ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ ΚΑΙ ΥΛΙΚΟΥ

"Εξειδικευμένη Αρχιτεκτονική, βασισμένη σε Αναδιατασσόμενη Λογική, για Επιτάχυνση Υπολογισμών Δευτεροταγούς Δομής Πρωτεϊνών με τον Αλγόριθμο Predator"

Δαγριτζίκος Παναγιώτης

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

Χανιά, Ιούνιος 2009

Ευχαριστίες

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

Θα ήθελα πρωτίστως να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Απόστολο Δόλλα για την υποστήριξή του, την καθοδήγηση και τις συμβουλές του αλλά και για την εμπιστοσύνη που έδειξε στις ικανότητές μου. Επίσης, θα ήθελα να ευχαριστήσω τον Αναπληρωτή Καθηγητή κ. Διονύσιο Πνευματικάτο και τον Επίκουρο Καθηγητή κ. Ιωάννη Παπαευσταθίου για την συμμετοχή τους ως μέλη της εξεταστικής επιτροπής.

Ένα μεγάλο ευχαριστώ και στους Διδακτορικούς Φοιτητές Γρηγόριο Χρυσό και Ευριπίδη Σωτηριάδη για την πολύ μεγάλη βοήθειά τους και την καθοδήγησή τους σε κάθε βήμα της πορείας μου από την αρχή μέχρι και την ολοκλήρωση αυτής της εργασίας.

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

Για τη φιλία και τη στήριξή τους θα ήθελα να ευχαριστήσω και όλους τους φίλους και συναδέλφους που ήταν δίπλα μου σε όλες τις φάσεις τις ζωής μου και με στήριξαν ανιδιοτελώς σε μεγάλο βαθμό.

Τέλος, θα ήθελα να πω ένα μεγάλο ευχαριστώ και να εκφράσω την αγάπη και την ευγνωμοσύνη μου στους γονείς μου και την αδερφή μου, που είναι πάντα δίπλα μου και με στηρίζουν στα όνειρα και τις αποφάσεις μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1 Γενικά…….9
1.2 Εισαγωγή στις πρωτεΐνες……9
1.3 Επιστημονική συνεισφορά……11
1.4 Δομή της διατριβής……12

ΚΕΦΑΛΑΙΟ 2

Πρόβλεψη δευτεροταγούς δομής πρωτεϊνών……13

2.1 Γενικά……13
2.2 Δευτεροταγής δομή πρωτεϊνών……13
2.3 Πρόβλεψη δευτεροταγούς δομής πρωτεΐνης……15
2.4 Αλγόριθμοι πρόβλεψης δευτεροταγούς δομής πρωτεΐνης……16

ΚΕΦΑΛΑΙΟ 3

Μελέτη του αλγορίθμου PREDATOR……19

3.1 Γενικά……19
3.2 Εισαγωγή στον αλγόριθμο PREDATOR……19
3.3 Ανάλυση βημάτων του αλγορίθμου PREDATOR……20
3.3.1 Στατιστική πάνω σε ζευγάρια υδρογονικών δεσμών σε τμήματα με α-έλικες ή β-γέφυρες……20
3.3.2 Παραγωγή των τάσεων των αμινοξέων για δημιουργία αντιπαράλληλων και παράλληλων β-φύλλων……21
3.3.3 Παραγωγή των τάσεων των αμινοξέων για δημιουργία α-ελίκων..23
3.3.4 Υπολογισμός τάσεων δευτεροταγούς δομής βασισμένων στη μέθοδο του πλησιέστερου γείτονα……23
3.3.5 Τάση για δημιουργία στροφής……24
3.3.6 Κανόνες για την εύρεση της δευτεροταγούς δομής των στοιχείων της πρωτεΐνης……24

ΚΕΦΑΛΑΙΟ 4

Μοντελοποίηση του αλγορίθμου PREDATOR……25

4.1 Γενικά……25

4.2 Μοντελοποίηση σε software……..25
4.3 Αναπαράσταση των αριθμών……27
4.3.1 Αριθμοί κινητής υποδιαστολής……27
4.3.2 Παράδειγμα πράξεων μεταξύ αριθμών κινητής υποδιαστολής……27
4.3.3 Πρόσθεση, αφαίρεση……27
4.3.4 Πολλαπλασιασμός……28
4.3.5 Διαίρεση……28
4.4 Περιγραφή των βημάτων του αλγορίθμου……28
4.4.1 Υπολογισμός $P_m^{Antipar}$……29
4.4.2 Υπολογισμός P_m^{par}……30
4.4.3 TakeBestBeta……31
4.4.4 Υπολογισμός Pmhelix……32
4.4.5 TakeBestAlpha……33
4.4.6 Υπολογισμός P_m^{turn}……33
4.4.7 Εύρεση ομόλογων, με την αλληλουχία εισόδου, πρωτεϊνών……34
4.4.8 PredictNN……35
4.4.9 Εφαρμογή κανόνων για πρόβλεψη της δομής……37

ΚΕΦΑΛΑΙΟ 5

Αρχιτεκτονική αλγορίθμου PREDATOR……40

5.1 Γενικά……40
5.2 Τμήμα Antipar……43
5.3 Τμήμα Par……46
5.4 Τμήμα Alpi……49
5.5 Τμήμα Turni……52
5.6 Τμήμα TakeBestAlpha……55
5.7 Τμήμα TakeBestBeta……57
5.8 Τμήμα Homologous……57
5.9 Τμήμα PredictNN……60
5.10 Τμήμα DecideStruct……64

ΚΕΦΑΛΑΙΟ 6

Αποτελέσματα – Μετρήσεις……67

6.1 Γενικά……67
6.2 Πολλαπλασιαστής και διαιρέτης ακέραιων αριθμών……67
6.3 Μνήμη BRAM……68
6.4 Ταυτοποίηση λειτουργίας……69
6.5 Απόδοση συστήματος……70
6.6 Αποτίμηση απόδοσης……71

ΚΕΦΑΛΑΙΟ 7

Συμπεράσματα, μελλοντικές επεκτάσεις και αλλαγές.....76

7.1 Γενικά.....76
7.2 Συμπεράσματα.....76
7.3 Μελλοντικές επεκτάσεις.....76

ΒΙΒΛΙΟΓΡΑΦΙΑ.....79

1 Εισαγωγή

1.1 Γενικά

Στο πρώτο κεφάλαιο αυτής της διατριβής, γίνεται μια εισαγωγή στις πρωτεΐνες και στο πώς αυτές δομούνται στο χώρο. Επιπλέον, γίνεται αναφορά στη σημασία που έχει η πρόβλεψη της δομής τους και ιδιαίτερα της δευτεροταγούς, με την οποία ασχολούμαστε εκτενέστερα.

Εικόνα 1-1: Η επιστήμη της βιοπληροφορικής

1.2 Εισαγωγή στις πρωτεΐνες

Ένα από τα σημαντικότερα προβλήματα που καλείται να επιλύσει η σύγχρονη βιοιατρική, είναι η πρόβλεψη της δομής μιας πρωτεΐνης γνωρίζοντας μόνο την αλληλουχία αμινοξέων από την οποία αποτελείται η πρωτεϊνική αλυσίδα.

Οι πρωτεΐνες είναι μεγάλα οργανικά μακρομόρια αποτελούμενα από αμινοξέα, τα οποία ενώνονται μεταξύ τους με πεπτιδικούς δεσμούς σχηματίζοντας μια γραμμική αλυσίδα, καλούμενη αλυσίδα πολυπεπτιδίων. Η αλληλουχία των αμινοξέων σε μια πρωτεΐνη καθορίζεται από την αντίστοιχη αλληλουχία των γονιδίων, τα οποία κωδικοποιούνται από το γενετικό κώδικα. Γενικά, ο γενετικός κώδικας προσδιορίζει τα 20 βασικά αμινοξέα (πίνακας 1-1), ενώ σε κάποιους οργανισμούς μπορεί να περιλαμβάνει τρία επιπλέον.

Πίνακας 1-1: Τα 20 βασικά αμινοξέα [1]

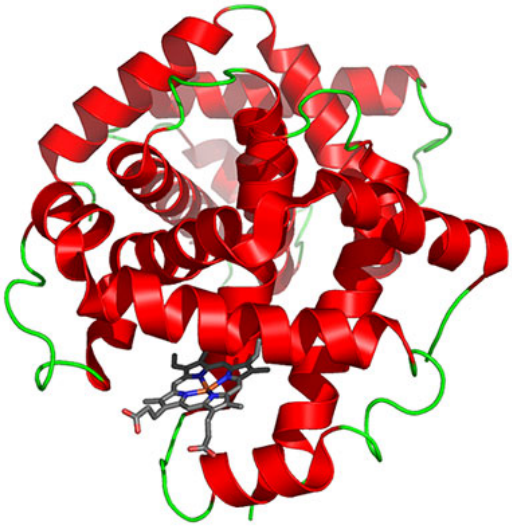
Ονομασία	Διεθνής σύντμηση	Ονομασία	Διεθνής σύντμηση
Αλανίνη	Ala	Λευκίνη	Leu
Αργιρίνη	Arg	Λυσίνη	Lys
Ασπαραγγίνη	Asn	Μεθιονίνη	Met
Ασπαρτικό οξύ	Asp	Φαινυλαλανίνη	Phe
Κυστεΐνη	Cys	Προλίνη	Pro
Γλουταμίνη	Gln	Σερίνη	Ser
Γλουταμικό οξύ	Glu	Θρεονίνη	Thr
Γλυκίνη	Gly	Τρυπτοφάνη	Trp
Ιστιδίνη	His	Τυροσίνη	Tyr
Ισολευκίνη	Ile	Βαλίνη	Val

Η δομή μιας πρωτεΐνης χωρίζεται σε τέσσερις βασικές κατηγορίες: Στην πρωτοταγή δομή, η οποία περιλαμβάνει απλά την ακολουθία των αμινοξέων που την αποτελούν. Η δευτεροταγής δομή περιλαμβάνει επαναλαμβανόμενες μικρότερες δομές οι οποίες συνδέονται μεταξύ τους με υδρογονικούς δεσμούς. Τα πιο συνήθη παραδείγματα τέτοιων δομών είναι οι άλφα-έλικες και τα βήτα φύλλα. Λόγω του ότι οι δευτεροταγείς δομές είναι τοπικές, πολλές δομές μαζί μπορεί να είναι παρούσες στο ίδιο πρωτεϊνικό μόριο. Η τριτοταγής δομή, είναι το σχήμα που παίρνει το μόριο της πρωτεΐνης και περιλαμβάνει τις αλληλοεπιδράσεις που φέρουν οι δευτεροταγείς δομές μεταξύ τους (Εικόνα 1-2). Η τριτοταγής δομή σταθεροποιείται με το σχηματισμό υδροφοβικού πυρήνα στο μόριο, ακόμα και από γέφυρες άλατος και υδρογονικούς δεσμούς και είναι υπεύθυνη για τον καθορισμό της βασικής λειτουργικότητας της πρωτεΐνης. Ο όρος «τριτοταγής δομή» συχνά καλείται και «αναδίπλωση» (protein folding). Τέλος, η τεταρτοταγής δομή είναι αποτέλεσμα της αλληλεπίδρασης δύο ή και περισσότερων πρωτεϊνικών μορίων τα οποία σχηματίζουν τα συμπλέγματα των πρωτεϊνών. Τέλος, μια πρωτεΐνη μπορεί να ολισθήσει μεταξύ των διαφορετικών αυτών δομών για να επιτελέσει τις βιολογικές της λειτουργίες, μια διαδικασία η οποία είναι αναστρέψιμη και καλείται αλλαγή σχηματισμού.

Δεδομένου ότι μια πρωτεΐνη απαρτίζεται από έναν πολύ μεγάλο αριθμό αμινοξέων, το πρόβλημα πρόβλεψης της δομής τους καθίσταται πολύπλοκο και αρκετά χρονοβόρο. Γι' αυτό το λόγο, απαιτούνται κάποιες θεωρητικές προσεγγίσεις για την εύρεση της δομής των αλληλουχιών για τις οποίες η πειραματική εξακρίβωση, για παράδειγμα μέσω κρυσταλλογράφησης, δεν είναι εφικτή.

Το ιδανικότερο σενάριο για την πρόβλεψη της δομής μιας πρωτεΐνης θα ήταν η ύπαρξη ενός αλγορίθμου ο οποίος θα είχε ως είσοδο την αλληλουχία των αμινοξέων που αποτελούν την πρωτεΐνη και θα επέστρεφε την ακριβή θέση τους στον τρισδιάστατο χώρο. Ωστόσο, μια τέτοια πρόβλεψη είναι εξαιρετικά δύσκολη ή ανέφικτη τουλάχιστον μέχρι σήμερα, λόγω των ανεπαρκών γνώσεων που υπάρχουν γύρω από το πώς αναδιπλώνονται οι πρωτεΐνες στο χώρο αλλά και λόγω της τεράστιας υπολογιστικής ισχύς που αυτή θα απαιτούσε. Αυτό συμβαίνει γιατί η αρχική ακολουθία των αμινοξέων υπόκειται σε μια σειρά από φυσικό-χημικές δυνάμεις σε ένα πολυσύνθετο μικροπεριβάλλον, το οποίο μπορεί να περιλαμβάνει και άλλες δομές, όπως ένζυμα, τα οποία εμποδίζουν πιθανές αναδιπλώσεις των πρωτεϊκών αλυσίδων και κάνουν έτσι πιο πολύπλοκη τη μελέτη τους.

Αν και έχουν γίνει κάποιες προσπάθειες για δημιουργία αλγορίθμων που προβλέπουν κατευθείαν τη δομή της πρωτεΐνης στο χώρο, οι περισσότερες μέθοδοι χρησιμοποιούν τη δευτεροταγή δομή της πρωτεΐνης ως βάση για την εύρεση της τελικής της μορφής. Έτσι, η πρόβλεψη της δευτεροταγούς δομής μιας πρωτεΐνης είναι ζωτικής σημασίας για την επιστημονική κοινότητα και εκεί έχει επικεντρωθεί ένα μεγάλο μέρος της έρευνας.



Εικόνα 1-2 : Τριτοταγής δομή αιμογλοβίνης [2]

1.3 Επιστημονική συνεισφορά

Σήμερα, υπάρχουν δεκάδες αλγόριθμοι οι οποίοι προβλέπουν τη δευτεροταγή δομή μιας πρωτεΐνης, καθένας από τους οποίους βασίζεται σε διαφορετική θεωρητική προσέγγιση και έχει διαφορετικά επίπεδα ακριβείας. Ενδεικτικά αναφέρουμε τους πιο διαδεδομένους και πιο ακριβείς αλγορίθμους: *GOR, JPRED, Meta-PP, PHD, PREDATOR, PredictProtein, PSIPRED και YASSPP*. Είναι αξιοσημείωτο ότι μέσα στο 2005, η ομάδα Computational Genomics του τμήματος Ιατρικής στο πανεπιστήμιο της Ουάσινγκτον, υλοποίησε τέσσερις νέους αλγορίθμους πρόβλεψης της δομής των πρωτεϊνών οι οποίοι ήταν διαθέσιμοι στο διαδίκτυο ως ένα τμήμα του προγράμματος PROTINFO[3]. Η ενέργεια αυτή απέσπασε θετικότατες κριτικές από την ιατρική επιστημονική κοινότητα και έδωσε ώθηση για εκτενέστερη και αποτελεσματικότερη μελέτη στο μείζον αυτό πρόβλημα της βιοιατρικής.

Η πρόβλεψη της δευτεροταγούς δομής των πρωτεϊνών με τον αλγόριθμο PREDATOR έχει αποδειχθεί ότι μπορεί να φτάσει στα επίπεδα ακρίβειας του 75%, γεγονός που τον καθιστά τον πιο ακριβή αλγόριθμο τουλάχιστον μέχρι σήμερα. Η εφαρμογή όμως των βημάτων του αλγορίθμου για κάθε ένα από τα αμινοξέα εισόδου απαιτεί μεγάλη υπολογιστική ισχύ και αρκετό χρόνο και δεδομένου ότι οι περισσότερες πρωτεΐνες που μελετούνται αποτελούνται από χιλιάδες αμινοξέα, μπορεί να φτάσει σε απαγορευτικά χρονικά επίπεδα.

Η παρούσα διπλωματική διατριβή καλύπτει τη μελέτη και το σχεδιασμό ολόκληρου του αλγορίθμου PREDATOR σε αναδιατασσόμενη προγραμματιζόμενη λογική (Field Programmable Gate Array-FPGA). Με την προσέγγιση αυτή μπορεί να επιτευχθεί σημαντική επιτάχυνση του χρόνου εκτέλεσης του αλγόριθμου και συνεπώς της πρόβλεψης της δομής της πρωτεΐνης, ανεξαρτήτως μεγέθους εισόδου.

1.4 Δομή της διατριβής

Στο **κεφάλαιο 2** της παρούσας διατριβής, αναλύεται ο τρόπος με τον οποίο δομούνται οι πρωτεΐνες στο χώρο, περιγράφονται συνοπτικά οι υπάρχοντες αλγόριθμοι που προβλέπουν τη δευτεροταγή δομή των πρωτεϊνών και γίνεται μια εισαγωγή στον αλγόριθμο PREDATOR.

Στο **κεφάλαιο 3**, περιγράφεται ο τρόπος με τον οποίο έγινε η μοντελοποίηση του PREDATOR σε software και γίνεται ανάλυση των βημάτων που απαρτίζουν τον αλγόριθμο.

Στο **κεφάλαιο 4,** παρουσιάζεται η αρχιτεκτονική που προτείνεται και γίνεται ανάλυση των επιμέρους υποσυστημάτων που χρησιμοποιήθηκαν, εξηγώντας ταυτόχρονα και τις σχεδιαστικές επιλογές που ακολουθήθηκαν.

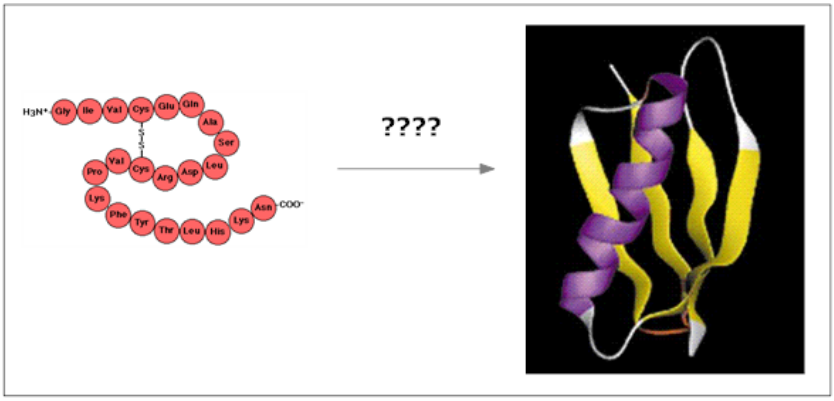
Στο **κεφάλαιο 5,** παρατίθενται πληροφορίες για την απόδοση της αρχιτεκτονικής υλοποιημένης σε FPGA και γίνεται σύγκριση με τους χρόνους εκτέλεσης που έχουν καταμετρηθεί στον προσωπικό υπολογιστή.

Τέλος, στο **κεφάλαιο 6,** αναφέρονται τα συμπεράσματα από τη συγκεκριμένη διατριβή και προτείνονται ιδέες για μελλοντική επέκτασης ή και βελτιώσεις της υπάρχουσας αρχιτεκτονικής.

2 Πρόβλεψη δευτεροταγούς δομής πρωτεϊνών

2.1 Γενικά

Στο κεφάλαιο αυτό περιλαμβάνεται το απαραίτητο θεωρητικό υπόβαθρο για τη δευτεροταγή δομή μιας πρωτεΐνης και γίνεται μια αναφορά στους αλγορίθμους που έχουν αναπτυχθεί για την πρόβλεψή της, χωρίζοντάς τους σε τρεις βασικές κατηγορίες ανάλογα με τη μέθοδο που ακολουθούν για να κάνουν την πρόβλεψη.



Εικόνα 2-1: Δευτεροταγής δομή πρωτεϊνών

2.2 Δευτεροταγής δομή πρωτεϊνών

Στη βιοχημεία και στη δομική βιολογία, η δευτεροταγής δομή αναπαριστά γενικά τον τρισδιάστατο σχηματισμό των τμημάτων των πολυμερών, όπως είναι οι πρωτεΐνες και τα νουκλεϊκά οξέα (DNA/RNA). Παρόλα αυτά, δεν περιγράφει τις ακριβείς ατομικές θέσεις των τμημάτων αυτών στον τρισδιάστατο χώρο, οι οποίες αποτελούν την τριτοταγή δομή. Η ιδέα της ύπαρξης δευτεροταγούς δομής στα βιολογικά μακρομόρια, ανήκει στον Kaj Ulrik

Linderstrøm-Lang [4], ο οποίος το 1952 σε μια διάλεξή του στο ιατρικό τμήμα του πανεπιστημίου του Stanford, πρωτοεισήγαγε τον όρο «δευτεροταγής δομή».

Η δευτεροταγής δομή στις πρωτεΐνες αποτελείται από τοπικές αλληλεπιδράσεις μεταξύ των αμινοξέων ανάμεσα στα οποία είτε παρεμβάλλονται υδρογονικοί δεσμοί είτε όχι. Οι συνηθέστερες δευτεροταγείς δομές είναι οι άλφα έλικες (α-helix) και τα βήτα φύλλα (β-sheet):

Η άλφα έλικα είναι αριστερόστροφος ή και δεξιόστροφος σχηματισμός σπείρας, ο οποίος μοιάζει με ελατήριο, όπου κάθε N-H ομάδα δανείζει έναν υδρογονικό δεσμό σε μια C=O ομάδα αμινοξέων που βρίσκεται τέσσερις θέσεις πιο πριν ($i + 4 \rightarrow i$ υδρογονικός δεσμός). Ο σχηματισμός αυτός της δευτεροταγούς δομής είναι επίσης γνωστός με την ονομασία Pauling-Corey-Branson (PCB) άλφα έλικα [5].

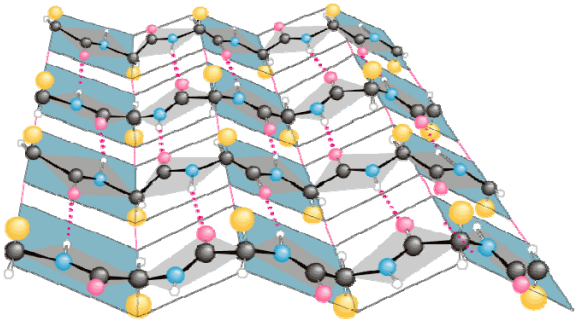
Τα βήτα φύλλα [6] είναι η δεύτερη συνηθέστερη μορφή δευτεροταγούς δομής μιας πρωτεΐνης και αποτελείται από βήτα λωρίδες οι οποίες συνδέονται παράπλευρα με τρεις ή και περισσότερους υδρογονικούς δεσμούς, σχηματίζοντας έτσι ένα περιστρεφόμενο φύλλο. Η αλληλεπίδραση των β-φύλλων εμπλέκεται στο σχηματισμό πρωτεϊνικών συνονθυλευμάτων όπου παρατηρούνται σε κάποιες σπάνιες ασθένειες, όπως για παράδειγμα η αμυλοείδωση (amyloidoses).

Κάποιες άλλες δομές όπως η άλφα ή η 3_{10} έλικα, συναντούνται πολύ σπάνια στις πρωτεΐνες, αλλά είναι σημαντικοί παράγοντες ως διαμεσολαβητές στην αναδίπλωση της πρωτεΐνης. Η τυχαία σπείρα (random coil) δεν είναι μια πραγματική δευτεροταγής δομή πρωτεΐνης, αλλά υποδηλώνει το σχηματισμό εκείνο που στερείται της παρουσίας γνωστής δευτεροταγούς δομής.

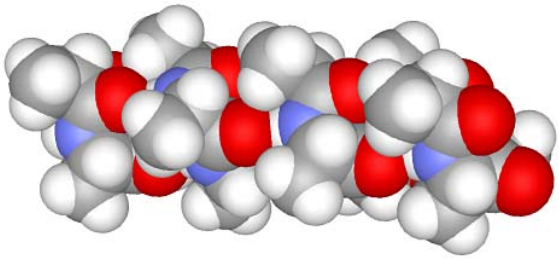
Τα αμινοξέα ποικίλουν ανάλογα με την ικανότητά τους να συνθέτουν τα βασικά στοιχεία που αποτελούν τη δευτεροταγή δομή της πρωτεΐνης. Σε αντίθεση με το γενικό κανόνα, κάποιες πρωτεΐνες όπως η προλίνη και η γλυκίνη, είναι γνωστές ως «καταστροφείς ελίκων» λόγω του ότι αντί να σχηματίζουν, εκείνες εμποδίζουν το σωστό σχηματισμό του κυρίου μέρους της άλφα έλικας. Επιπλέον, κάποια αμινοξέα, αν και σπανίως, προτιμούν να σχηματίζουν μεταξύ τους β-κλώνους (b-strands), δομές οι οποίες είναι διαφορετικές από τις προαναφερθείσες. Οι β-κλώνοι είναι μια ομάδα μήκους 5-10 αμινοξέων των οποίων η πεπτιδική αλυσίδα είναι πλήρως επεκταμένη.

Παρόλα αυτά, οι προτιμήσεις που έχουν κάποια αμινοξέα για το σχηματισμό διαφορετικών δομών από τις συνήθεις, δεν αποτελεί τροχοπέδη για την ανάπτυξη αξιόπιστων μεθόδων που μπορούν να προβλέψουν τη δευτεροταγή δομή της πρωτεΐνης.

Στις εικόνες 2-2 και 2-3 δίνεται μια αναπαράσταση της μορφής που έχουν οι βασικές δευτεροταγείς δομές της πρωτεΐνης στο χώρο.



Εικόνα 2-2: Μορφή β-φύλλου στο χώρο



Εικόνα 2-3: Μορφή α-έλικας στο χώρο

2.3 Πρόβλεψη δευτεροταγούς δομής πρωτεΐνης

Μια σημαντική υπόθεση της πρόβλεψης όλων των δομών των πρωτεϊνών είναι το γεγονός ότι η αλληλουχία των αμινοξέων προσδιορίζει μοναδικά τη δομή μιας πρωτεΐνης στο χώρο. Η απόδειξη ότι η δομή της πρωτεΐνης καθορίζεται από τα αμινοξέα δόθηκε από τον Christian B. Anfinsen (1916-1995), ο οποίος για την ανακάλυψή του αυτή τιμήθηκε με το βραβείο Nobel το 1972 [7].

Η παραπάνω υπόθεση γίνεται βάσιμη με τα ακόλουθα αποδεικτικά στοιχεία: Αν με κάποιο τρόπο η πρωτεΐνη «ξετυλιχθεί» in vitro με τέτοιο τρόπο έτσι ώστε να μην υπάρχουν άλλες ουσίες και στη συνέχεια αφεθεί, τότε μέσα σε χρονικό διάστημα ενός δευτερολέπτου η πρωτεΐνη θα πάρει την ίδια τρισδιάστατη μορφή που είχε και προηγουμένως . Με αυτό τον τρόπο αποδεικνύεται ότι όλη η πληροφορία για τη δομή της πρωτεΐνης βρίσκεται στην αλληλουχία των αμινοξέων. Όπως γίνεται σε κάθε κανόνα, έτσι και σε αυτόν, υπάρχουν πρωτεΐνες όπου δεν ισχύει το παραπάνω, λόγω του ότι κάποιες από αυτές χρειάζονται «βοηθητικά μόρια» για να αναδιπλωθούν.

