραπάνω. Έπειτα εισάγεται στον καταχωρητή ολίσθησης, και πάλι ως ροή δεδομένων, η βάση γενετικών δεδομένων και γίνεται ο έλεγχος στα περιεχόμενα των μνημών με τον τρόπο που περιγράφηκε παραπάνω. Οι «απαντήσεις» της μνήμης, δηλαδή τα '1' ή τα '0' που επιστρέφουν οι μνήμες περνούν μέσα από την πύλη AND. Όταν η έξοδος της πύλης είναι '1' τότε έχουμε πιθανή «επιτυχία». Ειδάλλως έχουμε σίγουρη «αποτυχία». Σε κάθε κύκλο ρολογιού, η μονάδα ελέγχου ελέγχει για μία νέα πιθανή επιτυχία. Το datapath της προτεινόμενης αρχιτεκτονικής φαίνεται στο **Error! Reference source not found.**.

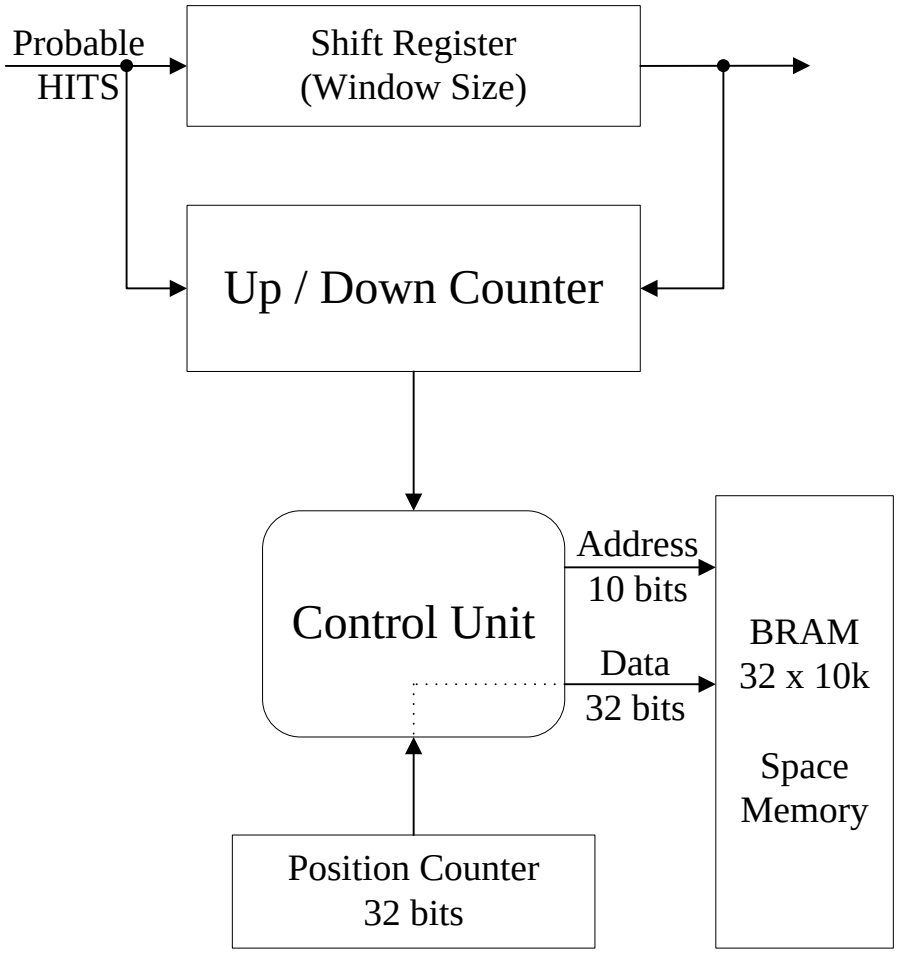


Σχήμα 1-4: Datapath αρχιτεκτονικής TUC PreBLAST

Το δεύτερο κομμάτι του TUC PreBLAST, θα ονομάζεται κομμάτι ελέγχου. Το κομμάτι ελέγχου (1 καταχωρητής ολίσθησης, 2 μετρητές μια μνήμη και μια μονάδα ελέγχου που υλοποιείται από μία μηχανή πεπερασμένων καταστάσεων) έχει την ακόλουθη λειτουργικότητα. Τα αποτελέσματα εισάγονται σε κάθε κύκλο στον καταχωρητή ολίσθησης της μονάδας ελέγχου. Αν εισαχθεί ένα '1', τότε ένας από τους 2 μετρητές, ο μετρητής κατωφλιού, αυξάνεται. Αν ένα '1', ολισθήσει εκτός του καταχωρητή ολίσθησης, τότε ο μετρητής αυτός μειώνεται. Η τελική τιμή του μετρητή είναι ο αριθμός των πιθανών «επιτυχιών» που εντοπίστηκαν στο παράθυρο της βάσης γενετικών δεδομένων που εξετάζεται εκείνες τις στιγμές. Ο άλλος μετρητής, ο μετρητής θέσεις, αυξάνεται κάθε φορά κάθε που εισάγεται στο σύστημα ένας νέος χαρακτήρας από τη βάση γενετικών δεδομένων. Όταν η τιμή του πρώτου μετρητή ξεπεράσει την προκαθορισμένη τιμή του κατωφλιού, τότε η μονάδα ελέγχου γράφει στη μνήμη της την τιμή του καταχωρητή θέσης, σηματοδοτώντας την αρχή της περιοχής υψηλής ομοιότητας μεταξύ βάσης γενετικών δεδομένων και ερωτήματος. Όταν η τιμή του πρώτου μετρητή πέσει κάτω από την τιμή του κατωφλιού, τότε η μονάδα ελέγχου και πάλι γράφει την τιμή του καταχωρητή θέσης στη μνήμη της, σηματοδοτώντας το τέλος της περιοχής υψηλής ομοιότητας μεταξύ βάσης γενετικών δεδομένων και ερωτήματος. Με τον τρόπο αυτό, σηματοδοτούνται όλες οι περιοχές υψηλής ομοιότητας μεταξύ βάσης γενετικών δεδομένων και ερωτήματος.

Το datapath της προτεινόμενης αρχιτεκτονικής φαίνεται στο Σχήμα 1-4, ενώ το control path της φαίνεται στο Σχήμα 1-5.

Γενικά, οι αποδόσεις που μετρήθηκαν από τη λειτουργία του TUC PreBLAST, ήταν αρκετά ικανοποιητικές. Ειδικά αν ληφθεί υπόψη ότι έστω και με μικρή μείωση του τελικού διαστήματος της βάσης γενετικών δεδομένων, το οποίο επεξεργάζεται ο αλγόριθμος BLAST, επιτυγχάνεται εξοικονόμηση χρόνου λόγω μείωσης των απαιτούμενων υπολογισμών.



Σχήμα 1-5: Control Path της αρχιτεκτονικής TUC PreBLAST

1.4 Φίλτρα Bloom

1.4.1 Εισαγωγή

Το φίλτρο Bloom [**29**], το οποίο γεννήθηκε σαν ιδέα από τον Burton H. Bloom το 1970, είναι μια αποδοτική, όσον αφορά το χώρο, πιθανοκρατική δομή δεδομένων που χρησιμοποιείται για να εξακριβωθεί το κατά πόσον ένα στοιχείο είναι μέλος ενός συνόλου. Είναι πιθανό να συμβούν false positives, αλλά αδύνατον να συμβούν false negatives. Στοιχεία μπορούν να προστεθούν στο σύνολο αλλά δε μπορούν να αφαιρεθούν. Όσο περισσότερα στοιχεία εισάγονται στο σύνολο, τόσο περισσότερο μεγαλώνει η πιθανότητα για false positives.

1.4.2 Αλγόριθμος Φίλτρου Bloom

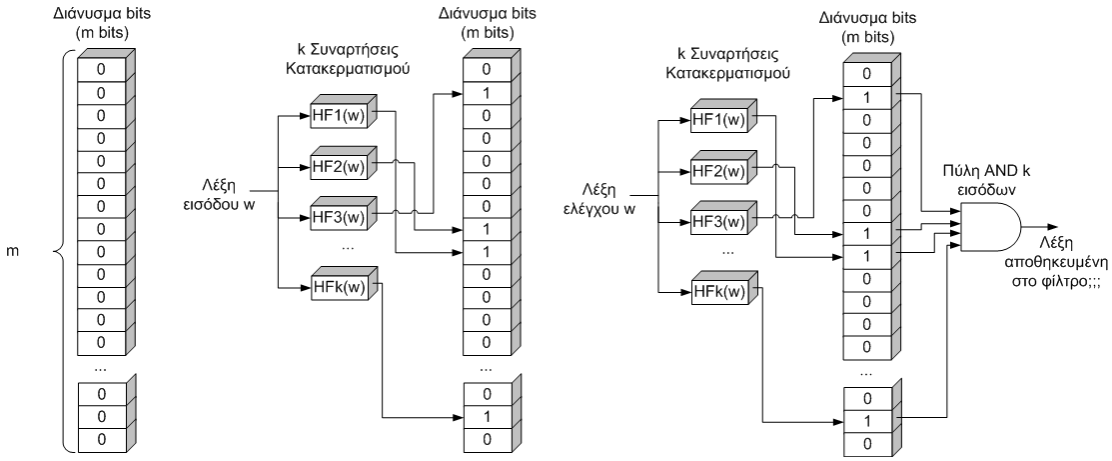
Ένα κενό φίλτρο Bloom είναι ένας μονοδιάστατος πίνακας μεγέθους m bits, όπου όλα του τα στοιχεία είναι αρχικοποιημένα στο 0. Έχουν οριστεί, επίσης, k συναρτήσεις κατακερματισμού κάθε μια από τις οποίες αντιστοιχίζει μια τιμή κλειδί σε μία από τις m θέσεις του πίνακα.

Για να προσθέσουμε ένα νέο στοιχείο στον πίνακα, το περνάμε μέσα και από τις k συναρτήσεις κατακερματισμού ώστε να πάρουμε τις k θέσεις του πίνακα. Τέλος, θέτουμε το bit στις διευθύνσεις που βρήκαμε αμέσως πριν, σε 1.

Για να εξακριβώσουμε το κατά πόσο ένα στοιχείο βρίσκεται στο σύνολο, περνάμε αυτό το στοιχείο από τις k συναρτήσεις κατακερματισμού και παίρνουμε k θέσεις του πίνακα. Αν οποιοδήποτε bit σ' αυτές τις θέσεις είναι 0, τότε το στοιχείο σίγουρα δεν ανήκει στο σύνολο, γιατί αν ανήκε τότε όλα τα bits θα γινόντουσαν 1 κατά τη στιγμή της εισαγωγής του. Αν όλα τα bit σ' αυτές τις θέσεις είναι 1, τότε είτε το στοιχείο ανήκει στο σύνολο είτε τα bits έχουν γίνει 1 κατά την εισαγωγή άλλων στοιχείων.

Αφαίρεση στοιχείου δεν από αυτό το φίλτρο Bloom, δεν υποστηρίζεται. Το στοιχείο αντιστοιχεί σε k bits και αν και το να κάνουμε αυτά τα k bits 0 αρκεί για να το αφαιρέσουμε, αυτό μπορεί να οδηγήσει στην ταυτόχρονη αφαίρεση ακόμα ενός ή περισσοτέρων στοιχείο που αντιστοιχούν και σε αυτό το bit. Σημειωτέων ότι δεν υπάρχει τρόπος να καθορίσουμε αν έχουν εισαχθεί στο σύνολο τέτοια στοιχεία. Μια τέτοια αφαίρεση, θα οδηγούσε σε πιθανότητα false negative, το οποίο δεν επιτρέπεται.

Η απεικόνιση της λειτουργίας ενός φίλτρου Bloom, φαίνεται στο Σχήμα 1-6.



Σχήμα 1-6: Λειτουργία φίλτρου Bloom, αρχικοποίηση (αριστερά), κωδικοποίηση λέξης w (μέση), ελεγχος ύπαρξης λέξης w (δεξιά)

1.4.3 Πλεονεκτήματα ως προς το χώρο και το χρόνο

Παρόλο που το φίλτρο Bloom διακινδυνεύει false positives, έχει σημαντικά χωρικά πλεονεκτήματα απέναντι σε άλλες δομές αναπαράστασης συνόλων. Αντίθετα με τις δομές αυτές, το φίλτρο Bloom με ένα μικρό ποσοστό σφάλματος και ένα βέλτιστο αριθμό συναρτήσεων κατακερματισμού, μπορεί να απαιτήσει μόνο λίγα bits για την αποθήκευση κάθε στοιχείου του συνόλου, ανεξάρτητα από το μέγεθος του κάθε στοιχείου. Επίσης, με την προσθήκη λίγων ακόμα bits για την αποθήκευση του στοιχείου, μπορεί να μειώσει αρκετές φορές το μέγεθός του.

Επίσης, άξιο αναφοράς είναι και το γεγονός ότι ο χρόνος που απαιτείται είτε για την προσθήκη στοιχείου είτε για τον έλεγχο της ύπαρξής του μέσα στο σύνολο, είναι σταθερός και είναι πλήρως ανεξάρτητος από τον αριθμό των στοιχείων που βρίσκονται ήδη στο σύνολο. Μάλιστα σε επίπεδο hardware το φίλτρο Bloom είναι πολύ αποδοτικό χρονικά, μιας και οι k αναζητήσεις είναι ανεξάρτητες και μπορούν να εκτελούνται παράλληλα.

Για την καλύτερη κατανόηση της χωρικής αποτελεσματικότητας του φίλτρου Bloom, θα συγκρίνουμε το γενικό φίλτρο Bloom με την ειδική περίπτωση όπου χρησιμοποιούμε μόνο μια συνάρτηση κατακερματισμού, k=1. Στη δεύτερη περίπτωση, για να κρατήσουμε το ποσοστό των false positives χαμηλό, πρέπει να κρατήσουμε τον αριθμό των bits που τίθενται στο 1, όσο το δυνατόν μικρότερο, κάτι που σημαίνει ότι πρέπει να χρησιμοποιηθεί πίνακας μεγάλου μήκους. Αντιθέτως, το γενικό φίλτρο Bloom, όπου υπάρχουν περισσότερες της μιας συναρτήσεις κατακερματισμού, μπορούμε να έχουμε περισσότερα bits στο 1, διατηρώντας παράλληλα ένα χαμηλό ποσοστό false positives. Διαλέγοντας, όμως, κατάλληλο αριθμό και είδος συναρτήσεων κατακερματισμού αλλά και κατάλληλο μέγεθος πίνακα, μειώνουμε τον αριθμό bits που

θα πάνε στο 1 και η κατανομή τους θα γίνει πιο τυχαία κάτι που βελτιώσει ακόμα περισσότερο την απόδοση του φίλτρου.

1.4.4 Πιθανότητα για false positives

Έστω ότι μια συνάρτηση κατακερματισμού διαλέγει κάθε θέση του πίνακα με ίση πιθανότητα. Η πιθανότητα ένα συγκεκριμένο bit να μην γίνει 1 από κάποια συγκεκριμένη συνάρτηση κατακερματισμού κατά την εισαγωγή ενός στοιχείου είναι

$$1-\frac{1}{m}$$

Η πιθανότητα, το συγκεκριμένο bit να μη γίνει 1 από καμιά από τις συναρτήσεις κατακερματισμού είναι

$$\left(1-\frac{1}{m}\right)^{k}$$

Αν έχουν εισαχθεί n στοιχεία, τότε η πιθανότητα ένα bit να είναι ακόμα στο 0 είναι

$$\left(1-\frac{1}{m}\right)^{kn}$$

Οπότε, η πιθανότητα ένα bit να είναι 1 είναι

$$1-\left(1-\frac{1}{m}\right)^{kn}$$

Κάθε μια από τις k θέσεις του πίνακα που υπολογίζουν οι συναρτήσεις κατακερματισμού έχει γίνει 1 με πιθανότητα ίση με την παραπάνω. Η πιθανότητα, όλα αυτά να είναι 1, κάτι που θα έδειχνε ότι πιθανότατα το στοιχείο που αναζητάτε βρίσκεται στο σύνολο, είναι

$$\left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^{k} \approx 1-e^{-kn/m}{}^{k}$$

Είναι φανερό ότι η πιθανότητα για false positives μειώνεται όσο αυξάνεται το m ενώ αυξάνεται όσο αυξάνεται το n. Για δεδομένα m και n, η τιμή του k που ελαχιστοποιεί την πιθανότητα είναι

$$\frac{m}{n}\ln 2 \approx \frac{9m}{13n} \approx 0.7\frac{m}{n}$$

το οποίο δίνει πιθανότητα για false positive ίση με

$$\left(\frac{1}{2}\right)^{k} \approx 0.6185^{m/n}$$

1.5 Πολυεπεξεργαστικό σύστημα υλοποιημένο σε FPGA (M.PL.EM.)

Το πολυεπεξεργαστικό ενσωματωμένο σύστημα M.PL.EM. [**30**], [**31**], το οποίο έχει υλοποιηθεί με τη χρήση Xilinx Microblaze soft-processors σε μια Xilinx Virtex-II Pro FPGA, μπορεί να χρησιμοποιηθεί σε διάφορες εφαρμογές που μπορούν να εκτελεστούν με παραλληλισμό.

1.5.1 Βασικές Μονάδες του M.PL.EM.

Ένα γενικό block diagram του πολυεπεξεργαστικού συστήματος M.PL.EM. φαίνεται στο Σχήμα 1-7. Η βασική μονάδα του πολυεπεξεργαστικού συστήματος είναι ένας soft-core επεξεργαστής Xilinx MicroBlaze. Ο Xilinx MicroBlaze είναι ένας 32 bit, RISC, soft-core επεξεργαστής, ο οποίος εκμεταλλεύεται συγκεκριμένα πλεονεκτήματα των Xilinx FPGAs. Ο Xilinx MicroBlaze είναι ένας επεξεργαστής Harvard-Architecture ο οποίος περιλαμβάνει και 32 bit Δίαυλο Εντολών (Instruction Bus) αλλά και 32 bit Δίαυλο Δεδομένων (Data Bus). Τρεις τύποι διαύλων υποστηρίζονται: δίαυλος FSL, δίαυλος LMB και δίαυλος OPB.

1.5.2 Υλοποίηση του Πολυεπεξεργαστικού Συστήματος M.PL.EM.

Η αρχιτεκτονική του πολυεπεξεργαστικού συστήματος M.PL.EM. υλοποιήθηκε με τη χρήση των εργαλείων της Xilinx EDK και ISE. Τα τροποποιημένα περιφερειακά αναπτύχθηκαν με τη χρήση της γλώσσας VHDL, ενώ το λογισμικό των επεξεργαστών υλοποιήθηκε με τη χρήση της γλώσσας προγραμματισμού C. Στόχος της υλοποίησης ήταν να χρησιμοποιηθεί η FPGA XC2VP30 της οικογένειας Virtex-II Pro, με speed-grade -7 και η FPGA XC5VLX330T της οικογένειας Virtex5, με speed grade -2.

Το πολυεπεξεργαστικό σύστημα M.PL.EM. κάνει χρήση:

- Μέχρι 80 Xilinx MicroBlaze soft-core επεξεργαστών
- Εξωτερικής μνήμης Cypress SRAM των 4MB
- Μέχρι 14 διαύλους OPB
- Πληθώρας γεφυρών OPB to OPB
- Ως και 84 blocks μνήμης BRAM
- Μιας RS232 περιφερειακού για λόγους I/O

Σύμφωνα με την τοπολογία που επιλέχθηκε, κύριος σκοπός ήταν η δημιουργία ενός πολυεπεξεργαστικού συστήματος που θα αξιοποιούσε εξ' ολοκλήρου τους πόρους που παρέχει μια FPGA. Στην FPGA της οικογένειας Virtex II Pro, χώρεσαν 14 MicroBlaze επεξεργαστές, ενώ σ' αυτήν της οικογένειας Virtex5 χώρεσαν μέχρι 80 MicroBlaze επεξεργαστές.

Κάθε MicroBlaze επεξεργαστής συνδέεται με ένα περιφερειακό FSL, με τη χρήση μιας ζεύξης FSL. Η σύνδεση δε γίνεται με τη χρήση διαύλου OPB, επειδή ο ρυθμός μεταφοράς δεδομένων μεταξύ περιφερειακού και επεξεργαστή είναι πολύ μεγάλος, κάτι το οποίο δε μπορεί να υποστηρίξει ο δίαυλος OPB.

Όσον αφορά τα τροποποιημένα FSL περιφερειακά, περιέχουν μια Dual Port BRAM βάθους 4096 λέξεων και πλάτους 32 bits, η οποία αποθηκεύει τα δεδομένα που έρχονται από την εξωτερική μνήμη και ενός ελεγκτή, ο οποίος καθορίζει την επικοινωνία του περιφερειακού με τον επεξεργαστή και την SRAM.

Ο SRAM Controller υλοποιεί έναν αλγόριθμο Round Robin για την αρχικοποίηση των περιφερειακών FSL με τα δεδομένα. Επίσης, επεξεργάζεται τα δεδομένα με τέτοιο τρόπο ώστε να διευκολύνει την επεξεργασία τους από του MicroBlaze επεξεργαστές.

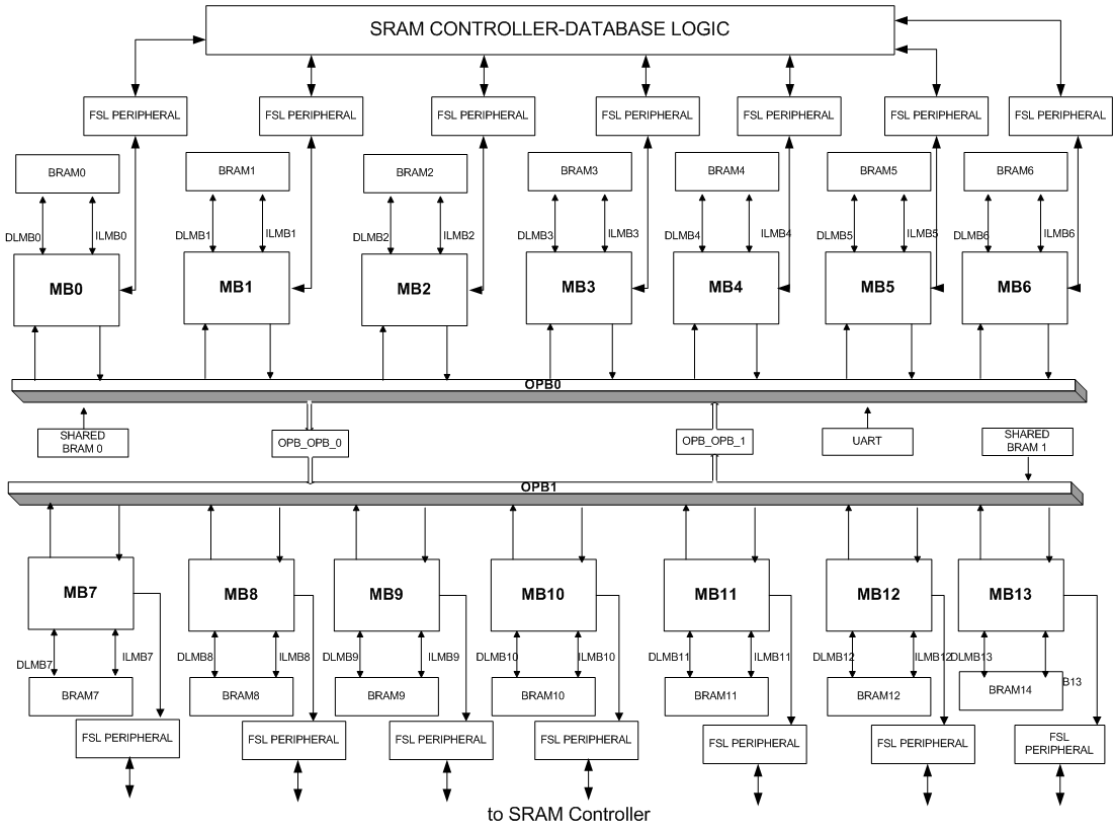
Για την επαλήθευση της ορθής λειτουργίας αλλά και τη μέτρηση της απόδοσης του πολυεπεξεργαστικού συστήματος M.PL.EM. χρησιμοποιήθηκε ο αλγόριθμος BLAST και ειδικότερα η παραλλαγή BLASTn. Σχεδιάστηκε παραλλαγή της υλοποίηση του αλγορίθμου ώστε να ταιριάζει στο πολυεπεξεργαστικό σύστημα και μετρήθηκαν οι αποδόσεις σε σχέση με την υλοποίηση του αλγορίθμου για έναν υπολογιστή γενικού σκοπού.

1.5.3 Λειτουργία του Πολυεπεξεργαστικού Συστήματος M.PL.EM.

Το πολυεπεξεργαστικό σύστημα M.PL.EM. λειτουργεί με τον τρόπο που περιγράφεται ακολούθως. Αρχικά το σύστημα βρίσκεται σε με κατάσταση όπου γίνεται το reset του συστήματος. Μετά από κάποιους κύκλους και μόλις ολοκληρωθεί το reset του συστήματος, κάθε επεξεργαστής αρχίζει να εκτελεί το λογισμικό του, το οποίο είναι ίδιο για όλους τους επεξεργαστές. Ο Ελεγκτής της SRAM (SRAM Controller) αρχίζει να δουλεύει μόνο όταν όλοι οι επεξεργαστές είναι έτοιμοι να δεχτούν δεδομένα από τη μνήμη.

Έπειτα, ο SRAM Controller ξεκινάει την αρχικοποίηση των μνημών όλων των FSL περιφερειακών. Όταν ολοκληρωθεί η αρχικοποίηση της μνήμης του εκάστοτε περιφερειακού FSL, ο επεξεργαστής του ξεκινάει την ανάγνωση των δεδομένων. Μόλις ολοκληρωθεί η επεξεργασία των δεδομένων από τον επεξεργαστή, το περιφερειακό FSL ζητάει νέα δεδομένα από τον SRAM Controller. Δεδομένου ότι οι BRAMs που χρησιμοποιούνται, έχουν δύο πόρτες, την ίδια στιγμή που δέχεται τα δεδομένα η μνήμη του περιφερειακού FSL, γίνεται ταυτόχρονα και επεξεργασία των δεδομένων που είναι ήδη τοποθετημένα στη μνήμη από τον επεξεργαστή. Η παραπάνω διαδικασία επαναλαμβάνεται μέχρις ότου σταλούν όλα τα δεδομένα της μνήμης SRAM στις μνήμες

των περιφερειακών FSL. Με τη μέθοδο αυτή όλοι οι επεξεργαστές δουλεύουν παράλληλα καθώς επεξεργάζονται διαφορετικά δεδομένα την ίδια στιγμή.



Σχήμα 1-7: Block Diagram πολυεπεξεργαστικού συστήματος

1.6 Συνεισφορά της Παρούσας Εργασίας

Η παρούσα εργασία έχει δύο στόχους. Πρώτον να βελτιώσει τα χαρακτηριστικά και την λειτουργία του TUC PreBLAST και δεύτερον να δημιουργήσει ένα ολοκληρωμένο σύστημα επίλυσης του αλγόριθμου BLAST με την ολοκλήρωση της αρχιτεκτονικής με το πολυεπεξεργαστικό σύστημα MPLEM.

Όσο αφορά τον TUC PreBLAST, η συγκεκριμένη εργασία, προσπαθεί να τον κάνει ακόμα πιο αποδοτικό. Ουσιαστικά, ο TUC PreBLAST υστερεί στον τρόπο με τον οποίο αποθηκεύονται τα '1', που προέρχονται από το εκάστοτε ερώτημα, στις μνήμες και στον τρόπο με τον οποίο ελέγχονται έπειτα τα δεδομένα των μνημών σε σύγκριση με την βάση των γενετικών δεδομένων. Ο TUC PreBLAST χρησιμοποιεί μια λογική η οποία μπορεί να θεωρηθεί ότι παράγει τυχαία αποτελέσματα, τεμαχίζοντας με κάποιο «τυχαίο» τρόπο το w-mer και παίρνοντας τις επιμέρους δεκαπεντάδες bits. Αποτυγχάνει, όμως, να εξαφανίσει τη συσχέτιση ανάμεσα στο w-mer και στις επιμέρους δεκαπεντάδες bits.

Αυτόματα προκύπτει η ανάγκη μιας μεθόδου που θα κάνει καλύτερο κατακερματισμό των δεδομένων που προκύπτουν από τα w-mers, στις μνήμες. Τα φίλτρα Bloom, είναι τελικά η μέθοδος που θα χρησιμοποιηθεί στη συγκεκριμένη εργασία ώστε να βελτιωθεί περεταίρω η απόδοση του TUC PreBLAST. Τα φίλτρα Bloom, μάλιστα, θα επιτρέψουν και μια πιο άπληστη προσέγγιση της υλοποίησης του TUC PreBLAST, στην οποία αφενός τα w-mers δε θα σπάνε σε επιμέρους κομμάτια, αφετέρου οι μνήμες που θα χρησιμοποιηθούν θα είναι μικρότερες. Σκοπός αυτών των άπληστων επιλογών, είναι η ενσωμάτωση όσο το δυνατόν περισσότερων μονάδων prefiltering, υλοποιημένων με τη χρήση φίλτρων Bloom, σε ένα σύστημα παράλληλης επεξεργασίας υλοποιημένο σε αναδιατασσόμενη λογική, το οποίο επιλύει τον αλγόριθμο BLAST.

Από την άλλη, το M.PL.EM. είναι ένα σύστημα που επιλύει τον αλγόριθμο BLASTn με τη χρήση αναδιατασσόμενης λογικής. Η επιτάχυνση που προσφέρει είναι ήδη αρκετά μεγάλη σε σχέση με τις υλοποιήσεις σε Λογισμικό. Ωστόσο η επιτάχυνση θα ήταν ακόμα μεγαλύτερη, αν τελικά εφαρμοζόταν ο αλγόριθμος BLASTn μόνο στο κομμάτι της βάσης γενετικών δεδομένων, στο οποίο υπάρχει μεγαλύτερη πιθανότητα να εμφανίζονται λύσεις του αλγορίθμου.

Η συγκεκριμένη εργασία, θα ενσωματώσει μια μονάδα prefiltering υλοποιημένη με φίλτρα Bloom στο M.PL.EM. ώστε να μετρηθεί η τελική επιτάχυνση, η οποία θα προκύψει από τη βελτιστοποίηση αυτή.

Συγκεκριμένα, η συνεισφορά της εργασίας είναι:

- Αλλαγή του λογισμικού που δημιουργήθηκε για το TUC PreBLAST ώστε να γίνει η ενσωμάτωση των φίλτρων Bloom. Το λογισμικό αυτό θα λειτουργήσει και ως μέθοδος πιστοποίησης της ορθής λειτουργίας του φίλτρου που θα υλοποιηθεί σε γλώσσα VHDL.
- Υλοποίηση σε VHDL, μονάδας prefiltering βάσεως γενετικών δεδομένων.
- Ενσωμάτωση της μονάδας prefiltering στον πολυεπεξεργαστή M.PL.EM.
- Αποτίμηση και σύγκριση των αποτελεσμάτων του TUC PreBLAST με τη χρήση Bloom φίλτρων και χωρίς.
- Αποτίμηση και σύγκριση αποτελεσμάτων του M.PL.EM. με τη χρήση Bloom φίλτρων και χωρίς.

2 Μελέτη

2.1 Φίλτρα Bloom στον TUC PreBLAST

2.1.1 Προεπεξεργαστής

Από τη θεωρία πάνω στην οποία βασίζονται τα φίλτρα Bloom προκύπτει ότι με τη χρήση τους, είναι δυνατόν να επιτευχθεί καλής ποιότητας κατακερματισμός, σε συνάρτηση, πάντα, με τις παραμέτρους των φίλτρων. Αν γίνει αλλαγή της μονάδας μνημών του TUC PreBLAST με μια μονάδα η οποία υλοποιεί φίλτρα Bloom, λόγω της καλύτερης ποιότητας κατακερματισμού, υπάρχει πιθανότητα να βελτιωθεί ακόμα περισσότερο η απόδοση του prefiltering.

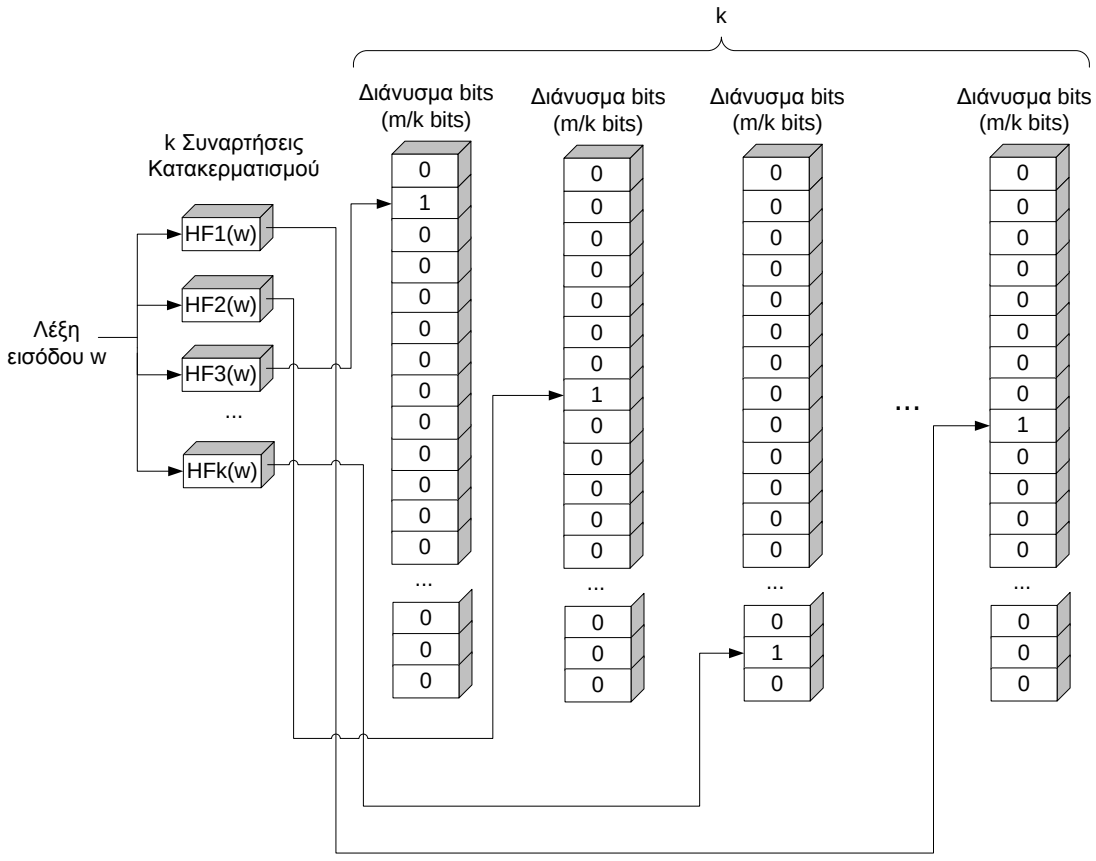
Υλοποιείται ένα φίλτρο Bloom που σε γενικές γραμμές έχει την ακόλουθη λειτουργικότητα. Λαμβάνει σαν είσοδο ολόκληρο το ερώτημα και το τεμαχίζει σε 12άδες χαρακτήρων (2 bits για κάθε χαρακτήρα, επομένως 24 bits), οι οποίες ονομάζονται w-mers. Οι k συναρτήσεις κατακερματισμού που έχουν επιλεγεί και για τις οποίες θα γίνει λόγος αργότερα, μετασχηματίζουν κάθε μία από αυτές τις 24άδες των bits σε k ποσότητες μεγέθους n bits. Ο αριθμός n των bits κάθε ποσότητας, αποτελεί αντικείμενο της μελέτης της παρούσας εργασίας. Οι k n-άδες των bits θα αποτελέσουν τις διευθύνσεις εκείνες, στις οποίες θα τοποθετηθεί το "1" στις k μνήμες της μονάδας φίλτρων Bloom.

Αφού γίνει η αρχικοποίηση του φίλτρου με τα w-mers, θα γίνει έλεγχος της βάσης γενετικών δεδομένων για αντιστοιχίες με το ερώτημα. Κατ' αντιστοιχία, η βάση γενετικών δεδομένων θα τεμαχιστεί σε 12άδες χαρακτήρων (24 bits). Αυτές οι 24άδες των bits θα μετασχηματιστούν από τις k συναρτήσεις κατακερματισμού που έχουν επιλεγεί, σε κ n-άδες bits. Αυτές οι k n-άδες θα αποτελέσουν τις διευθύνσεις εκείνες, στις οποίες θα γίνει έλεγχος για το κατά πόσο υπάρχει το "1" στις k μνήμες της μονάδας φίλτρων Bloom. Οι «απαντήσεις» των μνημών θα περάσουν από μια πύλη AND και εφόσον όλες είναι θετικές, δηλαδή "1", τότε το συγκεκριμένο w-mer της βάσης γενετικών δεδομένων, πιθανότατα αποτελεί και w-mer του ερωτήματος.

Τελικά, η έξοδος του προεπεξεργαστή θα είναι ένα σύνολο αριθμών, που θα υποδηλώνουν τις θέσεις της βάσης γενετικών δεδομένων στις οποίες εντοπίστηκαν αντιστοιχίες με το ερώτημα. Τέλος, με την εφαρμογή παραθύρου και threshold, ο προεπεξεργαστής καταλήγει σε ένα μικρότερο κομμάτι της βάση γενετικών δεδομένων, το οποίο όμως περιέχει όλες τις ακριβείς αντιστοιχίες με το ερώτημα

συμπεριλαμβανομένων και κάποιων εσφαλμένων αντιστοιχιών, οι οποίες όμως είναι πολύ λιγότερες σε αριθμό από αυτές που προκύπτουν από τον TUC PreBLAST.

Για τη συγκεκριμένη τροποποίηση, δε χρησιμοποιήθηκε το φίλτρο Bloom που παρουσιάζεται στο Σχήμα 1-6, αλλά μια παραλλαγή αυτού, η οποία ονομάζεται διαμερισμένο φίλτρο Bloom. Η διαφορά του από το κανονικό είναι ότι κάθε συνάρτηση κατακερματισμού οδηγεί σε μία ξεχωριστή μνήμη. Το διαμερισμένο φίλτρο Bloom χρησιμοποιεί μνήμες μικρότερου μεγέθους, ωστόσο έχει την ίδια πιθανότητα false-positive με αυτό που παρουσιάζεται στο Σχήμα 1-6. Αναλυτικότερα, το διαμερισμένο φίλτρο Bloom φαίνεται στο Σχήμα 2-1.



Σχήμα 2-1: Διαμερισμένο φίλτρο Bloom

2.1.2 Συναρτήσεις κατακερματισμού

Η επιλογή των συναρτήσεων κατακερματισμού θα μπορούσε να γίνει αυθαίρετα. Πιθανότατα οποιοδήποτε είδος τέτοιας συνάρτησης θα μπορούσε να εξυπηρετήσει το σκοπό.

Για το συγκεκριμένο πείραμα υπήρξαν 2 κύριες επιλογές όσον αφορά τις συναρτήσεις κατακερματισμού. Η πρώτη βασίζεται σε μια τεχνική κατακερματισμού που υλοποιείται με κάποιους πίνακες κατακερματισμού και κάποιες απλές συναρτήσεις

μετατροπής. Η δεύτερη βασίζεται στη θεωρία των πρώτων πολυωνύμων. Ακολουθεί σύντομη περιγραφή των δύο τεχνικών.

Τελικά, προτιμήθηκε η μέθοδος που χρησιμοποιεί τις ιδιότητες των πρώτων πολυωνύμων, λόγω του ότι αυτή ήταν ήδη μια μέθοδος που έχει πιστοποιηθεί για το σκοπό αυτό.

2.1.2.1 Πίνακες Κατακερματισμού και Συναρτήσεις Μετατροπής

Θεωρούμε μια συνάρτηση κατακερματισμού που θα έχει τη μορφή που φαίνεται παρακάτω

$$h_i\ X\ = d_{i1} \cdot x_1 \oplus d_{i2} \cdot x_2 \oplus ... \oplus d_{ib} \cdot x_b$$

όπου $X = x_1, x_2, ..., x_b$, να είναι τα δεδομένα που θα προωθηθούν στη συνάρτηση για κατακερματισμό. Το σύνολο των $d_{i1}, d_{i2}, ..., d_{ib}$, ο πίνακας κατακερματισμού, είναι ένα σύνολο από τυχαίες τιμές τόσων bits όσα θέλουμε να είναι κα τα bit στα οποία θέλουμε η συνάρτηση κατακερματισμού να μειώσει την αρχική μας πληροφορία (εδώ από 24 bits σε 14, 15 και 16).

Πιο απλά, αυτή η συνάρτηση κατακερματισμού για κάθε bit του X που είναι 1, γίνεται XOR με το τρέχον αποτέλεσμα η τυχαία τιμή που σχετίζεται με το bit αυτό.

2.1.2.2 Πρώτα Πολυώνυμα

Ένα πρώτο πολυώνυμο είναι το ελάχιστο πολυώνυμο ενός αρχικού στοιχείου της επέκτασης πεδίου $GF\ p^m$. Με άλλα λόγια, ένα πολυώνυμο $F\ X$ με συντελεστές στο $GF\ p = Z/pZ$, είναι πρώτο πολυώνυμο αν έχει μια ρίζα a στο $GF\ p^m$ τέτοια ώστε $0, 1, a, a^2, ..., a^{p^{m-2}}$ να είναι ολόκληρο το πεδίο $GF\ p^m$ και επιπλέον $F\ X$ να είναι το πολυώνυμο μικρότερου βαθμού που έχει το a ως ρίζα.

Επειδή όλα τα ελάχιστα πολυώνυμα δεν αναλύονται σε πρωτοβάθμιους παράγοντες, τότε και τα πρώτα πολυώνυμα είναι δεν αναλύονται επίσης. Ένα πρώτο πολυώνυμο πρέπει να έχει ένα μη μηδενικό σταθερό όρο, ειδάλλως θα ήταν δυνατό να διαιρεθεί με το x. Επίσης, όλα τα πρώτα πολυώνυμα έχουν μονό αριθμό όρων, γιατί αλλιώς θα μπορούσαν να διαιρεθούν από το x+1. Τέλος, οι ρίζες ενός πρώτου πολυωνύμου έχουν βαθμό $p^m - 1$.

Τα πρώτα πολυώνυμα έχουν τις εξής χρήσεις:

- Στην αναπαράσταση των στοιχείων ενός πεπερασμένου πεδίου. Αν a στο $GF\ p^m$ είναι μια ρίζα ενός πρώτου πολυωνύμου $F\ X$, τότε από τη στιγμή που ο βαθμός της ρίζας a είναι $p^m - 1$, αυτό σημαίνει ότι όλα τα στοιχεία του $GF\ p^m$, μπορούν να αναπαρασταθούν ως διαδοχικές δυνάμεις της a:

$$GF\ p^m = 0, 1, a, a^2, \ldots, a^{p^{m-2}}$$

  Όταν όλα αυτά τα στοιχεία μειωθούν με το υπόλοιπο της διαίρεσης με το $F\ X$, παρέχουν την πολυωνυμική βάση αναπαράστασης όλων των στοιχείων στο πεδίο.
- Δημιουργία τυχαίου bit. Τα πρώτα πολυώνυμα ορίζουν μια επαναληπτική σχέση, η οποία μπορεί να χρησιμοποιηθεί για να δημιουργήσει ψευδοτυχαία bits. Ουσιαστικά, κάθε καταχωρητής ολίσθησης γραμμικής ανάδρασης με μέγιστο κύκλο (ο οποίος είναι $2^{lfsr\ length} - 1$) σχετίζεται με ένα πρώτο πολυώνυμο.

Στην περίπτωσή μας, αξιοποιείται η δεύτερη χρήση των πρώτων πολυωνύμων.

Η έρευνα που έγινε, οδηγήθηκε με βασικό στόχο την επίτευξη όσο το δυνατόν μεγαλύτερου prefiltering της βάσης με τη μικρότερη δυνατή ποσότητα μνήμης.

Οι μετρήσεις έγιναν με βάση τα configurations που φαίνονται στον ακόλουθο πίνακα

- 2 συναρτήσεις κατακερματισμού, 2 μνήμες 14 bit διευθύνσεων (2^{14} διευθύνσεις)
- 4 συναρτήσεις κατακερματισμού, 4 μνήμες, 14 bit διευθύνσεων (2^{14} διευθύνσεις)
- 8 συναρτήσεις κατακερματισμού, 8 μνήμες, 16 bit διευθύνσεων (2^{16} διευθύνσεις)
- 16 συναρτήσεις κατακερματισμού, 16 μνήμες, 15 bit διευθύνσεων (2^{15} διευθύνσεις)

Η επιλογή των παραπάνω configurations προκύπτουν από τη διαθέσιμη ποσότητα μνήμης στις FPGAs Virtex II Pro και Virtex 5.

Για τα παραπάνω configuration μνήμης, έγιναν πειράματα για ολόκληρο αλλά και τεμαχισμένο ερώτημα σε διαφορετικά μεγέθη.

Για τις μετρήσεις χρησιμοποιήθηκε μια «δύσκολη» περίπτωση, το Chr13, το οποίο δεν ανταποκρίνεται τόσο καλά, σε αντίθεση με άλλα χρωμοσώματα, στη μέθοδο prefiltering που χρησιμοποιήθηκε.

2.2 Ολόκληρο Ερώτημα

2.2.1 2 συναρτήσεις κατακερματισμού, 2 μνήμες 14 bit διευθύνσεων

Σ' αυτό το configuration, χρησιμοποιούνται 2 συναρτήσεις κατακερματισμού, οι οποίες οδηγούν σε διευθύνσεις 2 μνημών με 14 bit διευθύνσεων, δηλαδή βάθους 2^{14} η κάθε μια. Το ισοδύναμο αυτής της μνήμης σε BRAMs είναι 1 BRAM.

Σύμφωνα με το παραπάνω configuration, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-1 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-2.

Όπως φαίνεται από τα στοιχεία του πίνακα, το τελικό ποσοστό της, προς επεξεργασία, βάσης παραμένει στο 100% για όλες τις τιμές του παραθύρου.

2.2.2 4 συναρτήσεις κατακερματισμού, 4 μνήμες 14 bit διευθύνσεων

Σ' αυτό το configuration, χρησιμοποιούνται 4 συναρτήσεις κατακερματισμού, οι οποίες οδηγούν σε διευθύνσεις 2 μνημών με 14 bit διευθύνσεων, δηλαδή βάθους 2^{14} η κάθε μια. Το ισοδύναμο αυτής της μνήμης σε BRAMs είναι 2 BRAM.

Σύμφωνα με το παραπάνω configuration, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-3 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-2.

Όπως φαίνεται από τα στοιχεία του πίνακα, το τελικό ποσοστό της, προς επεξεργασία, βάσης παραμένει στο 100% για όλες τις τιμές του παραθύρου.

2.2.3 8 συναρτήσεις κατακερματισμού, 8 μνήμες 16 bit διευθύνσεων

Σ' αυτό το configuration, χρησιμοποιούνται 8 συναρτήσεις κατακερματισμού, οι οποίες οδηγούν σε διευθύνσεις 8 μνημών με 16 bit διευθύνσεων, δηλαδή βάθους 2^{16} η κάθε μια. Το ισοδύναμο αυτής της μνήμης σε BRAMs είναι 16 BRAM.

Σύμφωνα με το παραπάνω configuration, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-4 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-2.

Όπως φαίνεται από τα στοιχεία του πίνακα, το τελικό ποσοστό της, προς επεξεργασία, βάσης παραμένει στο 100% για όλες τις τιμές του παραθύρου.

2.2.4 16 συναρτήσεις κατακερματισμού, 16 μνήμες 14 bit διευθύνσεων

Σ' αυτό το configuration, χρησιμοποιούνται 16 συναρτήσεις κατακερματισμού, οι οποίες οδηγούν σε διευθύνσεις 16 μνημών με 14 bit διευθύνσεων, δηλαδή βάθους 2^{14} η κάθε μια. Το ισοδύναμο αυτής της μνήμης σε BRAMs είναι 8 BRAM.

Σύμφωνα με το παραπάνω configuration, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-5 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-2.

Όπως φαίνεται από τα στοιχεία του πίνακα, το τελικό ποσοστό της, προς επεξεργασία, βάσης παραμένει στο 100% για όλες τις τιμές του παραθύρου.

2.2.5 Συμπεράσματα

Όπως φαίνεται, για όλα τα configurations, το τελικό space της βάσης είναι όσο θα έπρεπε να είναι, δεδομένου του διαστήματος που δίνουν τα true positive hits, ωστόσο παραμένει αρκετά μεγάλο, κάτι το οποίο δεν είναι επιθυμητό.

2.3 Τεμαχισμένο Ερώτημα

Παρόλο που τα μεγάλα ερωτήματα είναι επιθυμητά, παρατηρούμε ότι η μέθοδος prefiltering που χρησιμοποιείται δεν μπορεί να τα χειριστεί κατάλληλα.

Η ιδέα που προκύπτει είναι να γίνει «σπάσιμο» του ερωτήματος σε μικρότερα κομμάτια και παράλληλη επεξεργασία των επιμέρους κομματιών.

Δεδομένου των παραπάνω configurations, υλοποιούνται εκ νέου τα πειράματα στα οποία, πλέον, το ερώτημα είναι σπασμένο σε κομμάτια των 500 και 250 χαρακτήρων. Για κάποια configurations έχουν μελετηθεί και ερωτήματα διαφορετικού μεγέθους όπως 2000 και 1000 χαρακτήρες.

Στα πειράματα αυτά δε μετριούνται τα πιθανά hits ούτε το ποσοστό επιτυχίας hit, λόγω των επικαλύψεων που λαμβάνονται υπόψη. Για τον υπολογισμό του τελικού διαστήματος, όμως, οι επικαλύψεις αυτές αφαιρούνται με τη χρήση των κατάλληλων εργαλείων.