Δεδομένου ότι η πρόβλεψη της δευτεροταγούς δομής της πρωτεΐνης είναι από τα πρώτα και από τα πιο σημαντικά προβλήματα που αντιμετωπίστηκαν από υπολογιστικές τεχνικές, υπάρχουν δεκάδες αναπτυγμένες μέθοδοι που χρησιμοποιούνται για την επίλυση του προβλήματος.

Όπως έχουμε ήδη αναφέρει, λόγω του ότι ολόκληρη η πληροφορία για το σχηματισμό της δευτεροταγούς δομής περιλαμβάνεται στην πρωτεύουσα αλληλουχία των αμινοξέων, ακόμα και μια μικρή ομάδα αμινοξέων που αποτελεί τμήμα μιας πρωτεΐνης, θα αποκτήσει μια συγκεκριμένη μορφή δευτεροταγούς δομής. Έτσι, πολλοί αλγόριθμοι εξετάζουν ένα «παράθυρο» αλληλουχίας 13-17 αμινοξέων, υποθέτοντας ότι το κεντρικό αμινοξύ στο παράθυρο θα υιοθετήσει ένα σχηματισμό που καθορίζεται από τα γειτονικά αμινοξέα. Για τις άλφα έλικες, αυτό το παράθυρο έχει μήκος 5-40 αμινοξέα και για τα β-φύλλα το μήκος του είναι 5-10 αμινοξέα.

2.4 Αλγόριθμοι πρόβλεψης δευτεροταγούς δομής πρωτεΐνης

Οι τρεις πιο γνωστές μέθοδοι για την πρόβλεψη της πρωτοταγούς δομής της πρωτεΐνης, περιλαμβάνουν:

- *Τις μεθόδους Chou-Fasman και GOR*
- *Μοντέλα νευρωνικών δικτύων*
- *Μεθόδους του πλησιέστερου γείτονα* (nearest neighbor methods)

Chou-Fasman και GOR

Το 1974, ο Chou και ο Fasman [8], [9] ανέπτυξαν σε 15 πρωτεΐνες, των οποίων η δομή είχε γίνει γνωστή μέσω της κρυσταλλογράφησης με ακτίνες Χ, μια στατιστική μέθοδο βασισμένη στην τάση των αμινοξέων να αναπτύξουν δευτεροταγείς δομές. Τα στατιστικά αυτά στοιχεία προκύπτουν από κάποιες συγκεκριμένες στερεοχημικές και φυσικοχημικές ιδιότητες των αμινοξέων. Με το πέρασμα των χρόνων, τα στοιχεία αυτά τελειοποιήθηκαν μιας και κρυσταλλογραφήθηκαν εκατοντάδες ακόμα πρωτεΐνες.

Σε αντίθεση με τους Chou-Fasman οι οποίοι υπέθεσαν ότι κάθε αμινοξύ ξεχωριστά επηρεάζει τη δευτεροταγή δομή της πρωτεΐνης, η ομάδα GOR (Garnier, Osguthorpe, Robson), έλαβε υπόψη την επιρροή που ασκούν τα γειτονικά αμινοξέα στη δομή που θα σχηματίσει το κεντρικό αμινοξύ[8], [10]. Για παράδειγμα, αν ένα αμινοξύ γειτνιάζει με αμινοξέα που προτιμούν να σχηματίζουν έλικα, η πιθανότητα να σχηματίζει κι αυτό έλικα είναι πολύ μεγάλη παρόλο που η προτίμησή του για το σχηματισμό έλικας μπορεί να είναι μικρή. Έτσι, αντί να λαμβάνονται υπόψη οι τάσεις ενός και μόνο αμινοξέος, υπολογίστηκαν οι πιθανότητες όλων των τύπων αμινοξέων για το σχηματισμό α-έλικας και β-φύλλου.

Τα προγράμματα που χρησιμοποιούν την παραπάνω μέθοδο είναι τα: DPM[11], DSC[12], GOR IV [13].

Μοντέλα νευρωνικών δικτύων

Ένα νευρωνικό δίκτυο αποτελείται από μια υπολογιστική μηχανή εκμάθησης, η οποία παρέχει στις υπολογιστικές μεθόδους την ικανότητα για «μάθηση» σε μια προσπάθεια να προσομοιωθούν οι τρόποι με τους οποίους επικοινωνούν οι εγκεφαλικοί νευρώνες κατά τη διάρκεια της μάθησης . Οι υπολογιστές εκπαιδεύονται να αναγνωρίζουν πρότυπα (pattern recognition)σε γνωστές δευτεροταγείς δομές πρωτεϊνών, χρησιμοποιώντας για εκμάθηση μη ομόλογες δομές και κάνοντας εν τέλει δοκιμές σε πρωτεΐνες όπου η δομή τους είναι γνωστή.

Τα νευρωνικά δίκτυα είναι δυνατό να πετύχουν ακρίβεια κοντά στο 73% σε περιπτώσεις όπου η πρωτεϊνική δομή αποτελείται από τις τρεις γνωστές καταστάσεις (α-έλικα, β-φύλλο, τυχαία σπείρα). Ο λόγος που επιτυγχάνεται τόσο μεγάλη ακρίβεια έγκειται στην ικανότητα της μεθόδου να «ταιριάξει» την πρωτεΐνη της οποίας η δομή είναι άγνωστη, με άλλες πρωτεΐνες της ίδιας οικογένειας, στις οποίες και ανακαλύπτει πρωτεϊνικά τμήματα τα οποία είναι ίδια και στις δύο περιπτώσεις. Δυστυχώς, τα νευρωνικά δίκτυα λειτουργούν ως «μαύρα κουτιά», λόγω του ότι αν και τα βάρη του πίνακα βαρών είναι γνωστά στο χρήστη, οι παράμετροι του νευρωνικού δικτύου είναι κρυφοί και δεν είναι βιολογικά μεταφράσιμοι ή ενδιαφέροντες για το χρήστη. Συμπερασματικά, ενώ τα νευρωνικά δίκτυα είναι πανίσχυρα

εργαλεία για την πρόβλεψη της δομής των πρωτεϊνών, δε δίνουν κάποια πληροφορία γύρω από τη διαδικασία αναγνώρισης των μορίων, γεγονός που τα καθιστά δύσκολα στη χρήση τους. Τα προγράμματα που χρησιμοποιούν την παραπάνω μέθοδο είναι τα: PHD[14], PSIPRED[15], NNPREDICT[16].

Μεθόδοι του πλησιέστερου γείτονα (nearest neighbor methods)

Η βασική ιδέα της μεθόδου του πλησιέστερου γείτονα είναι η πρόβλεψη της δευτεροταγούς δομής του κεντρικού αμινοξέος ενός τμήματος μιας πρωτεΐνης βασισμένη σε πρωτεΐνες όπου η δευτεροταγή δομή τους είναι γνωστή. Αυτό γίνεται με την εύρεση όμοιων αλληλουχιών μεταξύ πρωτεϊνών που η δομή τους είναι γνωστή και τμήματος της άγνωστης πρωτεΐνης. Χρησιμοποιώντας τη γνωστή δευτεροταγή δομή των πρωτεϊνών αυτών και με τη χρήση βαρών ανάλογα με το ποια πρωτεΐνη είχε τη μεγαλύτερη ομοιότητα με την άγνωστη, γίνεται η πρόβλεψη της δευτεροταγούς δομής.

Για παράδειγμα, με την ολίσθηση ενός παραθύρου καθορισμένου μεγέθους πάνω σε 400 περίπου πρωτεΐνες γνωστής δομής οι οποίες δεν είναι ομόλογες μεταξύ τους και καταγράφοντας στη συνέχεια τη δευτεροταγή δομή του κεντρικού αμινοξέος στο παράθυρο, δημιουργείται μια λίστα μικρών πρωτεϊνικών τμημάτων με γνωστή δομή. Στη συνέχεια, ένα παράθυρο ίδιου μεγέθους επιλέγεται στην πρωτεΐνη με την άγνωστη δομή και συγκρίνεται με τη λίστα των προαναφερθέντων πρωτεϊνών, βρίσκοντας έτσι τα 50 καλύτερα «ταιριάσματα». Η συχνότητα με την οποία το κεντρικό αμινοξύ σε κάθε ένα από τα 50 κομμάτια θα σχηματίσει μια συγκεκριμένη δευτεροταγής δομή, χρησιμοποιείται τελικά για την πρόβλεψη της δευτεροταγούς δομής του αμινοξέος σε όλη την αλληλουχία.

Η ποικιλία στις μεθόδους του πλησιέστερου γείτονα, προκύπτει από τον καθορισμό των εκάστοτε παραμέτρων, όπως το πώς ορίζεται η ομοιότητα ή και το μήκος του παραθύρου που θα χρησιμοποιηθεί. Τα προγράμματα που χρησιμοποιούν την παραπάνω μέθοδο είναι τα: SOPM[17], SOPMA[18], NNSSP[19], PREDATOR.[20]

Πίνακας 2-1: Τα ποσοστά επιτυχίας οκτώ δημοφιλών αλγορίθμων πρόβλεψης δευτεροταγούς δομής πρωτεΐνης [21]

Όνομα	Ακρίβεια (%)
PHDsec	>72
NSSP	>71
SOPM	>70
DSC	70
SSPRED	>70
PREDATOR	75
MultiPredict	>65
NNPREDICT	>65

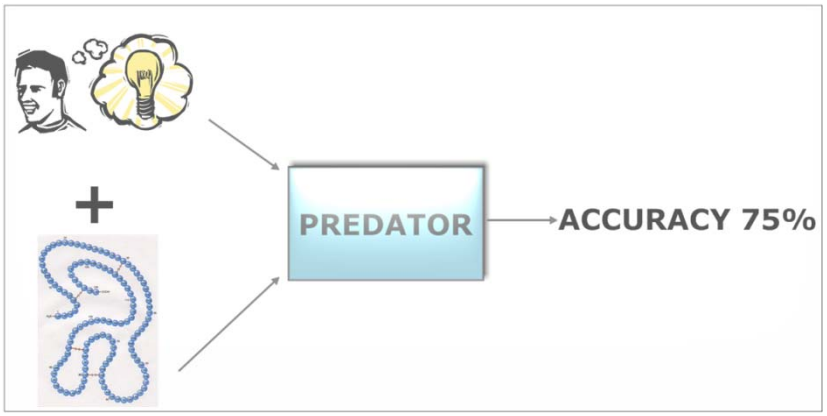
Στο πίνακα 2-1 παρατηρούμε ότι ο πιο ακριβής αλγόριθμος όσον αφορά την πρόβλεψή του, είναι ο PREDATOR. Το γεγονός αυτό, σε συνδυασμό με το ότι ο PREDATOR είναι ένας από

τους αλγορίθμους του Bioperf [22], μια σουίτα για την αξιολόγηση Αρχιτεκτονικής Υπολογιστών υψηλών επιδόσεων σε εφαρμογές Βιοπληροφορικής, μας ώθησε στο να τον επιλέξουμε για την υλοποίηση σε υλικό (hardware).

3 Μελέτη αλγορίθμου PREDATOR

3.1 Γενικά

Στο κεφάλαιο αυτό γίνεται αναλυτική περιγραφή του αλγορίθμου PREDATOR. Η περιγραφή αυτή χωρίζεται σε έξι στάδια για την πιο εύκολη κατανόησή του και για να απλοποιηθεί η διαδικασία της μοντελοποίησης, η οποία θα περιγραφεί στο επόμενο κεφάλαιο.



Εικόνα 3-1: Ο PREDATOR είναι ο πιο ακριβής αλγόριθμος πρόβλεψης δευτεροταγούς δομής μιας πρωτεΐνης

3.2 Εισαγωγή στον αλγόριθμο PREDATOR

Ο σχηματισμός των δευτεροταγών δομικών στοιχείων στις πρωτεΐνες καθορίζεται από μια πολύπλοκη αλληλεπίδραση φυσικών δυνάμεων και εξαρτάται τόσο από τις τοπικές όσο και από τις πιο μακρινές αλληλεπιδράσεις. Στη δουλειά αυτή χρησιμοποιούνται οι πληροφορίες από συγκεκριμένες μη-τοπικές αλληλεπιδράσεις που σχετίζονται με το σχηματισμό δευτεροταγών δομών για την πρόβλεψη της δομής της πρωτεΐνης. Οι υδρογονικοί δεσμοί αναπαριστούν έναν από τους πιο σχετικούς τύπους αυτών των αλληλεπιδράσεων. Οι α-έλικες και τα β-φύλλα ανακαλύφθηκαν από την ομάδα του Pauling (1951) [23] με τη χρήση στερεοχημικών περιορισμών. Πρόσφατες έρευνες τονίζουν το σπουδαίο ρόλο των υδρογονικών δεσμών στο

σχηματισμό μιας πρωτεΐνης. Η πιο αποδεκτή μέθοδος εύρεσης της δευτεροταγούς δομής της πρωτεΐνης από τις ατομικές συντεταγμένες των πρωτεϊνών (Kabsch και Sander [8],1983) βασίζεται στην αναγνώριση των δομών που σχηματίζουν οι υδρογονικοί δεσμοί.

Για να επιτευχθεί η μεγαλύτερη δυνατή ακρίβεια πρόβλεψης είναι αναγκαίο να γίνει συγκερασμός της μεθόδου εύρεσης των δομικών δευτεροταγών στοιχείων από τις ατομικές συντεταγμένες και της πρόβλεψης της δομής μόνο από τα δεδομένα που δίνει η ίδια η πρωτεϊνική αλληλουχία. Με την ανίχνευση των δομών που σχηματίζουν οι υδρογονικοί δεσμοί είναι εφικτή η πρόβλεψη της δομής της πρωτεΐνης με ακρίβεια 68%, ενώ με τη χρήση και των ομόλογων πρωτεϊνών, η ακρίβεια ανέρχεται περίπου στο 75%. Η ρουτίνα που χρησιμοποιεί τις παραπάνω μεθόδους ονομάζεται PREDATOR.

Για την εκπαίδευση και τη δοκιμή του αλγορίθμου, αρχικά έγινε χρήση ενός συνόλου από 125 πρωτεϊνικές αλυσίδες, οι οποίες δημοσιεύθηκαν από τους Rost και Sander (1993). Για την τελική εκπαίδευση χρησιμοποιήθηκαν 559 πρωτεϊνικές αλυσίδες των οποίων η δομή είχε βρεθεί από την κρυσταλλογράφησή τους με ακτίνες Χ. Οι πρωτεΐνες αυτές έχουν μέγιστη ομοιότητα μεταξύ τους 25% και το ελάχιστο μήκος τους ανέρχεται στα 50 αμινοξέα [20].

3.3 Ανάλυση βημάτων του αλγορίθμου PREDATOR

Η τεχνική πρόβλεψης της δευτεροταγούς δομής που περιγράφουμε εδώ, βασίζεται στην «προτίμηση» που έχουν τα αμινοξέα για να σχηματίσουν α-έλικες, β-φύλλα ή σπείρες. Έτσι, υπολογίζονται οι τάσεις που έχουν τα αμινοξέα σε κάθε θέση που βρίσκονται. Τρεις από αυτές βασίζονται σε μακρινές αλληλεπιδράσεις μεταξύ αμινοξέων με υδρογονικούς δεσμούς σε παράλληλα ή αντιπαράλληλα φύλλα αλλά και σε έλικες. Τρεις επιπλέον τιμές που καθορίζονται με βάση το ποσοστό ομοιότητας ενός τμήματος της πρωτεΐνης με εκείνες τις πρωτεΐνες όπου η δομή τους είναι γνωστή, υπολογίζονται για την έλικα, τα φύλλα και τη σπείρα. Τέλος, μια στατιστικώς υπολογιζόμενη τάση υπολογίζεται με τη χρήση ενός ολισθαίνοντος παραθύρου μήκους τεσσάρων αμινοξέων. Τέλος, γίνεται χρήση κάποιων κατωφλίων για τον καθορισμό της δομής που έχει το συγκεκριμένο πρωτεϊνικό τμήμα. Σύμφωνα με τα [20], [24], [25], [26], στα επόμενα υποκεφάλαια θα περιγράψουμε αναλυτικά τα προαναφερθέντα βήματα.

3.3.1. Στατιστική πάνω σε ζευγάρια υδρογονικών δεσμών σε τμήματα με α-έλικες ή β-γέφυρες

Σε αυτή την εργασία λαμβάνονται υπόψη τρία κύρια είδη υδρογονικών β-γεφυρών δύο αντιπαράλληλων και μιας παράλληλης (Εικόνα 3-2, 3-3, 3-4). Τα αμινοξέα που περικλείονται μεταξύ της β-γέφυρας καλούνται εσωτερικά, είτε όταν σχηματίζουν υδρογονικούς δεσμούς μεταξύ τους μέσα στην κύρια καρβοξυλική αλυσίδα, είτε όταν γειτνιάζουν με δύο αμινοξέα καθένα από τα οποία συμμετέχει σε έναν υδρογονικό δεσμό. Οι τάσεις των αμινοξέων για τη συμμετοχή σε αυτές τις β-γέφυρες υπολογίζονται ως εξής:

$$\Omega_{AB}^{i} = ln(\frac{N_{AB}^{i>6}}{N_{AB}^{>6}}),$$

όπου τα A και B υποδηλώνουν ένα ζεύγος αμινοξέων, το i= I, II, III και αναφέρεται στον τύπο της γέφυρας, $N_{AB}^{i>6}$ είναι ο αριθμός των φορών όπου ένα ζευγάρι αμινοξέων συναντάται σε ένα σε μια β-γέφυρα τύπου i σε ένα σύνολο από πρωτεϊνικές δομές και $N_{AB}^{>6}$ είναι η συχνότητα με την οποία το συγκεκριμένο ζεύγος αμινοξέων ανήκει στις πρωτεϊνικές δομές σε εύρος

μεγαλύτερο από 6 αμινοξέα, ένα κατώφλι το οποίο χρησιμοποιείται για να εξαλείψει την πιθανότητα ύπαρξης πολύ μικρών στροφών στα δεδομένα. Ομοίως, οι τάσεις δύο αμινοξέων με απόσταση 4 για να σχηματίσουν υδρογονικούς δεσμούς α-έλικας υπολογίζεται ως:

$$\Omega_{AB}^{\alpha} = ln(\frac{N_{AB}^{\alpha,4}}{N_{AB}^{4}}),$$

όπου $N_{AB}^{\alpha,4}$ είναι ο αριθμός των αμινοξέων με υδρογονικούς δεσμούς τύπου A και B και N_{AB}^{4} η συχνότητα με την οποία τα αμινοξέα τύπου A και B βρίσκονται στην πρωτεϊνική αλυσίδα σε απόσταση τεσσάρων αμινοξέων.

3.3.2. Παραγωγή των τάσεων των αμινοξέων για δημιουργία αντιπαράλληλων και παράλληλων β-φύλλων

Για μια δοσμένη πρωτεϊνική αλυσίδα, μήκους N αμιονοξέων, υπολογίζουμε δύο πίνακες, έναν για την ανίχνευση αντιπαράλληλων και έναν άλλο για την ανίχνευση παράλληλων β-γεφυρών, τα στοιχεία των οποίων υπολογίζονται ως εξής:

$$M_{mn}^{Antipar} = max(\sum_{k=0}^{w-1} \Omega_{m+k,(N+W-1)-k}^{\substack{\{i=I,\alpha\nu\ k\ \acute{\alpha}\rho\tau\iota o\varsigma \\ i=II,\alpha\nu\ k\ \pi\varepsilon\rho\iota\tau\tau\acute{o}\varsigma\}}}, \sum_{k=0}^{w-1} \Omega_{m+k,(N+W-1)-k}^{\substack{\{i=I,\alpha\nu\ k\ \pi\varepsilon\rho\iota\tau\tau\acute{o}\varsigma \\ i=II,\alpha\nu\ k\ \acute{\alpha}\rho\tau\iota o\varsigma\}}}),$$

$$\mu\varepsilon\ m = [W, N-W]\ \kappa\alpha\iota\ n = [m+D, N-W]$$

και

$$M_{mn}^{Par} = max(\sum_{k=0}^{w-1} \Omega^{i=III}_{\substack{m+k,n+k\ \alpha\nu\ k\ \acute{\alpha}\rho\tau\iota o\varsigma \\ n+k,m+k\ \alpha\nu\ k\ \pi\varepsilon\rho\iota\tau\tau\acute{o}\varsigma}}, \sum_{k=0}^{w-1} \Omega^{i=III}_{\substack{m+k,n+k\ \alpha\nu\ k\ \pi\varepsilon\rho\iota\tau\tau\acute{o}\varsigma \\ n+k,m+k\ \alpha\nu\ k\ \acute{\alpha}\rho\tau\iota o\varsigma}}),$$

$$\mu\varepsilon\ m = [0, N-2W-D]\ \kappa\alpha\iota\ n = [m+D+W, N-W]$$

όπου τα m και n είναι οι αρχικές θέσεις των αμινοξέων δύο με αναδιπλούμενων τμημάτων μήκους W, το D είναι η ελάχιστη απόσταση μεταξύ των τερματικών τμημάτων διαδοχικών φύλλων(ίση με 6) και το i συμβολίζει τον τύπο της γέφυρας. Με άλλα λόγια τα $M_{mn}^{Antipar}$ και M_{mn}^{Par} δείχνουν την πιθανότητα δύο τμήματα της αλληλουχίας που βρίσκονται στις θέσεις m και n, να ανήκουν αντίστοιχα σε μια παράλληλη ή αντιπαράλληλη β-γέφυρα.

Η τελική μονοδιάστατη τάση για το σχηματισμό φύλλου με θέση εκκίνησης του παραθύρου την m, προκύπτει με τον εξής τρόπο:

Για τους αντιπαράλληλους και παράλληλους πίνακες που αναφέρθηκαν, βρίσκουμε τη μέγιστη τιμή κάθε στήλης, δηλαδή τους πίνακες:

$$\Pi_m^{Antipar} = max\ (M_{mn}^{Antipar})\ ,\ n=[0,N]$$

και

$$\Pi_m^{Par} = max\ (M_{mn}^{Par})\ ,\ n=[0,N]$$

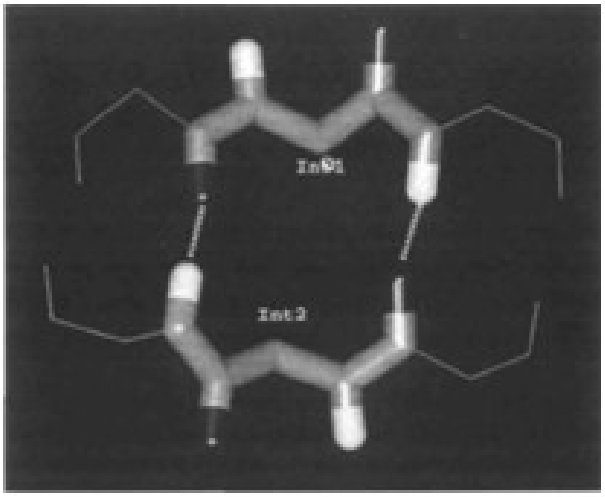
Κάθε αμινοξύ σε μια θέση m, παίρνει τη μέγιστη τιμή του πίνακα Π για κάθε παράθυρο που το περιλαμβάνει. Έτσι οι τάσεις για κάθε αμινοξύ βρίσκονται από τον τύπο:

$$P_m^{Antipar} = \max\ (\Pi_i^{Antipar}),\ \text{με i=[m-W+1,m]}$$

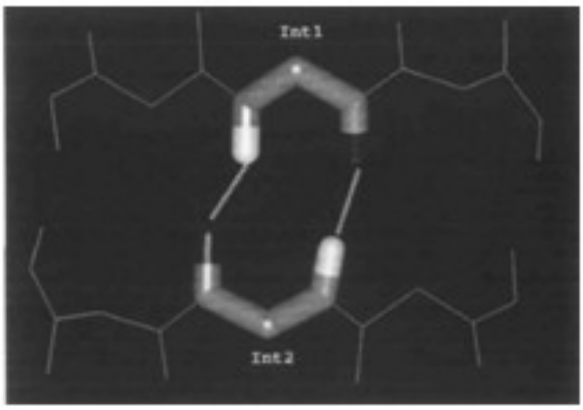
και

$$P_m^{Par} = \max\ (\Pi_i^{Par}),\ \text{με i=[m-W+1,m]}$$

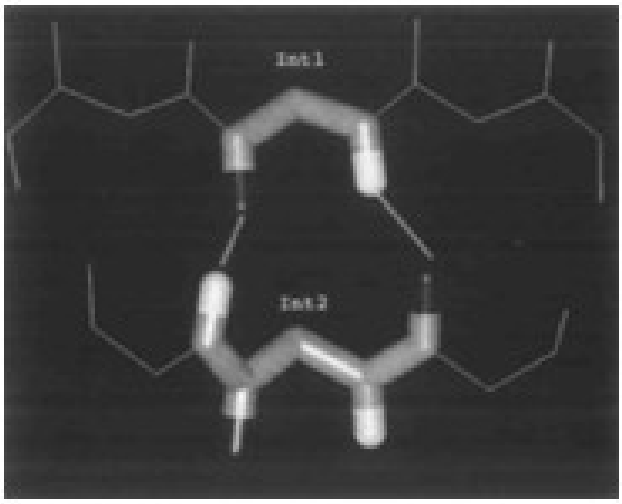
Ένα παράθυρο τεσσάρων αμινοξέων για τα β-φύλλα αποδεικνύεται στατιστικά ότι είναι βέλτιστο.



Εικόνα 3-2: Αντιπαράλληλη β-γέφυρα τύπου I



Εικόνα 3-3: Αντιπαράλληλη β-γέφυρα τύπου II



Εικόνα 3-4: Παράλληλη β-γέφυρα τύπου ΙΙΙ

3.3.3. Παραγωγή των τάσεων των αμινοξέων για δημιουργία α-ελίκων

Οι τιμές για τις τάσεις των αμινοξέων για το σχηματισμό α-έλικας παράγονται από τις τιμές του Ω_{AB}^{α}, βρίσκοντας απλά το μέσο όρο τους με τη χρήση ενός παραθύρου μήκους W=7.Δηλαδή,

$$P_m^{helix} = \frac{\sum_{k=0}^{W} \Omega_{m+k,m+k+4}^{\alpha}}{W}, \; m=[1,N-W-4]$$

3.3.4. Υπολογισμός τάσεων δευτεροταγούς δομής βασισμένων στη μέθοδο του πλησιέστερου γείτονα

Ένα σύνολο τάσεων p_m^{α} , p_m^{β} , p_m^{coil} για α-έλικες, β-φύλλα και σπείρες αντίστοιχα, προκύπτει από την εφαρμογή της μεθόδου του πλησιέστερου γείτονα, όπως αυτή περιγράφηκε από την ομάδα του Zhang (1992). Για κάθε τμήμα της αλληλουχίας με παράθυρο δεκατριών αμινοξέων, τα 25 ομοιότερα τμήματα με τη γνωστή δομή βρίσκονται στην πρωτεϊνική βάση δεδομένων. Η μέτρηση της ομοιότητας μεταξύ δύο πρωτεϊνικών τμημάτων βασίζεται στην στατιστική εξαγωγή της προτίμησης των αμινοξέων για τη συμμετοχή στις τρεις βασικές δομικές καταστάσεις (α-έλικα, β-φύλλα, σπείρα), καθώς και στην επιρροή που ασκούν τα γειτονικά αμινοξέα σε ένα παράθυρο μήκους 13 αμινοξέων. Οι τάσεις των πλησιέστερων γειτόνων για το κεντρικό αμινοξύ στο παράθυρο υπολογίζονται με βάση την ακόλουθη εξίσωση:

$$P_m^s = \frac{C_m^s}{Score_m^{Av,s}}, \; s=\alpha, \beta, coil,$$

όπου το C_m^s είναι το ποσοστό των 25 γειτονικών τμημάτων με κεντρικό αμινοξύ το m το οποίο βρίσκεται στην κατάσταση s και το $Score_m^{Av,s}$ συμβολίζει τη μέση γειτονική απόσταση του

συγκεκριμένου τμήματος από τους γείτονες που βρίσκονται στην κατάσταση s. Με αυτόν τον τρόπο, οι τάσεις είναι ανάλογες του αριθμού των ταυτίσεων ανάμεσα στους 25 «πλησιέστερους γείτονες» με τη γνωστή δομή.

3.3.5. Τάση για δημιουργία στροφής

Για κάθε τμήμα αποτελούμενο από 4 αμινοξέα, με θέση εκκίνησης την m σε μια δοσμένη πρωτεϊνική αλληλουχία, η τάση του για τη δημιουργία β-στροφών, υπολογίζεται από το μέσο όρο των πιθανοτήτων για τη δημιουργία στροφής που δημοσίευσαν οι Hutchinson και Thornton (1994), χρησιμοποιώντας ένα παράθυρο μήκους τεσσάρων αμινοξέων:

$$\Pi_i^{Turn} = T_i^I + T_{i+1}^{II} + T_{i+2}^{III} + T_{i+3}^{IV},$$

όπου $\boldsymbol{T^I, T^{II}, T^{III}}$ ***και*** $\boldsymbol{T^{IV}}$ είναι οι πιθανότητες να συμμετέχει ένα αμινοξύ που βρίσκεται στις θέσεις 1-4 αντίστοιχα, σε μια στροφή.

3.3.6. Κανόνες για την εύρεση της δευτεροταγούς δομής των στοιχείων της πρωτεΐνης

Το τελευταίο βήμα του αλγορίθμου περιλαμβάνει τον υπολογισμό των 7 τιμών των τάσεων για κάθε αμινοξύ στην πρωτεϊνική αλυσίδα. Τρεις από αυτές *($P_m^{antipar}$, P_m^{par} , P_m^{helix})* βασίζονται στην πιθανότητα ύπαρξης αμινοξέων που συνδέονται μεταξύ τους με υδρογονικούς δεσμούς, τρεις ακόμα *(P_m^{α} , P_m^{β} , P_m^{coil})* υπολογίζονται από την ομοιότητα της αλληλουχίας στην πρωτεΐνη με την άγνωστη δομή με τα τμήματα των πρωτεϊνών που ανήκουν στη βάση δεδομένων του αλγορίθμου. Τέλος, η P_m^{turn} συμπεριφέρεται ως καθαρά στατιστική πιθανότητα για να συμμετάσχει ένα συγκεκριμένο αμινοξύ σε β-στροφή. Οι παραπάνω υπολογιζόμενες τιμές πρέπει να μετατραπούν σε ένα μονοδιάστατο πίνακα ο οποίος θα παίρνει τις εξής τελικές καταστάσεις: *'H' (helix), 'E' (extended / sheet), 'C' (coil)*.

Για τον υπολογισμό των παραπάνω τιμών ακολουθούνται οι εξής κανόνες:

- **Κανόνας 1:** Αν ($P_m^{antipar} > \tau^{antipar}$ ή $P_m^{par} > \tau^{par}$) ΚΑΙ $P_m^{helix} > \tau^{helix}$, τότε προβλέπεται φύλλο. Αν $P_m^{helix} > \tau^{helix}$, τότε προβλέπεται έλικα, αλλιώς σπείρα.
- **Κανόνας 2:** Αν ($P_m^{coil} > \tau^{coil}$) , τότε προβλέπεται σπείρα.
- **Κανόνας 3:** Αν ($P_m^{\beta} > \tau^{\beta}$) , τότε προβλέπεται φύλλο.
- **Κανόνας 4:** Αν ($P_m^{\alpha} > \tau^{\alpha}$) , τότε προβλέπεται έλικα.
- **Κανόνας 5:** Αν ($P_m^{turn} > \tau^{turn}$) , τότε προβλέπεται σπείρα.

<u>Σημείωση:</u> Οι τιμές των κατωφλίων $\tau^{antipar}$, τ^{par} , τ^{helix}, τ^{coil}, τ^{α} και τ^{β}, έχουν εξαχθεί εμπειρικά για την επίτευξη βέλτιστης ακρίβειας στην πρόβλεψη της δομής.

4 Μοντελοποίηση αλγορίθμου PREDATOR

4.1 Γενικά

Στο κεφάλαιο αυτό γίνεται μελέτη της μοντελοποίησης του αλγορίθμου και διαχωρίζονται τα βασικά στάδια από τα οποία αποτελείται, με σκοπό τον πιο εύκολο σχεδιασμό της αρχιτεκτονικής στο επόμενο κεφάλαιο.

Εικόνα 4-1: Μοντελοποίηση του αλγορίθμου

4.2 Μοντελοποίηση σε software

Για τη μοντελοποίηση του αλγορίθμου PREDATOR σε λογισμικό, χρησιμοποιήθηκε η γλώσσα προγραμματισμού C, έτσι ώστε να είναι πιο εύκολη η μετάβαση στη γλώσσα περιγραφής υλικού VHDL. Ο τρόπος με τον οποίο δομήθηκε ο κώδικας ήταν τέτοιος ώστε να είναι χωρισμένος σε όσο το δυνατόν περισσότερα και διακριτά τμήματα, ώστε να είναι περισσότερο κατανοητή η λειτουργικότητά του και απλούστερη η περαιτέρω σχεδίασή του.

Είσοδος αλγορίθμου

Όλες οι είσοδοι του αλγορίθμου δίνονται σε αρχεία txt, τα οποία διαβάζονται από το πρόγραμμα με τη χρήση μιας ρουτίνας και στη συνέχεια αποθηκεύονται στην κύρια μνήμη του υπολογιστή. Οι είσοδοι που χρησιμοποιούνται είναι οι εξής:

1. Ακολουθία αμινοξέων της πρωτεΐνης για την οποία θέλουμε να βρούμε τη δευτεροταγή δομή της. Η πρωτεΐνη μπορεί να έχει οποιοδήποτε μήκος από 50 μέχρι και 10,000 αμινοξέα. Τα αμινοξέα μπορούν να πάρουν τις τιμές ARNDCQEGHILKMFPSTWYVXZB, όπου και στη συνέχεια μετατρέπονται στους αντίστοιχους ακεραίους ανάλογα με τη θέση που καταλαμβάνουν στο άνωθεν διάνυσμα (π.χ για A=0, R=1 κ.ο.κ).

2. 559 πρωτεϊνικές αλυσίδες γνωστής δομής , η οποία έχει καθοριστεί μετά από κρυσταλλογράφηση με ακτίνες X. Η δομή αυτή των πρωτεϊνών είναι με τη μορφή:

    - STRIDE, στην οποία γίνεται χρήση των ατομικών συντεταγμένων για την εύρεση των βασικών τμημάτων της δομής της πρωτεΐνης με βάση την ενέργεια των υδρογονικών δεσμών και τις διεδρικές γωνίες στην κύρια αλυσίδα [25], [27].
    - DSSP, η οποία είναι όμοια με τη STRIDE, όμως εδώ γίνεται χρήση μόνο της ενέργειας των υδρογονικών δεσμών [25], [27].

    Η δομή του αρχείου αυτού περιλαμβάνει για κάθε πρωτεΐνη την ονομασία της, την αλληλουχία των αμινοξέων της και τη δομή της. Μια ενδεικτική μορφή αρχείου είναι η εξής:

    NM 1cop.brk E
    SQ MEQRITLKDYAMRFGQTKTAKDLGVYQSAINKAIHAGRKIFLTIN……
    SS __EEEEHHHHHHHH_HHHHHHHH___HHHHHHHHHH___EEEE ……

    <u>Σημείωση:</u> Ο χαρακτήρας '_' αντιστοιχεί σε σπείρα, ο χαρακτήρας 'H' σε έλικα και ο 'E' σε φύλλο. Στον αλγόριθμο που υλοποιήθηκε σε C, τα εξαγόμενα αποτελέσματα αφορούν πρωτεΐνες που αναπαρίστανται με τη μορφή STRIDE. Πολύ απλά ο χρήστης θα μπορούσε να χρησιμοποιήσει και τη DSSP μορφή αν το επέλεγε στην αρχή του προγράμματος.

1. Τέσσερις πίνακες εξαρτήσεων (propensity tables), μεγέθους 20x20 με ακέραιες τιμές, που δίνουν την πιθανότητα για κάθε ζεύγος αμινοξέων να συμμετάσχει στο σχηματισμό:

    - Αντιπαράλληλων β-bridge τύπου I
    - Αντιπαράλληλων β-bridge τύπου II
    - Αντιπαράλληλων β-bridge τύπου III
    - Υδρογονικών δεσμών

2. Έναν τρισδιάστατο πίνακα 13x20x20 με ακέραιες τιμές, ο οποίος περιέχει τις εξαρτήσεις για κάθε ζεύγος που βρίσκεται σε ένα παράθυρο 13 αμινοξέων. Ο πίνακας αυτός χρησιμοποιείται στην εφαρμογή της μεθόδου του πλησιέστερου γείτονα (Nearest Neigbour, NN).

3. Έναν διάνυσμα επτά θέσεων, το οποίο περιλαμβάνει τις τιμές των κατωφλίων για την εφαρμογή των επτά κανόνων που θα χρησιμοποιηθούν για να γίνει η τελική πρόβλεψη της δομής της πρωτεΐνης.

4.3 Αναπαράσταση των αριθμών

4.3.1. Αριθμοί κινητής υποδιαστολής

Για την αναπαράσταση των αριθμών που χρησιμοποιεί ο αλγόριθμος, χρησιμοποιήθηκε η αριθμητική σταθερής υποδιαστολής (fixed-point arithmetic) [28]. Η αριθμητική σταθερής υποδιαστολής είναι μια μορφή αναπαράστασης πραγματικών αριθμών, οι οποίοι έχουν ένα προκαθορισμένο αριθμό ψηφίων πριν και μετά την (νοητή) υποδιαστολή. Ο λόγος που χρησιμοποιείται αυτή η αριθμητική, είναι διότι απαιτείται λιγότερο υπολογιστικό κόστος μιας και οι πράξεις που γίνονται μεταξύ των αριθμών είναι ουσιαστικά σε ακέραια μορφή και έτσι γίνεται αποφυγή της χρήσης κινητής υποδιαστολής.

Οι αριθμοί σταθερής υποδιαστολής είναι χρήσιμοι στο να αναπαριστούν κλασματικές τιμές, συνήθως με βάση το 2 ή το 10, με κόστος όμως την πιθανή απώλεια πληροφορίας. Για το λόγο αυτό έγιναν σχολαστικοί έλεγχοι για το αν η ακρίβεια που χάνεται μπορεί να μεταβάλλει την έξοδο του αλγορίθμου, γεγονός όμως που δεν ίσχυε.

4.3.2. Παράδειγμα πράξεων μεταξύ αριθμών κινητής υποδιαστολής

Για να μετατρέψουμε έναν αριθμό από πραγματικό (float) σε fixed, θα πρέπει να τον πολλαπλασιάσουμε με μια δύναμη του 2. Για το συγκεκριμένο αλγόριθμο επιλέχθηκε το 2^{16} = 65536, λόγω του ότι ήταν ο πιο μικρός αριθμός που μπορούσε να χρησιμοποιηθεί ώστε να μην έχουμε απώλειες στην ακρίβεια.

4.3.3. Πρόσθεση, αφαίρεση

Για την εκτέλεση της πρόσθεσης (και κατ' επέκταση της αφαίρεσης), προσθέτουμε τους αριθμούς ως ακέραιους, χωρίς να επέμβουμε στο αποτέλεσμα. Έστω ότι θέλουμε να προσθέσουμε τους αριθμούς fixed_1 και fixed_2. Τότε το αποτέλεσμα θα ισούται με:

fixed_3 = fixed_1 + fixed_2

4.3.4. Πολλαπλασιασμός

Για τον πολλαπλασιασμό, εκτελείται κανονικά η πράξη με τη διαφορά ότι το αποτέλεσμα ολισθαίνει δεξιά τόσες θέσεις όσες και η δύναμη του 2 που χρησιμοποιήθηκε για τη μετατροπή σε fixed point.

Στη συγκεκριμένη περίπτωση, το αποτέλεσμα ολισθαίνεται κατά 16 θέσεις. Δηλαδή έστω ότι θέλουμε να πολλαπλασιάσουμε τους αριθμούς fixed_1 και fixed_2. Τότε το αποτέλεσμα θα ισούται με:

fixed_3 = (fixed_1 * fixed_2) >> 16

4.3.5. Διαίρεση

Για τη διαίρεση, εκτελείται κανονικά η πράξη με τη διαφορά ότι ο διαιρετέος ολισθαίνει αριστερά τόσες θέσεις όσες και η δύναμη του 2 που χρησιμοποιήθηκε για τη μετατροπή σε fixed point.

Στη συγκεκριμένη περίπτωση, ο διαιρετέος ολισθαίνει κατά 16 θέσεις. Έστω ότι θέλουμε να διαιρέσουμε τους αριθμούς fixed_1 και fixed_2. Τότε το αποτέλεσμα θα ισούται με:

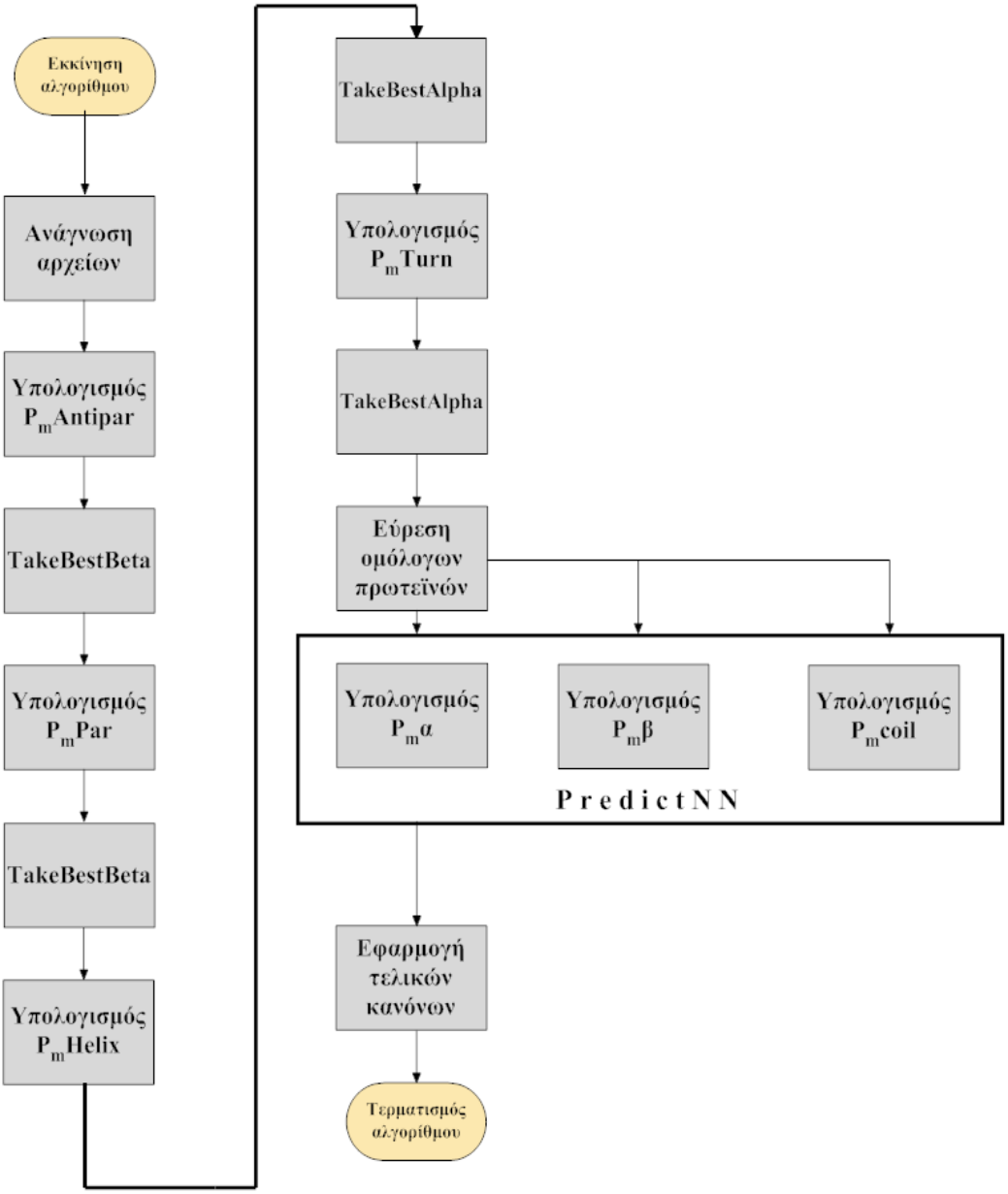
fixed_3 = (fixed_1 << 16) / fixed_2

4.4 Περιγραφή των βημάτων του αλγορίθμου

Στην εικόνα 4-2 αναπαρίσταται σχηματικά μια σύνοψη των ενεργειών που κάνει ο αλγόριθμος για να φτάσει στην πρόβλεψη της δομής μιας πρωτεΐνης. Θα χωρίσουμε τον αλγόριθμο σε επτά βήματα για την καλύτερη κατανόησή του. Τα βήματα αυτά είναι συνοπτικά:

- Ο υπολογισμός των επτά «τάσεων» (propensities)για κάθε ένα από τα αμινοξέα εισόδου:
  - $P_m^{Antipar}$
  - P_m^{Par}
  - P_m^{Helix}
  - P_m^{turn}
  - P_m^{α} , P_m^{β} , P_m^{coil}
- Η εύρεση των πρωτεϊνών που είναι ομόλογες με την πρωτεΐνη εισόδου.
- Η εφαρμογή των επτά κανόνων για την τελική πρόβλεψη της δομής.

<u>Σημείωση:</u> Θεωρούμε ότι N είναι το μήκος της πρωτεΐνη της οποίας θα προβλεφθεί η δομή και Iseq η ακολουθία εισόδου των αμινοξέων τα οποία έχουν μετατραπεί σε ακέραιους αριθμούς.



Εικόνα 4-2: Συνοπτική περιγραφή των βημάτων του αλγορίθμου

4.4.1 Υπολογισμός $P_m^{Antipar}$

- Στόχος: Υπολογισμός του Score για την πιθανότητα ύπαρξης αντιπαράλληλης γέφυρας μεταξύ των αμινοξέων και αποθήκευσή τους στο διάνυσμα Anti, με αριθμό στοιχείων ίσο με N.
- Είσοδος: Πίνακας εξαρτήσεων για αντιπαράλληλες β-γέφυρες τύπου I και II

Για τον υπολογισμό του $\mathbf{P_m^{Antipar}}$ (Εικόνα 4-3), παίρνουμε ανά τετράδες όλα τα ζεύγη αμινοξέων που έχουν απόσταση στην Iseq πάνω από 3. Τα ζεύγη αυτά έχουν τιμές:

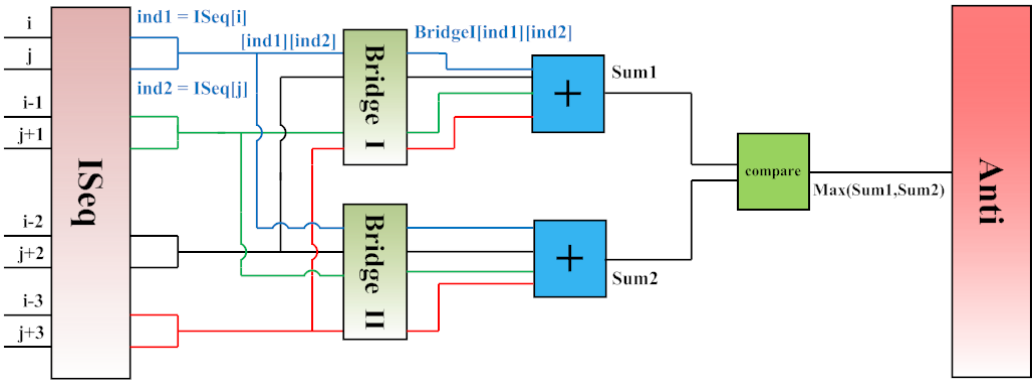
$$ind1 = Iseq[i - k]$$

$$\kappa\alpha\iota$$

$$ind2 = Iseq\,[\,j + k\,],$$

$$\mu\varepsilon\ i = [4\,,\,N\,)\,,\ j = [\,i{+}4,\ N{-}4\,)\ \kappa\alpha\iota\ k = [0,\,4)\,,\ \mu\varepsilon\ i,\,j,\,k \in N$$

Κάθε ζεύγος αμινοξέων ind1, ind2 αντιπροσωπεύει τις συντεταγμένες οι οποίες προσδιορίζουν από ένα στοιχείο στον πίνακα με τις αντιπαράλληλες β-γέφυρες τύπου I και II. Έπειτα, αθροίζουμε τις 4 παραπάνω τιμές, δημιουργώντας έτσι δύο αθροίσματα, τα Sum1 και Sum2, τα οποία περιλαμβάνουν τις τιμές από τους πίνακες Bridge I και Bridge II αντίστοιχα. Τέλος, βρίσκουμε το μεγαλύτερο από τα δύο αθροίσματα το οποίο και συγκρίνουμε με αυτά που θα προκύψουν από όλες τις τιμές των j. Με τον τρόπο αυτό προκύπτει το κάθε ένα από τα στοιχεία του διανύσματος Anti. Αφού εκτελεστούν όλες οι επαναλήψεις για όλες τις τιμές που παίρνει το i και συμπληρωθούν τα στοιχεία του Anti, καλείται η συνάρτηση TakeBestBeta, η οποία όπως θα δούμε διαμορφώνει κατάλληλα το διάνυσμα Anti για να είναι έτοιμο για την εφαρμογή των κανόνων του αλγορίθμου.



Εικόνα 4-3: Τρόπος εκτέλεσης τμήματος antipar (για μία επανάληψη)

4.4.2. Υπολογισμός P_m^{par}

- Στόχος: Υπολογισμός του Score για την πιθανότητα ύπαρξης παράλληλης γέφυρας μεταξύ των αμινοξέων και αποθήκευσή τους στο διάνυσμα Pari, με αριθμό στοιχείων ίσο με N.
- Είσοδος: Πίνακας εξαρτήσεων για παράλληλες β-γέφυρες τύπου III

Για τον υπολογισμό του P_m^{par}(Εικόνα 4-4), ακολουθούμε παρόμοια διαδικασία με αυτή για τον υπολογισμό του $P_m^{Antipar}$, με τη διαφορά ότι παίρνουμε ανά τετράδες όλα τα ζεύγη αμινοξέων που έχουν απόσταση στην Iseq πάνω από 9. Τα ζεύγη αυτά έχουν τιμές:

$$ind1 = Iseq[i + k]$$

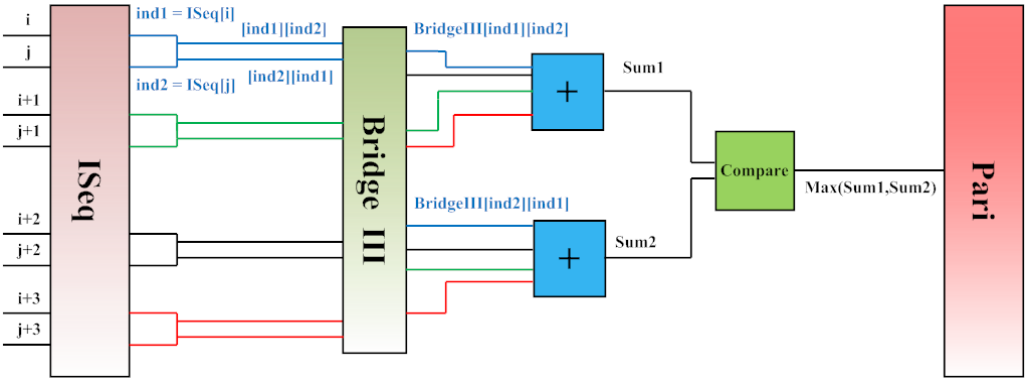
$$\kappa\alpha\iota$$

$$ind2 = Iseq\,[\,j + k\,],$$

$$\mu\varepsilon\ i = [0\,,\,N\text{-}4)\,,\ j = [\,i+9,\,N\text{-}4\,)\ \kappa\alpha\iota\ k = [0,\,4)\,,\ \mu\varepsilon\ i,\,j,\,k \in N$$

Ομοίως με το τμήμα Antipar, κάθε ζεύγος αμινοξέων αντιπροσωπεύει τις συντεταγμένες οι οποίες προσδιορίζουν τα στοιχεία στον πίνακα με την παράλληλη β-γέφυρα τύπου III. Στη συνέχεια, αθροίζουμε τις 4 τιμές για κάθε k και για συγκεκριμένα i, j, δημιουργώντας έτσι δύο αθροίσματα, τα Sum1 και Sum2, τα οποία περιλαμβάνουν τις τιμές από τον πίνακα Bridge III, με τη διαφορά ότι οι όροι του Sum1 είναι οι BridgeIII [ind1][ind2], ενώ για το Sum2 είναι οι BridgeIII [ind2][ind1]. Τέλος, βρίσκουμε το μεγαλύτερο από τα δύο αθροίσματα το οποίο και συγκρίνουμε με αυτά που θα προκύψουν από όλες τις δυνατές τιμές των j.

Με τον τρόπο αυτό προκύπτει το κάθε ένα από τα στοιχεία του διανύσματος Pari. Αφού εκτελεστούν όλες οι επαναλήψεις για όλες τις τιμές που παίρνει το i και συμπληρωθούν τα στοιχεία του Pari, καλείται η συνάρτηση TakeBestBeta την οποία και θα αναλύσουμε στη συνέχεια.



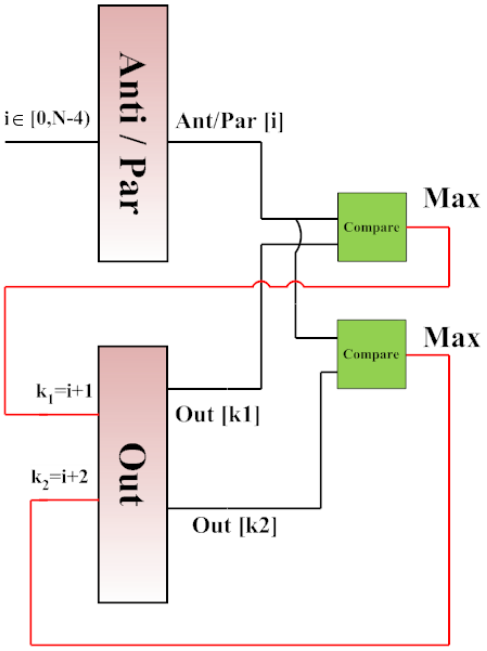
Εικόνα 4-4: Τρόπος εκτέλεσης τμήματος par (για μία επανάληψη)

4.4.3. TakeBestBeta

Αφού ολοκληρωθούν τα 2 παραπάνω βήματα και συμπληρωθούν τα διανύσματα Anti και Pari αντίστοιχα, καλούμε τη συνάρτηση TakeBestBeta (Εικόνα 4-5) η οποία μετασχηματίζει τα διανύσματα με τον εξής τρόπο:

(1) Αρχικοποιεί με μηδενικά ένα διάνυσμα Out μήκους N.
(2) Για i = [0,N-4) και k=[i+1, i+3), συγκρίνει τις τιμές των διανυσμάτων Anti / Par [i] και Out[k]. Αν το Anti/Par [i] είναι μεγαλύτερο από το Out[k], τότε το Out[k] παίρνει την τιμή του Anti/Par [i].