2.3.1 2 συναρτήσεις κατακερματισμού, 2 μνήμες 14 bit διευθύνσεων

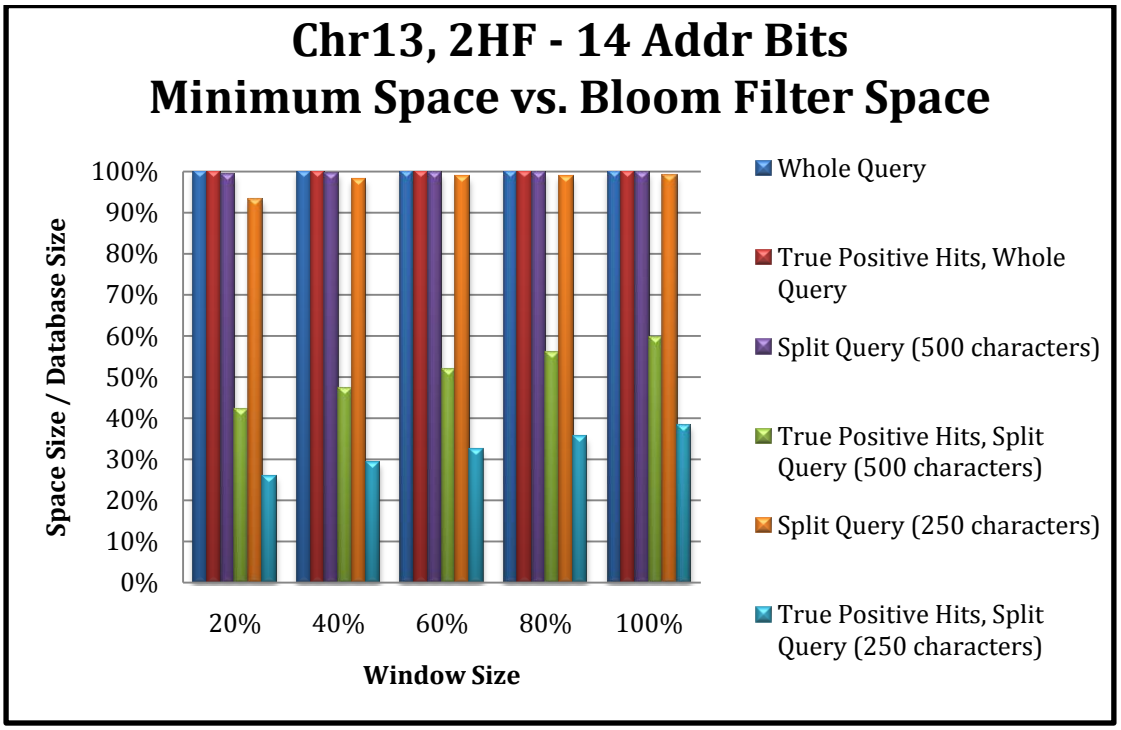
Σ' αυτό το configuration, χρησιμοποιούνται 2 συναρτήσεις κατακερματισμού, οι οποίες οδηγούν σε διευθύνσεις 2 μνημών με 14 bit διευθύνσεων, δηλαδή βάθους 2^{14} η κάθε μια. Το ισοδύναμο αυτής της μνήμης σε BRAMs είναι 1 BRAM.

Σύμφωνα με το παραπάνω configuration, για το σπασμένο σε 500 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-6 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-7.

Ομοίως, για σπασμένο, σε 250 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-8 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-9.

Όπως φαίνεται από τους πίνακες, τα πρώτα δεδομένα είναι ενθαρρυντικά, μιας και με την ελάχιστη ποσότητα μνήμης και με την τεχνική του σπασίματος του ερωτήματος, γίνεται δυνατή η, έστω και μικρή, μείωση του τελικού διαστήματος.

Ενδεικτικές είναι και οι γραφικές αναπαραστάσεις των δεδομένων που προέκυψαν και φαίνονται στο Σχήμα 2-2, οι οποίες επιβεβαιώνουν αυτό που αναφέρθηκε ακριβώς παραπάνω.



Σχήμα 2-2: Chr13, 2HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space

Στο Σχήμα 2-2 φαίνονται τα αποτελέσματα του φίλτρου που υλοποιήθηκε όσον αφορά τα τελικά διαστήματα, σε σχέση με τα βέλτιστα αποτελέσματα, τα οποία θα επιτυγχάνονταν αν είχαμε μόνο true positive hits χρησιμοποιώντας και πάλι τη μέθοδο του σπασμένου ερωτήματος.

2.3.2 4 συναρτήσεις κατακερματισμού, 4 μνήμες 14 bit διευθύνσεων

Σ' αυτό το configuration, χρησιμοποιούνται 4 συναρτήσεις κατακερματισμού, οι οποίες οδηγούν σε διευθύνσεις 4 μνημών με 14 bit διευθύνσεων, δηλαδή βάθους 2^{14} η κάθε μια. Το ισοδύναμο αυτής της μνήμης σε BRAMs είναι 2 BRAM.

Σύμφωνα με το παραπάνω configuration, για το σπασμένο σε 1000 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-10 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-11.

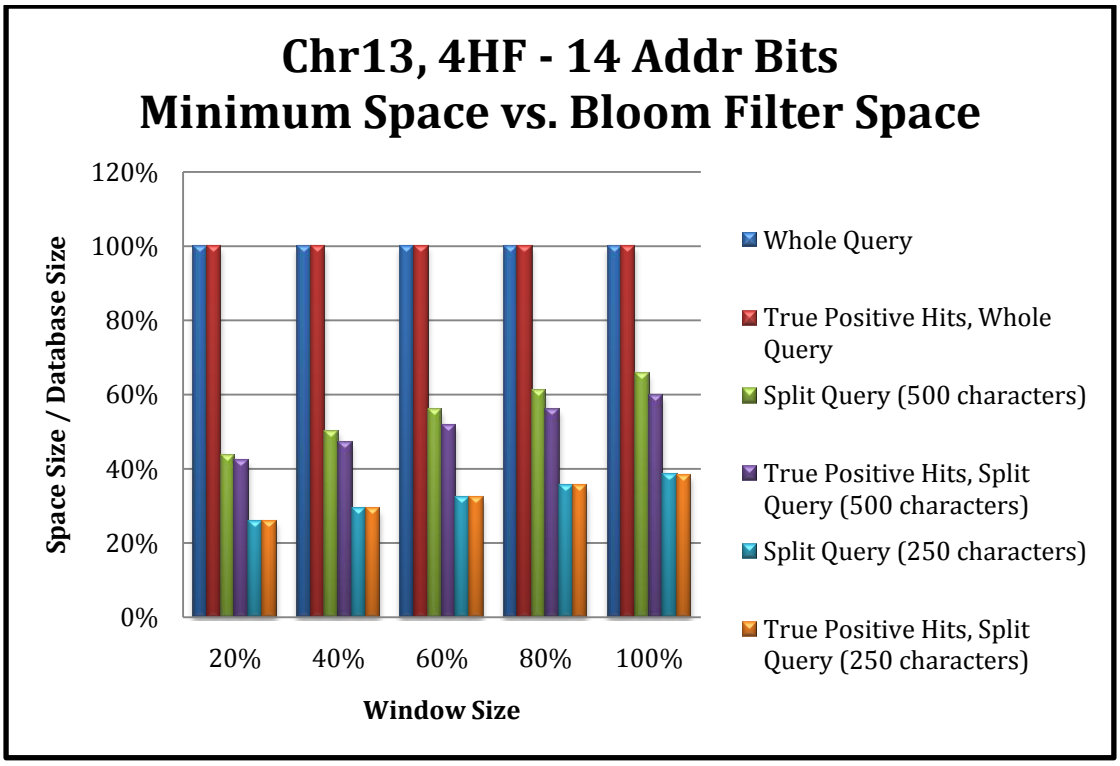
Ομοίως, για σπασμένο, σε 500 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-12 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το

τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-13.

Ομοίως, για σπασμένο, σε 250 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-14 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-15.

Όπως φαίνεται από τους πίνακες, τα δεδομένα είναι απόλυτα ικανοποιητικά, μιας και προσθέτοντας στο προηγούμενο configuration μία ακόμα BRAM και συνεχίζοντας τη χρήση της τεχνικής του σπασίματος του ερωτήματος, γίνεται δυνατό να πλησιαστεί με πολύ μικρή απόκλιση το βέλτιστο διάστημα – το οποίο επιτυγχάνεται αν χρησιμοποιήσουμε την τεχνική του σπασίματος του ερωτήματος και μόνο true positive hits - από αυτό που δίνει το φίλτρο που υλοποιήθηκε.

Ενδεικτικές είναι και οι γραφικές αναπαραστάσεις των δεδομένων που προέκυψαν και φαίνονται στο Σχήμα 2-3, οι οποίες επιβεβαιώνουν αυτό που αναφέρθηκε ακριβώς παραπάνω.



Σχήμα 2-3: Chr13, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space

Στο Σχήμα 2-3 φαίνονται τα αποτελέσματα του φίλτρου που υλοποιήθηκε όσον αφορά τα τελικά διαστήματα, σε σχέση με τα βέλτιστα αποτελέσματα, τα οποία θα επιτυγχάνονταν αν είχαμε μόνο true positive hits χρησιμοποιώντας και πάλι τη μέθοδο του σπασμένου ερωτήματος.

2.3.3 8 συναρτήσεις κατακερματισμού, 8 μνήμες 16 bit διευθύνσεων

Σ' αυτό το configuration, χρησιμοποιούνται 8 συναρτήσεις κατακερματισμού, οι οποίες οδηγούν σε διευθύνσεις 8 μνημών με 16 bit διευθύνσεων, δηλαδή βάθους 2^{16} η κάθε μια. Το ισοδύναμο αυτής της μνήμης σε BRAMs είναι 16 BRAM.

Σύμφωνα με το παραπάνω configuration, για το σπασμένο σε 2000 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-16 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-17.

Ομοίως, για σπασμένο, σε 1000 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-18 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-19.

Ομοίως, για σπασμένο, σε 500 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-20 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-21.

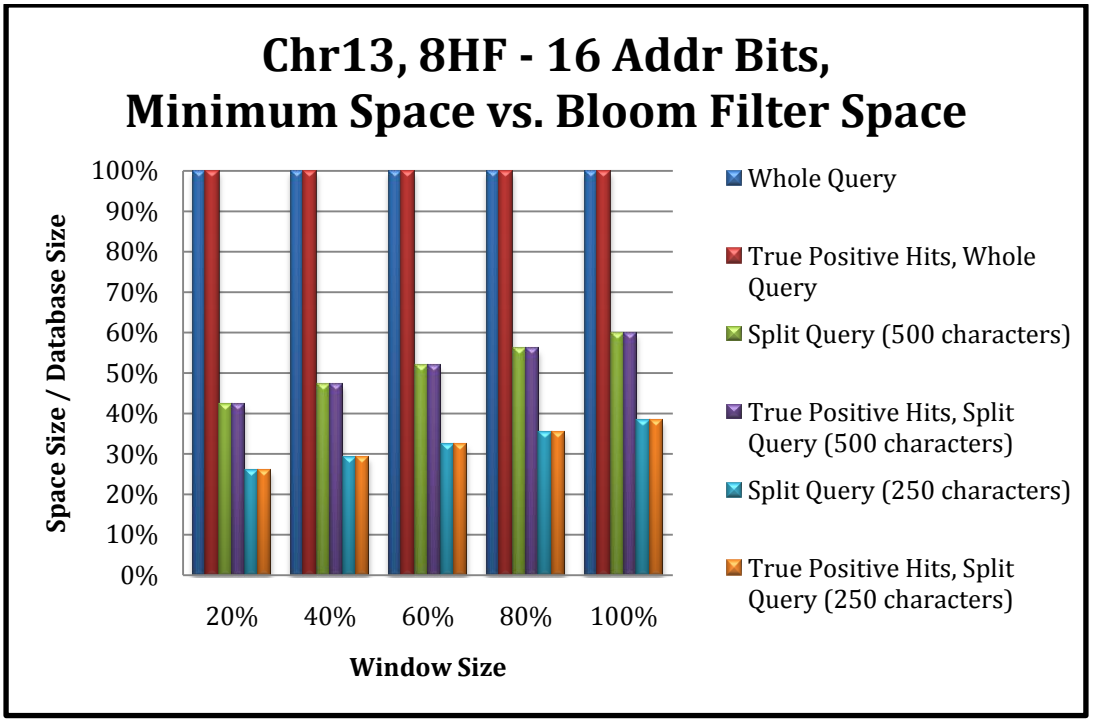
Ομοίως, για σπασμένο, σε 250 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-22 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-23.

Όπως φαίνεται από τους πίνακες, τα δεδομένα είναι, ξανά, απόλυτα ικανοποιητικά, κάτι το οποίο δεν ήταν παρά αναμενόμενο. Η αύξηση της μνήμης ήταν αρκετά μεγάλη κάτι το οποίο οδήγησε στο να επιτευχθεί απόλυτη ταύτιση των διαστημάτων που εξάγει το φίλτρο που υλοποιήθηκε και εκείνων τα οποία θεωρούνται

βέλτιστα και εξάγονται με το να χρησιμοποιήσουμε την τεχνική του σπασίματος του ερωτήματος και μόνο true positive hits.

Το κόστος σε μνήμη είναι πολύ μεγάλο σε σχέση το κέρδος σε μείωση του διαστήματος που τελικά αποκομίζεται. Αυτό κάνει το συγκεκριμένο configuration όχι και τόσο αποδεκτό.

Ενδεικτικές είναι και οι γραφικές αναπαραστάσεις των δεδομένων που προέκυψαν και φαίνονται στο Σχήμα 2-4, οι οποίες επιβεβαιώνουν αυτό που αναφέρθηκε ακριβώς παραπάνω.



Σχήμα 2-4: Chr13, 8HF - 16 Addr Bits, Minimum Space vs. Bloom Filter Space

Στο Σχήμα 2-4 φαίνονται τα αποτελέσματα του φίλτρου που υλοποιήθηκε όσον αφορά τα τελικά διαστήματα, σε σχέση με τα βέλτιστα αποτελέσματα, τα οποία θα επιτυγχάνονταν αν είχαμε μόνο true positive hits χρησιμοποιώντας και πάλι τη μέθοδο του σπασμένου ερωτήματος.

2.3.4 16 συναρτήσεις κατακερματισμού, 16 μνήμες 14 bit διευθύνσεων

Σ' αυτό το configuration, χρησιμοποιούνται 16 συναρτήσεις κατακερματισμού, οι οποίες οδηγούν σε διευθύνσεις 16 μνημών με 14 bit διευθύνσεων, δηλαδή βάθους 2^{14} η κάθε μια. Το ισοδύναμο αυτής της μνήμης σε BRAMs είναι 8 BRAM.

Σύμφωνα με το παραπάνω configuration, για το σπασμένο σε 1000 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-24 σε αντιπαραβολή

με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-25.

Ομοίως, για σπασμένο, σε 500 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-26 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-27.

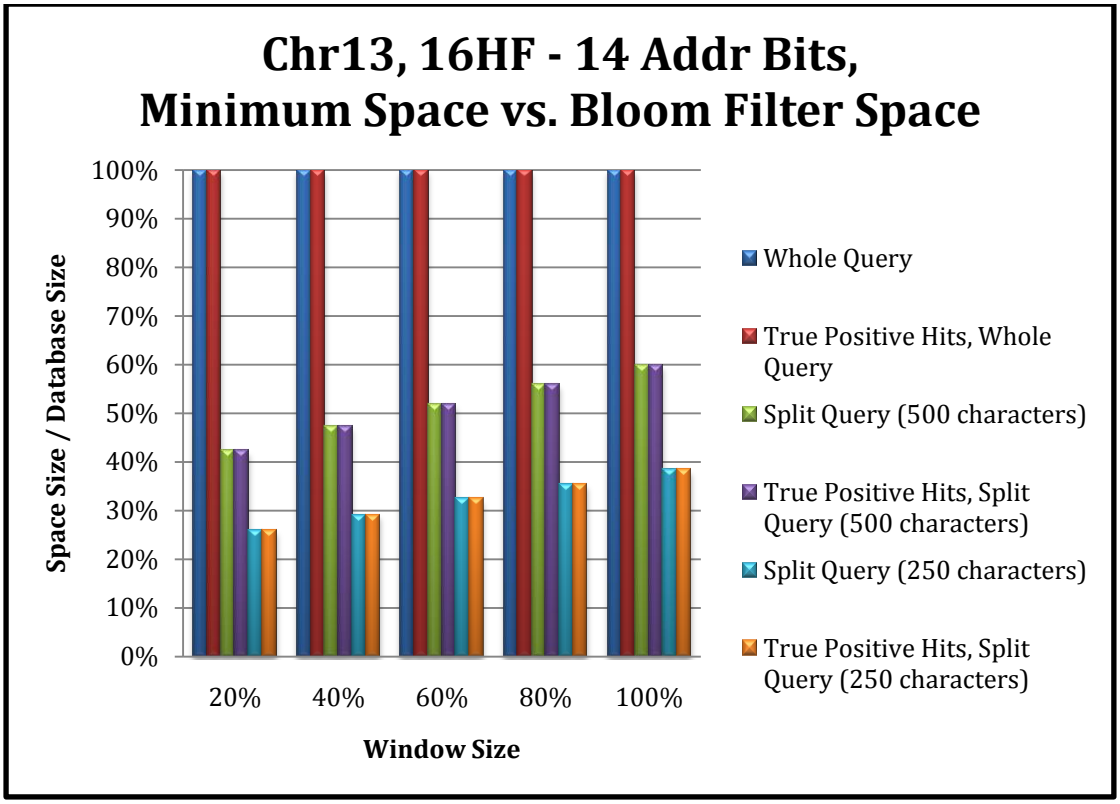
Ομοίως, για σπασμένο, σε 250 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-28 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-29.

Όπως φαίνεται από τους πίνακες, τα δεδομένα είναι, ξανά, απόλυτα ικανοποιητικά, κάτι το οποίο δεν ήταν παρά αναμενόμενο. Η αύξηση της μνήμης ήταν αρκετά μεγάλη κάτι το οποίο οδήγησε στο να επιτευχθεί απόλυτη ταύτιση των διαστημάτων που εξάγει το φίλτρο που υλοποιήθηκε και εκείνων τα οποία θεωρούνται βέλτιστα και εξάγονται με το να χρησιμοποιήσουμε την τεχνική του σπασίματος του ερωτήματος και μόνο true positive hits.

Το κόστος σε μνήμη είναι πολύ μεγάλο σε σχέση το κέρδος σε μείωση του διαστήματος που τελικά αποκομίζεται. Αυτό κάνει και το συγκεκριμένο configuration όχι και τόσο αποδεκτό.

Ενδεικτικές είναι και οι γραφικές αναπαραστάσεις των δεδομένων που προέκυψαν και φαίνονται στο Σχήμα 2-5, οι οποίες επιβεβαιώνουν αυτό που αναφέρθηκε ακριβώς παραπάνω.

Στο Σχήμα 2-5Σχήμα 2-4 φαίνονται τα αποτελέσματα του φίλτρου που υλοποιήθηκε όσον αφορά τα τελικά διαστήματα, σε σχέση με τα βέλτιστα αποτελέσματα, τα οποία θα επιτυγχάνονταν αν είχαμε μόνο true positive hits χρησιμοποιώντας και πάλι τη μέθοδο του σπασμένου ερωτήματος.



Σχήμα 2-5: Chr13, 16HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space

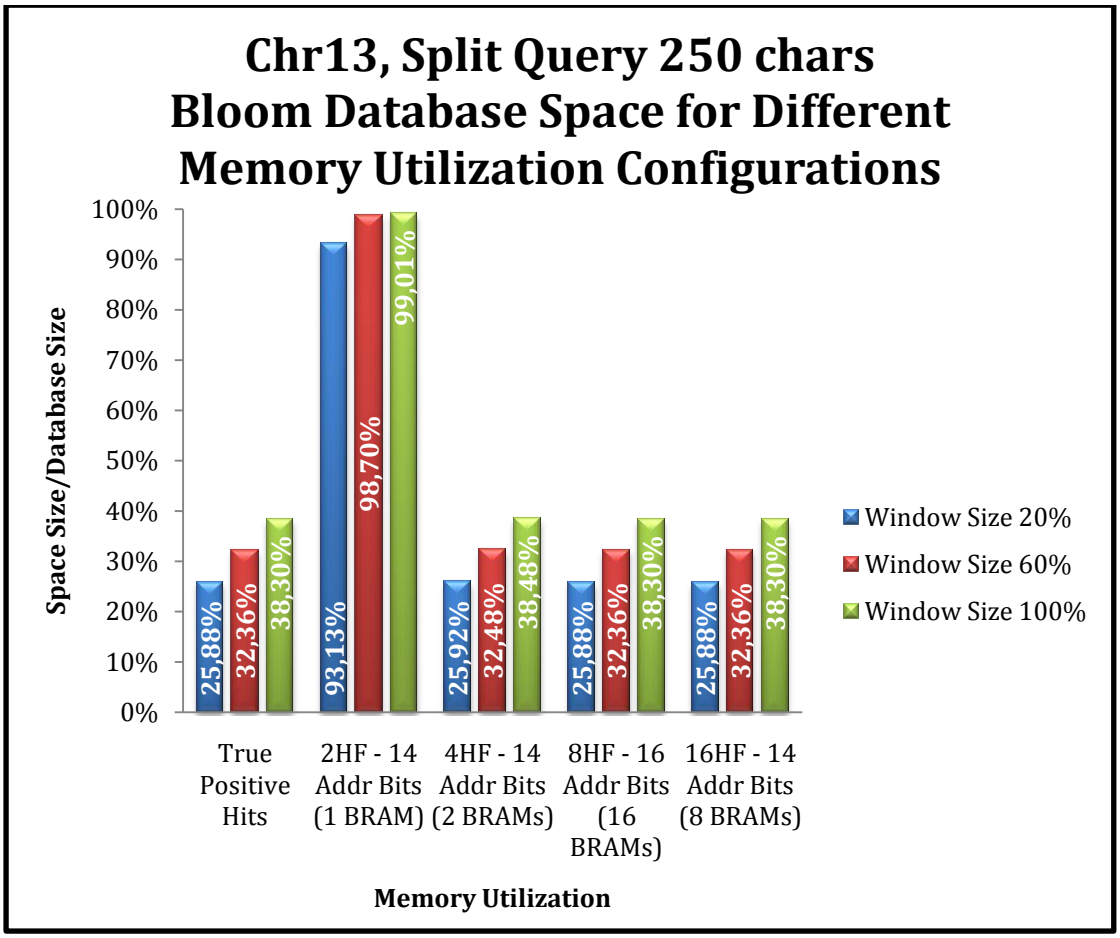
2.3.5 Συμπεράσματα

Όπως φαίνεται από τους πίνακες και τις γραφικές, όσο πέφτει το μέγεθος του ερωτήματος, για τα περισσότερα configurations, το τελικό space της βάσης πέφτει και μάλιστα πλησιάζει πάρα πολύ ικανοποιητικά το ελάχιστο space, το οποίο προκύπτει εφαρμόζοντας τη διαδικασία του σπασίματος και για τα true positive hits μόνο.

Άμεση συνεπαγωγή του παραπάνω, είναι ότι εφόσον η μέθοδος του σπασίματος είναι αποδοτική όσον αφορά το prefiltering, επόμενη παράμετρος ώστε να επιλέξουμε το καλύτερο configuration είναι με τι ποσότητα μνήμης πετυχαίνουμε τα καλύτερα αποτελέσματα.

Τα configurations μας απαιτούν 1 BRAM, 2 BRAMs, 8 BRAMs και 16 BRAMs. Από τους πίνακες είναι εύκολο να διακρίνει κανείς ότι για 8 και 16 BRAMs τα αποτελέσματα είναι ακριβώς τα ίδια και μάλιστα το space είναι το ελάχιστο δυνατό δεδομένου του space που δίνουν τα true positive hits. Επομένως, εφόσον έπρεπε να χρησιμοποιηθεί ένα από τα δύο configurations, το πιο λογικό θα ήταν να γίνει χρήση αυτού που καταλαμβάνει 8 BRAMs. Αν γίνει, όμως, πιο προσεκτική παρατήρηση των πινάκων, είναι εύκολο να διαπιστώσει κανείς ότι με το 1/4 την μνήμης αυτής (για το configuration με τα 2 BRAMs) επιτυγχάνονται spaces που διαφέρουν κατά ελάχιστο (της τάξης του 0,5%-1,5%) από τα optimal spaces. Το configuration του 1 BRAM δεν

αποδίδει τα προσδοκώμενα αποτελέσματα με μέγεθος ερωτήματος μεγαλύτερο ή ίσο των 250 χαρακτήρων. Τα αποτελέσματα θα ήταν καλύτερα αν μειωνόταν ακόμα περισσότερο το μέγεθος του ερωτήματος. Το Σχήμα 2-6 κάνει την επιλογή του κατάλληλου configuration μνήμης, ακόμα πιο ξεκάθαρο.



Σχήμα 2-6: Bloom Database Space for Different Memory Utilization Configurations

2.4 4 Συναρτήσεις Κατακερματισμού, 4 Μνήμες 14 bit Διευθύνσεων

Φαίνεται να είναι αποδοτικότερη η χρήση της διπλάσιας μνήμης (2 BRAMs), γι' αυτό επιλέχθηκε το συγκεκριμένο configuration για να γίνει μια σειρά πιο εκτεταμένων πειραμάτων σε διάφορες βάσεις χρωμοσωμάτων και σε διάφορα ερωτήματα.