Αφού ολοκληρωθούν όλες οι επαναλήψεις για κάθε τιμή του i, το τελικό διάνυσμα Anti/Par εξισώνεται με το Out που έχει δημιουργηθεί προηγουμένως και έτσι έχουμε το τελικό διάνυσμα.

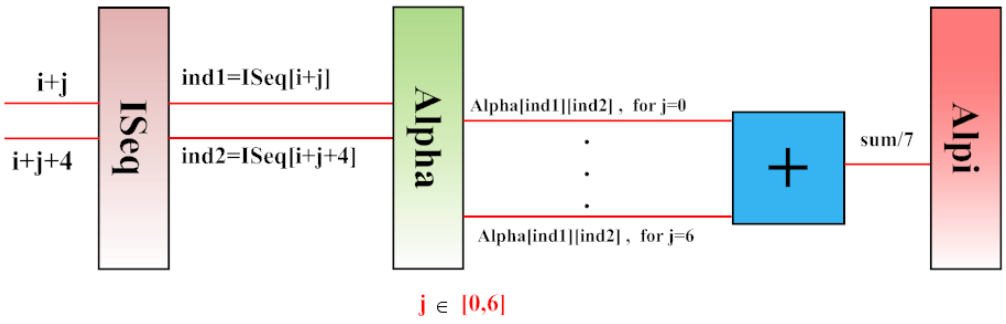


Εικόνα 4-5: Τρόπος εκτέλεσης τμήματος TakeBestBeta (για μία επανάληψη)

4.4.4. Υπολογισμός P_m^{helix}

- Στόχος: Υπολογισμός του Score για την ύπαρξη έλικας μεταξύ των αμινοξέων και αποθήκευσή του στο διάνυσμα Alpi.
- Είσοδος: πίνακας εξαρτήσεων Alpha για έλικες

Για τον υπολογισμό του διανύσματος Alpi (Εικόνα 4-6), υπολογίζουμε κάθε φορά τις τιμές ind1= ISeq [i + j] και ind2= ISeq [i + j + 4]. Για j ∈ [0,7) δημιουργούνται 7 ζευγάρια τιμών [ind1,ind2] τα οποία αποτελούν τις συντεταγμένες για την προσπέλαση στον πίνακα Alpha. Αφού αθροίσουμε τις τιμές Alpha[ind1][ind2] και για τις 7 τιμές του j και διαιρέσουμε το αποτέλεσμα με το 7, προκύπτει η μία τιμή για το διάνυσμα Alpi. Εκτελώντας τα παραπάνω για i ∈ [0,N-11) συμπληρώνονται όλα τα στοιχεία του Alpi και εν συνεχεία καλείται η συνάρτηση TakeBestAlpha την οποία και θα αναλύσουμε παρακάτω.



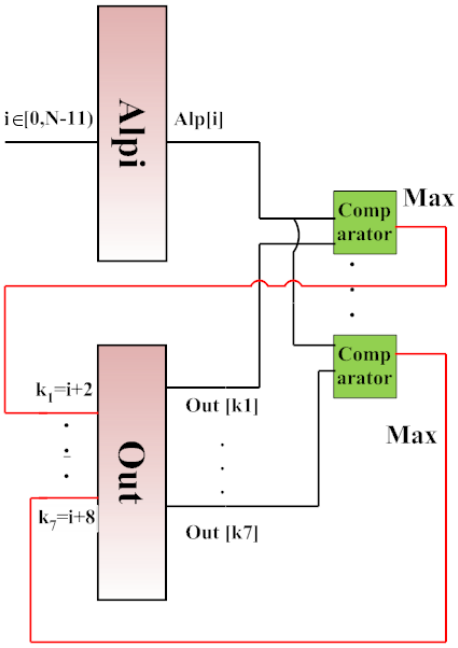
Εικόνα 4-6: Τρόπος εκτέλεσης τμήματος helix (για μία επανάληψη)

4.4.5. TakeBestAlpha

Αφού ολοκληρωθεί ο υπολογισμός των P_m^{helix} και συμπληρωθεί το διάνυσμα Alpi, καλούμε τη συνάρτηση TakeBestAlpha (Εικόνα 4-7) η οποία μετασχηματίζει το Alpi με τον εξής τρόπο:

(1) Αρχικοποιεί με μηδενικά ένα διάνυσμα Out μήκους N.
(2) Για i = [0,N-11) και k=[i+2, i+9), συγκρίνει τις τιμές των διανυσμάτων Alpi [i] και Out[k] για όλες τις τιμές του k. Αν το Alpi [i] είναι μεγαλύτερο από το Out[k], τότε το Out[k] παίρνει την τιμή του Alpi [i].

Αφού ολοκληρωθούν όλες οι επαναλήψεις για κάθε τιμή του i, το τελικό διάνυσμα Alpi εξισώνεται με το Out που έχει δημιουργηθεί προηγουμένως και έτσι έχουμε το τελικό διάνυσμα.



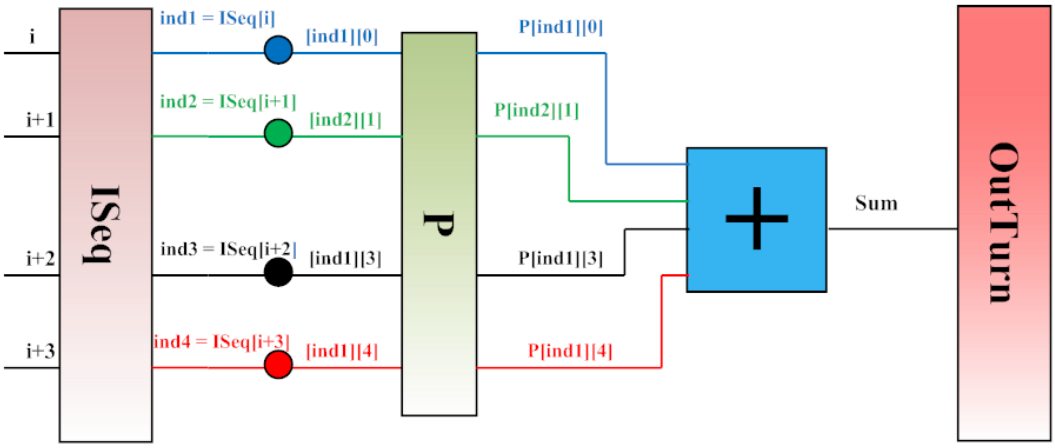
Εικόνα 4-7: Τρόπος εκτέλεσης τμήματος TakeBestAlpha (για μία επανάληψη)

4.4.6. Υπολογισμός P_m^{turn}

- Στόχος: Υπολογισμός του Score για την ύπαρξη turn μεταξύ των αμινοξέων και αποθήκευσή του στο διάνυσμα OutTurn.
- Είσοδος: πίνακας εξαρτήσεων για turn (αποθήκευση στον 20x20 πίνακα P)

Για τον υπολογισμό κάθε τιμής του διανύσματος OutTurn (Εικόνα 4-8), γίνεται χρήση ενός ολισθαίνοντος παραθύρου μήκους 4 αμινοξέων. Με αυτόν τον τρόπο παίρνουμε κάθε φορά τις τιμές ind1=ISeq[i], ind2=ISeq[i+1], ind3=ISeq[i+2], ind4=ISeq[i+3] και στη συνέχεια προσπελαύνουμε τον πίνακα P στις θέσεις P[ind1][0], P[ind2][1], P[ind3][2], P[ind4][3], όπου και αθροίζουμε τις τιμές που προκύπτουν. Η παραπάνω διαδικασία γίνεται μέχρι το παράθυρο

να φτάσει στο τέλος της ακολουθίας εισόδου όπου και τερματίζεται (κάθε φορά ολισθαίνει κατά μία θέση). Τέλος, καλείται η συνάρτηση TakeBestAlpha και έτσι το διάνυσμα OutTurn παίρνει την τελική του μορφή για την εφαρμογή των κανόνων.



Εικόνα 4-8: Τρόπος εκτέλεσης τμήματος Turn (για μία επανάληψη)

4.4.7. Εύρεση ομόλογων, με την αλληλουχία εισόδου, πρωτεϊνών

- Είσοδος: Πίνακας με τις 559 πρωτεΐνες γνωστής δομής.
- Έξοδος: Το διάνυσμα Homol, 559 θέσεων, στο οποίο γράφεται 1 σε περίπτωση που βρεθεί πρωτεΐνη ομόλογη με την αρχική και 0 σε αντίθετη περίπτωση. Επίσης το διάνυσμα KnownAsn μήκους ίσου με αυτό της εισόδου, στο οποίο γράφεται η δομή της επτάδας που ήταν όμοια με αυτή της εισόδου.

Για να βρεθούν οι ομόλογες πρωτεΐνες με την αρχική, συγκρίνουμε όλες τις πιθανές επτάδες διαδοχικών αμινοξέων της πρωτεΐνης εισόδου, με όλες τις πιθανές επτάδες κάθε πρωτεΐνης της βάσης. Αν σε μία πρωτεΐνη βάσης βρεθούν πάνω από 13 όμοιες επτάδες, τότε η πρωτεΐνη αυτή θεωρείται ομόλογη με την αρχική. Στη συνέχεια, για την ομόλογη πρωτεΐνη, γράφουμε στο διάνυσμα KnownAsn τη δομή της στη θέση εκείνη που εμφάνισε ομοιότητα με την πρωτεΐνη εισόδου. Οι υπόλοιπες θέσεις του KnownAsn συμπληρώνονται με '?'. Στους πίνακες 4-1 και 4-2 γίνεται ένα παράδειγμα για τη καλύτερη κατανόηση του τρόπου εύρεσης των ομόλογων πρωτεϊνών. Πιο συγκεκριμένα, στον πίνακα 4-1 σημειώνονται με χρώμα τα σημεία εκείνα που υπάρχει ομοιότητα μεταξύ της πρωτεΐνης βάσης και της εισόδου.

Πίνακας 4-1: Παράδειγμα εύρεσης ομόλογης πρωτεΐνης με αυτή της εισόδου (1/2)

Πρωτεΐνη βάσης	...KTSAATLWRHWPRIKKARPDLPVLVMSAQNTFMTAIKASEKGAYDYLPKPFDLTEL...
Πρωτεΐνη εισόδου	...WPRIKKAMQTDLTLMITGAYDYLPKPFDLTELKTSAATLWRHGRALAEPKR...
Δομή πρωτεΐνης βάσης	____EEEE___EEEEEE__EEEEEE_HHHHHHH__EEEEE_____EEE____HHHHHHH...

Αφού η πρωτεΐνη βάσης έχει 14 ομοιότητες με την πρωτεΐνη εισόδου, θεωρείται ομόλογη. Επομένως, στο τελευταίο βήμα αντιγράφουμε τη δομή της στις θέσεις που παρατηρήθηκε ομοιότητα και το διάνυσμα KnownAsn φαίνεται στον πίνακα 4-2.

Πίνακας 4-2: Παράδειγμα εύρεσης ομόλογης πρωτεΐνης με αυτή της εισόδου (2/2)

Πρωτεΐνη εισόδου	**...**WPRIKKAMQTDLTLMITGAYDYLPKPFDLTELKTSAATLWRHGRALAEPKR...
KnownAsn	...____EEEE ??????????EEEEE_HHHHHHH__EEEEE_____??????????...

4.4.8. PredictNN

- Είσοδος: Ο πίνακας με την αλληλουχία των 559 πρωτεϊνών γνωστής δομής , ο πίνακας με τη δομή τους και ο πίνακας MbrMat, ο οποίος περιέχει τις εξαρτήσεις για κάθε ζεύγος που βρίσκεται σε ένα παράθυρο 13 αμινοξέων.
- Έξοδος: Τα διανύσματα ALpNN, BetaNN και CoilNN τα οποία θα χρησιμοποιηθούν για την εφαρμογή των κανόνων.

Για κάθε αμινοξύ της εισόδου ακολουθείται η εξής διαδικασία:

Γίνεται προσπέλαση του MbrMat στις θέσεις f0 = MbrMat[0][p1[0]], f1 = MbrMat[1][p1[1]] ,... , f12 = MbrMat[12][p1[12]], με p1[0] το τρέχον στοιχείο της εισόδου, p1[1] το επόμενο κ.ο.κ.

Έπειτα, γίνεται αρχικοποίηση των διανυσμάτων Score, Structure και Position τα οποία είναι 25 θέσεων. Για κάθε αμινοξύ της βάσης δημιουργείται το άθροισμα CurrScore, το οποίο ισούται με f0[p2[0]] + f1[p2[1]] + ... + f12[p2[12]], όπου p2[0] το τρέχον αμινοξύ της βάσης, p2[1] το επόμενο κ.ο.κ.

Αν το παραπάνω υπολογιζόμενο άθροισμα είναι μικρότερο από τη μεταβλητή CurrMax (αρχικοποιημένη σε 98304), τότε στο διάνυσμα Score (στη θέση πο υποδεικνύει η μεταβλητή IndMax, την οποία θα εξηγήσουμε αργότερα), γράφεται το CurrMax, στο διάνυσμα Structure (στην ίδια θέση με το Score) γράφεται ο αριθμός της πρωτεΐνης βάσης όπου βρέθηκε και στο διάνυσμα Position γράφεται η θέση του αμινοξέος στη συγκεκριμένη πρωτεΐνη βάσης.

Στη συνέχεια σαρώνεται το διάνυσμα Score και βρίσκεται η μέγιστη τιμή η οποία πρέπει να είναι ταυτόχρονα μεγαλύτερη της CurrMax. Η θέση που βρέθηκε η τιμή αυτή ανατίθεται στη μεταβλητή IndMax.

Η παραπάνω διαδικασία επαναλαμβάνεται για όλα τα αμινοξέα όλων των πρωτεϊνών της βάσης. Αφού ολοκληρωθεί, για κάθε στοιχείο του παραγόμενου διανύσματος Score, αναζητούμε τη δομή που έχει στη θέση StrDb[Structure[i]][Position[i]], όπου i η θέση που εξετάζουμε κάθε φορά στο διάνυσμα Score.

Έπειτα, γίνεται καταμέτρηση του πλήθους των 'H','E','C' που αντιστοιχούν στο διάνυσμα Score και αποθήκευσή του στις μεταβλητές count_H, count_E και count_C αντίστοιχα. Για κάθε 'H' που βρίσκεται, αυξάνεται η τιμή μιας μεταβλητής score_H κατά score_H + score[i]. Αντίστοιχα, για κάθε 'E' και 'C' αυξάνεται η μεταβλητή score_E κατά score_E + score[i] και score_C κατά score_C + score[i].

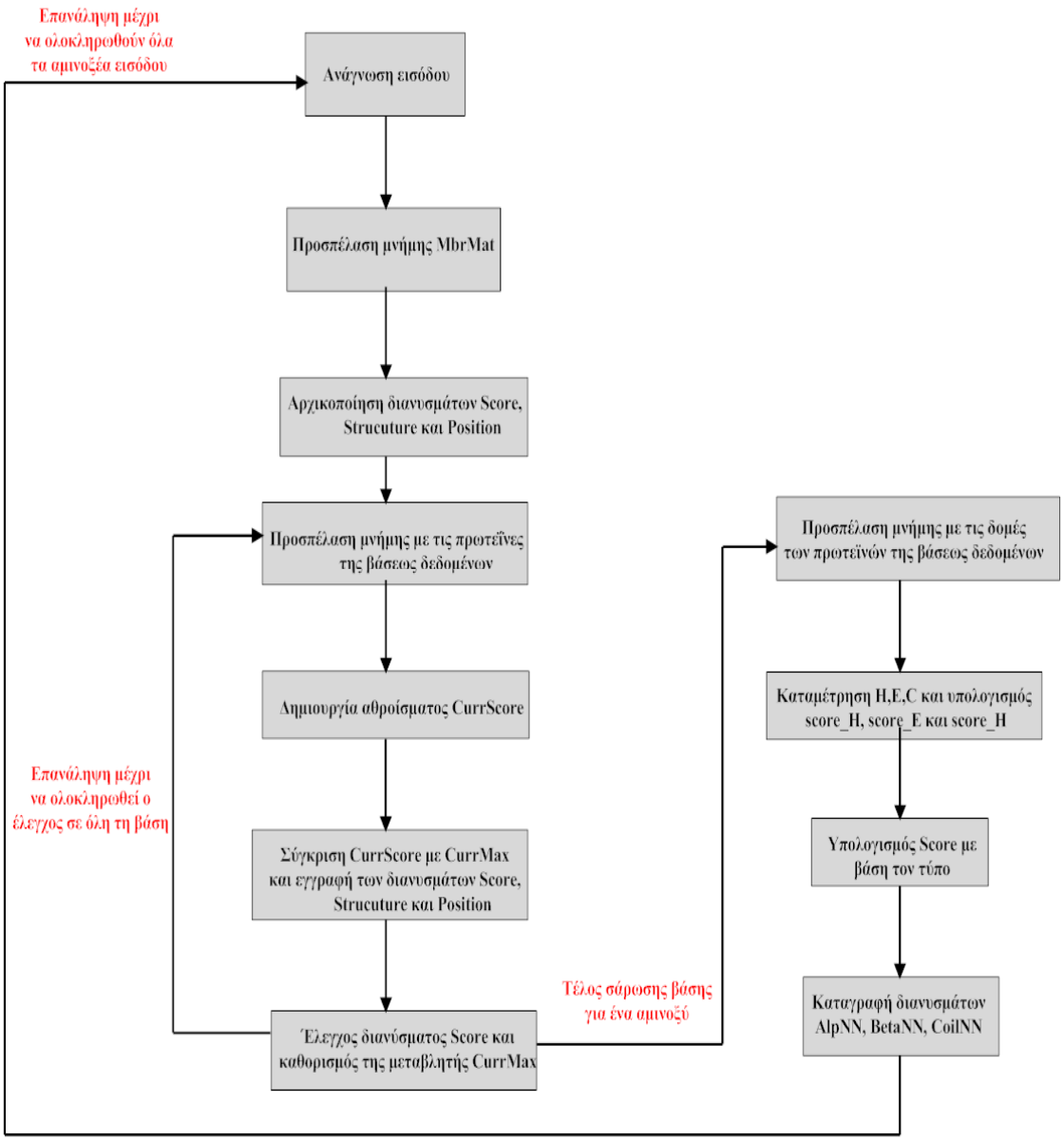
Στο τελευταίο βήμα, υπολογίζεται η τιμές που παίρνουν τα διανύσματα AlpNN, BetaNN και CoilNN στη θέση i, οι οποίες είναι ίσες με:

$$AlpNN\ [i] = \frac{count_H^2}{15\ x\ Score_H}$$

$$BetaNN\ [i] = \frac{count_E^2}{15\ x\ Score_E}$$

$$CoilNN\ [i] = \frac{count_C^2}{15\ x\ Score_C}$$

Η παραπάνω διαδικασία εκτελείται για όλα τα αμινοξέα εισόδου και εν τέλει συμπληρώνονται τα διανύσματα εξόδου AlpNN, BetaNN και CoilNN (Εικόνα 4-9).



Εικόνα 4-9: Συνοπτική περιγραφή των βημάτων για το τμήμα PredictNN του αλγορίθμου

4.4.9. Εφαρμογή κανόνων για πρόβλεψη της δομής

- Είσοδος: Το διάνυσμα Threshold με τις τιμές $\tau^{Antipar}$, τ^{par}, τ^{helix}, τ^{coil}, τ^{α}, τ^{β} , τ^{Turn} να είναι τα κατώφλια που χρησιμοποιούμε για την εφαρμογή των κανόνων. Επίσης όλα τα διανύσματα που υπολογίστηκαν στα προηγούμενα βήματα του αλγορίθμου, δηλαδή τα Anti, Par, Alpi, OutTurn, AlpNN, BetaNN, CoilNN και το KnownAsn.
- Έξοδος: Το διάνυσμα FinalPrediction στο οποίο γράφεται η τελική πρόβλεψη του αλγορίθμου στη εξής μορφή.

H = έλικα

E = φύλλο

C = σπείρα

Για την ολοκλήρωση της πρόβλεψης της δομής της πρωτεΐνης, εφαρμόζουμε τους εξής 7 κανόνες :

Κανόνας 1.

3 Για i = 0 μέχρι i < N

3.6 Αν Alp [i] ≥ τ^{helix}, FinalPrediction[i] = helix αλλιώς

3.7 Αν Anti[i] ≥ $\tau^{Antipar}$ ή Par[i] ≥ τ^{Par} FinalPrediction[i] = sheet αλλιώς

3.7.1.1.1 FinalPrediction[i] = coil

Για κάθε στοιχείο του διανύσματος Alpi, αν αυτό είναι μεγαλύτερο από το τ^{helix}, τότε στην αντίστοιχη θέση στο διάνυσμα FinalPrediction, γράφεται 'H'. Διαφορετικά, αν κάποιο στοιχείο του Anti ή του Par είναι μεγαλύτερο από το $\tau^{antipar}$ και τ^{par} αντίστοιχα, τότε στην αντίστοιχη θέση στο διάνυσμα FinalPrediction γράφεται 'E'.

Κανόνας 2.

Κάνοντας χρήση ενός παραθύρου μήκους έξι, σαρώνουμε το διάνυσμα AlpNN και βρίσκουμε το μέσο όρο των έξι αυτών τιμών. Αν ο μέσος όρος είναι μεγαλύτερος από το τ^{a}, τότε στις αντίστοιχες θέσεις με αυτές που καταλαμβάνει το παράθυρο, γράφουμε στο διάνυσμα FinalPrediction το 'H'.

Κανόνας 3.

Χρησιμοποιώντας την ίδια λογική με τον κανόνα 2, κάνοντας χρήση ενός παραθύρου μήκους τέσσερα, σαρώνουμε το διάνυσμα BetaNN και βρίσκουμε το μέσο όρο των τεσσάρων αυτών τιμών. Αν ο μέσος όρος είναι μεγαλύτερος από το τ^{β}, τότε στις αντίστοιχες θέσεις με αυτές που καταλαμβάνει το παράθυρο, γράφουμε στο διάνυσμα FinalPrediction το 'E'.

Κανόνας 4.

Χρησιμοποιώντας την ίδια λογική με τους δύο παραπάνω κανόνες, κάνοντας χρήση ενός παραθύρου μήκους τρία, σαρώνουμε το διάνυσμα CoilNN και βρίσκουμε το μέσο όρο των τριών αυτών τιμών. Αν ο μέσος όρος είναι μεγαλύτερος από το τ^{coil}, τότε στις αντίστοιχες θέσεις με αυτές που καταλαμβάνει το παράθυρο, γράφουμε στο διάνυσμα FinalPrediction το 'C'.

Κανόνας 5.

Παίρνοντας όλες τις πιθανές διαδοχικές τετράδες στο διάνυσμα OutTurn, αν κάθε τιμή της τετράδας είναι μεγαλύτερη από το τ^{Turn} τότε στο διάνυσμα FinalPrediction στη θέση όπου βρίσκεται η τετράδα, γράφεται το 'C'.

Κανόνας 6.

Στις τρεις πρώτες και στις τρεις τελευταίες θέσεις του διανύσματος FinalPrediction γράφεται το 'C'.

Κανόνας 7.

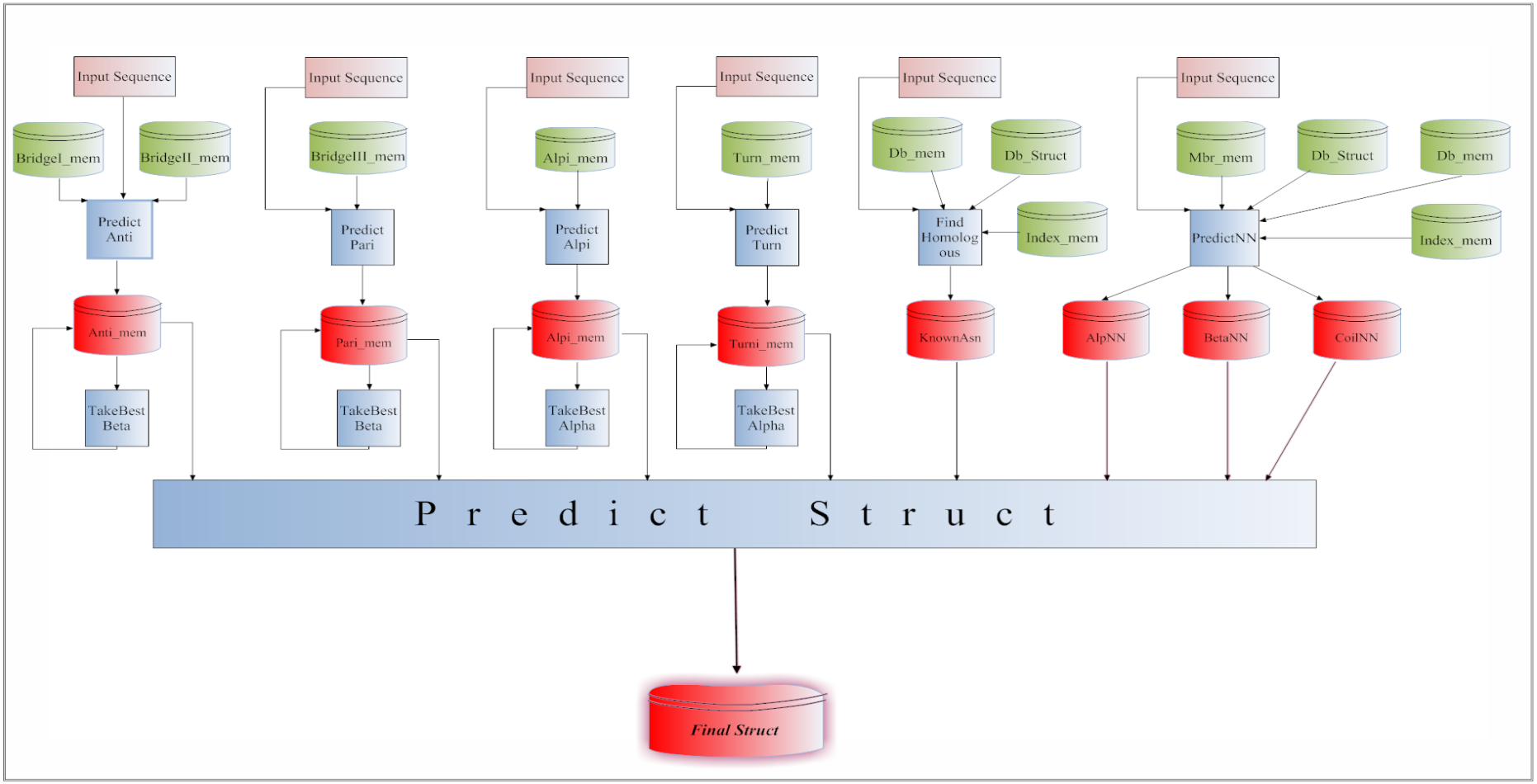
Στις θέσεις όπου το διάνυσμα KnownAsn έχει τιμή διάφορη του '?', τότε αντιγράφουμε τις τιμές του στο διάνυσμα FinalPrediction.