Στη συνέχεια παρουσιάζονται κάποιοι πίνακες και κάποιες γραφικές παραστάσεις, ίδιου τύπου με τα παραπάνω για κάθε χρωμόσωμα που μελετήθηκε.

2.4.1 Χρωμόσωμα 12

Σύμφωνα με το παραπάνω βέλτιστο configuration, από τον Πίνακας 6-30 ως τον Πίνακας 6-35 φαίνονται τα πειραματικά δεδομένα που προκύπτουν για το χρωμόσωμα 12.

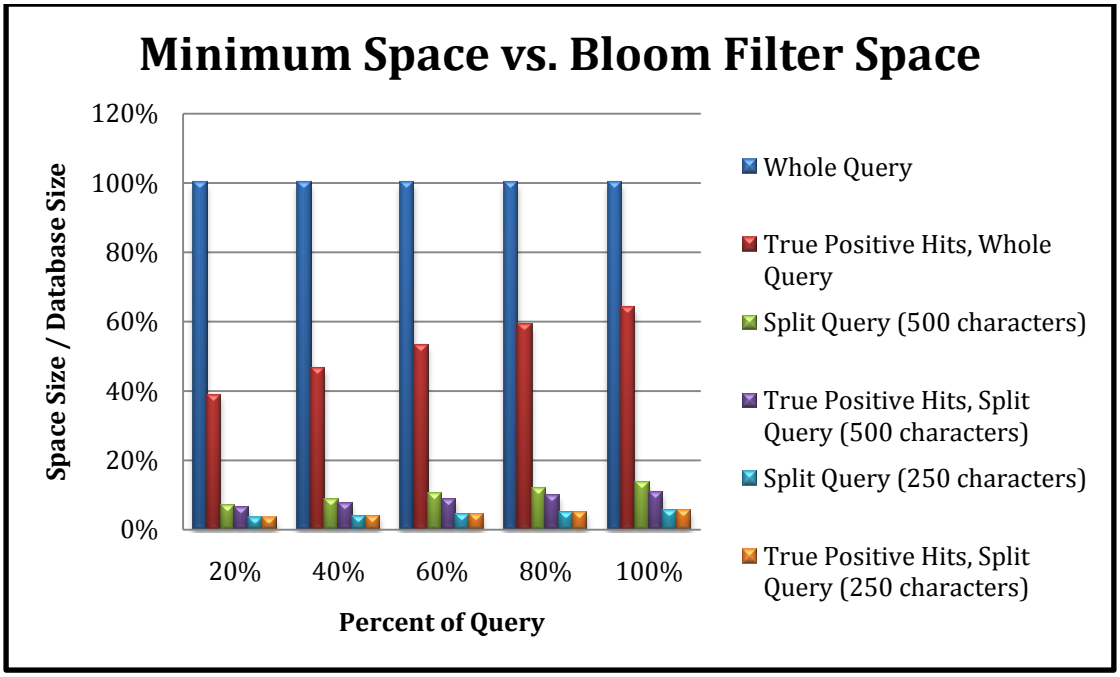
Για ολόκληρο το ερώτημα, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-30 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-31.

Ομοίως, για σπασμένο, σε 500 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-32 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-33.

Ομοίως, για σπασμένο, σε 250 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-34 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-35.

Ενδεικτικές είναι και οι γραφικές αναπαραστάσεις των δεδομένων που προέκυψαν και φαίνονται στο Σχήμα 2-7, οι οποίες επιβεβαιώνουν αυτό που αναφέρθηκε ακριβώς παραπάνω.

Στο Σχήμα 2-7Σχήμα 2-4 φαίνονται τα αποτελέσματα του φίλτρου που υλοποιήθηκε όσον αφορά τα τελικά διαστήματα, σε σχέση με τα βέλτιστα αποτελέσματα, τα οποία θα επιτυγχάνονταν αν είχαμε μόνο true positive hits χρησιμοποιώντας και πάλι τη μέθοδο του σπασμένου ερωτήματος.



Σχήμα 2-7 Chr12, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space

2.4.2 Χρωμόσωμα 14

Σύμφωνα με το παραπάνω βέλτιστο configuration, από τον Πίνακας 6-36 ως τον Πίνακας 6-41 φαίνονται τα πειραματικά δεδομένα που προκύπτουν για το χρωμόσωμα 14.

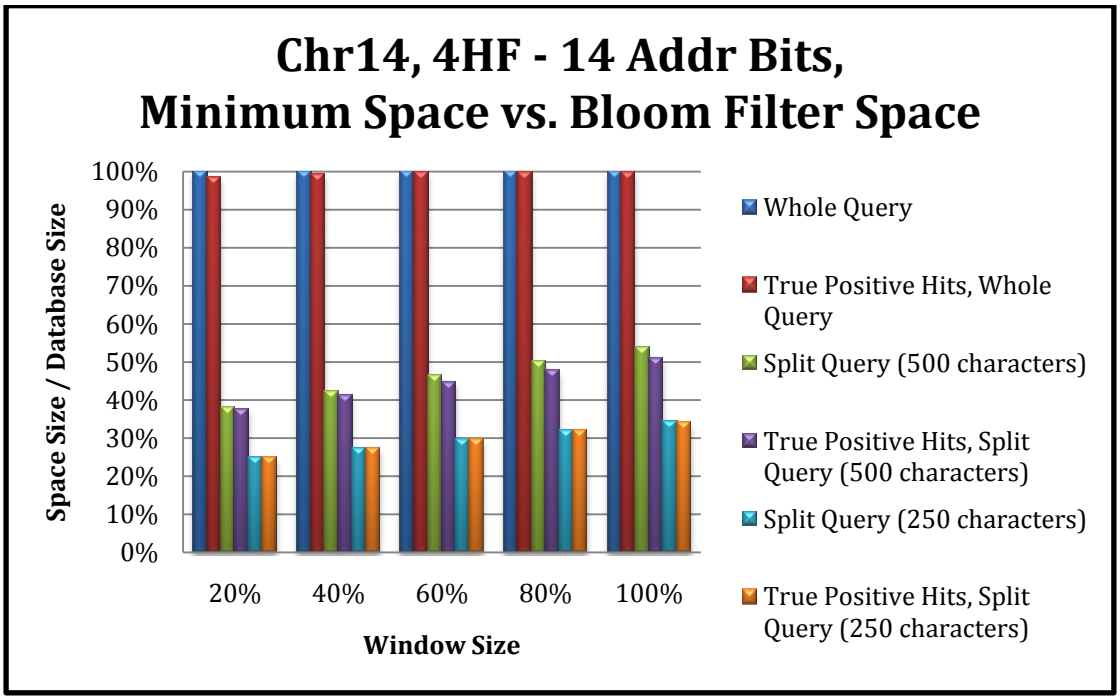
Για ολόκληρο το ερώτημα, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-36 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-37.

Ομοίως, για σπασμένο, σε 500 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-38 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-39.

Ομοίως, για σπασμένο, σε 250 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-40 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-41.

Ενδεικτικές είναι και οι γραφικές αναπαραστάσεις των δεδομένων που προέκυψαν και φαίνονται στο Σχήμα 2-8, οι οποίες επιβεβαιώνουν αυτό που αναφέρθηκε ακριβώς παραπάνω.

Στο Σχήμα 2-8Σχήμα 2-4 φαίνονται τα αποτελέσματα του φίλτρου που υλοποιήθηκε όσον αφορά τα τελικά διαστήματα, σε σχέση με τα βέλτιστα αποτελέσματα, τα οποία θα επιτυγχάνονταν αν είχαμε μόνο true positive hits χρησιμοποιώντας και πάλι τη μέθοδο του σπασμένου ερωτήματος.



Σχήμα 2-8: Chr14, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space

2.4.3 Χρωμόσωμα 17

Σύμφωνα με το παραπάνω βέλτιστο configuration, από τον Πίνακας 6-42 ως τον Πίνακας 6-47 φαίνονται τα πειραματικά δεδομένα που προκύπτουν για το χρωμόσωμα 17.

Για ολόκληρο το ερώτημα, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-42 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-43.

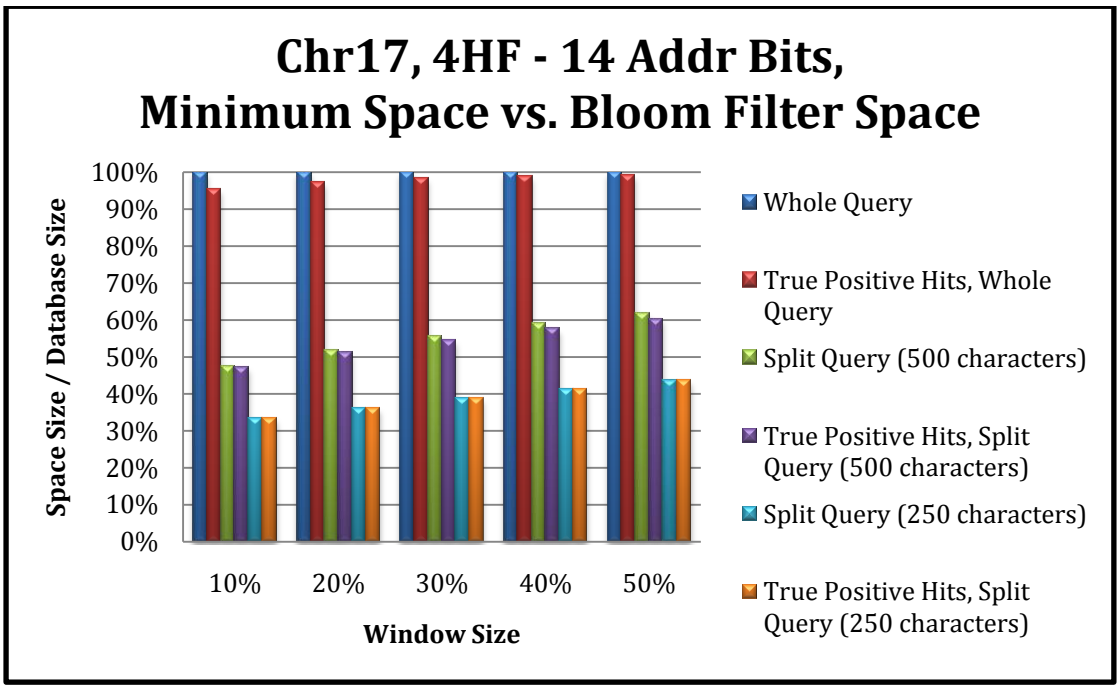
Ομοίως, για σπασμένο, σε 500 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-44 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο

τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-45.

Ομοίως, για σπασμένο, σε 250 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-46 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-47.

Ενδεικτικές είναι και οι γραφικές αναπαραστάσεις των δεδομένων που προέκυψαν και φαίνονται στο Σχήμα 2-9, οι οποίες επιβεβαιώνουν αυτό που αναφέρθηκε ακριβώς παραπάνω.

Στο Σχήμα 2-9Σχήμα 2-4 φαίνονται τα αποτελέσματα του φίλτρου που υλοποιήθηκε όσον αφορά τα τελικά διαστήματα, σε σχέση με τα βέλτιστα αποτελέσματα, τα οποία θα επιτυγχάνονταν αν είχαμε μόνο true positive hits χρησιμοποιώντας και πάλι τη μέθοδο του σπασμένου ερωτήματος.



Σχήμα 2-9: Chr17, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space

2.4.4 Χρωμόσωμα X

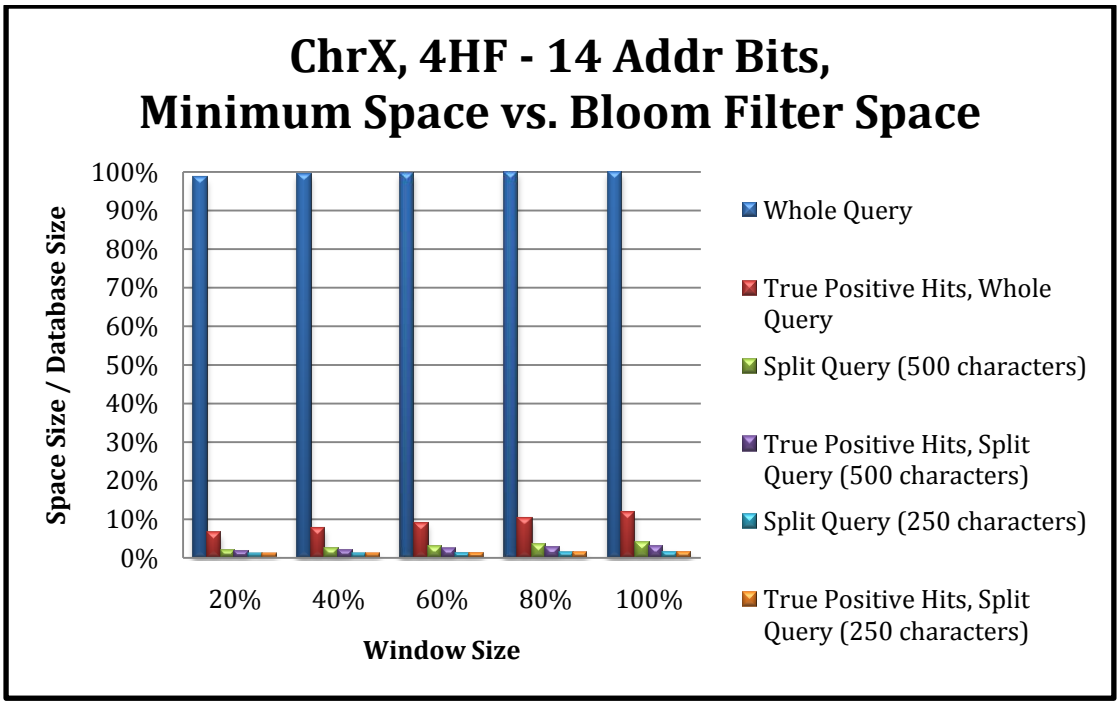
Σύμφωνα με το παραπάνω βέλτιστο configuration, από τον Πίνακας 6-48 ως τον Πίνακας 6-53 φαίνονται τα πειραματικά δεδομένα που προκύπτουν για το χρωμόσωμα X.

Για ολόκληρο το ερώτημα, προκύπτουν τα δεδομένα που φαίνονται στον Πίνακας 6-48 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-49.

Ομοίως, για σπασμένο, σε 500 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-50 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-51.

Ομοίως, για σπασμένο, σε 250 χαρακτήρες ερώτημα, προκύπτουν τα δεδομένα του Πίνακας 6-52 σε αντιπαραβολή με τα δεδομένα που προκύπτουν αν μετρηθεί το τελικό διάστημα της βάσης χρησιμοποιώντας και πάλι το σπασμένο ερώτημα και μόνο τα true positive hits χωρίς καθόλου false positive hits, τα οποία φαίνονται στον Πίνακας 6-53.

Ενδεικτικές είναι και οι γραφικές αναπαραστάσεις των δεδομένων που προέκυψαν και φαίνονται στο Σχήμα 2-10, οι οποίες επιβεβαιώνουν αυτό που αναφέρθηκε ακριβώς παραπάνω.



Σχήμα 2-10 ChrX, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space

Στο Σχήμα 2-10 φαίνονται τα αποτελέσματα του φίλτρου που υλοποιήθηκε όσον αφορά τα τελικά διαστήματα, σε σχέση με τα βέλτιστα αποτελέσματα, τα οποία

θα επιτυγχάνονταν αν είχαμε μόνο true positive hits χρησιμοποιώντας και πάλι τη μέθοδο του σπασμένου ερωτήματος.

2.4.5 Συμπεράσματα

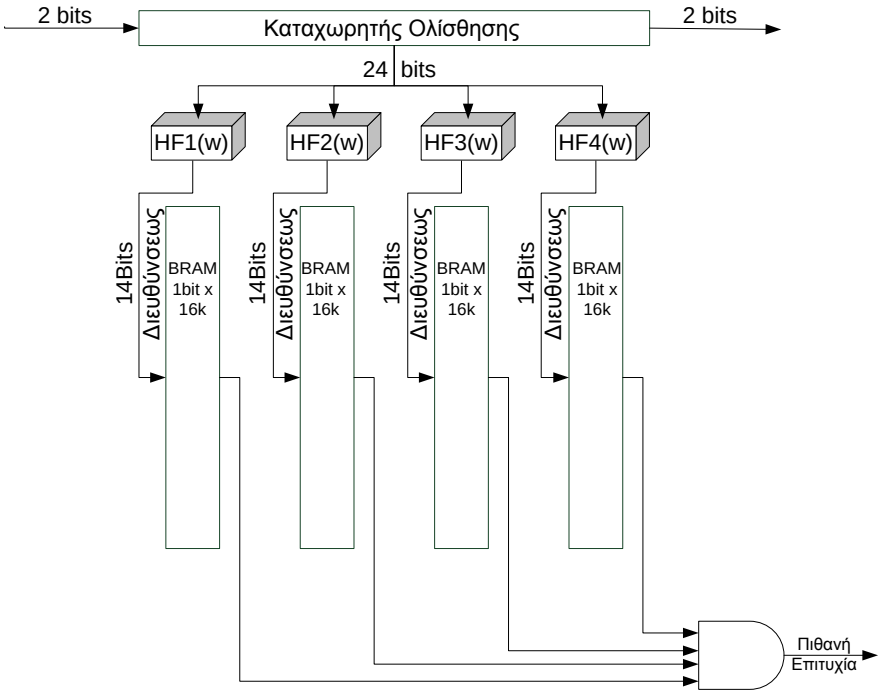
Από τις πιο πάνω γραφικές παραστάσεις, αλλά και από τους πίνακες με τα δεδομένα, φαίνεται πως το επιλεγμένο configuration αποδίδει τα αναμενόμενα και σε πιο «εύκολες» περιπτώσεις, δηλαδή σε περιπτώσεις όπου το ερώτημα είναι εξ αρχής μικρό.

3 Σχεδίαση και Υλοποίηση

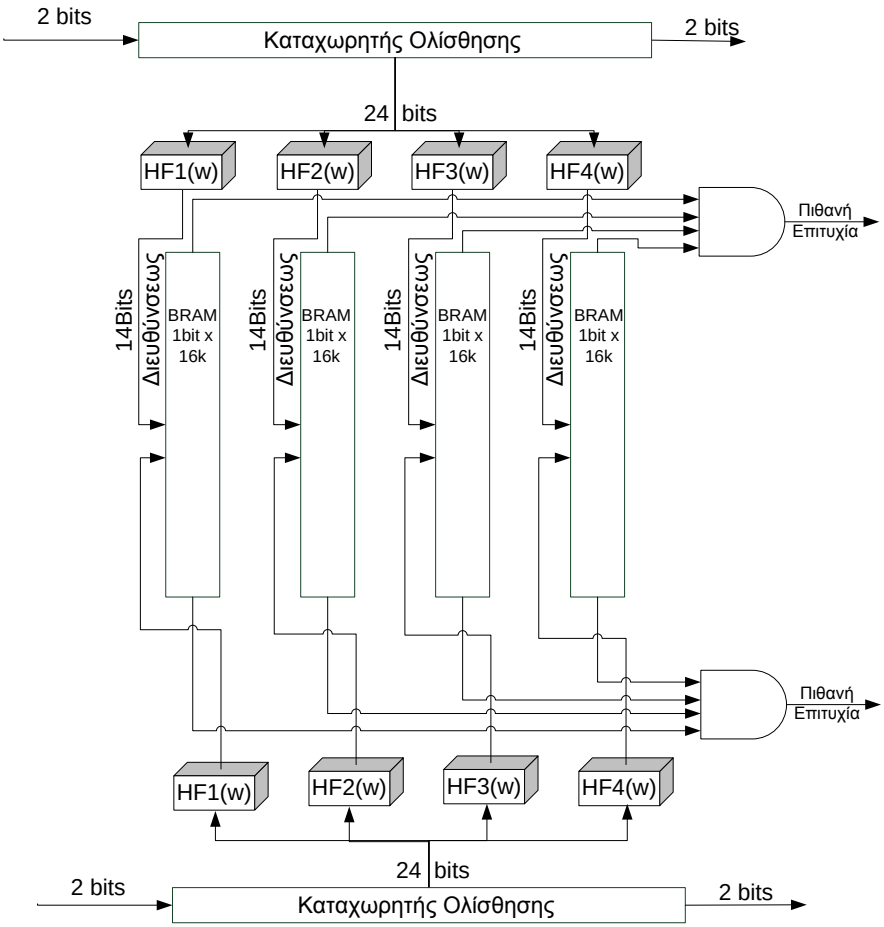
3.1 Αρχιτεκτονική TUC PreBLAST με χρήση φίλτρων Bloom

Όπως και στον TUC PreBLAST, έτσι και σ' αυτή την παραλλαγή που προτείνει αυτή η εργασία, η είσοδος της αρχιτεκτονικής είναι μια ροή δεδομένων. Κάθε νέο κύκλο ρολογιού, ένας νέος χαρακτήρας. Το datapath αποτελείται από ένα καταχωρητή ολίσθησης, μήκους 24 bits (12 χαρακτήρες x 2 bits ο καθένας), 4 μονάδες που υλοποιούν τις συναρτήσεις κατακερματισμού και 4 μνήμες RAM πλάτους 1 bit και βάθους 2^{14}. Ο καταχωρητής ολίσθησης, σε κάθε κύκλο ρολογιού, εισάγει ένα νέο χαρακτήρα, και τα νέα w-mers που προκύπτουν, τροφοδοτούν με τις κατάλληλες ποσότητες από bits τις μονάδες των συναρτήσεων κατακερματισμού. Οι μονάδες αυτές παράγουν κάποιες άλλες, κατακερματισμένες ποσότητες από bits, οι οποίες θα χρησιμοποιηθούν για να διευθυνσιοδοτηθούν οι μνήμες του συστήματος. Αν σε κάθε μία από τις διευθύνσεις αυτές και στην αντίστοιχη μνήμη, είναι αποθηκευμένο το '1', τότε προκύπτει σαν έξοδος μια πιθανή επιτυχία. Το μέγεθος του καταχωρητή ολίσθησης, οφείλεται στο μέγεθος του κάθε w-mer. Σε κάθε κύκλο ρολογιού, αφού εισάγεται ένας νέος χαρακτήρας στον καταχωρητή ολίσθησης, ο τελευταίος πραγματοποιεί ολίσθηση 2 θέσεων. Τα bits του καταχωρητή ολίσθησης, περνάνε μέσα από τις μονάδες των συναρτήσεων κατακερματισμού και παράγονται 4 διευθύνσεις, οι οποίες διευθυνσιοδοτούν τις μνήμες RAM με 14 bits την κάθε μία. Οι απαντήσεις των μνημών RAM είναι 1 bit η κάθε μια και αφού προκύψουν, το χειρισμό τους αναλαμβάνει μια πύλη AND. Αν το αποτέλεσμα της λογικής πράξης AND είναι '1', τότε παράγεται μια πιθανή επιτυχία. Δύο σχεδιάσεις υλοποιήθηκαν, οι οποίες διαφέρουν στο είδος της μνήμης RAM. Στο Σχήμα 3-1 και στο Σχήμα 3-2 φαίνονται οι σχεδιάσεις των datapath που προτείνονται. Η χρήση μνημών RAM 2 εισόδων καταφέρνει και μας δίνει 2 προεπεξεργαστές ανά μονάδα TUC PreBLAST.

Το control path της αρχιτεκτονικής του TUC PreBLAST δεν άλλαξε καθόλου. Η διαδικασία εφαρμογής παραθύρων και κατωφλιού, παραμένει ο ίδιος. Η σχεδίαση του control path είναι αυτή που φαίνεται στο Σχήμα 1-5.



Σχήμα 3-1: Το datapath της αρχιτεκτονικής TUC PreBLAST με φίλτρο Bloom και μνήμες RAM μιας εισόδου



Σχήμα 3-2: Το datapath της αρχιτεκτονικής TUC PreBLAST με φίλτρο Bloom και μνήμες RAM δύο θυρών

3.2 Αρχιτεκτονική πολυεπεξεργαστικού συστήματος M.PL.EM. με ενσωμάτωση TUC PreBLAST με χρήση φίλτρου Bloom

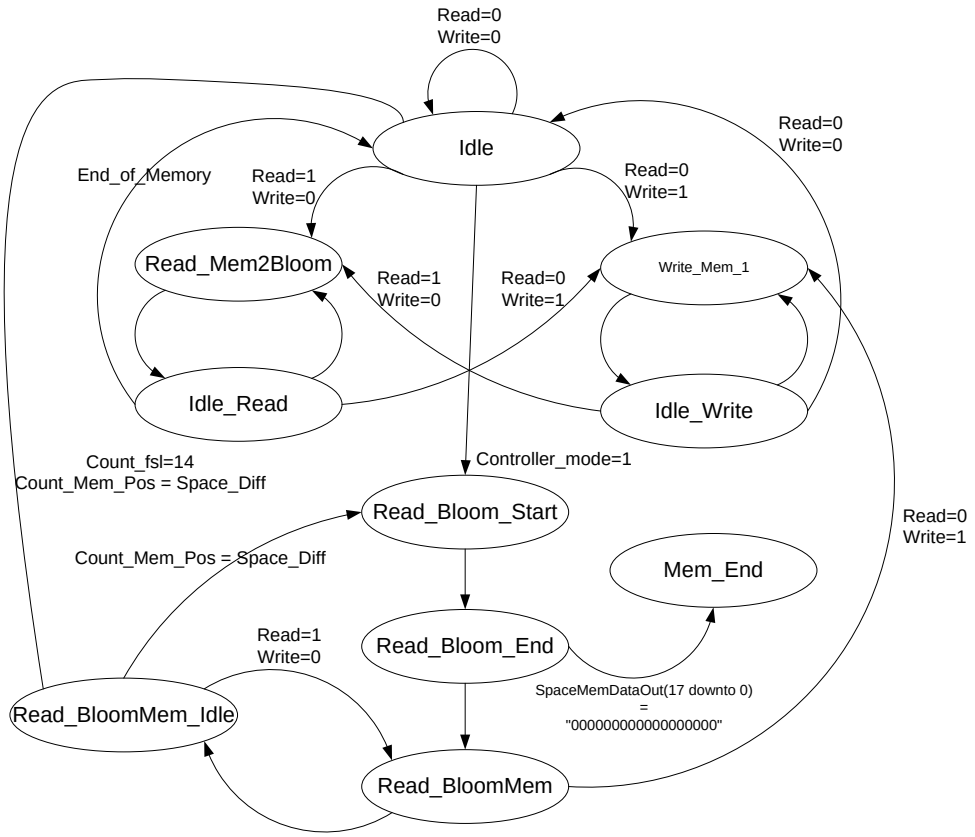
Η αρχιτεκτονική που προτείνει αυτή η εργασία για την ενσωμάτωση της παραλλαγής του TUC PreBLAST με τη χρήση φίλτρου Bloom, στο πολυεπεξεργαστικό σύστημα MP.LE.M., διαφέρει από την αρχική αρχιτεκτονική του πολυεπεξεργαστικού συστήματος M.PL.EM. σε κάποια καίρια σημεία. Η ενσωμάτωση του TUC PreBLAST εστιάζεται στην τροποποίηση του SRAM Controller. Πιο συγκεκριμένα, πριν ο SRAM Controller στείλει δεδομένα από τη μνήμη SRAM στις μνήμες των περιφερειακών FSL, τα στέλνει πρώτα στον TUC PreBLAST. Ο TUC PreBLAST, τα επεξεργάζεται με τον τρόπο που παρουσιάστηκε παραπάνω και γράφει τα αποτελέσματα στη μνήμη της μονάδας ελέγχου του, τη Space Memory. Μόλις ο TUC PreBLAST ολοκληρώσει τη λειτουργία του, ο SRAM Controller αναλαμβάνει να διαβάσει τα διαστήματα των περιοχών ενδιαφέροντος που υπολόγισε ο TUC PreBLAST και να υπολογίσει ποια δεδομένα θα στείλει στο κάθε περιφερειακό FSL. Ο SRAM Controller, στέλνει τα δεδομένα μιας περιοχής ενδιαφέροντος στη μνήμη ενός περιφερειακού FSL, ώστε να τα επεξεργαστεί ο επεξεργαστής του και να πραγματοποιήσει τις πιθανές επεκτάσεις του 3ου βήματος του αλγορίθμου BLAST.

Γίνεται ξεκάθαρο ότι πλέον ο SRAM Controller πρέπει να δουλεύει σε 2 καταστάσεις. Στην πρώτη κατάσταση διαβάζει τη μνήμη SRAM Και στέλνει τα δεδομένα στον TUC PreBLAST. Μόλις ο TUC PreBLAST επεξεργαστεί την είσοδο που του στέλνει ο SRAM Controller και εξάγει τις περιοχές ενδιαφέροντος, δίνει σήμα στον SRAM Controller να περάσει στην κατάσταση 2. Στην κατάσταση αυτή, ο SRAM Controller ζητάει από τον TUC PreBLAST να του στείλει τα αποτελέσματα της προηγούμενης επεξεργασίας, ώστε να στείλει στις μνήμες των περιφερειακών FSL τα δεδομένα των περιοχών ενδιαφέροντος.

Όπως αναφέρθηκε και παραπάνω, για την διαχείριση της επικοινωνίας μεταξύ της μνήμης SRAM και των περιφερειακών FSL, σχεδιάστηκε για το πολυεπεξεργαστικό σύστημα M.PL.EM. ένας ελεγκτής, ο οποίος καθορίζει τον τρόπο με τον οποίο αποστέλλονται τα δεδομένα από τη μνήμη SRAM στις μνήμες των περιφερειακών FSL. Ο SRAM Controller του πολυεπεξεργαστικού συστήματος M.PL.EM. συνδεόταν από τη μια πλευρά με τη μνήμη SRAM και από την άλλη πλευρά με τα περιφερειακά FSL. Ο SRAM Controller υλοποιήθηκε με τη χρήση μιας Μηχανής Πεπερασμένων Καταστάσεων (FSM). Η FSM του πολυεπεξεργαστικού συστήματος M.PL.EM. έπρεπε να τροποποιηθεί ώστε να γίνει η ενσωμάτωση του TUC PreBLAST στο πολυεπεξεργαστικό σύστημα M.PL.EM.

Στο Σχήμα 3-3 παρουσιάζεται ο SRAM Controller μετά τις τροποποιήσεις που έγιναν για την αρχιτεκτονική που προτείνει αυτή η εργασία. Οι καταστάσεις Read_Mem2Bloom και Idle_Read αντικαθιστούν τις καταστάσεις Read_Mem και Idle_Read, στις οποίες γινόταν η ανάγνωση των δεδομένων της μνήμης και η αποστολή τους στα FSL περιφερειακά. Οι νέες καταστάσεις διαβάζουν και πάλι τα δεδομένα από τη μνήμη, ωστόσο τα στέλνουν στον TUC PreBLAST ώστε να τα επεξεργαστεί και να υπολογίσει τις περιοχές ενδιαφέροντος. Επίσης προστέθηκαν οι καταστάσεις Read_Bloom_Start, Read_Bloom_End, Read_BloomMem και Read_BloomMem_Idle, η λειτουργία των οποίων θα παρουσιασθεί παρακάτω.

Ο SRAM Controller που παρουσιάζεται, υλοποιεί έναν αλγόριθμο Round Robin για την αρχικοποίηση των BRAMs των περιφερειακών FSL. Η μνήμη SRAM είναι ήδη προφορτωμένη με τα δεδομένα της βάσης γενετικών δεδομένων, ενώ οι μνήμες του TUC PreBLAST είναι ήδη φορτωμένες με το ερώτημα. Υποστηρίζεται και η δυνατότητα εγγραφής στις μνήμες αυτές, παρόλο που στα πλαίσια της συγκεκριμένης εργασίας δε χρησιμοποιείται.

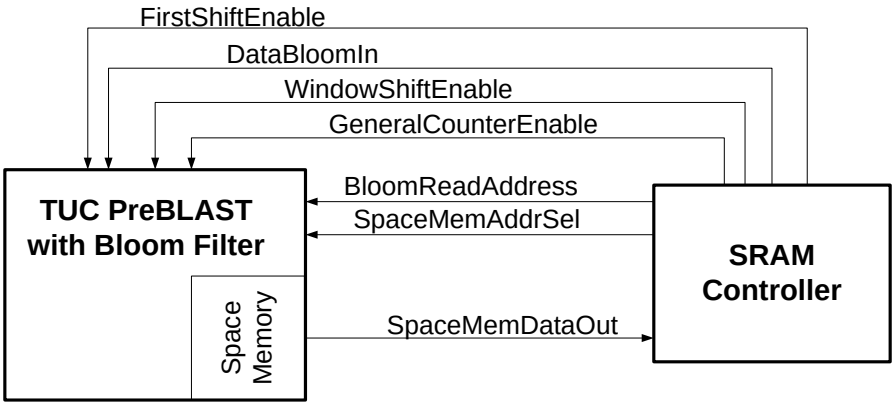


Σχήμα 3-3: Η νέα FSM του SRAM Controller

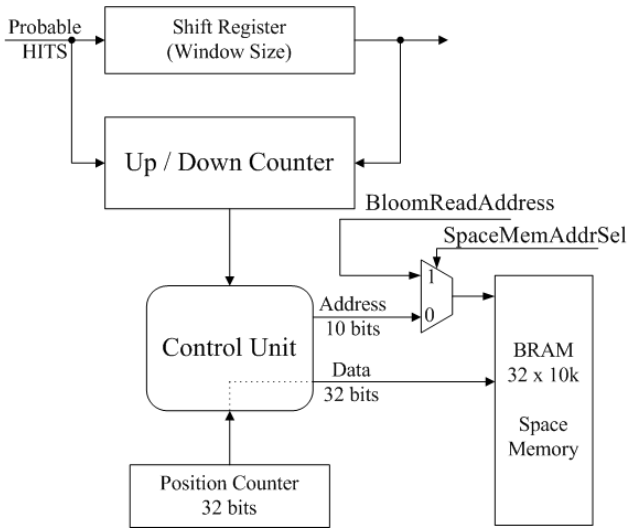
Πιο συγκεκριμένα, ο SRAM Controller λειτουργεί με τον ακόλουθο τρόπο. Αρχικά βρίσκεται στην κατάσταση Idle και σε Controller_Mode 0. Στην κατάσταση αυτή, ο SRAM Controller περιμένει για μια αίτηση ανάγνωσης ή εγγραφής, καθώς προχωρά και στην αρχικοποίηση των κατάλληλων μετρητών για την υλοποίηση του αλγορίθμου Round Robin. Όταν προκύψει μια αίτηση ανάγνωσης από κάποιο περιφερειακό FSL, ο SRAM Controller μεταβαίνει από την κατάσταση Idle στην κατάσταση Read_Mem2Bloom. Τότε ο SRAM Controller, διαβάζει τα δεδομένα της μνήμης SRAM και τα στέλνει στον TUC PreBLAST. Υπάρχουν τέσσερις μετρητές, ένας ο οποίος μετράει τον αριθμό των περιφερειακών που θα πάρουν δεδομένα, ένας που κρατάει τη θέση μνήμης από την οποία διαβάζουμε στην SRAM, ένας που μετράει τη διεύθυνση που θα σταλεί στο FSL περιφερειακό και ένας που μετράει τη διεύθυνση που θα σταλεί στη μνήμη των αποτελεσμάτων του TUC PreBLAST. Μετά από αυτό ο SRAM Controller θέτει τα σήματα ελέγχου της SRAM ως εξής: CE_b=0, OE_n=0, WE_n=1, BLE_n=0, BHE_n=0 [**32**]. Έπειτα μεταβαίνει στην κατάσταση Idle_Read. Αυτή η κατάσταση είναι υποχρεωτική εξαιτίας της λειτουργικότητας της μνήμης SRAM. Όταν στείλει όλα τα δεδομένα της μνήμης και ο TUC PreBLAST ολοκληρώσει την επεξεργασία τους, ο SRAM Controller γυρνάει στην κατάσταση Idle, ενώ περνάει σε Controller_Mode 1. Έπειτα μεταβαίνει στην κατάσταση Read_Bloom_Start. Στην κατάσταση αυτή, ζητάει από τον TUC PreBLAST να του δώσει τη διεύθυνση της αρχής μιας περιοχής ενδιαφέροντος στη βάση γενετικών δεδομένων. Έπειτα μεταβαίνει στην κατάσταση Read_Bloom_End όπου ζητάει από τον TUC PreBLAST να του δώσει τη διεύθυνση του τέλους της περιοχής ενδιαφέροντος στη βάση γενετικών δεδομένων. Μόλις διαβάσει και τη διεύθυνση του τέλους της περιοχής ενδιαφέροντος, περνάει στην κατάσταση Read_BloomMem. Στην κατάσταση αυτή, ο SRAM Controller δίνει στη μνήμη SRAM τις διευθύνσεις που πήρε από τον TUC PreBLAST και αφού θέσει τα σήματα ελέγχου της SRAM όμοια με προηγουμένως, αρχίζει και γράφει τα δεδομένα προς επεξεργασία στη μνήμη του περιφερειακού FSL. Εν τω μεταξύ έχει θέσει και το κατάλληλο σήμα write_enable για το περιφερειακό FSL που θα αρχικοποιηθεί. Μετά από αυτά, ο SRAM Controller μεταβαίνει στην κατάσταση Read_BloomMem_Idle. Αυτή η κατάσταση είναι και πάλι υποχρεωτική εξαιτίας της λειτουργικότητας της μνήμης SRAM. Μόλις γίνει η εγγραφή στη μνήμη του περιφερειακού FSL, επιστρέφει ο SRAM Controllers στην κατάσταση Read_BloomMem. Η εναλλαγή των τελευταίων καταστάσεων εκτελείται τόσες φορές όσοι είναι και οι χαρακτήρες της περιοχής ενδιαφέροντος. Η ίδια διαδικασία επαναλαμβάνεται μέχρι να γεμίσουν και τα 14 περιφερειακά με δεδομένα από τις 14 πρώτες περιοχές ενδιαφέροντος που υπολόγισε ο TUC PreBLAST. Αν έχουν προκύψει παραπάνω από 14 περιοχές ενδιαφέροντος, ο SRAM Controller περιμένει

μέχρι να ολοκληρώσει κάποιο περιφερειακό FSL και έπειτα ακολουθεί την διαδικασία που περιγράφηκε. Όταν τελειώσουν οι περιοχές ενδιαφέροντος, ο SRAM Controller μεταβαίνει στην κατάσταση Mem_End στην οποία μένει για πάντα.

Τα υπόλοιπα δομικά στοιχεία του πολυεπεξεργαστικού συστήματος M.PL.EM. καθώς και το σύστημα των μνημών του TUC PreBLAST δεν τροποποιήθηκαν. Ωστόσο έπρεπε να τροποποιηθεί το control path του TUC PreBLAST. Η αλλαγή που έπρεπε να γίνει ήταν να ενσωματωθεί ένας πολυπλέκτης ακριβώς πριν από την πόρτα διευθύνσεων της Space Memory, ώστε όταν ο SRAM Controller βρισκεται σε Controller_Mode 0, η μνήμη αυτή να παίρνει διευθύνσεις προορισμού από τον TUC PreBLAST, ενώ όταν βρίσκεται σε Controller_Mode 1, η μνήμη να παίρνει διευθύνσεις ανάγνωσης από τον SRAM Controller. Ένα block διάγραμμα της διασύνδεσης μεταξύ του SRAM Controller και του TUC PreBLAST φαίνεται στο Σχήμα 3-4, ενώ η αλλαγή που έγινε στο Control Path του TUC PreBLAST, φαίνεται στο Σχήμα 3-5.



Σχήμα 3-4: Διασύνδεση SRAM Controller και TUC PreBLAST



Σχήμα 3-5: Νέο Controll Path του TUC PreBLAST

4 Μέτρηση Αποδόσεων

4.1 TUC PreBLAST με χρήση Φίλτρου BLOOM

Η σχεδίαση υλοποιήθηκε, έγινε place and route και εξομοιώθηκε πλήρως. Για την υλοποίηση χρησιμοποιήθηκε μια FGPA της οικογένειας Xilinx Virtex 5 και συγκεκριμένα η XC5VLX330T. Οι μετρήσεις έγιναν σε περιβάλλον Xilinx ISE 10.1, σε λειτουργικό σύστημα Ubuntu 8.04 και σε έναν υπολογιστή Intel Core2Duo E8400, με συχνότητα 3GHz και 2GB RAM. Ο Πίνακας 4-1 δείχνει την κατανομή των πόρων για ένα προεπεξεργαστή και 108 παράλληλους προεπεξεργαστές στην περίπτωση που χρησιμοποιούνται μνήμες RAM μονής θύρας, ενώ ο Πίνακας 4-2 δείχνει την κατανομή των πόρων για μια μονάδα προεπεξεργασίας, δηλαδή για 2 και 81 παράλληλες μονάδες προεπεξεργασίας, δηλαδή 162 προεπεξεργαστές συνολικά, στην περίπτωση που χρησιμοποιούνται μνήμες RAM δύο θυρών. Οι πολλαπλοί επεξεργαστές λειτουργούν όλοι παράλληλα και βρίσκονται όλοι σε ένα ολοκληρωμένο. Η σχεδίαση περιορίζεται στους 108 και 162 προεπεξεργαστές αντίστοιχα, λόγω του συνολικού αριθμού των διαθέσιμων BRAMs της συγκεκριμένης FPGA. Είναι αξιοσημείωτο ότι μπορεί να χρησιμοποιούνται οι BRAMs στο σύνολό τους, ωστόσο πολύ μικρό ποσοστό των LUTs χρησιμοποιείται. Ο Πίνακας 4-3 και ο Πίνακας 4-4 δείχνουν τις αντίστοιχες συχνότητες ρολογιού και το throughput του κάθε συστήματος.

Πίνακας 4-1: Κατανεμημένοι Πόροι για τον TUC PreBLAST με χρήση Φίλτρων BLOOM με μνήμες RAM μονής θύρας

Αριθμός Προεπεξεργαστών	LUTs/Μονάδα	BRAM/ Μονάδα
1	381	3
108	8781	324
Συνολικοί Πόροι FPGA	207360	324
Ποσοστό Κάλυψης (1 προεπεξεργαστής)	0,02%	1%
Ποσοστό Κάλυψης (108 προεπεξεργαστές)	4,23%	100%

Πίνακας 4-2: Κατανεμημένοι Πόροι για τον TUC PreBLAST με χρήση Φίλτρων BLOOM με μνήμες RAM δύο θυρών

Αριθμός Προεπεξεργαστών	LUTs/Μονάδα	BRAM/ Μονάδα
2	504	4
162	14104	324
Συνολικοί Πόροι FPGA	207360	324
Ποσοστό Κάλυψης (2 προεπεξεργαστής)	0,02%	1.23%
Ποσοστό Κάλυψης (162 προεπεξεργαστές)	6,8%	100%

Πίνακας 4-3: Ρολόι Συστήματος και Throughput του TUC PreBLAST με χρήση Φίλτρων BLOOM με μνήμες RAM μονής θύρας

Αριθμός Προεπεξεργαστών	Ταχύτητα Ρολογιού (MHz)	Throughput Χαρακτήρες/Sec (10^6)	Throughput Bits/Sec (10^6)
1	300,842	300,842	601,684
108	300,842	32490,936	64981,872

Πίνακας 4-4: Ρολόι Συστήματος και Throughput του TUC PreBLAST με χρήση Φίλτρων BLOOM με μνήμες RAM δύο θυρών

Αριθμός Προεπεξεργαστών	Ταχύτητα Ρολογιού (MHz)	Throughput Χαρακτήρες/Sec (10^6)	Throughput Bits/Sec (10^6)
2	300,842	601,684	1203,368
162	300,842	48736,404	97472,808

Για τα πειράματα που πραγματοποιήθηκαν με τεμαχισμένο ερώτημα, έχουν προκύψει τα αντίστοιχα throughputs. Για ένα προεπεξεργαστή με μνήμες RAM μονής θύρας και ερώτημα τεμαχισμένο σε κομμάτια των 1000, των 500 και των 250 χαρακτήρων, τα throughputs φαίνονται στον Πίνακας 4-5. Για 108 προεπεξεργαστές με μνήμες RAM μονής θύρας και ερώτημα τεμαχισμένο σε κομμάτια των 1000, των 500 και των 250 χαρακτήρων, τα throughputs φαίνονται στον Πίνακας 4-6. Για μια μονάδα προεπεξεργασίας, επομένως για δύο προεπεξεργαστές με μνήμες RAM δύο θυρών και ερώτημα τεμαχισμένο σε κομμάτια των 1000, των 500 και των 250 χαρακτήρων, τα throughputs φαίνονται στον Πίνακας 4-7. Για 81 μονάδες προεπεξεργασίας, επομένως για 162 προεπεξεργαστές συνολικά με μνήμες RAM δύο θυρών και ερώτημα τεμαχισμένο σε κομμάτια των 1000, των 500 και των 250 χαρακτήρων, τα throughputs

φαίνονται στον Πίνακας 4-8. Στους χρόνους εκτέλεσης, συμπεριλαμβάνεται και ο χρόνος εκτέλεσης του NCBI BLAST, ο οποίος αναλαμβάνει να εκτελέσει τις επεκτάσεις από τη στιγμή που ο TUC PreBLAST θα του δώσει τις περιοχές ενδιαφέροντος. Ο χρόνος εκτέλεσης του NCBI BLAST μετρήθηκε με το πρόγραμμα VTune Peformance Analyzer v9.0 της INTEL, σε έναν υπολογιστή γενικού σκοπού, με λειτουργικό σύστημα Windows XP SP3, επεξεργαστή Intel Core2Duo στα 2,2GHz και μνήμη RAM 2GB. Το πείραμα που εκτελέσθηκε έχει σα βάση γενετικών δεδομένων το chr13 του χιμπατζή και σαν ερώτημα ένα κομμάτι του αντίστοιχου χρωμοσώματος του ανθρώπου. Ο τελικός χρόνος εκτέλεσης του πειράματος είναι ίσος με 36,34 sec.

Πίνακας 4-5: Χρόνοι Εκτέλεση NCBI BLAST και TUC PreBLAST για ένα προεπεξεργαστή με μνήμες RAM μονής θύρας

Μέγεθος Ερωτήματος (χαρακτήρες)	Συνολικός Χρόνος Εκτέλεσης NCBI BLAST και TUC PreBLAST (sec)
Ολόκληρο Ερώτημα	36.668
1000	38.253
500	27.534
250	31.016

Πίνακας 4-6: Χρόνοι Εκτέλεση NCBI BLAST και TUC PreBLAST για 108 προεπεξεργαστές με μνήμες RAM μονής θύρας

Μέγεθος Ερωτήματος (χαρακτήρες)	Συνολικός Χρόνος Εκτέλεσης NCBI BLAST και TUC PreBLAST (sec)
Ολόκληρο Ερώτημα	36.343
1000	33.991
500	19.007
250	14.286

Πίνακας 4-7: Χρόνοι Εκτέλεση NCBI BLAST και TUC PreBLAST για 2 προεπεξεργαστές με μνήμες RAM δύο θυρών

Μέγεθος Ερωτήματος (χαρακτήρες)	Συνολικός Χρόνος Εκτέλεσης NCBI BLAST και TUC PreBLAST (sec)
Ολόκληρο Ερώτημα	36.504
1000	36.121
500	23.269
250	22.649

Πίνακας 4-8: Χρόνοι Εκτέλεση NCBI BLAST και TUC PreBLAST για 162 προεπεξεργαστές με μνήμες RAM δύο θυρών

Μέγεθος Ερωτήματος (χαρακτήρες)	Συνολικός Χρόνος Εκτέλεσης NCBI BLAST και TUC PreBLAST (sec)
Ολόκληρο Ερώτημα	36.342
1000	33.990
500	19.006
250	14.285

Αξίζει να γίνει σύγκριση των αποδόσεων του TUC PreBLAST με τη χρήση φίλτρων BLOOM και χωρίς. Δεδομένου ότι δεν υπάρχουν πληροφορίες για τους χρόνους εκτέλεσης, θα γίνει σύγκριση μόνο των πόρων που καταναλώνουν οι δύο αρχιτεκτονικές, το throughput αλλά και τα τελικά διαστήματα των περιοχών ενδιαφέροντος της βάσης γενετικών δεδομένων. Η αρχιτεκτονική του TUC PreBLAST χωρίς τη χρήση φίλτρων Bloom χρησιμοποιεί 4 BRAMs για τον κατακερματισμό των w-mers του ερωτήματος και 1 ακόμα BRAM για την αποθήκευση των διαστημάτων. Δεδομένου ότι η FPGA που χρησιμοποιήθηκε, η XC5VLX330T, διαθέτει 324 BRAMs, υπολογίζεται ότι χωράει 64 τέτοιους προεπεξεργαστές. Αντίστοιχα, η αρχιτεκτονική του TUC PreBLAST με τη χρήση φίλτρου Bloom χρησιμοποιεί 2 BRAMs για τον κατακερματισμό των w-mers του ερωτήματος και 1 ακόμα BRAM για την αποθήκευση των διαστημάτων. Αντίστοιχα, η XC5VLX330T μπορεί να χωρέσει 108 τέτοιους προεπεξεργαστές. Ο Πίνακας 4-9 δείχνει τη σύγκριση των αποδόσεων των δύο αρχιτεκτονικών.

Πίνακας 4-9: Σύγκριση πόρων και αποδόσεων των δύο αρχιτεκτονικών με μνήμες RAM μονής θύρας

TUC PreBLAST				TUC PreBLAST με χρήση Φίλτρων BLOOM			
Αριθμός Προεπεξεργαστών	LUTs/ Μονάδα	BRAMs / Μονάδα	Throughput Χαρακτήρες/sec (10^6)	Αριθμός Προεπεξεργαστών	LUTs/ Μονάδα	BRAMs / Μονάδα	Throughput Χαρακτήρες/sec (10^6)
1	105	5	232,32	**1**	381	3	300,842
64	3780	320	8973,64	**108**	8781	324	32490,936

Η αρχιτεκτονική του TUC PreBLAST χωρίς τη χρήση φίλτρων Bloom αλλά με τη χρήση μνημών RAM διπλής εισόδου χρησιμοποιεί 4 BRAMs για τον κατακερματισμό των w-mers του ερωτήματος και 2 ακόμα BRAMs για την αποθήκευση των διαστημάτων. Δεδομένου ότι η FPGA που χρησιμοποιήθηκε, η XC5VLX330T, διαθέτει 324 BRAMs, υπολογίζεται ότι χωράει 54 τέτοιες μονάδες προεπεξεργασίας που ουσιαστικά περιέχουν δύο προεπεξεργαστές η κάθε μία, άρα 108 προεπεξεργαστές συνολικά. Αντίστοιχα, η αρχιτεκτονική του TUC PreBLAST με τη χρήση φίλτρου Bloom και μνημών RAM δύο θυρών χρησιμοποιεί 2 BRAMs για τον κατακερματισμό των w-mers του ερωτήματος και 2 ακόμα BRAMs για την αποθήκευση των διαστημάτων. Αντίστοιχα, η XC5VLX330T μπορεί να χωρέσει 81 τέτοιες μονάδες προεπεξεργασίας που ουσιαστικά περιέχουν 2 προεπεξεργαστές η κάθε μία, άρα 162 προεπεξεργαστές συνολικά. Ο Πίνακας 4-10 δείχνει τη σύγκριση των αποδόσεων των δύο αρχιτεκτονικών.