5 Αρχιτεκτονική αλγορίθμου PREDATOR

5.1 Γενικά

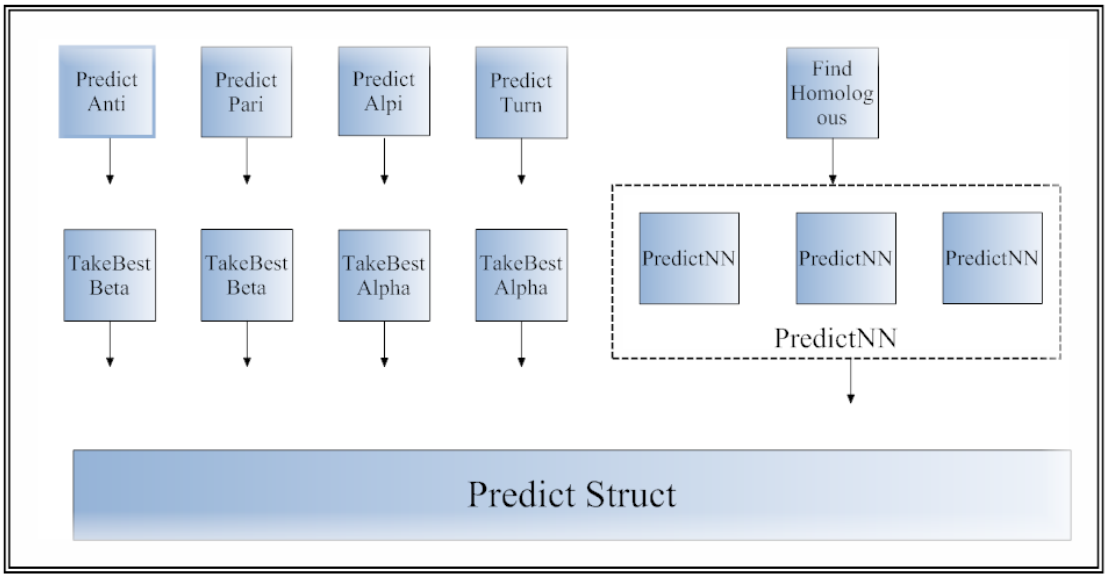
Στο κεφάλαιο αυτό παρουσιάζεται η αρχιτεκτονική του αλγορίθμου PREDATOR, με ιδιαίτερη έμφαση στην ανάλυση των βασικών υπολογιστικών κομματιών και στις σχεδιαστικές επιλογές που ακολουθήθηκαν.

Μια συνοπτική απεικόνιση της αρχιτεκτονικής που ακολουθήθηκε φαίνεται στην εικόνα 5-1.



Εικόνα 5-2: Αρχιτεκτονική αλγορίθμου PREDATOR

Στην εικόνα 5-2 αναπαρίσταται η χρονολογική σειρά που εκτελούνται τα τμήματα της σχεδίασης. Έτσι παρατηρούμε ότι τα τμήματα PredictAnti, PredictPari, PredictAlpi, PredictTurn και FindHomologous ξεκινούν την ίδια χρονική στιγμή, τα τμήματα TakeBestBeta, TakeBestBeta, TakeBestAlpha, TakeBestBeta και PredictNN ξεκινούν όταν τελειώσουν τα αντίστοιχα προηγούμενά τους και το τμήμα PredictStruct εκτελείται τελευταίο.



Εικόνα 5-2: Χρονολογική αναπαράσταση εκτέλεσης των τμημάτων της αρχιτεκτονικής του αλγορίθμου

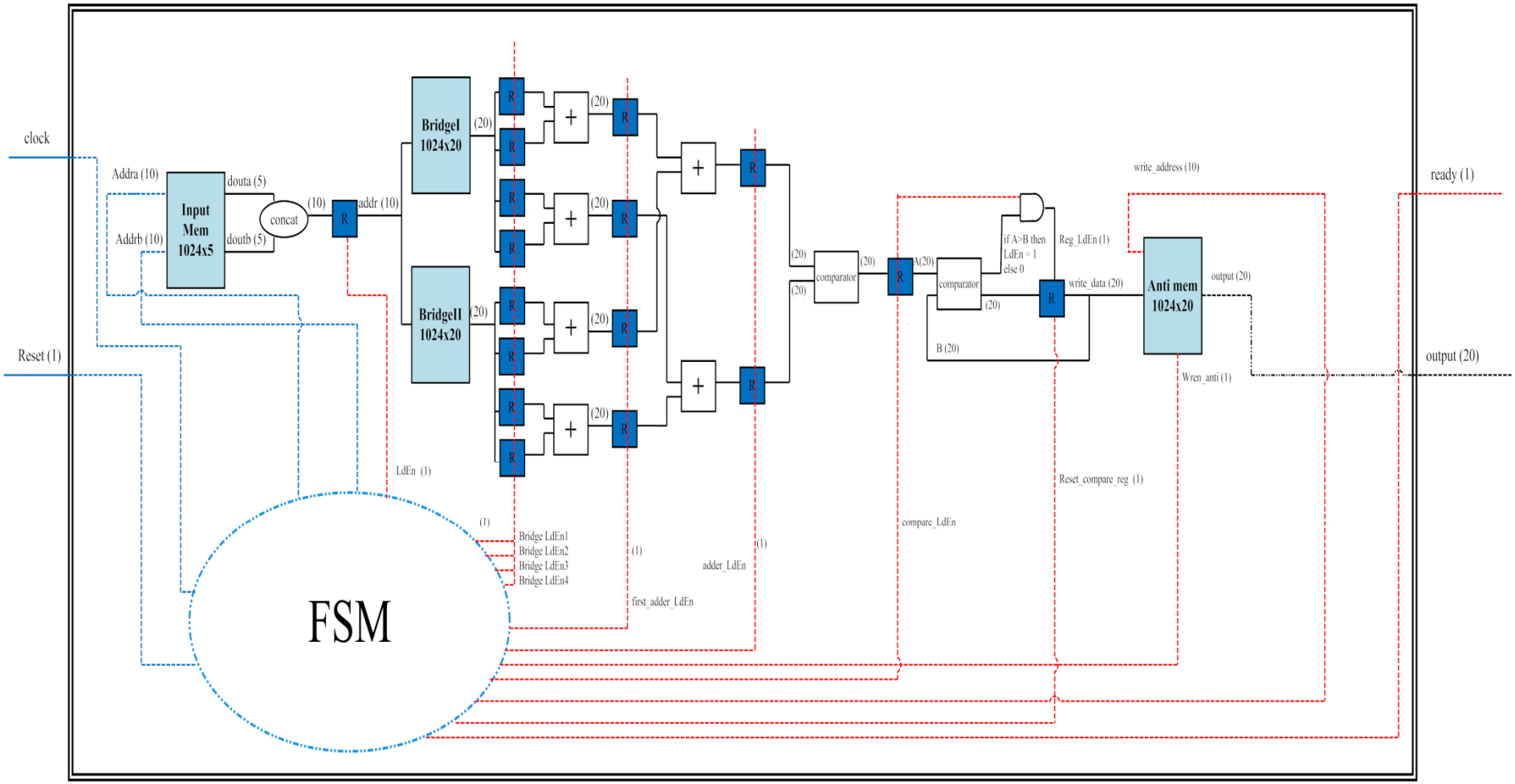
Στις επόμενες ενότητες θα αναλύσουμε κάθε κομμάτι της αρχιτεκτονικής ξεχωριστά, περιγράφοντας τον τρόπο με τον οποίο αυτά σχεδιάστηκαν.

5.2 Τμήμα Antipar

Η αρχιτεκτονική του τμήματος Antipar (εικόνα 5-3) χωρίζεται σε επτά στάδια:

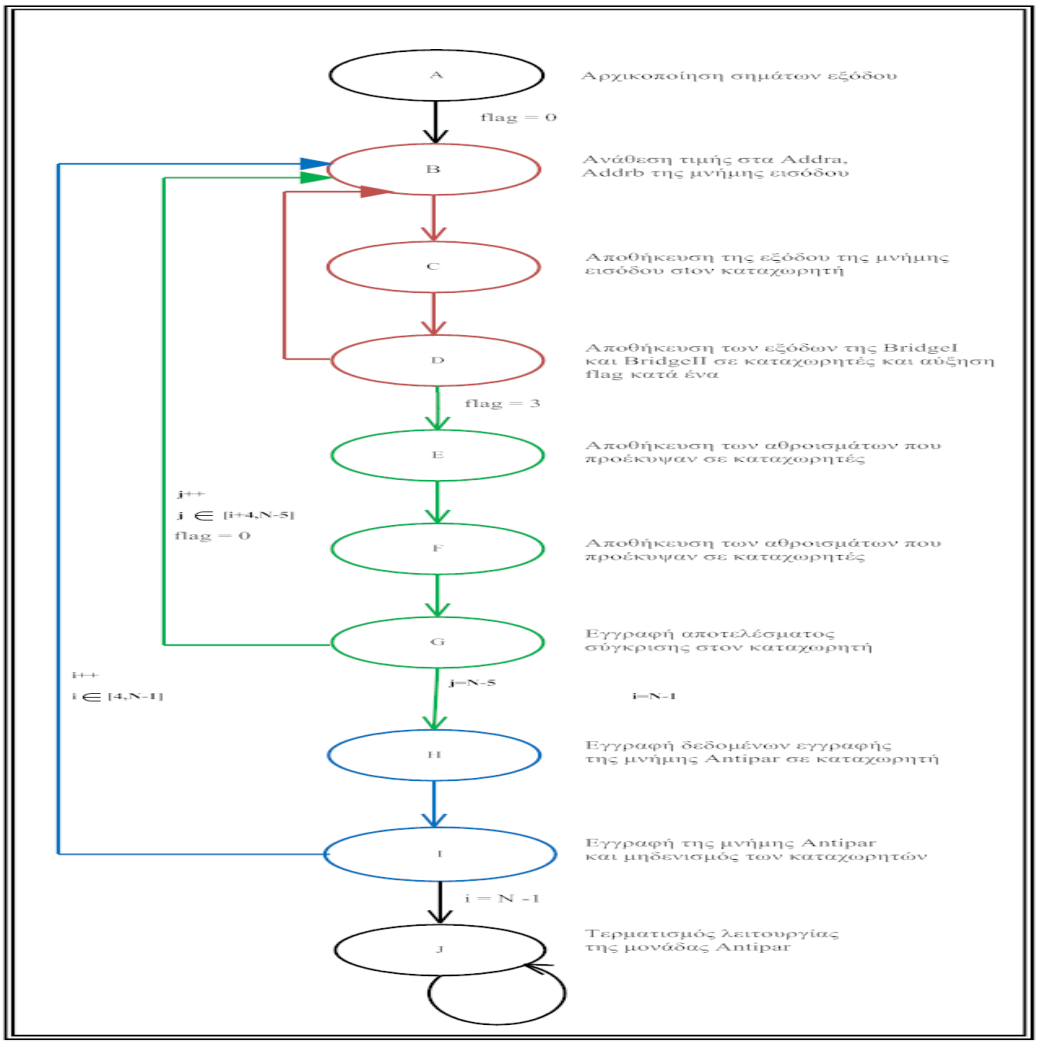
1ο στάδιο: Αρχικά γίνεται ανάγνωση της μνήμης εισόδου με βάση τις διευθύνσεις που δίνονται από την FSM ελέγχου. Οι έξοδοι της μνήμης, αφού ενωθούν σε ένα διάνυσμα των 10 bits, λειτουργούν ως διεύθυνση για την προσπέλαση των μνημών BridgeI και BridgeII.

2ο στάδιο: Γίνεται ανάγνωση των μνημών BridgeI και BridgeII και εγγραφή των εξόδων των μνημών στους καταχωρητές.
3ο στάδιο: Αθροίζονται οι τιμές των εξόδων του δεύτερου σταδίου ανά δύο. Δηλαδή αθροίζουμε μεταξύ τους τις εξόδους των καταχωρητών 1 και 2, καθώς και αυτές των καταχωρητών 3 και 4 για τις μνήμες BridgeI και BridgeII αντίστοιχα. Τέλος, εγγράφουμε τα αποτελέσματα σε καταχωρητές.
4ο στάδιο: Αθροίζουμε τις εξόδους του πρώτου με τον τρίτο καταχωρητή, καθώς και του δεύτερου με τον τέταρτο και εγγράφουμε τα αποτέλεσμα σε δύο νέους καταχωρητές.
5ο στάδιο: Γίνεται σύγκριση των εξόδων των δύο προαναφερθέντων καταχωρητών και εγγράφεται σε έναν άλλο καταχωρητή ο μεγαλύτερος από τους δύο αριθμούς.
6ο στάδιο: Γίνεται σύγκριση της εξόδου του παραπάνω καταχωρητή με αυτή που είχε στον προηγούμενο κύκλο ρολογιού. Αν η έξοδός του στον τρέχοντα κύκλο είναι μεγαλύτερη από αυτή στον προηγούμενο, τότε γράφεται στον καταχωρητή η νέα τιμή.
7ο στάδιο: Γίνεται εγγραφή στη μνήμη Anti_mem με διεύθυνση και σήμα ενεργοποίησης εγγραφής που προέρχονται από την FSM ελέγχου και δεδομένα εγγραφής την έξοδο του καταχωρητή.



Εικόνα 5-3: Αρχιτεκτονική τμήματος Antipar

Για να λειτουργήσει σωστά το μονοπάτι δεδομένων είναι απαραίτητη η παρουσία μιας μονάδας ελέγχου. Η μονάδα αυτή αποτελείται από σύνολο πεπερασμένων καταστάσεων (Finite State Machines-FSMs), οι οποίες είναι οργανωμένες σε ιεραρχία. Το σύνολο των μηχανών αυτών δημιουργεί τα σωστά σήματα για τους πολυπλέκτες στις εισόδους και στις εξόδους των βασικών υπολογιστικών μονάδων σε κάθε φάση λειτουργίας της αρχιτεκτονικής καθώς και για εγγραφή ή ανάγνωση από τις μνήμες.



Εικόνα 5-4: Μονάδα ελέγχου για το τμήμα Antipar

Η FSM ελέγχου (εικόνα 5-4) για το συγκεκριμένο κομμάτι, χωρίζεται σε 3 στάδια:

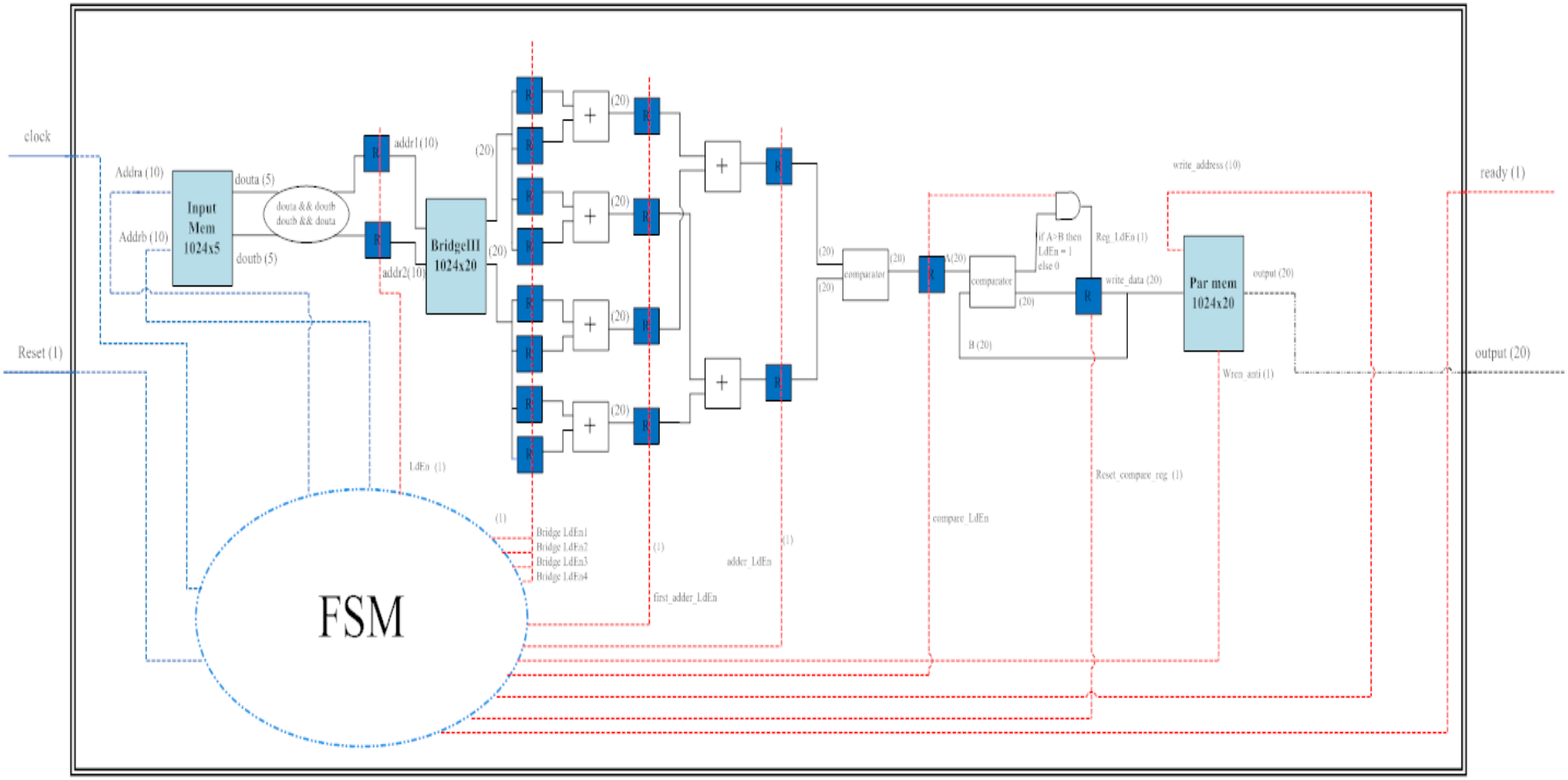
Το **πρώτο** στάδιο (B-D) είναι ουσιαστικά αυτό που καθορίζει το βασικό υπολογιστικό κομμάτι όπου γίνεται ανάγνωση της μνήμης εισόδου και έπειτα των μνημών BridgeI και BridgeII. Στη συνέχεια, υπολογίζεται το άθροισμα των τεσσάρων τιμών που προκύπτουν από κάθε μνήμη οι οποίες αντιστοιχούν σε μία συγκεκριμένη τιμή i, j.
Στο **δεύτερο** στάδιο (E-G) γίνεται η σύγκριση των διαφόρων τιμών που υπολογίστηκαν στο πρώτο στάδιο για κάθε τιμή του συνόλου j και για μια ορισμένη τιμή του i. Έπειτα, γίνεται αποθήκευση του αποτελέσματος της σύγκρισης σε καταχωρητή.

Στο **τρίτο** και τελευταίο στάδιο, γίνεται εγγραφή της μνήμης Antipar με το μεγαλύτερο αριθμό που έχει προκύψει από τις συγκρίσεις και ουσιαστικά ολοκληρώνεται η i-οστή επανάληψη.

5.3 Τμήμα Par

Όπως και στο κομμάτι Antipar, έτσι και στο Par, η αρχιτεκτονική του (εικόνα 5-5) χωρίζεται σε επτά στάδια:

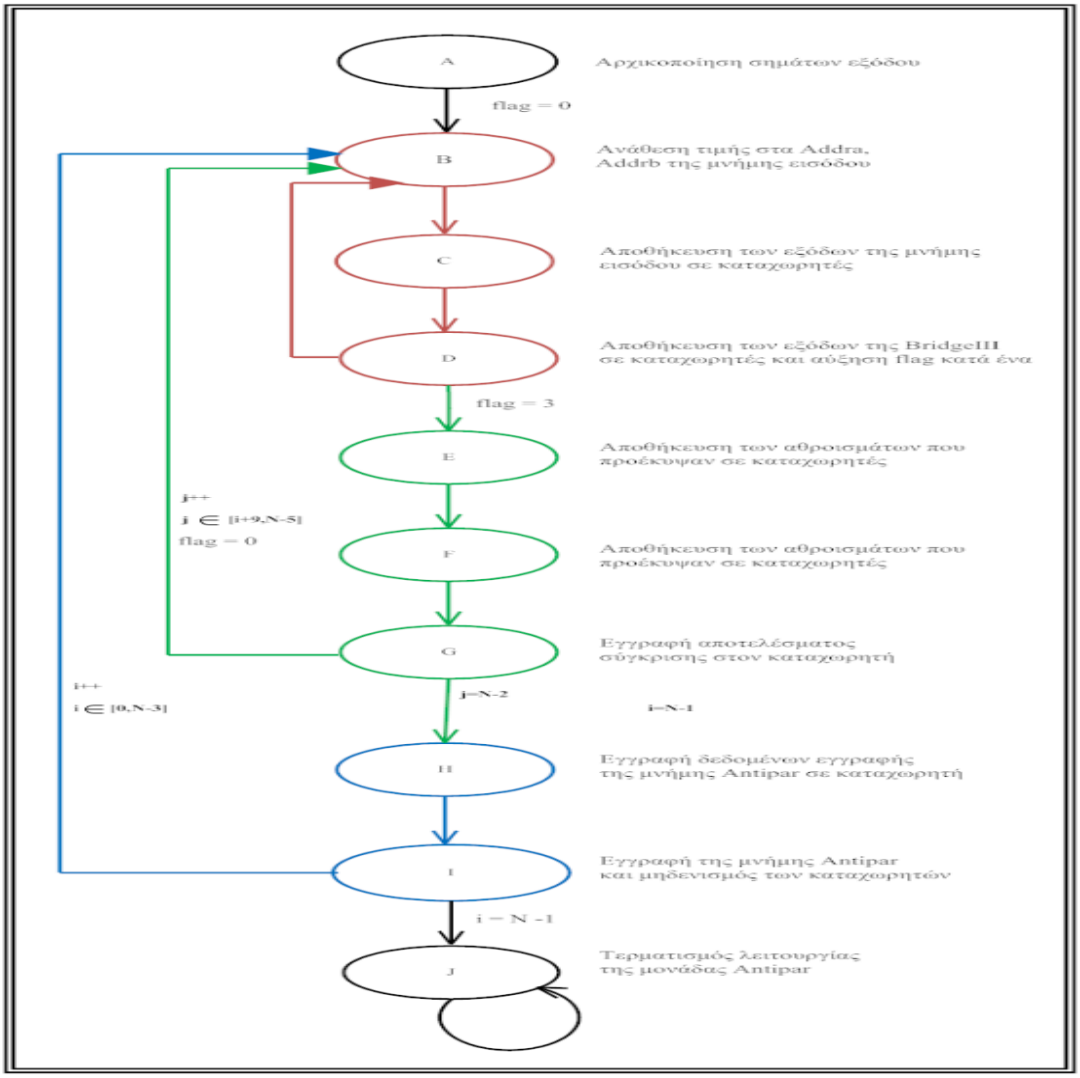
1ο στάδιο: Αρχικά γίνεται ανάγνωση της μνήμης εισόδου με βάση τις διευθύνσεις που δίνονται από την FSM ελέγχου. Οι έξοδοι της μνήμης, αφού ενωθούν σε δύο διανύσματα των 10 bits, με τη δεύτερη έξοδο να ακολουθεί την πρώτη και αντίστροφα, λειτουργούν ως διευθύνσεις για την προσπέλαση της μνήμης BridgeIII.
2ο στάδιο: Γίνεται ανάγνωση της μνήμης BridgeIII και εγγραφή των εξόδων της στους καταχωρητές.
3ο στάδιο: Αθροίζονται οι τιμές των εξόδων του δεύτερου σταδίου ανά δύο. Δηλαδή αθροίζουμε μεταξύ τους τις εξόδους των καταχωρητών 1 και 2, καθώς και αυτές των καταχωρητών 3 και 4 που προέρχονται από τη BridgeIII. Τέλος, εγγράφουμε τα αποτελέσματα σε καταχωρητές.
4ο στάδιο: Αθροίζουμε τις εξόδους του πρώτου με τον τρίτο καταχωρητή, καθώς και του δεύτερου με τον τέταρτο και εγγράφουμε τα αποτέλεσμα σε δύο νέους καταχωρητές.
5ο στάδιο: Γίνεται σύγκριση των εξόδων των δύο προαναφερθέντων καταχωρητών και εγγράφεται σε έναν άλλο καταχωρητή ο μεγαλύτερος από τους δύο αριθμούς.
6ο στάδιο: Γίνεται σύγκριση της εξόδου του παραπάνω καταχωρητή με αυτή που είχε στον προηγούμενο κύκλο ρολογιού. Αν η έξοδός του στον τρέχοντα κύκλο είναι μεγαλύτερη από αυτή στον προηγούμενο, τότε γράφεται στον καταχωρητή η νέα τιμή.
7ο στάδιο: Γίνεται εγγραφή στη μνήμη Par_mem με διεύθυνση και σήμα ενεργοποίησης εγγραφής που προέρχονται από την FSM ελέγχου και δεδομένα εγγραφής την έξοδο του καταχωρητή.



Εικόνα 5-5: Αρχιτεκτονική τμήματος Par

Η FSM ελέγχου (εικόνα 5-6) χωρίζεται σε 3 στάδια:

Το **πρώτο** στάδιο (B-D) είναι ουσιαστικά αυτό που καθορίζει το βασικό υπολογιστικό κομμάτι όπου γίνεται ανάγνωση της μνήμης εισόδου και έπειτα της μνήμης BridgeIII. Στη συνέχεια, υπολογίζεται το άθροισμα των οκτώ τιμών που προκύπτουν από τη μνήμη οι οποίες αντιστοιχούν σε μία συγκεκριμένη τιμή i, j.
Στο **δεύτερο** στάδιο (E-G) γίνεται η σύγκριση των διαφόρων τιμών που υπολογίστηκαν στο πρώτο στάδιο για κάθε τιμή του συνόλου j και για μια ορισμένη τιμή του i. Έπειτα, γίνεται αποθήκευση του αποτελέσματος της σύγκρισης σε καταχωρητή.
Στο **τρίτο** και τελευταίο στάδιο, γίνεται εγγραφή της μνήμης Par με το μεγαλύτερο αριθμό που έχει προκύψει από τις συγκρίσεις και ουσιαστικά ολοκληρώνεται η i-οστή επανάληψη.

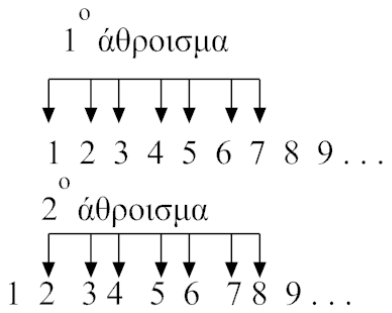


Εικόνα 5-6: Μονάδα ελέγχου για το τμήμα Par

5.4 Τμήμα Alpi

Η αρχιτεκτονική του τμήματος Alp (εικόνα 5-7) χωρίζεται σε τέσσερα στάδια:

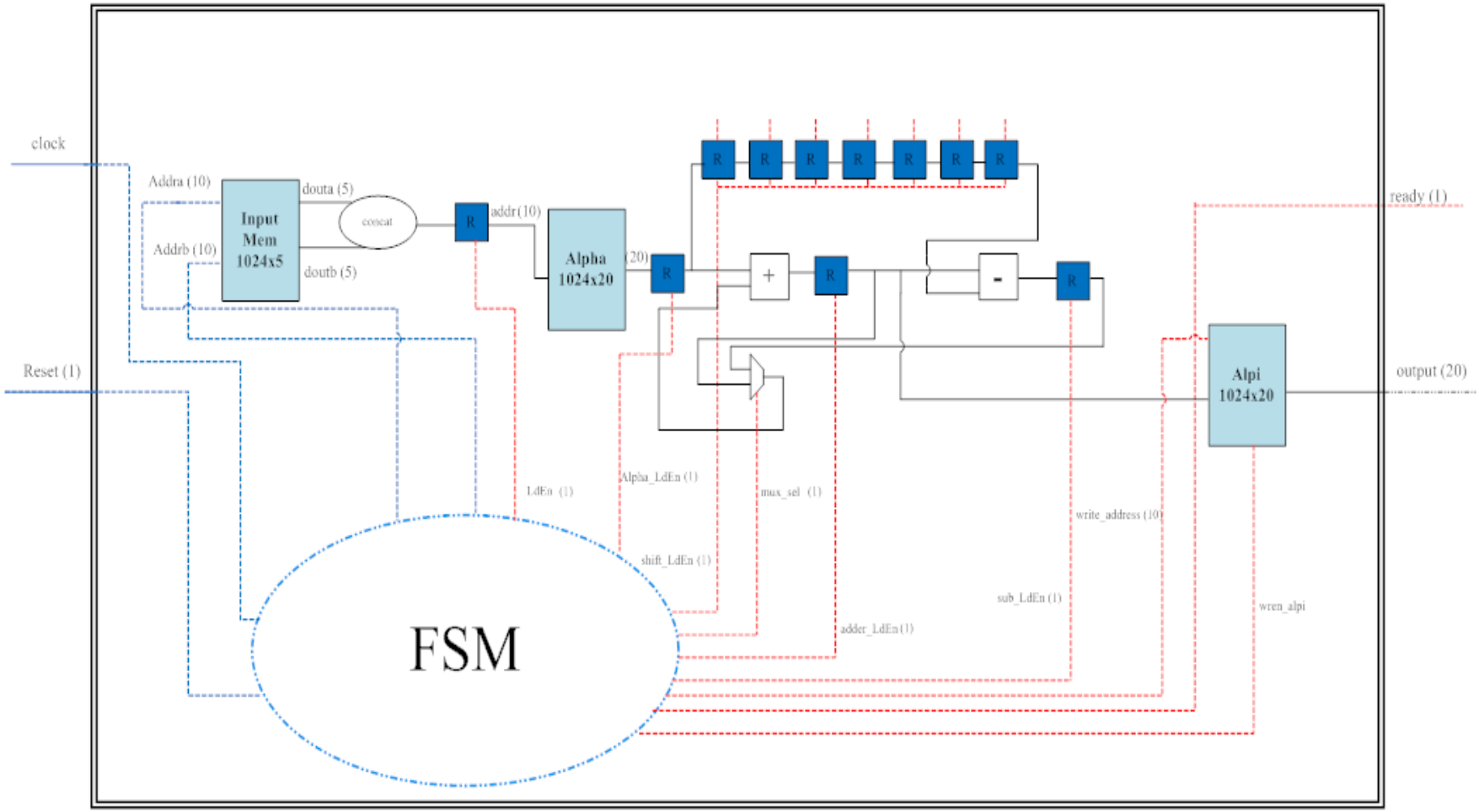
1ο στάδιο: Αρχικά γίνεται ανάγνωση της μνήμης εισόδου με βάση τις διευθύνσεις που δίνονται από την FSM ελέγχου. Οι έξοδοι της μνήμης, αφού ενωθούν σε ένα διάνυσμα των 10 bits, λειτουργούν ως διεύθυνση για την προσπέλαση της μνήμης Alpha.
2ο στάδιο: Γίνεται ανάγνωση της μνήμης Alpha και εγγραφή της εξόδου της μνήμης σε καταχωρητή.
3ο στάδιο: Αποθηκεύεται η έξοδος του 2ου σταδίου σε έναν καταχωρητή ολίσθησης 7 θέσεων. Ο λόγος που συμβαίνει αυτό, είναι ότι κάθε φορά διαβάζονται επτά τιμές από τη μνήμη Alpha και αθροίζονται μεταξύ τους. Στην επόμενη επανάληψη, αφαιρούμε την τιμή που διαβάστηκε πρώτη στην προηγούμενη επανάληψη και προσθέτουμε μια νέα (εικόνα 5-7).



Εικόνα 5-7: Παράδειγμα με τους όρους της πρόσθεσης για δύο διαδοχικές επαναλήψεις

Με τον τρόπο αυτό εξοικονομούνται 6 κύκλοι ρολογιού σε κάθε επανάληψη λόγω του ότι διαβάζεται από τη μνήμη μία τιμή κάθε φορά κι όχι επτά (προφανώς εξαιρείται η πρώτη επανάληψη, όπου διαβάζονται διαδοχικά οι τιμές από τη μνήμη Alpha μέχρι να «γεμίσουν» οι καταχωρητές ολίσθησης).

4ο στάδιο: Γίνεται εγγραφή στη μνήμη Alpi το άθροισμα που υπολογίστηκε στο 3ο στάδιο, με διεύθυνση και σήμα ενεργοποίησης εγγραφής που προέρχονται από την FSM ελέγχου και δεδομένα εγγραφής την έξοδο του καταχωρητή.

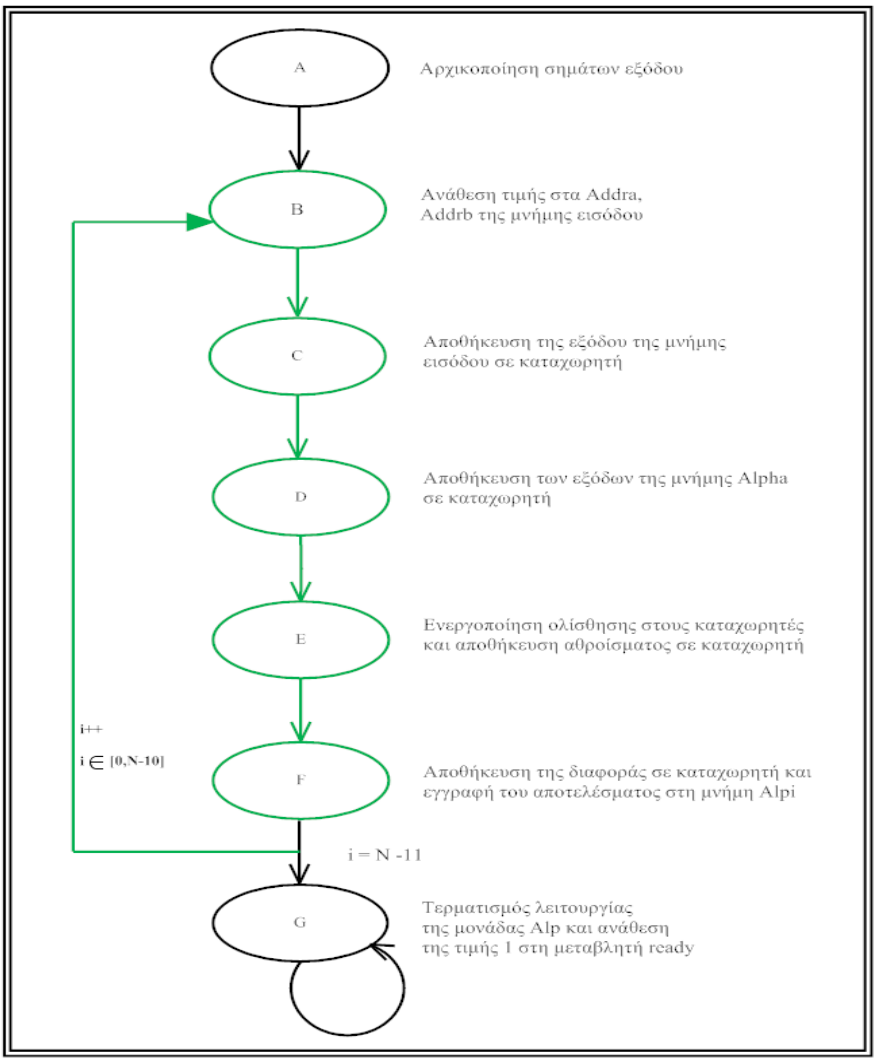


Εικόνα 5-8: Αρχιτεκτονική τμήματος Alpi

Η FSM ελέγχου (εικόνα 5-9) χωρίζεται σε 2 στάδια.

Το **πρώτο** στάδιο (B-F) είναι ουσιαστικά αυτό που καθορίζει το βασικό υπολογιστικό κομμάτι όπου γίνεται ανάγνωση της μνήμης εισόδου και έπειτα της μνήμης Alpha. Στη συνέχεια, υπολογίζεται το άθροισμα των τεσσάρων τιμών που προκύπτουν από κάθε μνήμη οι οποίες αντιστοιχούν σε μία συγκεκριμένη τιμή i, j. Έπειτα, γίνεται η αφαίρεση της εξόδου του καταχωρητή ολίσθησης από την έξοδο του αθροιστή και το αποτέλεσμα γράφεται σε καταχωρητή. Στον επόμενο κύκλο ρολογιού, προστίθεται στο αποτέλεσμα της αφαίρεσης η τιμή που διαβάστηκε από τη μνήμη Alpha και το άθροισμα γράφεται σε καταχωρητή.

Στο **δεύτερο** στάδιο, γίνεται εγγραφή της μνήμης Alpi με το άθροισμα που έχει υπολογιστεί προηγουμένως και ουσιαστικά ολοκληρώνεται η i-οστή επανάληψη.



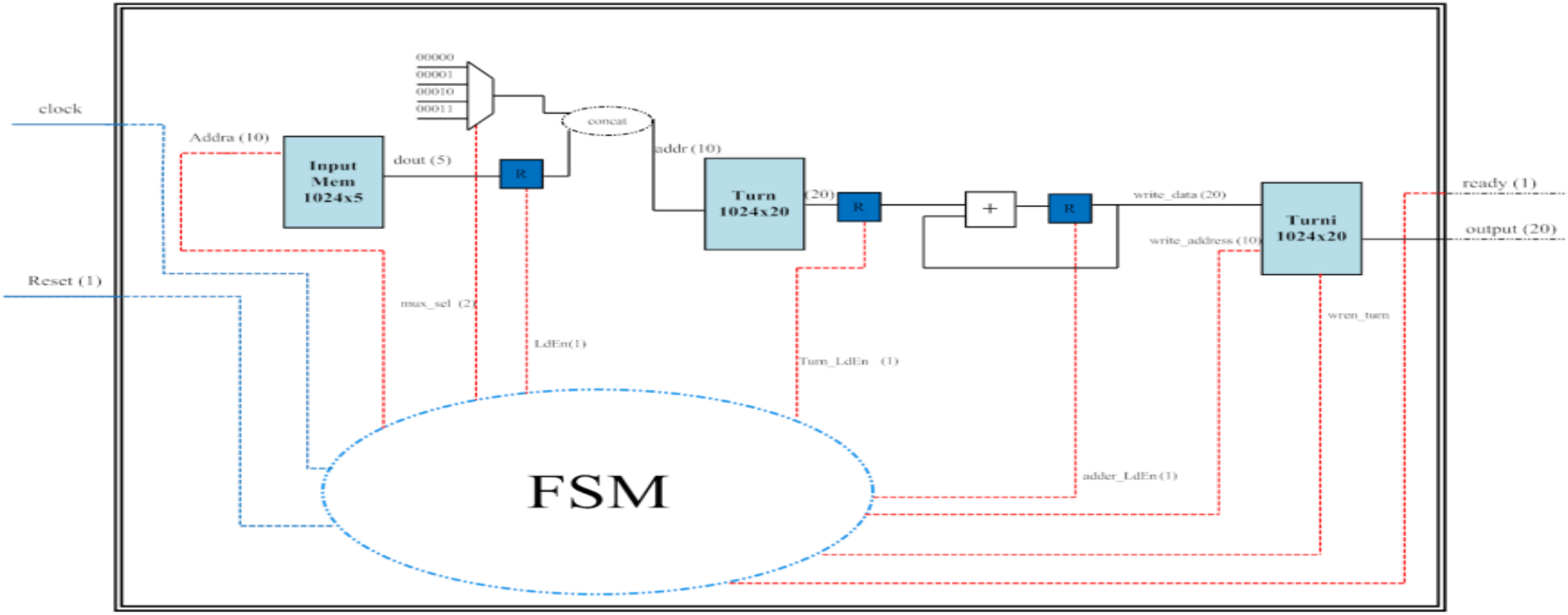
Εικόνα 5-9: Μονάδα ελέγχου για το τμήμα Alpi

5.5 Τμήμα Turni

Η αρχιτεκτονική του τμήματος Turni (εικόνα 5-10) χωρίζεται σε τέσσερα στάδια:

1^{ο} στάδιο: Αρχικά γίνεται ανάγνωση της μνήμης εισόδου με βάση τις διευθύνσεις που δίνονται από την FSM ελέγχου. Η έξοδος της μνήμης, αφού ενωθεί με την έξοδο του πολυπλέκτη σε ένα διάνυσμα των 10 bits, λειτουργεί ως διεύθυνση για την προσπέλαση της μνήμης Turn.
2^{ο} στάδιο: Γίνεται ανάγνωση της μνήμης Turn και εγγραφή της εξόδου της μνήμης σε καταχωρητή.
3^{ο} στάδιο: Εκτελείται η πρόσθεση της εξόδου του 2^{ου} σταδίου με αυτή του 3^{ου} και το αποτέλεσμα αποθηκεύεται σε καταχωρητή. Ουσιαστικά, για κάθε τιμή που εγγράφεται στη μνήμη Turni, απαιτούνται τέσσερις προσθέσεις, οπότε τα στάδια 1-3 επαναλαμβάνονται 4 φορές.

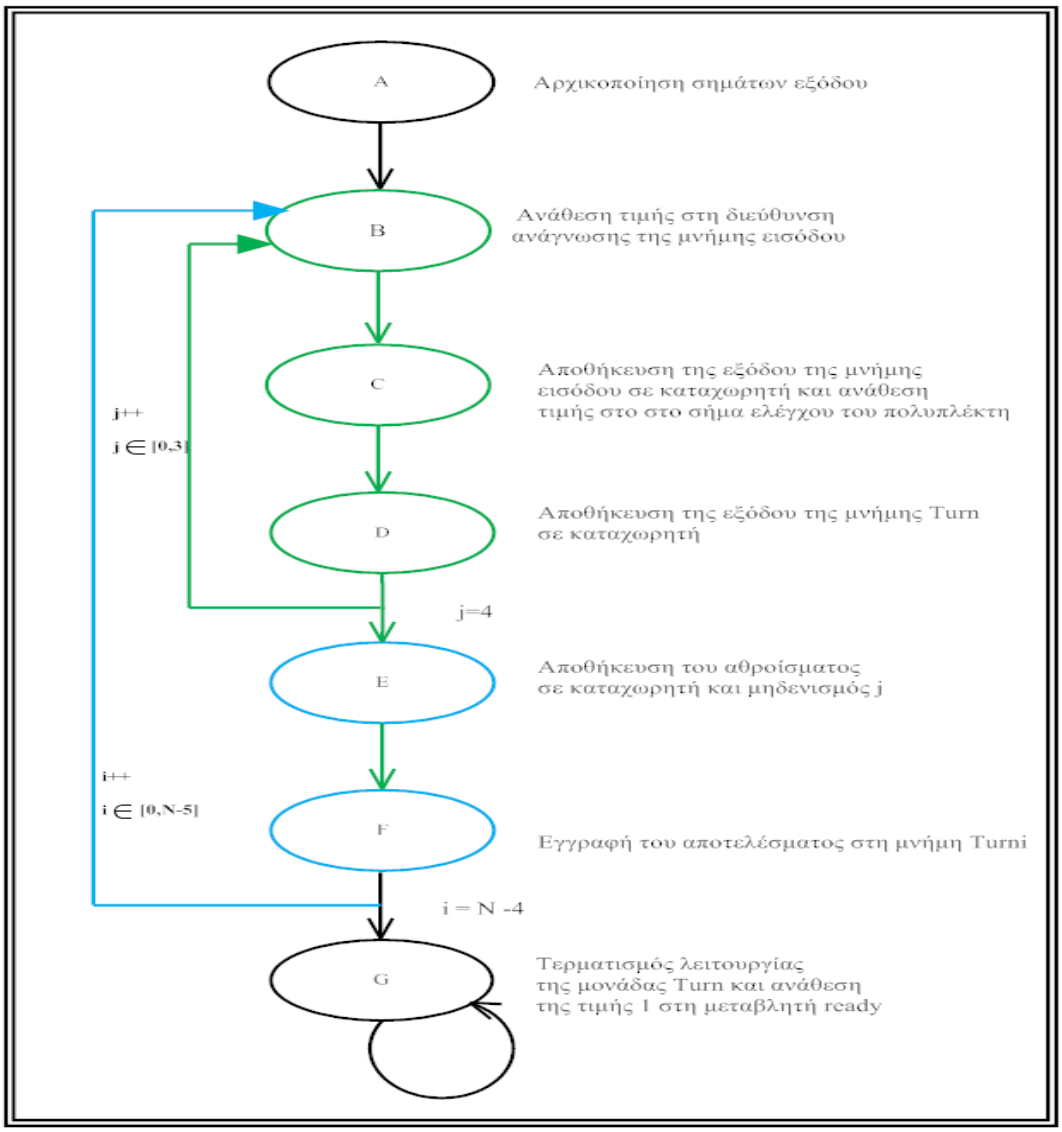
4^{ο} στάδιο: Γίνεται εγγραφή στη μνήμη Turni το άθροισμα που υπολογίστηκε στο 3^{ο} στάδιο, με διεύθυνση και σήμα ενεργοποίησης εγγραφής που προέρχονται από την FSM ελέγχου και δεδομένα εγγραφής την έξοδο του καταχωρητή.



Εικόνα 5-10: Αρχιτεκτονική τμήματος Turni

Η FSM ελέγχου (εικόνα 5-11) χωρίζεται σε 2 στάδια.

Το **πρώτο** στάδιο (B-D) είναι ουσιαστικά αυτό που καθορίζει το βασικό υπολογιστικό κομμάτι όπου γίνεται ανάγνωση της μνήμης εισόδου και έπειτα της μνήμης Turn. Στη συνέχεια, η διαδικασία επαναλαμβάνεται τέσσερις φορές και έτσι υπολογίζεται το άθροισμα των τεσσάρων τιμών που προκύπτουν από τη μνήμη Turn.
Στο **δεύτερο** στάδιο (E-F) γίνεται εγγραφή της μνήμης Turni με το άθροισμα που υπολογίστηκε στο πρώτο βήμα και ουσιαστικά ολοκληρώνεται η i-οστή επανάληψη.



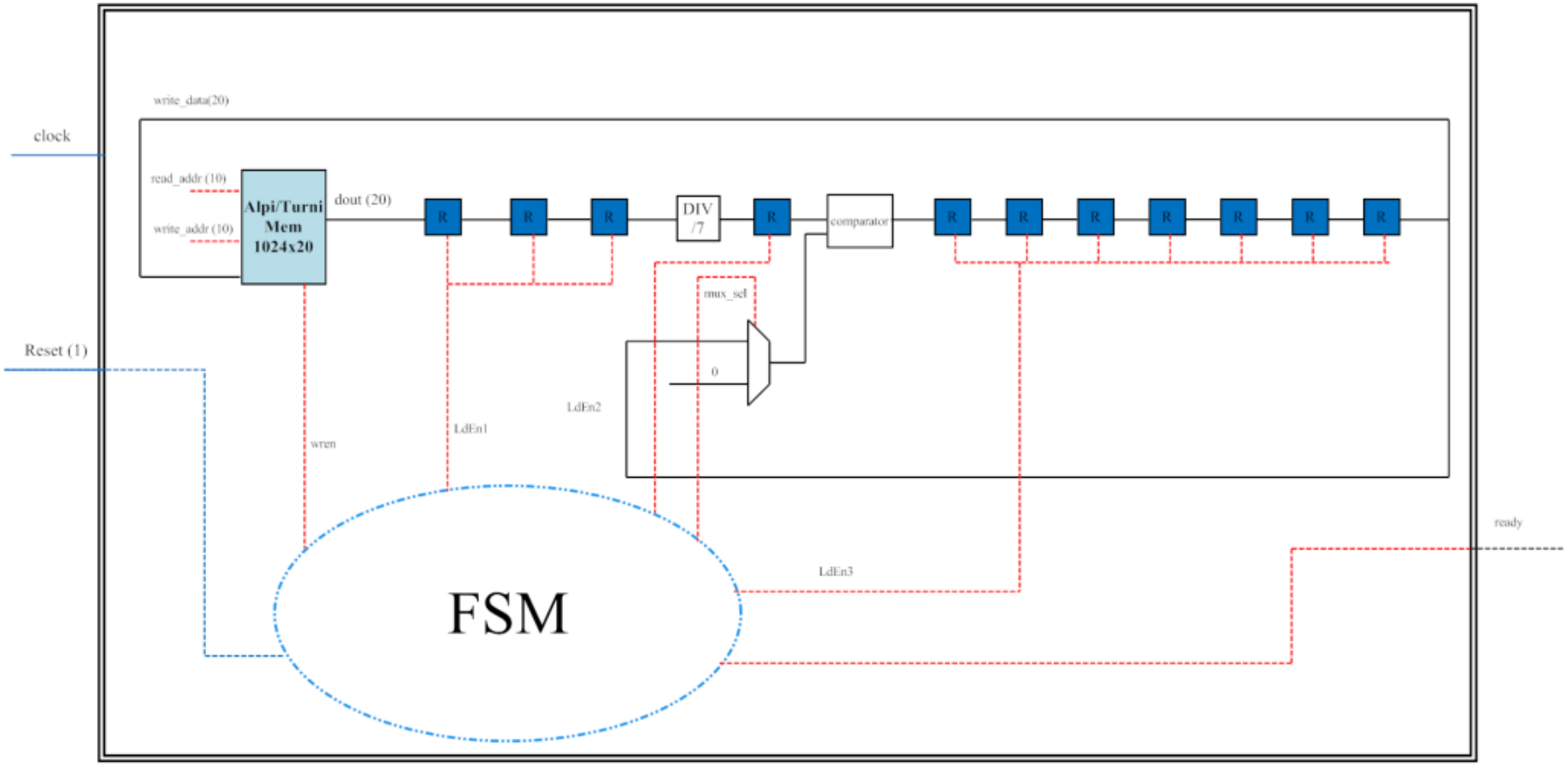
Εικόνα 5-11: Μονάδα ελέγχου για το τμήμα Turni

5.6 Τμήμα TakeBestAlpha

Στο κομμάτι αυτό της αρχιτεκτονικής (εικόνα 5-12) τροποποιούμε τη μνήμη Alpi και Turni κάθε φορά, έτσι ώστε οι τιμές των παραμέτρων Alp και Turn να είναι οι βέλτιστες για την εφαρμογή των κανόνων στη συνέχεια.

Πιο συγκεκριμένα, αφού διαβαστούν τρεις διαδοχικές τιμές από τη μνήμη εισόδου (Alpi_mem/Turni_mem), αποθηκεύονται σε καταχωρητές. Ο λόγος που διαβάζονται τρεις και όχι μία τιμή όπως επιβάλλει ο αλγόριθμος, είναι ότι υπάρχει περίπτωση να πανωγραφεί κάποια τιμή από την είσοδο λόγω του ότι τα αποτελέσματα γράφονται στην ίδια μνήμη με αυτή της εισόδου.
Στη συνέχεια, διαιρούμε την έξοδο της μνήμης με το 7 και συγκρίνουμε το αποτέλεσμα με τα δεδομένα που έχουν εγγραφεί στους καταχωρητές ολίσθησης. Αφού λοιπόν γίνουν και οι επτά συγκρίσεις, διαβάζουμε από τη μνήμη την επόμενη θέση και τέλος εγγράφουμε την έξοδο των καταχωρητών ολίσθησης (του τελευταίου καταχωρητή) στη μνήμη. Όπως προαναφέρθηκε, με το γεγονός ότι κρατάμε σε καταχωρητές τρεις διαδοχικές θέσεις μνήμης, αποφεύγουμε τον κίνδυνο να γραφεί ένα δεδομένο σε μια θέση μνήμης που θα διαβαζόταν σε επόμενο κύκλο.

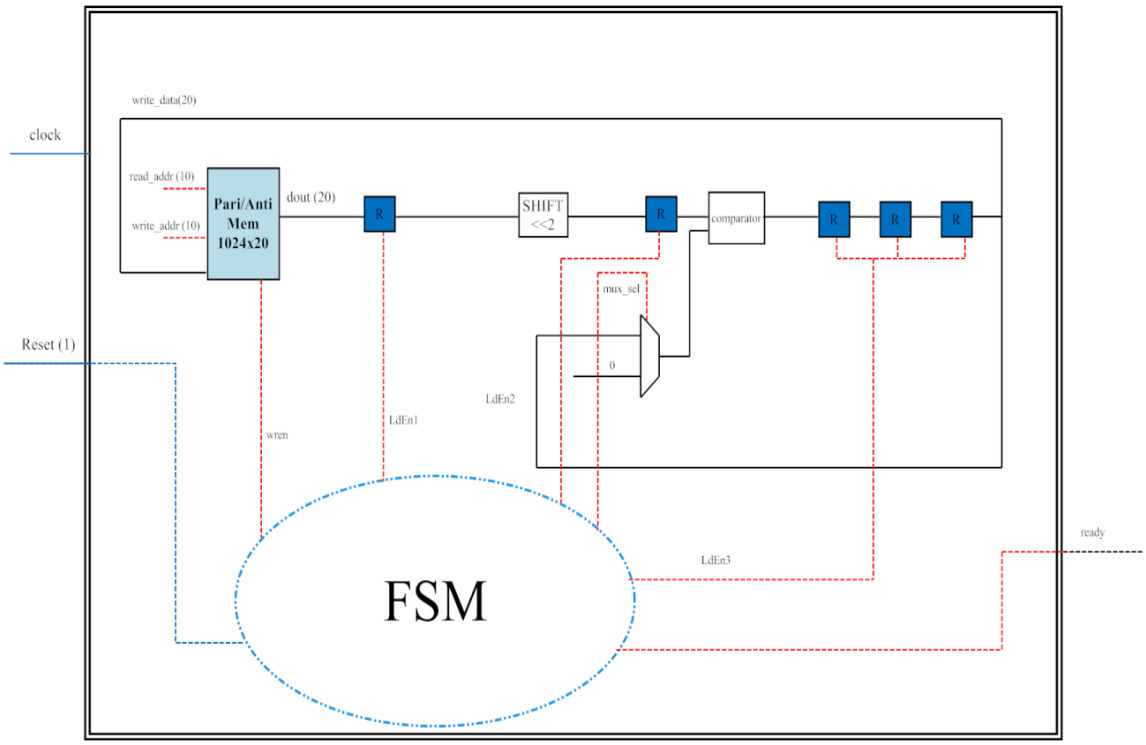
Η διαδικασία επαναλαμβάνεται για όλες τις διευθύνσεις που καθορίζει η FSM ελέγχου και τερματίζεται με την ανάθεση της τιμής 1 στην έξοδο Ready.