Πίνακας 4-10: Σύγκριση πόρων και αποδόσεων των δύο αρχιτεκτονικών με μνήμες RAM δύο θυρών

TUC PreBLAST				TUC PreBLAST με χρήση Φίλτρων BLOOM			
Αριθμός Προεπεξεργαστών	LUTs/ Μονάδα	BRAMs / Μονάδα	Throughput Χαρακτήρες/sec (10^6)	Αριθμός Προεπεξεργαστών	LUTs/ Μονάδα	BRAMs / Μονάδα	Throughput Χαρακτήρες/sec (10^6)
2	177	6	232,32	**2**	504	4	601,684
108	6397	324	8973,64	**162**	14104	324	48736,404

Το πιο ουσιαστικό κομμάτι της σύγκρισης των δυο αρχιτεκτονικών είναι αυτό της σύγκρισης των τελικών διαστημάτων των περιοχών ενδιαφέροντος που αυτές παράγουν. Ο Πίνακας 4-11 παρουσιάζει την αντιπαραβολή των διαστημάτων που παράγουν οι δύο αρχιτεκτονικές για κάποια από τα χρωμοσώματα πάνω στα οποία πραγματοποιήθηκαν πειράματα.

Πίνακας 4-11: Σύγκριση Τελικών Διαστημάτων των Περιοχών Ενδιαφέροντος που εξάγουν οι δύο αρχιτεκτονικές του TUC PreBLAST με και χωρίς το φίλτρο Bloom

Βάση Γενετικών Δεδομένων Χρωμοσώματος Χιμπατζή	Ερώτημα βασισμένο στο ανθρώπινο χρωμόσωμα	Ποσοστό της βάσης % για τεμαχισμένο ερώτημα TUC PreBLAST (4 BRAMs)			Ποσοστό της βάσης % για τεμαχισμένο ερώτημα TUC PreBLAST με χρήση φίλτρου Bloom (2 BRAMs)		
Split Query Size		**1000**	**500**	**250**	**1000**	**500**	**250**
Chr13	**Chr13q 12750 chars**	88.01%	56.79%	31.87%	91.91%	40.37%	24.13%
Chr14	**Chr14q 5511 chars**	69.16%	43.13%	26.19%	66.05%	35.81%	23.56%
Chr17	**Chr17q 3959 chars**	69.90%	48.58%	32.92%	65.86%	45.32%	31.83%

Παραπάνω, αναφέρθηκε ότι ο χρόνος εκτέλεσης του NCBI BLAST μετρήθηκε με το πρόγραμμα VTune Peformance Analyzer v9.0 της INTEL, σε έναν υπολογιστή γενικού σκοπού, με λειτουργικό σύστημα Windows XP SP3, επεξεργαστή Intel Core2Duo στα 2,2GHz και μνήμη RAM 2GB. Το πείραμα που εκτελέσθηκε έχει σα βάση γενετικών δεδομένων το chr13 του χιμπατζή, μεγέθους 98704794 χαρακτήρων και σαν ερώτημα ένα κομμάτι του αντίστοιχου χρωμοσώματος του ανθρώπου, μεγέθους 12750 χαρακτήρων. Ο τελικός χρόνος εκτέλεσης του πειράματος είναι ίσος με 36,34 sec. Αυτό δηλώνει ότι του throughput του συστήματος είναι 2716147 chars/sec. Αν ζητούσαμε από αυτό το σύστημα να τρέξει το NCBI BLAST Software για 1 δευτερόλεπτο στον επεξεργαστή, θα επεξεργαζόταν 2716147 χαρακτήρες. Αν περνούσαμε αυτούς τους χαρακτήρες, προηγουμένως, από ένα προεπεξεργαστή TUC PreBLAST με χρήση φίλτρου Bloom με τη χρήση μνημών RAM 2 δύο θυρών, θα χρειαζόταν χρόνος ίσος με 0,0045 sec για να γίνει η επεξεργασία τους, δεδομένου ότι το throughput του συστήματος αυτού μετρήθηκε να είναι 604684000 chars/sec. Προκύπτει, επομένως, ο Πίνακας 4-13, ο οποίος παρουσιάζει τους συνολικούς χρόνους εκτέλεσης του NCBI BLAST Software επαυξημένους με το overhead που εισάγει η χρήση του TUC PreBLAST με χρήση φίλτρου Bloom. Ο Πίνακας 4-12 στις δύο πρώτες στήλες παρουσιάζει τα σενάρια κατανάλωσης χρόνου του αλγορίθμου BLAST για το Βήμα 2 και το Βήμα 3. Η γραμμή κάτω από το Database Space, δηλώνει τι ποσοστό της αρχικής βάσης γενετικών δεδομένων έχει μείνει μετά τη εφαρμογή του prefiltering.

Ο Πίνακας 4-13, τελικά, παρουσιάζει τα speed ups του συνολικού συστήματος του επεξεργαστή που τρέχει το NCBI BLAST Software, αφού πρώτα γίνει prefiltering της βάσης γενετικ δεδομένων από τον TUC PreBLAST με τη χρήση φίλτρου Bloom. Τα speed ups προκύπτουν από τους συνολικούς χρόνους εκτέλεσης του συστήματος μετά την εφαρμογή του prefiltering, σε σχέση με την αρχική υπόθεση, δηλαδή ότι το σύστημα αναφοράς τρέχει το NCBI BLAST για 1 sec.

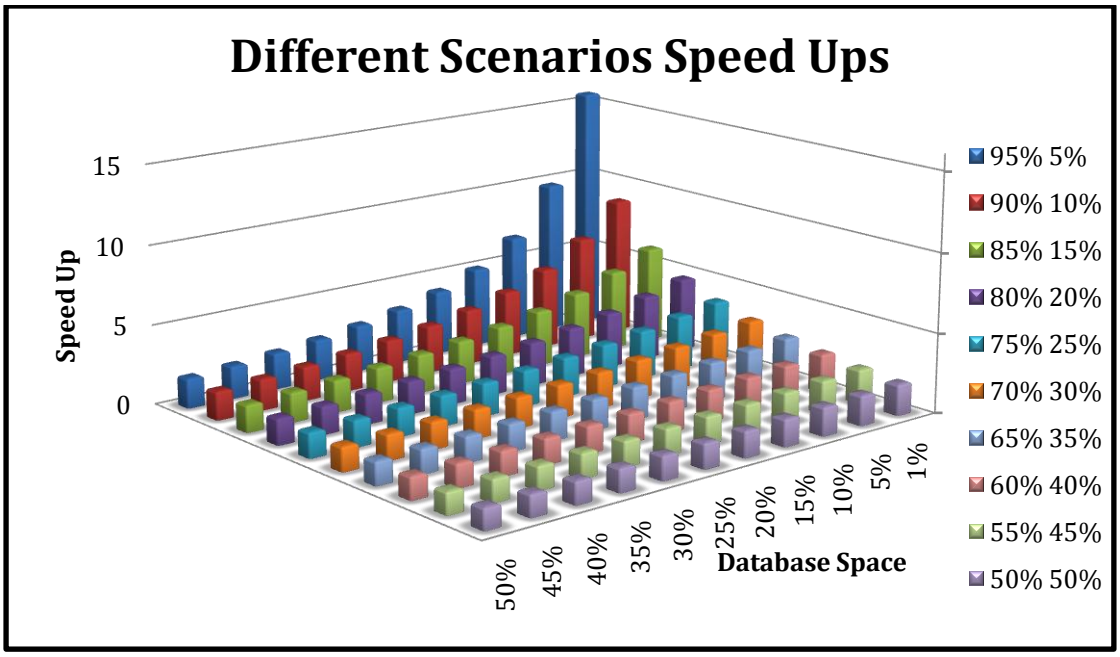
Πίνακας 4-12: Συνολικός Χρόνος Εκτέλεσης NCBI BLAST και TUC PreBLAST με χρήση φίλτρου Bloom

NCBI BLAST & TUC PreBLAST με φίλτρο Bloom (sec)												
		Database Space										
Step 2 (time load)	Step 3 (time load)	1%	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
95,00%	5,00%	0,064	0,102	0,150	0,197	0,245	0,292	0,340	0,387	0,435	0,482	0,530
90,00%	10,00%	0,114	0,150	0,195	0,240	0,285	0,330	0,375	0,420	0,465	0,510	0,555
85,00%	15,00%	0,163	0,197	0,240	0,282	0,325	0,367	0,410	0,452	0,495	0,537	0,580
80,00%	20,00%	0,213	0,245	0,285	0,325	0,365	0,405	0,445	0,485	0,525	0,565	0,605
75,00%	25,00%	0,262	0,292	0,330	0,367	0,405	0,442	0,480	0,517	0,555	0,592	0,630
70,00%	30,00%	0,312	0,340	0,375	0,410	0,445	0,480	0,515	0,550	0,585	0,620	0,655
65,00%	35,00%	0,361	0,387	0,420	0,452	0,485	0,517	0,550	0,582	0,615	0,647	0,680
60,00%	40,00%	0,411	0,435	0,465	0,495	0,525	0,555	0,585	0,615	0,645	0,675	0,705
55,00%	45,00%	0,460	0,482	0,510	0,537	0,565	0,592	0,620	0,647	0,675	0,702	0,730
50,00%	50,00%	0,510	0,530	0,555	0,580	0,605	0,630	0,655	0,680	0,705	0,730	0,755

Πίνακας 4-13: Speed Up NCBI BLAST και TUC PreBLAST με φίλτρου Bloom

Speed Up NCBI BLAST & TUC PreBLAST με φίλτρο Bloom												
		Database Space										
Step 2 (time load)	Step 3 (time load)	1%	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
95,00%	5,00%	15,622	9,803	6,688	5,076	4,090	3,424	2,945	2,584	2,301	2,075	1,889
90,00%	10,00%	8,809	6,688	5,141	4,175	3,515	3,035	2,670	2,384	2,153	1,963	1,803
85,00%	15,00%	6,134	5,076	4,175	3,546	3,082	2,725	2,442	2,212	2,022	1,862	1,726
80,00%	20,00%	4,706	4,090	3,515	3,082	2,743	2,472	2,250	2,064	1,907	1,771	1,654
75,00%	25,00%	3,817	3,424	3,035	2,725	2,472	2,262	2,085	1,934	1,803	1,689	1,589
70,00%	30,00%	3,210	2,945	2,670	2,442	2,250	2,085	1,944	1,820	1,711	1,614	1,528
65,00%	35,00%	2,770	2,584	2,384	2,212	2,064	1,934	1,820	1,718	1,627	1,546	1,472
60,00%	40,00%	2,436	2,301	2,153	2,022	1,907	1,803	1,711	1,627	1,552	1,483	1,419
55,00%	45,00%	2,174	2,075	1,963	1,862	1,771	1,689	1,614	1,546	1,483	1,424	1,371
50,00%	50,00%	1,963	1,889	1,803	1,726	1,654	1,589	1,528	1,472	1,419	1,371	1,325

Προκύπτει το τρισδιάστατο διάγραμμα που φαίνεται στο Σχήμα 4-1 και το οποίο περιγράφει καλύτερα τους παραπάνω πίνακες.



Σχήμα 4-1: 3-D Αναπαράσταση των Time-Load Scenarios και των Speed Up

Έστω για την περίπτωση που το time load είναι μοιρασμένο σε 80% και 20%, στα Βήματα 2 και 3 αντίστοιχα, του αλγορίθμου BLAST, όπως υποδεικνύεται στο [**15**]. Αν μετά το prefiltering, μείνει ένα 10% της βάσης γενετικών δεδομένων, ο νέος χρόνος εκτέλεσης του NCBI BLAST Software στον επεξεργαστή θα ήταν 0,28 sec, έναντι του 1 sec για την πλήρη βάση γενετικών δεδομένων. Αν στο χρόνο αυτό προστεθεί και το overhead που εισάγει η χρήση μιας μονάδας prefiltering, ο χρόνος εκτέλεσης θα είναι 0,285 sec. Αυτό μπορούμε να το δούμε και στον Πίνακας 4-12. Για το χρόνο αυτό, καταναλώνεται μικρή ποσότητα αναδιατασσόμενων πόρων, δηλαδή 504 LUTs και 4 BRAMs.

Για τη χειρότερη περίπτωση που μελετήθηκε, για το chr13, όπου το ερώτημα είναι 12750 χαρακτήρες, σε περίπτωση που σπάσουμε το ερώτημα σε κομμάτια των 250 χαρακτήρων, θα πρέπει να προστεθεί στον αρχικό χρόνο εκτέλεσης του NCBI BLAST Software στον επεξεργαστή, το overhead που θα προκύψει αν γίνει επεξεργασία και των 51 κομματιών του ερωτήματος το ένα μετά το άλλο από τη μονάδα prefiltering. Δηλαδή, αν t_{ex} είναι ο χρόνος εκτέλεσης του NCBI BLAST Software στον επεξεργαστή και t_o είναι ο χρόνος εκτέλεσης του prefiltering από τον TUC PreBLAST με τη χρήση φίλτρου Bloom, τότε ο συνολικός χρόνος εκτέλεσης του συστήματος θα είναι $t_{total} = t_{ex} + 51 \cdot t_o$. Το ελάχιστο τελικό διάστημα της βάσης γενετικών δεδομένων που προκύπτει για το

chr13, αν σπάσουμε το ερώτημα του σε κομμάτια των 250 χαρακτήρων είναι 24,13%. Επομένως, ο συνολικός χρόνος εκτέλεσης και του NCBI BLAST Software αλλά και του TUC PreBLAST με τη χρήση φίλτρου Bloom, για το παράδειγμα αυτό, θα είναι 22,649 sec. Είναι φανερό ότι ακόμα και στη χειρότερη περίπτωση, το speed up είναι της τάξης του 1,6x. Αν χρησιμοποιηθούν 51 μονάδες prefiltering, οι οποίες λειτουργούν παράλληλα, τότε στο συνολικό χρόνο εκτέλεσης στον επεξεργαστή, θα προστεθεί μόνο ένας χρόνος overhead, δηλαδή ο συνολικός χρόνος εκτέλεσης και του NCBI BLAST αλλά και του TUC PreBLAST με τη χρήση φίλτρου Bloom για το παράδειγμα αυτό θα είναι 0, sec. Επομένως προκύπτει speed up της τάξης του 2,544x. Και πάλι, η ποσότητα των αναδιατασσόμενων πόρων που χρησιμοποιήθηκε, 25704 LUTs και 204 BRAMs, είναι μικρή σε σχέση με το speed up που προσφέρει.

Αναλυτικότερα, τα δεδομένα για μερικές από τις πιο απαιτητικές περιπτώσεις φαίνονται στον Πίνακας 4-14.

Πίνακας 4-14: Speed Up για τις Απαιτητικές Περιπτώσεις Χρωμοσωμάτων (1 prefiltering unit)

		Execution Time - Speed Up (Time Load: Step2=80% - Step3=20%)					
		1 Prefiltering Unit					
Experiment	**NCBI BLAST Execution Time on PC (sec)**	**1000 chars**		**500 chars**		**250 chars**	
		Execution Time (sec)	**Speed Up**	**Execution Time (sec)**	**Speed Up**	**Execution Time (sec)**	**Speed Up**
Chr12	2,8	2,281	1,227	2,277	1,230	3,562	0,786
Chr13	36,34	36,121	1,006	23,270	1,562	22,649	1,604
Chr14	164,20	81,782	2,008	42,791	3,837	40,454	4,059
Chr17	180,42	89,411	2,018	46,119	3,912	42,817	4,214
ChrX	3,05	2,002	1,524	1,760	1,733	2,435	1,253

Πίνακας 4-15: Speed Up για τις Απαιτητικές Περιπτώσεις Χρωμοσωμάτων (81 prefiltering units)

Execution Time - Speed Up (Time Load: Step2=80% - Step3=20%)							
		81 Prefiltering Units (available)					
Experiment	**NCBI BLAST Execution Time on PC (sec)**	**1000 chars**		**500 chars**		**250 chars**	
		Execution Time (sec)	**Speed Up**	**Execution Time (sec)**	**Speed Up**	**Execution Time (sec)**	**Speed Up**
Chr12	2,8	1,382	2,026	0,702	3,991	0,634	4,419
Chr13	36,34	33,990	1,069	19,06	1,912	14,285	2,544
Chr14	164,20	80,880	2,030	40,986	4,006	36,993	4,439
Chr17	180,42	88,869	2,030	45,035	4,006	40,647	4,439
ChrX	3,05	1,505	2,026	0,764	3,990	0,690	4,419

4.2 Πολυεπεξεργαστικό σύστημα M.PL.EM. με ενσωμάτωση TUC PreBLAST με χρήση Φίλτρου BLOOM

Η σχεδίαση υλοποιήθηκε, έγινε place and route και εξομοιώθηκε πλήρως. Για την υλοποίηση χρησιμοποιήθηκε μια FGPA της οικογένειας Xilinx Virtex 2 Pro και συγκεκριμένα η XC2VP30. Οι μετρήσεις έγιναν σε περιβάλλον Xilinx ISE 7.1, σε λειτουργικό σύστημα Windows XP SP3 και σε έναν υπολογιστή Intel Core2Duo E8400 με συχνότητα 3GHz και 2GB RAM. Ο Πίνακας 4-16 δείχνει την κατανομή των πόρων για το πολυεπεξεργαστικό σύστημα M.PL.EM. που περιλαμβάνει 14 πυρήνες και ενσωματώνει και τον TUC PreBLAST Με τη χρήση φίλτρων BLOOM. Ο Πίνακας 4-17 δείχνει τις αντίστοιχες συχνότητες ρολογιού και το throughput του συστήματος.

Πίνακας 4-16: Κατανεμημένοι Πόροι για πολυεπεξεργαστικό σύστημα M.PL.EM. και τον TUC PreBLAST με χρήση Φίλτρων BLOOM (1 μονάδα prefiltering)

Αριθμός Επεξεργαστών	LUTs/Μονάδα	BRAM/ Μονάδα
14	23387	21
Συνολικοί Πόροι FPGA	27392	136
Ποσοστό Κάλυψης	85,4%	15,4%

Πίνακας 4-17: Ρολόι Συστήματος και Throughput του πολυεπεξεργαστικού συστήματος M.PL.EM. και του TUC PreBLAST με χρήση Φίλτρων BLOOM (1 μονάδα prefiltering)

Αριθμός Επεξεργαστών	Ταχύτητα Ρολογιού (MHz)	Throughput Χαρακτήρες/Sec (10^6)	Throughput Bits/Sec (10^6)
14	96,48	4527.3	9054.6

Ο Πίνακας 4-18 δείχνει τη σύγκριση των δύο αρχιτεκτονικών του MPLEM με και χωρίς τη χρήση του προεπεξεργαστή TUC PreBLAST όσων αφορά τους καταναλισκόμενους πόρους, ενώ ο Πίνακας 4-19 δείχνει τη σύγκριση των δύο αρχιτεκτονικών του MPLEM με και χωρίς τη χρήση του προεπεξεργαστή TUC PreBLAST όσων αφορά τους χρόνους εκτέλεσης. Και πάλι το πείραμα που χρησιμοποιείται για τη μέτρηση των χρόνων εκτέλεσης, έχει σα βάση γενετικών δεδομένων το chr13 του χιμπατζή και σαν ερώτημα ένα κομμάτι του αντίστοιχου χρωμοσώματος του ανθρώπου.

Οι χρόνοι εκτέλεσης των πειραμάτων έχουν μετρηθεί με κώδικα που εκτελεί τον αλγόριθμο BLAST και ο οποίος υλοποιήθηκε στο Εργαστήριο Μικροεπεξεργαστών και Υλικού. Δεν είναι βέλτιστος, για το λόγο του ότι γράφτηκε για τρέξει στον επεξεργαστή Microblaze.

Πίνακας 4-18: Σύγκριση των πόρων των δύο αρχιτεκτονικών

Αριθμός Επεξεργαστών	M.PL.EM.		M.PL.EM. & TUC PreBLAST (1 μονάδα prefiltering)	
	LUTs/ Μονάδα	BRAMs/ Μονάδα	LUTs/ Μονάδα	BRAMs/ Μονάδα
14	23010	18	23387	21

Ο Πίνακας 4-19 μας δείχνει κάτι σημαντικό. Με το prefiltering της βάσης γενετικών δεδομένων και με τη παροχή στο πολυεπεξεργαστικό σύστημα M.PL.EM. μόνο των περιοχών ενδιαφέροντος, επιτυγχάνεται speed up της τάξης του 7,5. Αυτό το speed up είναι σε σχέση με την προηγούμενη αρχιτεκτονική του MPLEM. Αν γίνει η σύγκριση με το αρχικό μηχάνημα αναφοράς για το MPLEM, ένα Intel Pentium 4 @ 3.2Ghz με μνήμη RAM 512MB και λειτουργικό σύστημα Windows XP SP2, το οποίο έτρεξε το μη βέλτιστο κώδικα που εκτελεί τον αλγόριθμο BLAST και στο οποίο ο χρόνος εκτέλεσης ήταν 283,22 δευτερόλεπτα, βλέπουμε ότι πετυχαίνουμε speed up της τάξης του 141x σε σχέση με αυτό. Η συνολική βελτίωση οφείλεται στο γεγονός ότι τελικά το πολυεπεξεργαστικό σύστημα M.PL.EM. επεξεργάζεται μόνο το 13,3% της αρχικής βάσης γενετικών δεδομένων.

Πίνακας 4-19: Σύγκριση του χρόνου εκτέλεσης των δύο αρχιτεκτονικών

Αριθμός Επεξεργαστών	MPLEM Software Intel Pentium 4 @ 3.2Ghz, 512 RAM, Windows XP SP2	M.PL.EM.	M.PL.EM. & TUC PreBLAST (1 μονάδα prefiltering)
	Χρόνος Εκτέλεσης (sec)	Χρόνος Εκτέλεσης (sec)	Χρόνος Εκτέλεσης (sec)
14	283.22	15.14	2.01

Φυσικά, για να θεωρηθεί πιο καλά τεκμηριωμένη η τιμή του speed up, πρέπει να γίνουν περαιτέρω και πιο εκτεταμένα πειράματα.

5 Συμπεράσματα και Μελλοντική Δουλειά

Σ' αυτή την εργασία, υλοποιήθηκε μια παραλλαγή του TUC PreBLAST, η οποία κάνει χρήση φίλτρου BLOOM ώστε να πετύχει καλύτερο κατακερματισμό των w-mers του ερωτήματος στις μνήμες του προεπεξεργαστή. Αυτό έχει σαν αποτέλεσμα το αποτελεσματικότερο prefiltering της βάσης γενετικών δεδομένων, επομένως και τη μείωση του τελικού διαστήματος της βάσης προς επεξεργασία. Αυτό οδηγεί σε συνολικό speed up από 1.5 ως 4.5, ανεξαρτήτως του επεξεργαστή που θα επιλύσει τον αλγόριθμο BLAST επεξεργαζόμενος τις περιοχές ενδιαφέροντος που θα προκύψουν.

Επίσης, ενσωματώνει αυτή την παραλλαγή του TUC PreBLAST με τη χρήση του φίλτρου Bloom στο πολυεπεξεργαστικό σύστημα M.PL.EM., το οποίο εν τέλει τροφοδοτείται με τις περιοχές ενδιαφέροντος της βάσης γενετικών δεδομένων, έτσι ώστε να επεξεργαστεί μόνο αυτές και όχι ολόκληρη τη βάση γενετικών δεδομένων. Αυτό επιφέρει ένα speed up της τάξης του 7.5 σε σχέση με την πρώτη υλοποίηση του M.PL.EM. και της τάξης του 141 σε σχέση με τον κώδικα του M.PL.EM. που επιλύει τον αλγόριθμο BLAST και ο οποίος είναι μη βέλτιστος, αφού γράφτηκε για να τρέξει σε ενσωματωμένο επεξεργαστή. Τα speed up αυτά προκύπτουν με μικρό κόστος σε πόρους.

Ο TUC PreBLAST θα μπορούσε να χρησιμοποιηθεί και με οποιοδήποτε άλλο σύστημα το οποίο επιλύει τον αλγόριθμο BLAST. Μετά από την επεξεργασία της βάσης γενετικών δεδομένων, θα τροφοδοτήσει το σύστημα με τις περιοχές ενδιαφέροντος ώστε οι επεκτάσεις του 3ου βήματος του αλγορίθμου να γίνουν μόνο για τις περιοχές αυτές. Ωστόσο, η επιλογή του M.PL.EM. έγινε λόγω του ότι η πλατφόρμα αυτή είχε αναπτυχθεί στο Εργαστήριο Μικροεπεξεργαστών και Υλικού του Πολυτεχνείου Κρήτης.