Εικόνα 5-12: Αρχιτεκτονική τμήματος TakeBestAlpha

5.7 Τμήμα TakeBestBeta

Η αρχιτεκτονική σε αυτό το κομμάτι του αλγορίθμου(εικόνα 5-13) είναι παρόμοια με αυτή για το τμήμα TakeBestAlpha με τη διαφορά ότι γίνονται τρεις αντί για επτά συγκρίσεις κάθε φορά και δε χρειάζεται να γίνεται ανάγνωση περισσότερων από μία θέσεων μνήμης.



Εικόνα 5-13: Αρχιτεκτονική τμήματος TakeBestBeta

5.8 Τμήμα Homologous

Στο κομμάτι αυτό της αρχιτεκτονικής (εικόνα 5-14) βρίσκονται οι πρωτεΐνες που είναι ομόλογες με την πρωτεΐνη εισόδου. Για το σκοπό αυτό, χρησιμοποιούνται ως είσοδο τέσσερις μνήμες:

- Η **Input_Mem**, 1019x35, η οποία περιέχει όλες τις επτάδες αμινοξέων της αλληλουχίας εισόδου
- Η **Db_Mem**, δύο θυρών, 20375x30 στην οποία είναι αποθηκευμένα όλα τα αμινοξέα των πρωτεϊνών της βάσης δεδομένων ανά εξάδες.

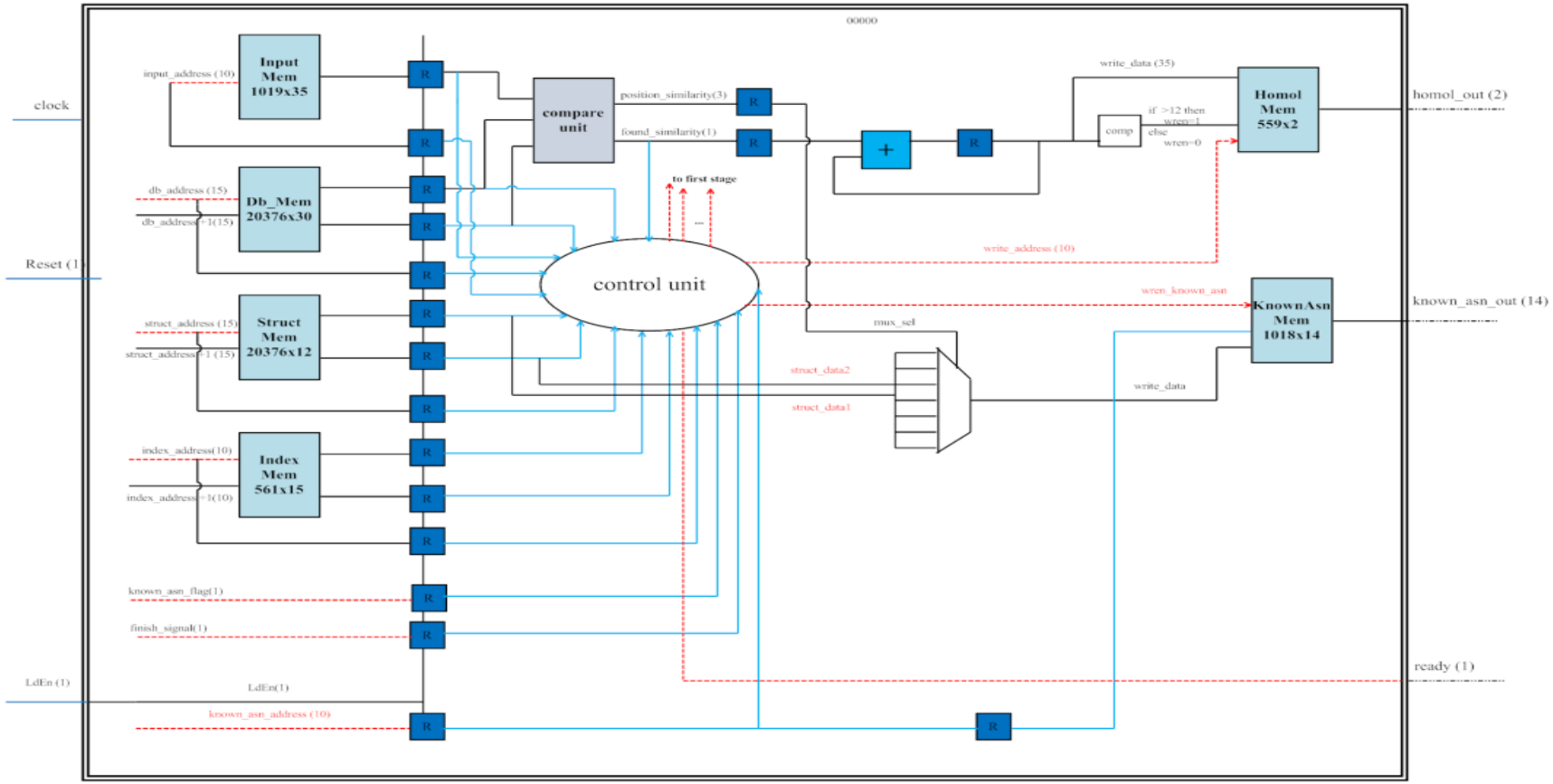
- Η **Struct_Mem**, δύο θυρών, 20375x15 στην οποία είναι αποθηκευμένη η δομή των αμινοξέων των πρωτεϊνών της βάσης δεδομένων ανά εξάδες.
- Η **Index_Mem**, δύο θυρών, 561x15 στην οποία είναι αποθηκευμένες οι διευθύνσεις της Db_Mem στις οποίες τελειώνει μια πρωτεΐνη από τη βάση δεδομένων.

Η βασική υπολογιστική μονάδα του τμήματος Homologous φαίνεται στην εικόνα 5-15. Η μονάδα σύγκρισης παίρνει ως είσοδο μία σειρά από τη μνήμη εισόδου (35 bits) και τη συγκρίνει με τις επτά 35άδες που προκύπτουν από τα 60 bits δεδομένων που διαβάζονται από την Db_Mem. Σε περίπτωση που έστω μία σύγκριση έχει ως αποτέλεσμα ισότητα μεταξύ των δεδομένων, τότε κρατάμε τη θέση που υπάρχει ισότητα και θέτουμε το σήμα OR_signal ίσο με 1.

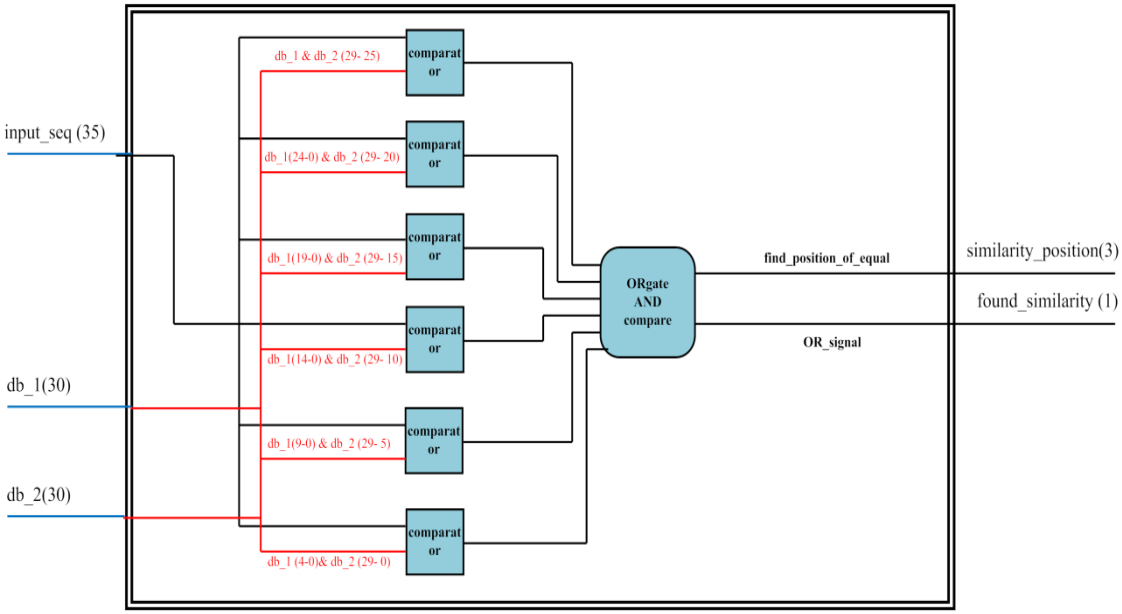
Στη συνέχεια, αθροίζουμε τα OR_signals που παίρνουμε σε κάθε κύκλο. Αν για μια συγκεκριμένη πρωτεΐνη από τη βάση, το άθροισμα βρεθεί μεγαλύτερο από 12, τότε η πρωτεΐνη αυτή θεωρείται ομόλογη. Αυτό έχει ως αποτέλεσμα την αντιγραφή της δομής των τμημάτων εκείνων που βρέθηκαν όμοια, στη μνήμη KnownAsn. Με τον τρόπο αυτό σταματά η διαδικασία για μία πρωτεΐνη από τη βάση και η μονάδα ελέγχου ξεκινά τη διαδικασία για την επόμενη, μέχρι να τελειώσουν και οι 559 πρωτεΐνες της βάσης δεδομένων.

Πρέπει να σημειωθεί ότι η μνήμη Index_mem δίνει την πληροφορία για το αν η πρωτεΐνη που εξετάζεται έφτασε στο τέλος της. Αυτό γίνεται όταν η έξοδός της συμπέσει με τη διεύθυνση ανάγνωσης από την Db_mem.

Η διαδικασία που περιγράφηκε πιο πάνω εκτελείται για όλες 35άδες της μνήμης εισόδου και όταν ολοκληρωθεί, το σήμα εξόδου ready γίνεται ίσο με 1.



Εικόνα 5-14: Αρχιτεκτονική τμήματος Homologous



Εικόνα 5-15: Μονάδα σύγκρισης για το τμήμα Homologous

5.9 Τμήμα PredictNN

Στο κομμάτι αυτό της αρχιτεκτονικής (εικόνα 5-16) βρίσκονται οι πρωτεΐνες που είναι ομόλογες με την πρωτεΐνη εισόδου. Για το σκοπό αυτό, χρησιμοποιούνται ως είσοδο δεκαεννιά μνήμες:

- Η **Input_Mem**, 1024x5, η οποία περιέχει την αλληλουχία εισόδου
- Η **Db_Mem**, μίας θύρας, 113535x5 στην οποία είναι αποθηκευμένα όλα τα αμινοξέα των πρωτεϊνών της βάσης δεδομένων.
- Η **Struct_Mem**, μίας θύρας, 113535x2 στην οποία είναι αποθηκευμένη η δομή των αμινοξέων των πρωτεϊνών της βάσης δεδομένων.
- Η **Index_Mem**, μίας θύρας, 561x17 στην οποία είναι αποθηκευμένες οι διευθύνσεις της Db_Mem στις οποίες τελειώνει μια πρωτεΐνη από τη βάση δεδομένων.
- Οι **Mbr_Mem,** οι οποίες είναι 14 μνήμες 640x20 που περιέχουν τα δεδομένα του τρισδιάστατου πίνακα MbrMatrix που περιγράψαμε στο προηγούμενο κεφάλαιο.
- Η **Homologous_Mem,** 559x2, η οποία περιλαμβάνει το αν οι πρωτεΐνες της βάσης είναι ομόλογες ή όχι με αυτή της εισόδου. Η μνήμη αυτή αποτελεί έξοδο του τμήματος homologous που περιγράψαμε στην ενότητα 5-8.

Το τμήμα αυτό της αρχιτεκτονικής χωρίζεται σε τέσσερα στάδια:

1° στάδιο: Γίνεται ανάγνωση δεκατριών διαδοχικών δεδομένων στη μνήμη εισόδου και στη μνήμη με την αλληλουχία των πρωτεϊνών βάσης.

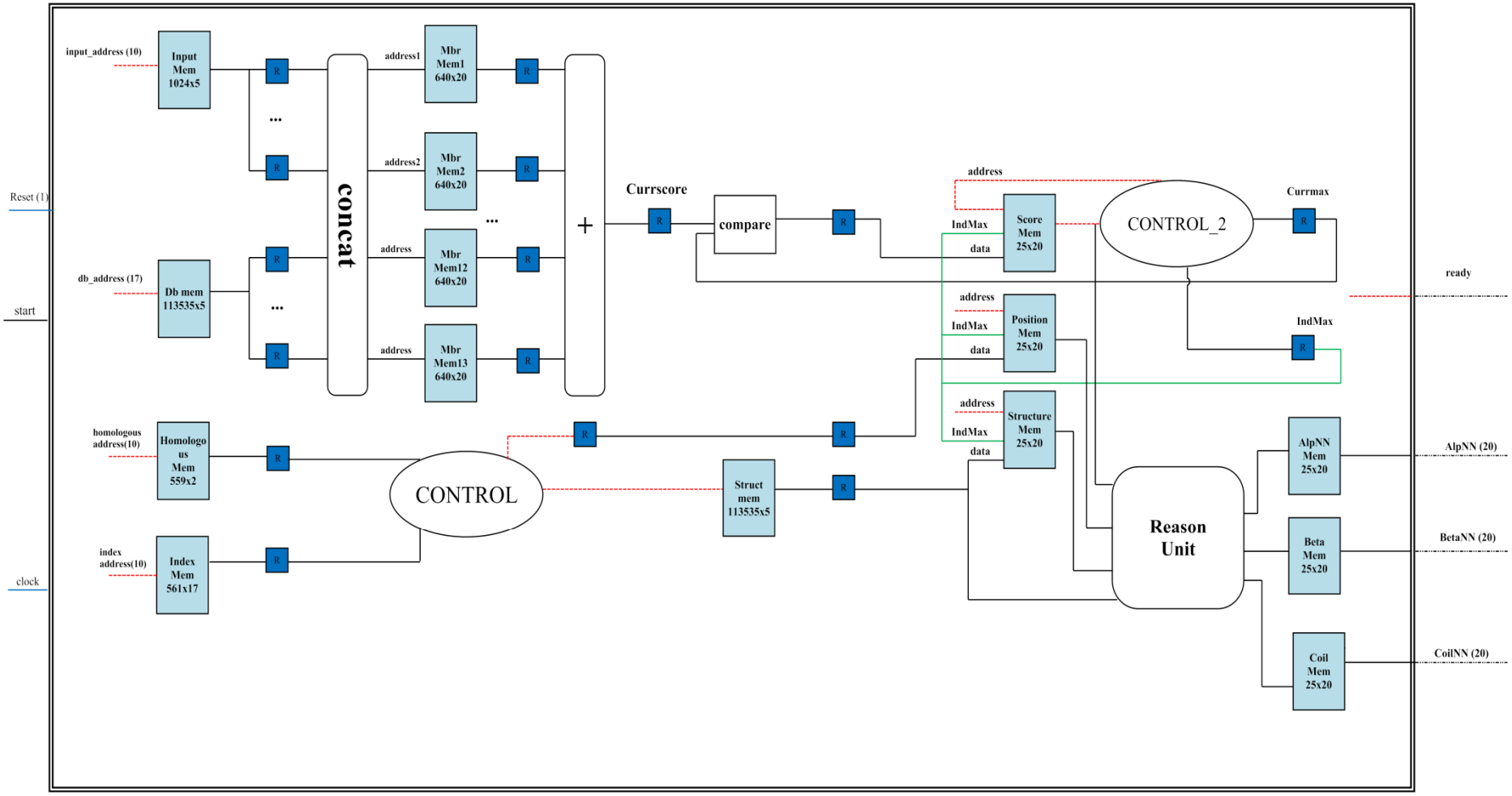
2° στάδιο: Γίνεται συνένωση των αντίστοιχων δεδομένων που διαβάστηκαν στο προηγούμενο στάδιο και το αποτέλεσμα που δημιουργείται λειτουργεί ως διευθυνσιοδότηση για τις μνήμες MbrMem (1-13).

3° στάδιο: Οι δεκατρείς έξοδοι των μνημών MbrMem αθροίζονται και το άθροισμα CurrScore που προκύπτει αποθηκεύεται σε καταχωρητή.

4ο στάδιο: Το CurrScore συγκρίνεται με το CurrMax (που προκύπτει από επόμενο στάδιο). Αν το CurrScore είναι μικρότερο από το CurrMax, τότε προχωράμε στο επόμενο στάδιο. Αν όχι, τότε η παραπάνω διαδικασία επαναλαμβάνεται για τις επόμενες 13άδες της μνήμης εισόδου και της μνήμης με τις πρωτεΐνες βάσης.

5° στάδιο: Το CurrScore και τα αντίστοιχα Position και Structure που είναι αποθηκευμένα σε καταχωρητές προηγούμενων σταδίων, γράφονται στις μνήμες Score, Position και Structure στη θέση που ορίζει το IndMax. Στη συνέχεια, εξετάζεται η μνήμη Score από τη μονάδα ελέγχου 2 (control_2). Πιο συγκεκριμένα, ελέγχεται αν κάποιο στοιχείο της μνήμης είναι μεγαλύτερο από το CurrMax. Αν βρεθεί κάτι τέτοιο, τότε η τιμή αυτή στη μνήμη Score αποτελεί το νέο Currmax και η θέση στην οποία βρέθηκε, το νέο IndMax. Σε περίπτωση που βρεθούν πάνω από μία τιμές μεγαλύτερες από το CurrMax, τότε αποθηκεύεται στον καταχωρητή CurrMax και IndMax η τελευταία τιμή.

Η παραπάνω διαδικασία συνεχίζεται μέχρι να ολοκληρωθούν όλα τα περιεχόμενα της μνήμης Db_mem. Τη λειτουργία αυτή την επιτελεί η μονάδα ελέγχου 1 (control) για τις πρωτεΐνες εκείνες που δεν είναι ομόλογες με αυτή της εισόδου. Όπως και στο τμήμα Homologous, έτσι και στο PredictNN, η μνήμη index_mem περιέχει τις διευθύνσεις που «τελειώνει» μια πρωτεΐνη από τη βάση.



Εικόνα 5-17: Αρχιτεκτονική για το τμήμα PredictNN

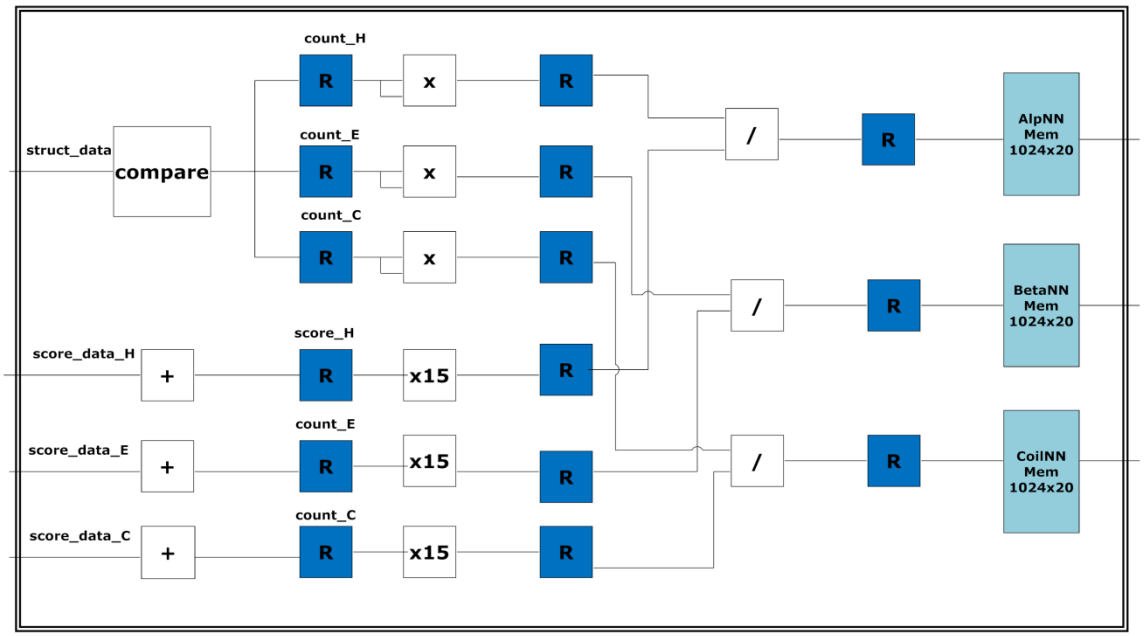
Όταν ολοκληρωθεί η μνήμη Db_mem για μια 13άδα από την αλληλουχία εισόδου, τότε το σύστημα «παγώνει» και εκτελείται η μονάδα Reason, το βασικό υπολογιστικό κομμάτι της οποίας φαίνεται στην εικόνα 5-17.

Η μονάδα Reason έχει ως είσοδο την έξοδο της μνήμης Struct_Mem για διευθύνσεις ανάγνωσης που δίνει η μονάδα ελέγχου 1. Αν η έξοδος της μνήμης Struct είναι 'H', τότε αυξάνεται η μετάβλητη Count_H στον καταχωρητή κατά ένα. Ομοίως για της περιπτώσεις όπου η έξοδος είναι 'E' και 'C'. Επιπλέον, προστίθενται μεταξύ τους τα δεδομένα στη μνήμη Score για τα οποία έχει βρεθεί ίδια δομή, το άθροισμα των οποίων αποθηκεύεται σε καταχωρητές (έστω Score_H, Score_E και Score_C τα αντίστοιχα αθροίσματα). Στο τελευταίο βήμα, υπολογίζεται η ποσότητα:

$$AlpNN = \frac{count_H^2}{15\ x\ Score_H}$$

$$BetaNN = \frac{count_E^2}{15\ x\ Score_E}$$

$$CoilNN = \frac{count_C^2}{15\ x\ Score_C}$$



Εικόνα 5-18: Βασικό υπολογιστικό κομμάτι μονάδος Reason

Όλη η παραπάνω διαδικασία επαναλαμβάνεται για όλες τις διαδοχικές 13άδες της Input_Mem και έτσι συμπληρώνονται οι μνήμες AlpNN_mem, BetaNN_mem και CoilNN_mem.

5.10 Τμήμα DecideStruct

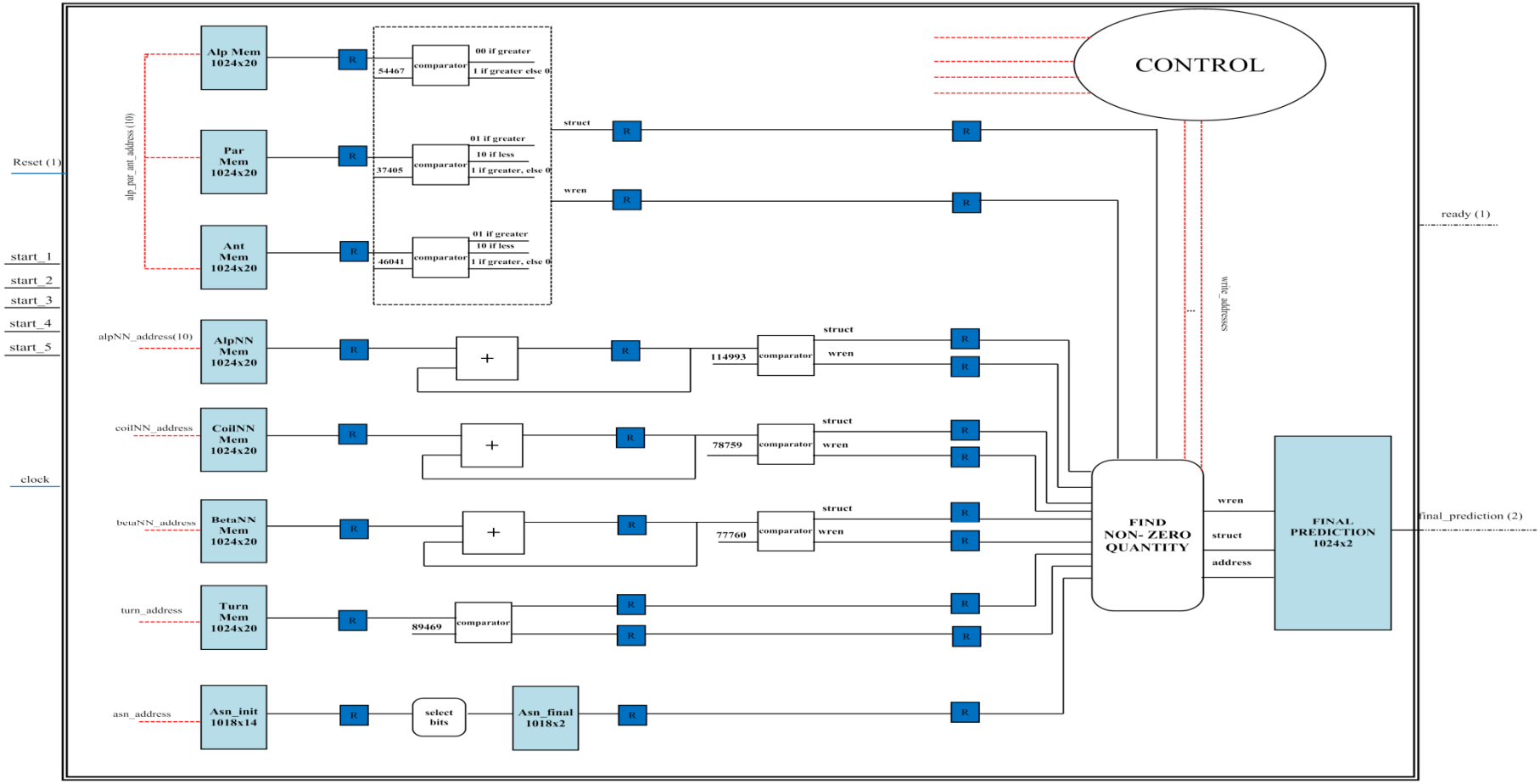
Στο τελευταίο αυτό κομμάτι της αρχιτεκτονικής (εικόνα 5-18), εφαρμόζονται οι επτά κανόνες που προαναφέρθηκαν για την εξαγωγή του τελικού αποτελέσματος. Βασική προϋπόθεση για την εκκίνηση του τμήματος αυτού είναι να έχουν τελειώσει όλα τα προηγούμενα (σήματα start 1-5 = 1).

Για την καλύτερη κατανόηση, η επεξήγηση της αρχιτεκτονικής θα γίνει με βάση τον εκάστοτε κανόνα που εφαρμόζεται.