Εκτός από την τελειοποίηση ενός πολύ καλού προεπεξεργαστή για το prefiltering της βάσης γενετικών δεδομένων επιτεύχθηκε και η δημιουργία ενός ενσωματωμένου συστήματος σε ολοκληρωμένο (system on a chip) το οποίο εκτελεί με ιδιαίτερα αποδοτικό τρόπο και τα 3 βήματα του αλγορίθμου BLAST.

Σαν μελλοντική δουλειά, προτείνεται μια λογική η οποία θα κάνει τη λειτουργία του συστήματος πιο δυναμική και ανεξάρτητη. Λογική, η οποία θα εξετάζει τις ιδιότητες του ερωτήματος αλλά και της βάσης γενετικών δεδομένων και βασιζόμενη στις ιδιότητες αυτές να επιλέγει κατάλληλη τιμή για το κατώφλι ώστε να μένουν εκτός κάποιες από τις αντιστοιχίες, οι οποίες δεν έχουν πολλές πιθανότητες να δώσουν κάποια επέκταση στο 3ο βήμα του αλγορίθμου BLAST.

Επίσης, προτείνεται η υλοποίηση μιας αρχιτεκτονικής, η οποία θα εξετάζει και πάλι κάποια από τα χαρακτηριστικά του ερωτήματος αλλά και της βάσης γενετικών δεδομένων ώστε να οδηγήσει το σύστημα prefiltering σε run-time reconfiguration, αναπροσαρμόζοντας τον τρόπο διευθυνσιοδότησης των διαθέσιμων μνημών ώστε να προκύψει ένα είτε πιο αποδοτικό configuration για το prefiltering είτε ένα configuration το οποίο δεν καταναλώνει άσκοπα τους πόρους της FPGA, οι οποίοι μπορούν να χρησιμοποιηθούν σε κάποιου άλλου είδους επεξεργασίας.

6 Παράρτημα Α

Στους πίνακες που ακολουθούν (Πίνακας 6-1 ως Πίνακας 6-53) εμφανίζονται τα πειραματικά δεδομένα της Μελέτης της συγκεκριμένης εργασίας. Επεξηγηματικά δίνονται ακολούθως κάποιοι όροι που χρησιμοποιούνται στους παρακάτω πίνακες:

- Size (characters): Μέγεθος της βάσης γενετικών δεδομένων και μέγεθος ερωτήματος σε χαρακτήρες.
- Window: Παράθυρο επεξεργασίας βάσης γενετικών δεδομένων (για ευκολία χρησιμοποιείται κάποιο ποσοστό του ερωτήματος).
- Percent of Query: Ο αριθμός των χαρακτήρων του ερωτήματος που τελικά χρησιμοποιείται για το δεδομένο παράθυρο.
- Space: Μέγεθος εξόδου του προεπεξεργαστή σε χαρακτήρες.
- Space/Database: Ποσοστό εξόδου προεπεξεργαστή ως προς το αρχικό μέγεθος της βάσης.

Πίνακας 6-1: Chr13, 2HF - 14 Addr Bits, Whole Query

Chr13, 2HF - 14 Addr Bits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	98717553	100,01%
Querry	chr13q	12775	20%	2555	98717551	100,01%
ncbi space	943085		30%	3832	98717553	100,01%
Possible Hits		73127168	40%	5110	98717551	100,01%
True positive hits		656217	50%	6387	98717548	100,01%
Hit Rate		0,90%	60%	7665	98717552	100,01%
			70%	8942	98717554	100,01%
			80%	10220	98717552	100,01%
			90%	11497	98717549	100,01%
			100%	12775	98717554	100,01%

Πίνακας 6-2: Chr13, 2HF - 14 Addr Bits, True positive hits, Whole Query

Chr13, 2HF - 14 Addr Bits, True positive hits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	98761427	100,06%
Query	chr13q	12775	20%	2555	98739991	100,04%
ncbi space	943085		30%	3832	98715251	100,01%
Possible Hits		656217	40%	5110	98715251	100,01%
True positive hits		656217	50%	6387	98715251	100,01%
Hit Rate		100,00%	60%	7665	98715251	100,01%
			70%	8942	98715251	100,01%
			80%	10220	98715251	100,01%
			90%	11497	98715251	100,01%
			100%	12775	98715251	100,01%

Πίνακας 6-3: Chr13, 4HF - 14 Addr Bits, Whole Query

Chr13, 4HF - 14 Addr Bits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	98717544	100,01%
Querry	chr13q	12775	20%	2555	98717544	100,01%
ncbi space	943085		30%	3832	98717544	100,01%
Possible Hits		15395477	40%	5110	98717544	100,01%
True positive hits		656217	50%	6387	98717544	100,01%
Hit Rate		4,26%	60%	7665	98717544	100,01%
			70%	8942	98717544	100,01%
			80%	10220	98717544	100,01%
			90%	11497	98717544	100,01%
			100%	12775	98717544	100,01%

Πίνακας 6-4: Chr13, 8HF - 16 Addr Bits, Whole Query

Chr13, 8HF - 16 Addr Bits, Whole Query						
		Size(characters)	Window	% of query	Space	Space/Database
Database	chr13	98704794	10%	1277	98753407	100,05%
Query	chr13q	12775	20%	2555	98729339	100,02%
ncbi space	943085		30%	3832	98716916	100,01%
Possible Hits		866826	40%	5110	98716916	100,01%
True positive hits		656217	50%	6387	98716916	100,01%
Hit Rate		75,70%	60%	7665	98716916	100,01%
			70%	8942	98716916	100,01%
			80%	10220	98716916	100,01%
			90%	11497	98716916	100,01%
			100%	12775	98716916	100,01%

Πίνακας 6-5: Chr13, 16HF - 14 Addr Bits, Whole Query

Chr13, 16HF - 14 Addr Bits, Whole Query						
		Size(characters)	Window	% of query	Space	Space/Database
Database	chr13	98704794	10%	1277	98717544	100,01%
Querry	chr13q	12775	20%	2555	98717544	100,01%
ncbi space	943085		30%	3832	98717544	100,01%
Possible Hits		15395477	40%	5110	98717544	100,01%
True positive hits		656217	50%	6387	98717544	100,01%
Hit Rate		4,26%	60%	7665	98717544	100,01%
			70%	8942	98717544	100,01%
			80%	10220	98717544	100,01%
			90%	11497	98717544	100,01%
			100%	12775	98717544	100,01%

Πίνακας 6-6: Chr13, 2HF - 14 Addr Bits, Split Query (500 characters)

Chr13, 2HF - 14 Addr Bits, Split Query (500 characters)						
		Size(characters)	Window	% of query	Space	Space/Database
Database	chr13	98704794	10%	1277	97985317	99,27%
Query	chr13q	500	20%	2555	98077273	99,36%
			30%	3832	98151907	99,44%
			40%	5110	98218598	99,51%
			50%	6387	98287270	99,58%
			60%	7665	98348800	99,64%
			70%	8942	98394122	99,69%
			80%	10220	98435151	99,73%
			90%	11497	98468205	99,76%
			100%	12775	98503696	99,80%

Πίνακας 6-7: Chr13, 2HF - 14 Addr Bits, True positive hits, Split Query (500 characters)

Chr13, 2HF - 14 Addr Bits, True positive hits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	39052579	39,57%
Query	chr13q	500	20%	2555	41718402	42,27%
			30%	3832	44223734	44,80%
			40%	5110	46639547	47,25%
			50%	6387	48947751	49,59%
			60%	7665	51127066	51,80%
			70%	8942	53250691	53,95%
			80%	10220	55243961	55,97%
			90%	11497	57177608	57,93%
			100%	12775	59030956	59,81%

Πίνακας 6-8: Chr13, 2HF - 14 Addr Bits, Split Query (250 characters)

Chr13, 2HF - 14 Addr Bits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	79265852	80,31%
Query	chr13q	250	20%	2555	91924429	93,13%
			30%	3832	95596002	96,85%
			40%	5110	96822080	98,09%
			50%	6387	97245403	98,52%
			60%	7665	97422294	98,70%
			70%	8942	97525491	98,81%
			80%	10220	97598780	98,88%
			90%	11497	97664322	98,95%
			100%	12775	97724841	99,01%

Πίνακας 6-9: Chr13, 2HF - 14 Addr Bits, True positive hits, Split Query (250 characters)

Chr13, 2HF - 14 Addr Bits, True positive hits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	23792525	24,10%
Query	chr13q	250	20%	2555	25541401	25,88%
			30%	3832	27170052	27,53%
			40%	5110	28814241	29,19%
			50%	6387	30426651	30,83%
			60%	7665	31940410	32,36%
			70%	8942	33469091	33,91%
			80%	10220	34911518	35,37%
			90%	11497	36371071	36,85%
			100%	12775	37800206	38,30%

Πίνακας 6-10: Chr13, 4HF - 14 Addr Bits, Split Query (1000 characters)

Chr13, 4HF - 14 Addr Bits, Split Query (1000 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	90724129	91,91%
Query	chr13q	1000	20%	2555	96351238	97,62%
			30%	3832	97725054	99,01%
			40%	5110	98170573	99,46%
			50%	6387	98337929	99,63%
			60%	7665	98428395	99,72%
			70%	8942	98487366	99,78%
			80%	10220	98529649	99,82%
			90%	11497	98563107	99,86%
			100%	12775	98588732	99,88%

Πίνακας 6-11: Chr13, 4HF - 14 Addr Bits, True positive hits, Split Query (1000 characters)

Chr13, 4HF - 14 Addr Bits, True positive hits, Split Query (1000 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	61080862	61,88%
Query	chr13q	1000	20%	2555	64614168	65,46%
			30%	3832	67774400	68,66%
			40%	5110	70659546	71,59%
			50%	6387	73261433	74,22%
			60%	7665	75617208	76,61%
			70%	8942	77790814	78,81%
			80%	10220	79742616	80,79%
			90%	11497	81531851	82,60%
			100%	12775	83188071	84,28%

Πίνακας 6-12: Chr13, 4HF - 14 Addr Bits, Split Query (500 characters)

Chr13, 4HF - 14 Addr Bits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	39849879	40,37%
Query	chr13q	500	20%	2555	43243229	43,81%
			30%	3832	46408626	47,02%
			40%	5110	49469740	50,12%
			50%	6387	52414620	53,10%
			60%	7665	55154801	55,88%
			70%	8942	57791191	58,55%
			80%	10220	60254824	61,05%
			90%	11497	62625972	63,45%
			100%	12775	64854406	65,71%

Πίνακας 6-13: Chr13, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)

Chr13, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	39052579	39,57%
Query	chr13q	500	20%	2555	41718402	42,27%
			30%	3832	44223734	44,80%
			40%	5110	46639547	47,25%
			50%	6387	48947751	49,59%
			60%	7665	51127066	51,80%
			70%	8942	53250691	53,95%
			80%	10220	55243961	55,97%
			90%	11497	57177608	57,93%
			100%	12775	59030956	59,81%

Πίνακας 6-14: Chr13, 4HF - 14 Addr Bits, Split Query (250 characters)

Chr13, 4HF - 14 Addr Bits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	23815513	24,13%
Query	chr13q	250	20%	2555	25587660	25,92%
			30%	3832	27235777	27,59%
			40%	5110	28896595	29,28%
			50%	6387	30527357	30,93%
			60%	7665	32060304	32,48%
			70%	8942	33605773	34,05%
			80%	10220	35065816	35,53%
			90%	11497	36539041	37,02%
			100%	12775	37981804	38,48%

Πίνακας 6-15: Chr13, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters)

Chr13, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	23792525	24,10%
Query	chr13q	250	20%	2555	25541401	25,88%
			30%	3832	27170052	27,53%
			40%	5110	28814241	29,19%
			50%	6387	30426651	30,83%
			60%	7665	31940410	32,36%
			70%	8942	33469091	33,91%
			80%	10220	34911518	35,37%
			90%	11497	36371071	36,85%
			100%	12775	37800206	38,30%

Πίνακας 6-16: Chr13, 8HF - 16 Addr Bits, Split Query (2000 characters)

Chr13, 8HF - 16 Addr Bits, Split Query (2000 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	83757975	84,86%
Query	chr13q	2000	20%	2555	87017518	88,16%
			30%	3832	89475377	90,65%
			40%	5110	91429891	92,63%
			50%	6387	92967477	94,19%
			60%	7665	94147557	95,38%
			70%	8942	95058928	96,31%
			80%	10220	95842731	97,10%
			90%	11497	96422065	97,69%
			100%	12775	96834466	98,11%

Πίνακας 6-17: Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (2000 characters)

Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (2000 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	83757975	84,86%
Query	chr13q	2000	20%	2555	87017518	88,16%
			30%	3832	89475377	90,65%
			40%	5110	91429891	92,63%
			50%	6387	92967477	94,19%
			60%	7665	94147557	95,38%
			70%	8942	95058928	96,31%
			80%	10220	95812567	97,07%
			90%	11497	96391901	97,66%
			100%	12775	96864630	98,14%

Πίνακας 6-18: Chr13, 8HF - 16 Addr Bits, Split Query (1000 characters)

Chr13, 8HF - 16 Addr Bits, Split Query (1000 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	61080862	61,88%
Query	chr13q	1000	20%	2555	64614168	65,46%
			30%	3832	67774400	68,66%
			40%	5110	70659546	71,59%
			50%	6387	73261433	74,22%
			60%	7665	75617208	76,61%
			70%	8942	77790814	78,81%
			80%	10220	79742616	80,79%
			90%	11497	81531851	82,60%
			100%	12775	83188071	84,28%

Πίνακας 6-19: Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (1000 characters)

Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (1000 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	61080862	61,88%
Query	chr13q	1000	20%	2555	64614168	65,46%
			30%	3832	67774400	68,66%
			40%	5110	70659546	71,59%
			50%	6387	73261433	74,22%
			60%	7665	75617208	76,61%
			70%	8942	77790814	78,81%
			80%	10220	79742616	80,79%
			90%	11497	81531851	82,60%
			100%	12775	83188071	84,28%

Πίνακας 6-20: Chr13, 8HF - 16 Addr Bits, Split Query (500 characters)

Chr13, 8HF - 16 Addr Bits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	39052579	39,57%
Query	chr13q	500	20%	2555	41718402	42,27%
			30%	3832	44223734	44,80%
			40%	5110	46639547	47,25%
			50%	6387	48947751	49,59%
			60%	7665	51127066	51,80%
			70%	8942	53250691	53,95%
			80%	10220	55243961	55,97%
			90%	11497	57177608	57,93%
			100%	12775	59030956	59,81%

Πίνακας 6-21: Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (500 characters)

Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	39052579	39,57%
Query	chr13q	500	20%	2555	41718402	42,27%
			30%	3832	44223734	44,80%
			40%	5110	46639547	47,25%
			50%	6387	48947751	49,59%
			60%	7665	51127066	51,80%
			70%	8942	53250691	53,95%
			80%	10220	55243961	55,97%
			90%	11497	57177608	57,93%
			100%	12775	59030956	59,81%

Πίνακας 6-22: Chr13, 8HF - 16 Addr Bits, Split Query (250 characters)

Chr13, 8HF - 16 Addr Bits, Split Query (250 characters)						
		Size(characters)	Window	% of query	Space	Space/Database
Database	chr13	98704794	10%	1277	23792525	24,10%
Query	chr13q	250	20%	2555	25541401	25,88%
			30%	3832	27170052	27,53%
			40%	5110	28814241	29,19%
			50%	6387	30426651	30,83%
			60%	7665	31940410	32,36%
			70%	8942	33469091	33,91%
			80%	10220	34911518	35,37%
			90%	11497	36371071	36,85%
			100%	12775	37800206	38,30%

Πίνακας 6-23: Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (250 characters)

Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (250 characters)						
		Size(characters)	Window	% of query	Space	Space/Database
Database	chr13	98704794	10%	1277	23792525	24,10%
Query	chr13q	250	20%	2555	25541401	25,88%
			30%	3832	27170052	27,53%
			40%	5110	28814241	29,19%
			50%	6387	30426651	30,83%
			60%	7665	31940410	32,36%
			70%	8942	33469091	33,91%
			80%	10220	34911518	35,37%
			90%	11497	36371071	36,85%
			100%	12775	37800206	38,30%

Πίνακας 6-24: Chr13, 16HF - 14 Addr Bits, Split Query (1000 characters)

Chr13, 16HF - 14 Addr Bits, Split Query (1000 characters)						
		Size(characters)	Window	% of query	Space	Space/Database
Database	chr13	98704794	10%	1277	61080862	61,88%
Query	chr13q	1000	20%	2555	64614168	65,46%
			30%	3832	67774400	68,66%
			40%	5110	70659546	71,59%
			50%	6387	73261433	74,22%
			60%	7665	75617208	76,61%
			70%	8942	77790814	78,81%
			80%	10220	79742616	80,79%
			90%	11497	81531851	82,60%
			100%	12775	83188071	84,28%

Πίνακας 6-25: Chr13, 16HF - 14 Addr Bits, True positive hits, Split Query (1000 characters)

Chr13, 16HF - 14 Addr Bits, True positive hits, Split Query (1000 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	61080862	61,88%
Query	chr13q	1000	20%	2555	64614168	65,46%
			30%	3832	67774400	68,66%
			40%	5110	70659546	71,59%
			50%	6387	73261433	74,22%
			60%	7665	75617208	76,61%
			70%	8942	77790814	78,81%
			80%	10220	79742616	80,79%
			90%	11497	81531851	82,60%
			100%	12775	83188071	84,28%

Πίνακας 6-26: Chr13, 16HF - 14 Addr Bits, Split Query (500 characters)

Chr13, 16HF - 14 Addr Bits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	39052579	39,57%
Query	chr13q	500	20%	2555	41718402	42,27%
			30%	3832	44223734	44,80%
			40%	5110	46639547	47,25%
			50%	6387	48947751	49,59%
			60%	7665	51127066	51,80%
			70%	8942	53250691	53,95%
			80%	10220	55243961	55,97%
			90%	11497	57177608	57,93%
			100%	12775	59030956	59,81%

Πίνακας 6-27: Chr13, 16HF - 14 Addr Bits, True positive hits, Split Query (500 characters)

Chr13, 16HF - 14 Addr Bits, True positive hits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	39052579	39,57%
Query	chr13q	500	20%	2555	41718402	42,27%
			30%	3832	44223734	44,80%
			40%	5110	46639547	47,25%
			50%	6387	48947751	49,59%
			60%	7665	51127066	51,80%
			70%	8942	53250691	53,95%
			80%	10220	55243961	55,97%
			90%	11497	57177608	57,93%
			100%	12775	59030956	59,81%

Πίνακας 6-28: Chr13, 16HF - 14 Addr Bits, Split Query (250 characters)

Chr13, 16HF - 14 Addr Bits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	23792525	24,10%
Query	chr13q	250	20%	2555	25541401	25,88%
			30%	3832	27170052	27,53%
			40%	5110	28814241	29,19%
			50%	6387	30426651	30,83%
			60%	7665	31940410	32,36%
			70%	8942	33469091	33,91%
			80%	10220	34911518	35,37%
			90%	11497	36371071	36,85%
			100%	12775	37800206	38,30%

Πίνακας 6-29: Chr13, 16HF - 14 Addr Bits, True positive hits, Split Query (250 characters)

Chr13, 16HF - 14 Addr Bits, True positive hits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr13	98704794	10%	1277	23792525	24,10%
Query	chr13q	250	20%	2555	25541401	25,88%
			30%	3832	27170052	27,53%
			40%	5110	28814241	29,19%
			50%	6387	30426651	30,83%
			60%	7665	31940410	32,36%
			70%	8942	33469091	33,91%
			80%	10220	34911518	35,37%
			90%	11497	36371071	36,85%
			100%	12775	37800206	38,30%

Πίνακας 6-30: Chr12, 4HF - 14 Addr Bits, Whole Query

Chr12, 4HF - 14 Addr Bits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr12	135675203	10%	316	135678316	100,00%
Querry	chr12q	3162	20%	632	135678316	100,00%
ncbi space	943085		30%	948	135678316	100,00%
Possible Hits		5666000	40%	1264	135678316	100,00%
True positive hits		97850	50%	1581	135678316	100,00%
Hit Rate		1,73%	60%	1897	135678316	100,00%
			70%	2213	135678316	100,00%
			80%	2529	135678316	100,00%
			90%	2845	135678316	100,00%
			100%	3162	135678316	100,00%

Πίνακας 6-31: Chr12, 4HF - 14 Addr Bits, True positive hits, Whole Query

Chr12, 4HF - 14 Addr Bits, True positive hits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr12	135675203	10%	316	46448436	34,24%
Query	chr12q	3162	20%	632	52575364	38,75%
ncbi space			30%	948	58200479	42,90%
Possible Hits		97850	40%	1264	63246079	46,62%
True positive hits		97850	50%	1581	67899356	50,05%
Hit Rate		100,00%	60%	1897	72220386	53,23%
			70%	2213	76233315	56,19%
			80%	2529	80007610	58,97%
			90%	2845	83632631	61,64%
			100%	3162	87019300	64,14%

Πίνακας 6-32: Chr12, 4HF - 14 Addr Bits, Split Query (500 characters)

Chr12, 4HF - 14 Addr Bits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr12	135675203	10%	316	8407951	6,20%
Query	chr12q	500	20%	632	9491576	7,00%
			30%	948	10587902	7,80%
			40%	1264	11678678	8,61%
			50%	1581	12769952	9,41%
			60%	1897	13882073	10,23%
			70%	2213	15064229	11,10%
			80%	2529	16188223	11,93%
			90%	2845	17330642	12,77%
			100%	3162	18513302	13,65%

Πίνακας 6-33: Chr12, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)

Chr12, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr12	135675203	10%	316	8072769	5,95%
Query	chr12q	500	20%	632	8826615	6,51%
			30%	948	9569919	7,05%
			40%	1264	10306745	7,60%
			50%	1581	11040179	8,14%
			60%	1897	11761320	8,67%
			70%	2213	12495876	9,21%
			80%	2529	13207940	9,73%
			90%	2845	13932783	10,27%
			100%	3162	14653229	10,80%

Πίνακας 6-34: Chr12, 4HF - 14 Addr Bits, Split Query (250 characters)

Chr12, 4HF - 14 Addr Bits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr12	135675203	10%	316	4289652	3,16%
Query	chr12q	250	20%	632	4700776	3,46%
			30%	948	5085141	3,75%
			40%	1264	5482576	4,04%
			50%	1581	5877846	4,33%
			60%	1897	6264990	4,62%
			70%	2213	6660399	4,91%
			80%	2529	7040757	5,19%
			90%	2845	7435551	5,48%
			100%	3162	7827934	5,77%

Πίνακας 6-35: Chr12, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters)

Chr12, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr12	135675203	10%	316	4278372	3,15%
Query	chr12q	250	20%	632	4680399	3,45%
			30%	948	5059172	3,73%
			40%	1264	5452016	4,02%
			50%	1581	5841937	4,31%
			60%	1897	6222170	4,59%
			70%	2213	6608906	4,87%
			80%	2529	6981842	5,15%
			90%	2845	7367580	5,43%
			100%	3162	7752522	5,71%

Πίνακας 6-36: Chr14, 4HF - 14 Addr Bits, Whole Query

Chr14, 4HF - 14 Addr Bits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr14	90582208	10%	551	90587697	100,01%
Querry	chr14q	5511	20%	1102	90587697	100,01%
ncbi space			30%	1653	90587697	100,01%
Possible Hits		14850586	40%	2204	90587697	100,01%
True positive hits		1514674	50%	2755	90587697	100,01%
Hit Rate		10,20%	60%	3306	90587697	100,01%
			70%	3857	90587696	100,01%
			80%	4408	90587696	100,01%
			90%	4959	90587697	100,01%
			100%	5511	90587697	100,01%

Πίνακας 6-37: Chr14, 4HF - 14 Addr Bits, True positive hits, Whole Query

Chr14, 4HF - 14 Addr Bits, True positive hits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr14	90582208	10%	551	88041949	97,20%
Query	chr14q	5511	20%	1102	89155177	98,42%
ncbi space			30%	1653	89739297	99,07%
Possible Hits		1514674	40%	2204	90051376	99,41%
True positive hits		1514674	50%	2755	90235114	99,62%
Hit Rate		100,00%	60%	3306	90355211	99,75%
			70%	3857	90443226	99,85%
			80%	4408	90492685	99,90%
			90%	4959	90522985	99,93%
			100%	5511	90544661	99,96%

Πίνακας 6-38: Chr14, 4HF - 14 Addr Bits, Split Query (500 characters)

Chr14, 4HF - 14 Addr Bits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr14	90582208	10%	551	32438993	35,81%
Query	chr14q	500	20%	1102	34541181	38,13%
			30%	1653	36508159	40,30%
			40%	2204	38417628	42,41%
			50%	2755	40279756	44,47%
			60%	3306	42048123	46,42%
			70%	3857	43790853	48,34%
			80%	4408	45442547	50,17%
			90%	4959	47053674	51,95%
			100%	5511	48642754	53,70%

Πίνακας 6-39: Chr14, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)

Chr14, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr14	90582208	10%	551	32131160	35,47%
Query	chr14q	500	20%	1102	33963769	37,49%
			30%	1653	35679957	39,39%
			40%	2204	37355540	41,24%
			50%	2755	38975727	43,03%
			60%	3306	40496939	44,71%
			70%	3857	41989924	46,36%
			80%	4408	43410138	47,92%
			90%	4959	44800880	49,46%
			100%	5511	46149040	50,95%

Πίνακας 6-40: Chr14, 4HF - 14 Addr Bits, Split Query (250 characters)

Chr14, 4HF - 14 Addr Bits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr14	90582208	10%	551	21337658	23,56%
Query	chr14q	250	20%	1102	22542572	24,89%
			30%	1653	23664070	26,12%
			40%	2204	24799433	27,38%
			50%	2755	25907269	28,60%
			60%	3306	26951892	29,75%
			70%	3857	28009265	30,92%
			80%	4408	29009925	32,03%
			90%	4959	30024874	33,15%
			100%	5511	31023497	34,25%

Πίνακας 6-41: Chr14, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters)

Chr14, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr14	90582208	10%	551	21327701	23,55%
Query	chr14q	250	20%	1102	22525644	24,87%
			30%	1653	23640035	26,10%
			40%	2204	24770258	27,35%
			50%	2755	25871566	28,56%
			60%	3306	26911126	29,71%
			70%	3857	27963227	30,87%
			80%	4408	28958965	31,97%
			90%	4959	29968877	33,08%
			100%	5511	30959827	34,18%

Πίνακας 6-42: Chr17, 4HF - 14 Addr Bits, Whole Query

Chr17, 4HF - 14 Addr Bits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr17	81665014	10%	395	81672312	100,01%
Querry	chr17q	3959	20%	791	81672708	100,01%
ncbi space	943085		30%	1187	81668950	100,00%
Possible Hits		7267054	40%	1583	81668950	100,00%
True positive hits		1532071	50%	1979	81668950	100,00%
Hit Rate		21,08%	60%	2375	81668950	100,00%
			70%	2771	81668950	100,00%
			80%	3167	81668950	100,00%
			90%	3563	81668950	100,00%
			100%	3959	81668950	100,00%

Πίνακας 6-43: Chr17, 4HF - 14 Addr Bits, True positive hits, Whole Query

Chr17, 4HF - 14 Addr Bits, True positive hits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr17	81665014	10%	316	76494919	93,67%
Query	chr17q	3959	20%	632	77815176	95,29%
ncbi space			30%	948	78712636	96,38%
Possible Hits		1514674	40%	1264	79364938	97,18%
True positive hits		1514674	50%	1581	79845592	97,77%
Hit Rate		100,00%	60%	1897	80235295	98,25%
			70%	2213	80514495	98,59%
			80%	2529	80741327	98,87%
			90%	2845	80914970	99,08%
			100%	3162	81048136	99,24%

Πίνακας 6-44: Chr17, 4HF - 14 Addr Bits, Split Query (500 characters)

Chr17, 4HF - 14 Addr Bits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr17	81665014	10%	395	37009168	45,32%
Query	chr17q	500	20%	791	38903065	47,64%
			30%	1187	40622429	49,74%
			40%	1583	42295266	51,79%
			50%	1979	43848978	53,69%
			60%	2375	45314548	55,49%
			70%	2771	46754056	57,25%
			80%	3167	48079902	58,87%
			90%	3563	49371674	60,46%
			100%	3959	50594171	61,95%

Πίνακας 6-45: Chr17, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)

Chr17, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr17	81665014	10%	316	36852546	45,13%
Query	chr17q	500	20%	632	38598003	47,26%
			30%	948	40190294	49,21%
			40%	1264	41719642	51,09%
			50%	1581	43153521	52,84%
			60%	1897	44495821	54,49%
			70%	2213	45797895	56,08%
			80%	2529	47009671	57,56%
			90%	2845	48186035	59,00%
			100%	3162	49301552	60,37%

Πίνακας 6-46: Chr17, 4HF - 14 Addr Bits, Split Query (250 characters)

Chr17, 4HF - 14 Addr Bits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr17	81665014	10%	395	25991108	31,83%
Query	chr17q	250	20%	791	27272419	33,40%
			30%	1187	28432706	34,82%
			40%	1583	29578553	36,22%
			50%	1979	30682630	37,57%
			60%	2375	31721989	38,84%
			70%	2771	32769985	40,13%
			80%	3167	33746411	41,32%
			90%	3563	34730678	42,53%
			100%	3959	35687048	43,70%

Πίνακας 6-47: Chr17, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters)

Chr17, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chr17	81665014	10%	316	25987730	31,82%
Query	chr17q	250	20%	632	27265630	33,39%
			30%	948	28422901	34,80%
			40%	1264	29565770	36,20%
			50%	1581	30667161	37,55%
			60%	1897	31704244	38,82%
			70%	2213	32750448	40,10%
			80%	2529	33725139	41,30%
			90%	2845	34706701	42,50%
			100%	3162	35661069	43,67%

Πίνακας 6-48: ChrX, 4HF - 14 Addr Bits, Whole Query

ChrX, 4HF - 14 Addr Bits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chrX	150212081	10%	165	145163090	96,64%
Querry	chrXq	1658	20%	331	147937029	98,49%
ncbi space	943085		30%	497	148770321	99,04%
Possible Hits		836287	40%	663	149198494	99,33%
True positive hits		23605	50%	829	149469804	99,51%
Hit Rate		2,82%	60%	994	149654896	99,63%
			70%	1160	149796530	99,72%
			80%	1326	149907555	99,80%
			90%	1492	149995391	99,86%
			100%	1658	150067404	99,90%

Πίνακας 6-49: ChrX, 4HF - 14 Addr Bits, True positive hits, Whole Query

ChrX, 4HF - 14 Addr Bits, True positive hits, Whole Query						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chrX	150212081	10%	165	8663220	5,77%
Querry	chrXq	1658	20%	331	9665136	6,43%
ncbi space	943085		30%	497	10674843	7,11%
Possible Hits		23605	40%	663	11661045	7,76%
True positive hits		23605	50%	829	12615746	8,40%
Hit Rate		100,00%	60%	994	13587267	9,05%
			70%	1160	14545964	9,68%
			80%	1326	15510556	10,33%
			90%	1492	16492746	10,98%
			100%	1658	17477613	11,64%

Πίνακας 6-50: ChrX, 4HF - 14 Addr Bits, Split Query (500 characters)

ChrX, 4HF - 14 Addr Bits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chrX	150212081	10%	165	2586281	1,72%
Query	chrXq	500	20%	331	2960050	1,97%
			30%	497	3333462	2,22%
			40%	663	3733528	2,49%
			50%	829	4108298	2,73%
			60%	994	4494423	2,99%
			70%	1160	4894276	3,26%
			80%	1326	5293689	3,52%
			90%	1492	5703596	3,80%
			100%	1658	6106720	4,07%

Πίνακας 6-51: ChrX, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)

ChrX, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chrX	150212081	10%	165	2454098	1,63%
Query	chrXq	500	20%	331	2706996	1,80%
			30%	497	2940341	1,96%
			40%	663	3172290	2,11%
			50%	829	3404856	2,27%
			60%	994	3630976	2,42%
			70%	1160	3865275	2,57%
			80%	1326	4098126	2,73%
			90%	1492	4329963	2,88%
			100%	1658	4560506	3,04%

Πίνακας 6-52: ChrX, 4HF - 14 Addr Bits, Actual Split Query (250 characters)

ChrX, 4HF - 14 Addr Bits, Actual Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chrX	150212081	10%	165	1317875	0,88%
Query	chrXq	250	20%	331	1448598	0,96%
			30%	497	1569257	1,04%
			40%	663	1694710	1,13%
			50%	829	1818735	1,21%
			60%	994	1939287	1,29%
			70%	1160	2062899	1,37%
			80%	1326	2182690	1,45%
			90%	1492	2305528	1,53%
			100%	1658	2429322	1,62%

Πίνακας 6-53: ChrX, 4HF - 14 Addr Bits, Actual True positive hits, Split Query (250 characters)

ChrX, 4HF - 14 Addr Bits, Actual True positive hits, Split Query (250 characters)						
		Size(characters)	**Window**	**% of query**	**Space**	**Space/Database**
Database	chrX	150212081	10%	165	1316192	0,88%
Query	chrXq	250	20%	331	1445681	0,96%
			30%	497	1566066	1,04%
			40%	663	1690106	1,13%
			50%	829	1812635	1,21%
			60%	994	1930804	1,29%
			70%	1160	2052073	1,37%
			80%	1326	2170865	1,45%
			90%	1492	2292289	1,53%
			100%	1658	2414615	1,61%

Αναφορές

[1] S. B. Needleman and C. D. Wunsch, "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins," *J. Mol. Biol.*, vol. 48, pp. 443-453, 1970.

[2] T. F. Smith and M. S. Waterman, "Identification of Common Molecular Subsequences," *J. Mol. Biol.*, vol. 48, pp. 195-197, 1981.

[3] W. Pearson and D. Lipman, "Improoved tools for biological sequence analysis," *Proc. Natl. Acad. Sci. USA*, vol. 85, pp. 2444-2448, 1988.

[4] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic Local Alignment Search Tool," *J. Mol. Biol.*, vol. 215, pp. 403-410, 1990.

[5] P. G. Higgs and T. K. Attwood, "Bioninformatics And Molecular Evolution," 2005.

[6] NCBI. [Online]. http://www.ncbi.nih.gov

[7] C. P. Sosa. Some Practical Suggestions for Performing NCBI BLAST Benchmarks on a pSeries 690 System. [Online]. http://www.redbooks.ibm.com/abstracts/redp0437.html?

[8] R. Radahakrishnan, R. Ali, R. Konchhar, K. Chadalavada, and R. Rajagopalan, "Performance Characterization of BLAST on 32-bit and 64-bit Dell PowerEdge Servers," *Dell Power Solutions*, Feb. 2005.

[9] A. B. f. P. G4/G5. [Online]. http://images.apple.com/acg/pdf/AGBLAST229-PerfData22Jun04.pdf

[10] D. Hoang et al., "FPGA Implementation of Systolic Sequence Alignment," in *2nd International Wokrshop on Field-Programmable Logic and Application, Lecture Notes in Computer Science*, vol. 705, 1992, pp. 183-191.

[11] D. Hoang et al., "Searching Genetic Databases on Splash 2," in *IEEE Workshop on FPGAs for Custom Computing Machines (FCCM)*, 1993, pp. 185-191.

[12] S. Guccione and E. Keller, "Gene Matching Using Jbits," in *12th International Conference on Field Programmable Logic and Applications, Lecture Notes in Computer Science*, vol. 2438, 2002, pp. 1168-1171.

[13] K. Puttegowda et al., "A Run-Time Reconfigurable System for Gene-Sequence Searching," in *16th International Conference on VLSI Design*, New Delhi, 2003, pp. 561-566.

[14] T. Oliver, B. Schmidt, and D. Maskel, "Reconfigurable Architecture for Bio-sequence Database Scanning on FPGAs," in *IEEE Trans. Circuits Syst. II*, vol. 52, 2005, pp. 851-

855.

[15] K. Muriki, K. Underwood, and R. Sass, "RC-BLAST: Towards an open source hardware implementation," in *International Workshop on High Performance Computational Biology*, Denver, 2005.

[16] M. C. Herbordt, J. Model, Y. Gu, B. Sukhwani, and T. VanCourt, "Single Pass, BLAST-Like, Approximate String Matching on FPGAs," in *14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM '06)*, 2006, pp. 217-226.

[17] P. Krishnamurthy, et al., "Biosequence Similarity Search on the Mercury System," in *IEEE 15th Int'l Conf. on Application-Specific Systems, Architecture and Processors*, September 2004, pp. 365-375.

[18] J. Lancaster, J. Buhler, and R. Chamberlain, "Acceleration of Ungapped Extension in Mercury BLAST," in *7th workshop on media and streaming processors*, Barcelona, November 12, 2005.

[19] D. Lavenier, L. Xinchun, and G. Georges, "Seed-based Genomic Sequence Comparison using a FPGA/FLASH Accelerator," in *IEEE International Conference on Field Programmable Technology (FPT '06)*, 2006, pp. 41-48.

[20] DeCypher. [Online]. http://www.timelogic.com/benchmark_blast.html

[21] C. Chang, "BLAST Implementation on BEE2," 2004.

[22] E. Sotiriades and A. Dollas, "A General Reconfigurable Architecture for the BLAST Algorithm," *The Journal of VLSI Signal Processing*, vol. 48, pp. 189-208, 2007.

[23] E. Sotiriades, C. Kozanitis, G. Chrysos, and A. Dollas, "Rapid Prototyping of a System-on-a-Chip for the BLAST Algorithm Implementation," in *17th International IEEE Workshop on Rapid System Prototyping RSP 2006, Computer Society Press*, Chania, Greece, 14-16 June 2006, pp. 223-229.

[24] E. Sotiriades, C. Kozanitis, and A. Dollas, "FPGA Based Architecture of DNA Sequence Comparison and Database Search," in *20th International Parallel and Distributed Processing Symposium, IPDPS 2006, at the 13th Reconfigurable Architectures Workshop*, Rhodes, Greece, 25-29 April, 2006, p. 193.

[25] E. Sotiriades, C. Kozanitis, and A. Dollas, "Some Initial Results on Hardware BLAST Acceleration with a Reconfigurable Architecture," in *20th International Parallel and Distributed Processing Symposium, IPDPS 2006, at the 5th IEEE International Workshop on High Performance Computational Biology (HiCOMB2006)*, Rhodes, Greece, 25-29 April, 2006, p. 251.

[26] P. Afratis, E. Sotiriades, G. Chrysos, S. Fytraki, and D. Pnevmatikatos, "A Rate-based Prefiltering Approach to BLAST Acceleration," *poster in International Conference on*

Field Programmable Logic and Applications (FPL '08).

[27] P. Afratis, E. Sotiriades, G. Chrysos, S. Fytraki, and D. Pnevmatikatos, "Preprocessor for BLAST Algorithm Data," *2nd Greek ECE Student Conference*, p. 12, 2008.

[28] E. Sotiriades, et al., "Design Space Exploration of Reconfigurable Acceleration for tha BLAST Algorithm," *Submitted at ACM Transactions on Embedded Computing Systems (TECS)*.

[29] B. H. Bloom, "Space/Time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422-426, 1970.

[30] G. Mplemenos and I. Papaefstathiou, "Building Large Multiprocessor Systems in State-of-the-art FPGAs".

[31] G. Mplemenos and I. Papaefstathiou, "MPLEM: An 80-processor FPGA Based Multiprocessor System".

[32] Cypress, *CY7C1041 256K x 16 Static RAM*.

Ευρετήριο Σχημάτων

Σχήμα 1-1: Το πρώτο βήμα του αλγορίθμου BLAST 9
Σχήμα 1-2: Το δεύτερο βήμα του αλγορίθμου BLAST 9
Σχήμα 1-3: Το τρίτο βήμα του αλγορίθμου BLAST 10
Σχήμα 1-4: Datapath αρχιτεκτονικής TUC PreBLAST 16
Σχήμα 1-5: Control Path της αρχιτεκτονικής TUC PreBLAST 18
Σχήμα 1-6: Λειτουργία φίλτρου Bloom, αρχικοποίηση (αριστερά), κωδικοποίηση λέξης w (μέση), ελεγχος ύπαρξης λέξης w (δεξιά) 20
Σχήμα 1-7: Block Diagram πολυεπεξεργαστικού συστήματος 25
Σχήμα 2-1: Διαμερισμένο φίλτρο Bloom 29
Σχήμα 2-2: Chr13, 2HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space 36
Σχήμα 2-3: Chr13, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space 37
Σχήμα 2-4: Chr13, 8HF - 16 Addr Bits, Minimum Space vs. Bloom Filter Space 39
Σχήμα 2-5: Chr13, 16HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space 41
Σχήμα 2-6: Bloom Database Space for Different Memory Utilization Configurations 42
Σχήμα 2-7 Chr12, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space 44
Σχήμα 2-8: Chr14, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space 45
Σχήμα 2-9: Chr17, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space 46
Σχήμα 2-10 ChrX, 4HF - 14 Addr Bits, Minimum Space vs. Bloom Filter Space 47
Σχήμα 3-1: Το datapath της αρχιτεκτονικής TUC PreBLAST με φίλτρο Bloom και μνήμες RAM μιας εισόδου 50
Σχήμα 3-2: Το datapath της αρχιτεκτονικής TUC PreBLAST με φίλτρο Bloom και μνήμες RAM δύο θυρών 50
Σχήμα 3-3: Η νέα FSM του SRAM Controller 52
Σχήμα 3-4: Διασύνδεση SRAM Controller και TUC PreBLAST 54
Σχήμα 3-5: Νέο Controll Path του TUC PreBLAST 54
Σχήμα 4-1: 3-D Αναπαράσταση των Time-Load Scenarios και των Speed Up 62

Ευρετήριο Πινάκων

Πίνακας 4-1: Κατανεμημένοι Πόροι για τον TUC PreBLAST με χρήση Φίλτρων BLOOM με μνήμες RAM μονής θύρας....55
Πίνακας 4-2: Κατανεμημένοι Πόροι για τον TUC PreBLAST με χρήση Φίλτρων BLOOM με μνήμες RAM δύο θυρών....56
Πίνακας 4-3: Ρολόι Συστήματος και Throughput του TUC PreBLAST με χρήση Φίλτρων BLOOM με μνήμες RAM μονής θύρας....56
Πίνακας 4-4: Ρολόι Συστήματος και Throughput του TUC PreBLAST με χρήση Φίλτρων BLOOM με μνήμες RAM δύο θυρών....56
Πίνακας 4-5: Χρόνοι Εκτέλεση NCBI BLAST και TUC PreBLAST για ένα προεπεξεργαστή με μνήμες RAM μονής θύρας....57
Πίνακας 4-6: Χρόνοι Εκτέλεση NCBI BLAST και TUC PreBLAST για 108 προεπεξεργαστές με μνήμες RAM μονής θύρας....57
Πίνακας 4-7: Χρόνοι Εκτέλεση NCBI BLAST και TUC PreBLAST για 2 προεπεξεργαστές με μνήμες RAM δύο θυρών....58
Πίνακας 4-8: Χρόνοι Εκτέλεση NCBI BLAST και TUC PreBLAST για 162 προεπεξεργαστές με μνήμες RAM δύο θυρών....58
Πίνακας 4-9: Σύγκριση πόρων και αποδόσεων των δύο αρχιτεκτονικών με μνήμες RAM μονής θύρας....59
Πίνακας 4-10: Σύγκριση πόρων και αποδόσεων των δύο αρχιτεκτονικών με μνήμες RAM δύο θυρών....59
Πίνακας 4-11: Σύγκριση Τελικών Διαστημάτων των Περιοχών Ενδιαφέροντος που εξάγουν οι δύο αρχιτεκτονικές του TUC PreBLAST με και χωρίς το φίλτρο Bloom....60
Πίνακας 4-12: Συνολικός Χρόνος Εκτέλεσης NCBI BLAST και TUC PreBLAST με χρήση φίλτρου Bloom....61
Πίνακας 4-13: Speed Up NCBI BLAST και TUC PreBLAST με φίλτρου Bloom....61
Πίνακας 4-14: Speed Up για τις Απαιτητικές Περιπτώσεις Χρωμοσωμάτων (1 prefiltering unit)....63
Πίνακας 4-15: Speed Up για τις Απαιτητικές Περιπτώσεις Χρωμοσωμάτων (81 prefiltering unit)....64
Πίνακας 4-16: Κατανεμημένοι Πόροι για πολυεπεξεργαστικό σύστημα M.PL.EM. και τον TUC PreBLAST με χρήση Φίλτρων BLOOM (1 μονάδα prefiltering)....65
Πίνακας 4-17: Ρολόι Συστήματος και Throughput του πολυεπεξεργαστικού συστήματος M.PL.EM. και του TUC PreBLAST με χρήση Φίλτρων BLOOM (1 μονάδα prefiltering)....65
Πίνακας 4-18: Σύγκριση των πόρων των δύο αρχιτεκτονικών....66
Πίνακας 4-19: Σύγκριση του χρόνου εκτέλεσης των δύο αρχιτεκτονικών....66
Πίνακας 6-1: Chr13, 2HF - 14 Addr Bits, Whole Query....70
Πίνακας 6-2: Chr13, 2HF - 14 Addr Bits, True positive hits, Whole Query....70
Πίνακας 6-3: Chr13, 4HF - 14 Addr Bits, Whole Query....70
Πίνακας 6-4: Chr13, 8HF - 16 Addr Bits, Whole Query....71
Πίνακας 6-5: Chr13, 16HF - 14 Addr Bits, Whole Query....71
Πίνακας 6-6: Chr13, 2HF - 14 Addr Bits, Split Query (500 characters)....71
Πίνακας 6-7: Chr13, 2HF - 14 Addr Bits, True positive hits, Split Query (500 characters)....72
Πίνακας 6-8: Chr13, 2HF - 14 Addr Bits, Split Query (250 characters)....72

Πίνακας 6-9: Chr13, 2HF - 14 Addr Bits, True positive hits, Split Query (250 characters) 72
Πίνακας 6-10: Chr13, 4HF - 14 Addr Bits, Split Query (1000 characters) 73
Πίνακας 6-11: Chr13, 4HF - 14 Addr Bits, True positive hits, Split Query (1000 characters) 73
Πίνακας 6-12: Chr13, 4HF - 14 Addr Bits, Split Query (500 characters) 73
Πίνακας 6-13: Chr13, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters) 74
Πίνακας 6-14: Chr13, 4HF - 14 Addr Bits, Split Query (250 characters) 74
Πίνακας 6-15: Chr13, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters) 74
Πίνακας 6-16: Chr13, 8HF - 16 Addr Bits, Split Query (2000 characters) 75
Πίνακας 6-17: Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (2000 characters) 75
Πίνακας 6-18: Chr13, 8HF - 16 Addr Bits, Split Query (1000 characters) 75
Πίνακας 6-19: Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (1000 characters) 76
Πίνακας 6-20: Chr13, 8HF - 16 Addr Bits, Split Query (500 characters) 76
Πίνακας 6-21: Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (500 characters) 76
Πίνακας 6-22: Chr13, 8HF - 16 Addr Bits, Split Query (250 characters) 77
Πίνακας 6-23: Chr13, 8HF - 16 Addr Bits, True positive hits, Split Query (250 characters) 77
Πίνακας 6-24: Chr13, 16HF - 14 Addr Bits, Split Query (1000 characters) 77
Πίνακας 6-25: Chr13, 16HF - 14 Addr Bits, True positive hits, Split Query (1000 characters) 78
Πίνακας 6-26: Chr13, 16HF - 14 Addr Bits, Split Query (500 characters) 78
Πίνακας 6-27: Chr13, 16HF - 14 Addr Bits, True positive hits, Split Query (500 characters) 78
Πίνακας 6-28: Chr13, 16HF - 14 Addr Bits, Split Query (250 characters) 79
Πίνακας 6-29: Chr13, 16HF - 14 Addr Bits, True positive hits, Split Query (250 characters) 79
Πίνακας 6-30: Chr12, 4HF - 14 Addr Bits, Whole Query 79
Πίνακας 6-31: Chr12, 4HF - 14 Addr Bits, True positive hits, Whole Query 80
Πίνακας 6-32: Chr12, 4HF - 14 Addr Bits, Split Query (500 characters) 80
Πίνακας 6-33: Chr12, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters) 80
Πίνακας 6-34: Chr12, 4HF - 14 Addr Bits, Split Query (250 characters) 81
Πίνακας 6-35: Chr12, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters) 81
Πίνακας 6-36: Chr14, 4HF - 14 Addr Bits, Whole Query 81
Πίνακας 6-37: Chr14, 4HF - 14 Addr Bits, True positive hits, Whole Query 82
Πίνακας 6-38: Chr14, 4HF - 14 Addr Bits, Split Query (500 characters) 82
Πίνακας 6-39: Chr14, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters) 82
Πίνακας 6-40: Chr14, 4HF - 14 Addr Bits, Split Query (250 characters) 83

Πίνακας 6-41: Chr14, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters) 83
Πίνακας 6-42: Chr17, 4HF - 14 Addr Bits, Whole Query 83
Πίνακας 6-43: Chr17, 4HF - 14 Addr Bits, True positive hits, Whole Query 84
Πίνακας 6-44: Chr17, 4HF - 14 Addr Bits, Split Query (500 characters) 84
Πίνακας 6-45: Chr17, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters) 84
Πίνακας 6-46: Chr17, 4HF - 14 Addr Bits, Split Query (250 characters) 85
Πίνακας 6-47: Chr17, 4HF - 14 Addr Bits, True positive hits, Split Query (250 characters) 85
Πίνακας 6-48: ChrX, 4HF - 14 Addr Bits, Whole Query 85
Πίνακας 6-49: ChrX, 4HF - 14 Addr Bits, True positive hits, Whole Query 86
Πίνακας 6-50: ChrX, 4HF - 14 Addr Bits, Split Query (500 characters) 86
Πίνακας 6-51: ChrX, 4HF - 14 Addr Bits, True positive hits, Split Query (500 characters) 86
Πίνακας 6-52: ChrX, 4HF - 14 Addr Bits, Actual Split Query (250 characters) 87
Πίνακας 6-53: ChrX, 4HF - 14 Addr Bits, Actual True positive hits, Split Query (250 characters) 87