1. Αρχικά, γίνεται ανάγνωση με κοινή διεύθυνση από τις μνήμες Alp, Par και Anti. Οι έξοδοι των μνημών συγκρίνονται με τα αντίστοιχα κατώφλια και εξάγεται η κατάλληλη δομή καθώς και το σήμα ενεργοποίησης εγγραφής της μνήμης. Τα δύο αυτά σήματα μεταφέρονται μέχρι το τελευταίο στάδιο, όπου και εισάγονται στη μονάδα Find_Non_Zero_Quality. Η διεύθυνση εγγραφής στη μνήμη προέρχεται από τη μονάδα ελέγχου.
2. Ο δεύτερος κανόνας ξεκινά με την ανάγνωση της μνήμης AlpNN. Αφού διαβαστούν οι 6 πρώτες τιμές από τη μνήμη, προστίθενται και το άθροισμα αυτό συγκρίνεται με το αντίστοιχο κατώφλι. Αν το άθροισμα είναι μεγαλύτερο από το κατώφλι, τότε το σήμα ενεργοποίησης εγγραφής της μνήμης γίνεται ίσο με 1 και η δομή που προκύπτει είναι η 'h' ("00"). Τα δύο αυτά σήματα μεταφέρονται μέχρι το τελευταίο στάδιο, όπου και εισάγονται στη μονάδα Find_Non_Zero_Quality. Η ίδια διαδικασία επαναλαμβάνεται για όλες τις διαδοχικές εξάδες που διαβάζονται από τη μνήμη.
3. Ομοίως με το δεύτερο κανόνα, με τη διαφορά ότι αθροίζονται τετράδες και η δομή που προκύπτει είναι η 'e' ("01").
4. Ομοίως με το δεύτερο κανόνα, με τη διαφορά ότι αθροίζονται τριάδες και η δομή που προκύπτει είναι η ˙c' ("10").
5. Στον πέμπτο κανόνα συγκρίνονται 4 διαδοχικές τιμές από τη μνήμη Turn. Αν αυτές οι τιμές είναι ίσες με το αντίστοιχο κατώφλι, τότε η δομή στην αντίστοιχη θέση που διαβάστηκαν οι τιμές είναι η 'c' ("10").
6. Πριν την εφαρμογή του έκτου κανόνα, πρέπει να μετατρέψουμε τη μνήμη Asn_init που είχε προκύψει από το τμήμα Homologous της αρχιτεκτονικής σε τέτοια μορφή ώστε κάθε θέση να έχει 2 bits αντί για 14. Αυτό γίνεται απλά διαβάζοντας τα 2 LSB bits κάθε σειράς της Asn_init και αποθηκεύοντας τα στην Asn_final. Στη συνέχεια, από τις θέσεις μνήμης όπου τα περιεχόμενα είναι διάφορα του «11» (άγνωστη δομή), αντιγράφουμε τα δεδομένα στις αντίστοιχες θέσεις της μνήμης FINAL PREDICTION.

Από τα βήματα 1-5, όλες οι έξοδοι που προκύπτουν για τη δομή, για το σήμα ενεργοποίησης εγγραφής της μνήμης και για τη διεύθυνση εγγραφής, εισάγονται στη μονάδα Find_Non_Zero_Quality όπου με τις κατάλληλες συγκρίσεις βρίσκονται οι μη-μηδενικές ποσότητες και εξάγονται στην τελική μνήμη.

Να σημειωθεί ότι όλες οι διευθύνσεις και τα σήματα ελέγχου, προέρχονται από τη μονάδα ελέγχου, η οποία ανάλογα με τον κανόνα που εφαρμόζεται κάθε φορά, ενεργοποιεί το κατάλληλο τμήμα της αρχιτεκτονικής (σήμα ενεργοποίησης καταχωρητή, διευθύνσεις ανάγνωσης και εγγραφής μνημών).



Εικόνα 5-18: Αρχιτεκτονική τμήματος DecideStruct

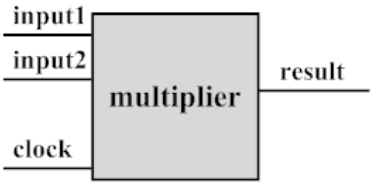
6 Αποτελέσματα - Μετρήσεις

6.1 Γενικά

Το παρόν κεφάλαιο περιγράφει τον τρόπο δημιουργίας των βασικών υποσυστημάτων που συνθέτουν την αρχιτεκτονική και παρουσιάζει τα αποτελέσματα της απόδοσης σε σχέση με το προσωπικό υπολογιστή (PC).

6.2 Πολλαπλασιαστής και διαιρέτης ακεραίων αριθμών

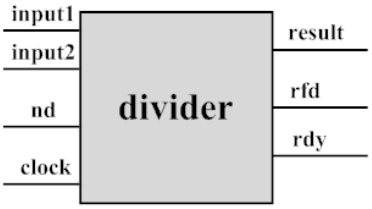
Οι πολλαπλασιαστές και οι διαιρέτες που χρησιμοποιήθηκαν στη σχεδίαση δημιουργήθηκαν με το εργαλείο Xilinx Core Generator 10.1. Οι διεπαφές των παραπάνω στοιχείων φαίνονται στους πίνακες 6-1 και 6-2.



Εικόνα 6-1: Δομική μονάδα πολλαπλασιαστή

Πίνακας 6-1: Διεπαφή πολλαπλασιαστή

Signal	Width	Type	Description
Input1	20	input	1st operand
Input2	20	input	2nd operand
clock	1	input	clock signal
Result	20	output	mult result



Εικόνα 6-2: Δομική μονάδα διαιρέτη

Πίνακας 6-2: Διεπαφή διαιρέτη

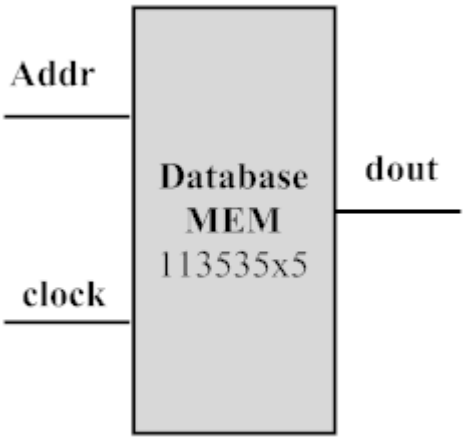
Signal	Width	Type	Description
Input1	20	input	divident
Input2	5	input	divisor
nd	1	input	new data
clock	1	input	clock signal
Result	20	output	quotient
rfd	1	output	ready for data
rfdy	1	output	result is ready

Κάθε πολλαπλασιαστής χρησιμοποιεί 2 DSP48E slices με σκοπό να αποφευχθεί το ενδεχόμενο να επηρεαστεί το κρίσιμο μονοπάτι (critical path). Έχει latency 15 κύκλων, λειτουργεί με ομοχειρία και δίνει ένα γινόμενο σε κάθε κύκλο (throughput=1). Κάθε διαιρέτης χρησιμοποιεί 9 DSP48E slices, λειτουργεί επίσης ομόχειρα και δίνει ένα αποτέλεσμα σε κάθε 3 κύκλους ρολογιού.

Χρησιμοποιήθηκαν συνολικά 18 πολλαπλασιαστές και 11 διαιρέτες για την υπολοίηση της αρχιτεκτονικής δεσμεύοντας έτσι ένα πολύ μικρό ποσοστό από τα διαθέσιμα DSP48E slices.

6.3 Μνήμη BRAM

Ο συνολικός αριθμός των BRAM blocks που υπάρχουν στην FPGA που επιλέχθηκε είναι 1032 blocks των 18kB ή 516 blocks των 36kB [29]. Οι μνήμες χρησιμοποιήθηκαν για να δοθούν στο σύστημα οι απαράιτητες είσοδοι. Ένα μεγάλο ποσοστό από τις διαθέσιμες BRAM's χρησιμοποιήθηκε για την αναπαράσταση της εισόδου που αφορά τη βάση δεδομένων με τις πρωτείνες όπου η δομή τους είναι γνωστή. Η μνήμη αυτή καταλαμβάνει 35 BRAM's στην FPGA και η διεπαφή της φαίνεται στην εικόνα 6-5.



Εικόνα 6-3: Διεπαφή μνήμης ROM

Η παραπάνω, είναι μια μνήμη ROM (read only memory) και κάθε της γραμμή αναπαριστά ένα αμινοξύ (5 bits) από την πρωτείνη της βάσης.

6.4 Ταυτοποίηση λειτουργίας

Η λειτουργία της αρχιτεκτονικής προσωμοιώθηκε με το πρόγραμμα ModelsimSE 6.3f. Για λόγους ευκολίας , αρχικά προσομοιώθηκε η λειτουργία των επιμέρους τμημάτων της αρχιτεκτονικής και διαπιστώθηκε η σωστή λειτουργία τους. Στη συνέχεια, ενώθηκαν όλα τα τμήματα και προσομοιώθηκε η συνολική λειτουργία της αρχιτεκτονικής.

Για τη διαπίστωση της σωστής λειτουργίας, χρησιμοποιήθηκαν εννιά διαφορετικές είσοδοι και συγκρίθηκαν τα αποτελέσματα με αυτά του προγράμματος PREDATOR του Bioperf. Αξίζει να σημειωθεί ότι υπήρχε πλήρης συμφωνία (99.9%) ανάμεσα στα αποτελέσματα της προσομοίωσης και αυτών του προγράμματος PREDATOR όπως και με αυτά της μοντελοποίησης σε software που υλοποιήθηκε.

Οι είσοδοι που χρησιμοποιήθηκαν προέρχονται από τις εισόδους που δίνει το Bioperf για τον έλεγχο της σωστής λειτουργίας οτυ προγράμματος, καθώς και συνδυασμός αυτών για να επιτύχουμε μεγαλύτερο μήκος στην αλληλουχία εισόδου. Τα πειράματα που διεξήχθησαν παρουσιάζονται στον πίνακα 6-3.

Πίνακας 6-3: Είσοδοι που χρησιμοποιήθηκαν για τον αλγόριθμο

Αριθμός πειράματος	Μήκος ακολουθίας	Πηγή
1	1024	τυχαία ακολουθία
2	1392	συνδυασμός εισόδων από το Bioperf
3	1979	συνδυασμός εισόδων από το Bioperf
4	2733	συνδυασμός εισόδων από το Bioperf
5	3267	συνδυασμός εισόδων από το Bioperf
6	5185	συνδυασμός εισόδων από το Bioperf
7	5625	συνδυασμός εισόδων από το Bioperf
8	6059	συνδυασμός εισόδων από το Bioperf
9	16843	συνδυασμός εισόδων από το Bioperf

6.5 Απόδοση συστήματος

Η αρχιτεκτονική που παρουσιάστηκε στο προηγούμενο κεφάλαιο είναι ιδιαίτερα απαιτητική σε μνήμη. Για το λόγο αυτό, επιλέχθηκε η FPGA XC5VSX240T της οικογένειας Virtex 5, η οποία έχει το μεγαλύτερο αριθμό block ram blocks (516 blocks των 36kB) από όλες τις FPGA's που υπάρχουν στην αγορά σήμερα. Έγινε χρήση της γλώσσας VHDL και του εργαλείου Xilinx Ise Design Suite 10.1 για την ανάπτυξη του συστήματος.

Το τελικο σύστημα, όπως αναφέρθηκε, αποτελείται από 3 υπολογιστικές μονάδες για το τμήμα PredictNN. Η σύνθεση πραγματοποιήθηκε με το εργαλείο Xilinx Ise 10.1 και προέκυψαν τα ακόλουθα αποτελέσματα, όσον αφορά τη συχνότητα ρολογιού και τους πόρους που καταναλώνει το σύστημα (Πίνακας 6-4).

Clock Frequency : 298.130 MHz

Πίνακας 6-4: Πόροι που χρησιμοποιούνται για την FPGA XC5VSX240T

Logic utilization	Used	Available	Utilization (%)
Number of slice registers	130589	149760	87
Number of slice LUTs	115652	149760	77
Number of fully used LUT-FF pairs	81563	127040	64
Number of Block Ram/ FIFO	480	516	93
Number of DSP48Es	135	1056	12

6.6 Αποτίμηση απόδοσης

Το πρόγραμμα για τον αλγόριθμο PREDATOR εκτελέστηκε σε PC με επεξεργαστή Intel Pentium Core 2 στα 2.4 GHz, με 3 Gb RAM και λειτουργικό σύστημα Windows XP SP3. Το συγκεκριμένο πρόγραμμα εκτελέστηκε 5 φορές για κάθε πείραμα και έγινε χρήση του προγράμματος Vtune για να μετρηθούν οι κύκλοι CPU που καταναλώνονται. Ο μέσος χρόνος εκτέλεσης για τις 5 μετρήσεις για το καθένα από τα 10 πειράματα φαίνεται στον πίνακα 6-5.

Πίνακας 6-5: Χρόνος εκτέλεσης για τον προσωπικό υπολογιστή

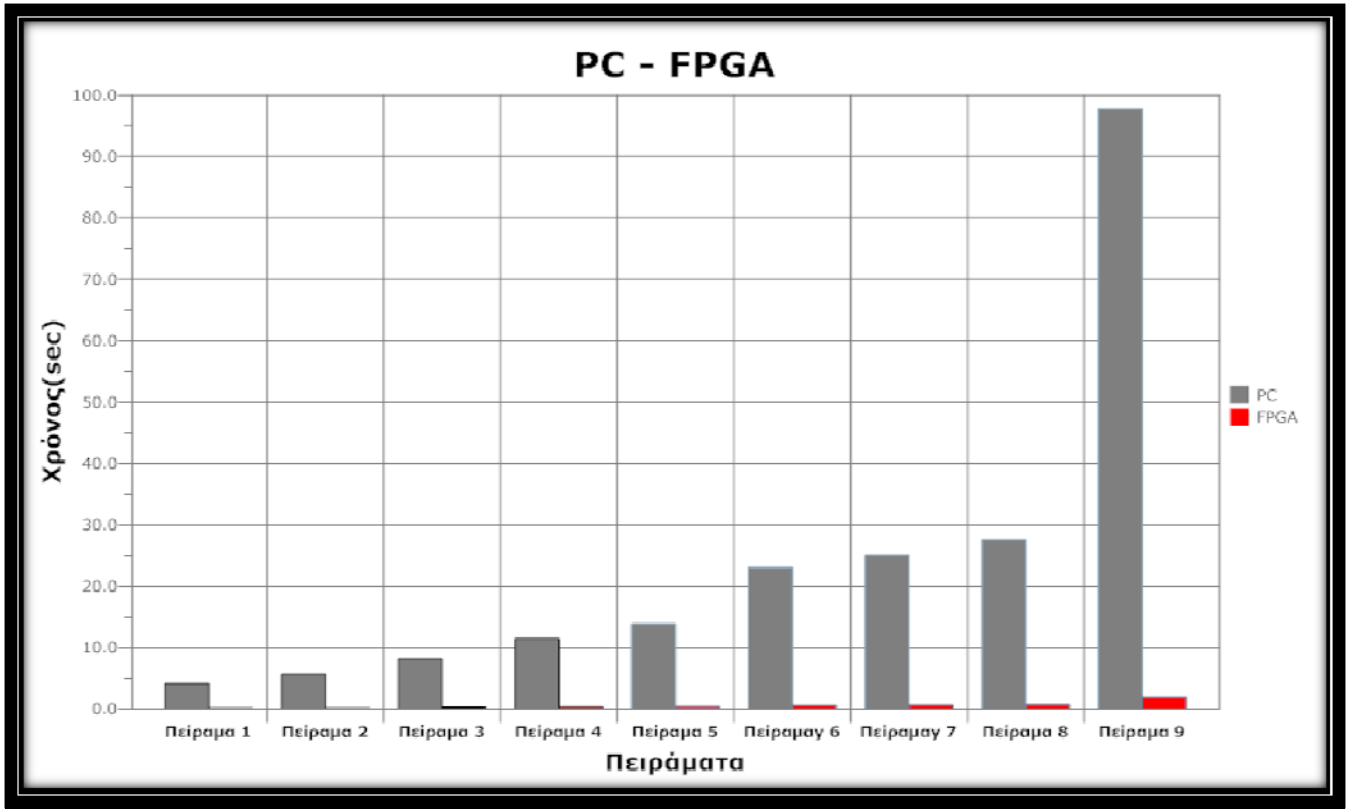
Αριθμός πειράματος	Χρόνος εκτέλεσης (sec)
1	4,140
2	5,625
3	8,109
4	11,453
5	13,859
6	23,062
7	25,094
8	27,609
9	97,790

Έπειτα, μετρήθηκαν οι κύκλοι ρολογιού που χρειάστηκε το σύστημα που υλοποιήθηκε στην FPGA για να επεξεργαστεί τον όγκο δεδομένων του πειράματος και από το σύνολο των κύκλων υπολογίστηκε ο εκτιμώμενος χρόνος εκτέλεσης (Πίνακας 6-6).

Πίνακας 6-6: Χρόνος εκτέλεσης για την FPGA και για προσωπικό υπολογιστή

Αριθμός πειράματος	Χρόνος εκτέλεσης FPGA(sec)	Χρόνος εκτέλεσης PC(sec)
1	0,1112	4,1040
2	0,1567	5,6250
3	0,2232	8,1090
4	0,3082	11,4530
5	0,3672	13,8590
6	0,5845	23,0620
7	0,6262	25,0940
8	0,6822	27,6090
9	1,894	97,7900

Συγκρίνοντας τους χρόνους εκτέλεσης για την FPGA και τον Pentium 4, προκύπτει το ακόλουθο διάγραμμα για κάθε ένα από τα εννιά πειράματα (Εικόνα 6-4).



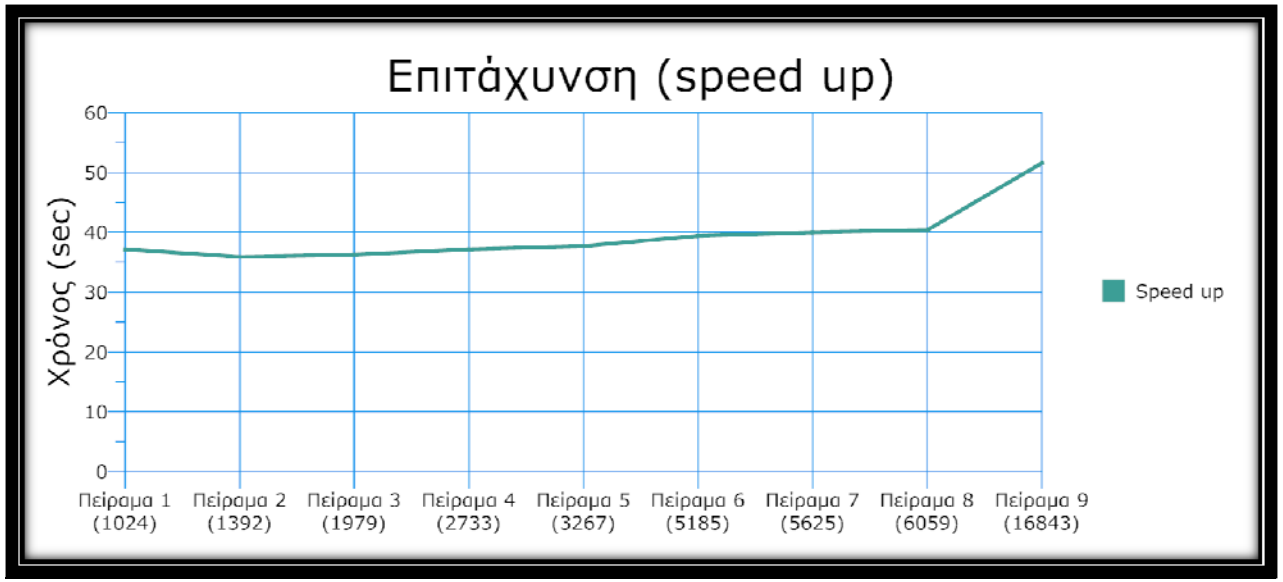
Εικόνα 6-4: Σύγκριση χρόνου εκτέλεσης προσωπικού υπολογιστή - FPGA

Συνδυάζοντας τις πληροφορίες των πινάκων 6-5 και 6-6, προκύπτει ο πίνακας 6-7 για την επιτάχυνση (speed up) που πραγματοποιείται από τη χρήση FPGA αντί για Pentium 4 για το συκεκριμένο πρόβλημα.

Πίνακας 6-7: Speed-up για τις διάφορες εισόδους

Αριθμός πειράματος	Μήκος αλληλουχίας	Speed up
1	1024	37,2
2	1392	35,9
3	1979	36,3
4	2733	37,2
5	3267	37,7
6	5185	39,4
7	5625	40,0
8	6059	40,4
9	16843	51,6

Τέλος, η μεταβολή του speed-up για τις εκτελέσεις των εννιά πειραμάτων αναπαρίσταται γραφικά στην εικόνα 6-5.



Εικόνα 6-5: Συνοπτική απεικόνιση speed-up ανάλογα με το μήκος εισόδου

7 Συμπεράσματα, μελλοντικές επεκτάσεις και αλλαγές

7.1 Γενικά

Σε αυτό το κεφάλαιο αναφέρονται τα συμπεράσματα και κάποιες ιδέες για επεκτάσεις της αρχιτεκτονικής.

7.2 Συμπεράσματα

Η διατριβή αυτή ασχολήθηκε με τη μελέτη του αλγορίθμου PREDATOR, ο οποίος προβλέπει τη δευτεροταγή δομή μιας πρωτεΐνης, και την υλοποίησή του εξ' ολοκλήρου σε αναδιατασσόμενη λογική. Ο στόχος ήταν το σύστημα αυτό να είναι αποδοτικότερο συγκριτικά με έναν προσωπικό υπολογιστή και στη συγκεκριμένη περίπτωση με έναν Intel Pentium Core 2 @2,40GHz.

Όπως παρατηρήθηκε από τα αποτελέσματα, ο στόχος εκτιμάται ότι ικανοποιείται σε πολύ μεγάλο βαθμό μιας και η υλοποίηση του αλγορίθμου σε hardware είναι, στη χειρότερη περίπτωση, 36 φορές πιο γρήγορη και μπορεί να γίνει ακόμα και 50 φορές πιο γρήγορη για εισόδους πολύ μεγάλου μήκους. Ένα άλλο σημαντικό πλεονέκτημα της υλοποίησης είναι ότι η επιτάχυνση του αλγορίθμου είναι σχεδόν σταθερή, γεγονός που εξασφαλίζει έναν πολύ μικρότερο χρόνο εκτέλεσης, ανεξαρτήτως μεγέθους εισόδου.

7.3 Μελλοντικές επεκτάσεις

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

Σημαντικές επεκτάσεις και αλλαγές που μπορούν να πραγματοποιηθούν είναι:

i. Δημιουργία κατάλληλου interface ώστε να μπορεί η αρχιτεκτονική να «επικοινωνήσει» με το πρόγραμμα PREDATOR για τη λήψη των απαραίτητων εισόδων για τη σωστή λειτουργία του αλγορίθμου. Με τον τρόπο αυτό η όλη διαδικασία θα γίνει πιο εύκολη για το χρήστη και δε θα χρειάζεται κάθε φορά η αρχικοποίηση των μνημών από τον ίδιο.

ii. Συνδυασμός της αρχιτεκτονικής του PREDATOR με άλλες αρχιτεκτονικές από το πακέτο BIOPERF που έχουν ήδη υλοποιηθεί στο εργαστήριο, έτσι ώστε να υπάρχει πλήρης ανάλυση για τη δομή μιας συγκεκριμένης πρωτεΐνης και για την πιθανή ομοιότητά της με άλλες.

iii. Καλύτερη εκμετάλλευση της χρησιμοποίησης των BRAM's, έτσι ώστε να γίνει εφικτή η ταυτόχρονη λειτουργία περισσότερων τμημάτων του αλγορίθμου, γεγονός που, στη χειρότερη περίπτωση, θα μείωνε κατά το ήμισυ την ταχύτητα εκτέλεσης του αλγορίθμου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Reeds PJ: "Dispensable and indispensable amino acids for humans". *J. Nutr.* **130** (7): 1835S–40S (2000).

[2] Hemoglobin1: http://www.3dchem.com/imagesofmolecules/Hemoglobin1.jpg

[3] PROTINFO: http://protinfo.compbio.washington.edu

[4] Kalman SM, Linderstrom-Lang K, Ottesen M, Richards FM. : "Degradation of ribonuclease by subtilisin". *Biochimica et Biophysica Acta* 16 (2): 297–99 (1955).

[5] Pauling L, Corey RB, Branson HR.. "The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain", *Proceedings of the National Academy of Sciences U.S.A.* 37: 235–40 (1951).

[6] FarzadFard, Fahim; *et al.*: "β-Sheet capping: Signals that initiate and terminate β-sheet formation". *Journal of Structural Biology* 161 (1): 101-110 (2008).

[7] The Nobel Prize in Chemistry Press Release (1972).

[8] Chou PY, Fasman GD.: "Prediction of protein conformation". Biochemistry, 13(2), p. 222-45 (1974).

[9] Mount D.M., Bioinformatics: "Sequence and Genome Analysis", Second Edition, Cold Spring Harbor Laboratory Press. ISBN 0879697121.

[10] Garnier J, Osguthorpe DJ, Robson B. : "Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins", J. Mol. Biol. 120:97-120 (1978).

[11] Gerald D. Fasman: "Prediction of protein structure and the principles of protein conformation", Springer, ISBN 0306431319, 9780306431319 (1989).

[12] Ross D. King , Mansoor Saqi , Roger Sayle , Michael J. E. Sternberg : "DSC: Public Domain Protein Secondary Structure Prediction" (1997).

[13] Garnier J, Gibrat JF and Robson B.: "GOR method for predicting protein secondary structure from amino acid sequence", Methods Enzymol, 266, 540-53. (1996)

[14] Burkhard Rost , Chris Sander and Reinhard Schneider : "PHD - an automatic mail server for protein secondary structure prediction" (1993).

[15] Jones DT.: "Protein secondary structure prediction based on position-specific scoring matrices". J. Mol. Biol. 292: 195-202 (1999).

[16] D. G. Kneller, F. E. Cohen and R. Langridge: "Improvements in Protein Secondary Structure Prediction by an Enhanced Neural Network" (1990).

[17] Geourjon C, Deléage G.: "SOPM: a self-optimized method for protein secondary structure prediction", Institut de Biologie et de Chimie des Protéines, UPR 412-CNRS, Lyon, France.

[18] Geourjon C, Deléage G.: "SOPMA: significant improvements in protein secondary structure prediction by consensus prediction from multiple alignments.", Institut de Biologie et de Chimie des Protéines, UPR 412-CNRS, Lyon, France.

[19] Solovyev V.V.,Salamov A.A.: "Secondary structure prediction based on discriminant analysis in Computer analysis of Genetic macromolecules", World Scientific, p.352-364 (1994).

[20] Dmitrij Frishman and Patrick Argos: Incorporation of long-distance interactions into a secondary structure prediction algorithm. Protein Engineering, vol 9, p.133-142 (1996).

[21] Protein Structure Prediction: http://cmgm.stanford.edu/WWW/www_predict.html

[22] D.A. Bader, Y. Li, T. Li, V. Sachdeva: "BioPerf: A Benchmark Suite to Evaluate High-Performance Computer Architecture on Bioinformatics Applications", *The IEEE International Symposium on Workload Characterization (IISWC 2005)*, Austin, TX, October 6-8 (2005).

[23] L. Pauling and R.B Corey: "Configurations of polypeptide chains with favored orientations of the polypeptide around single bonds: Two pleated sheets.", Proc. Natl. Acad. Sci. Wash., 37:729-740 (1951).

[24] Dmitrij Frishman and Patrick Argos: "Incorporation of long-distance interactions into a secondary structure prediction algorithm", Protein Engineering, vol 9, p. 133-142 (1996).

[25] Dmitrij Frishman and Patrick Argos: "Knowledge-Based Protein Secondary Structure Assignment", (1995).

[26] Frishman,D & Argos,P: "Knowledge-based secondary structure assignment.", Proteins: structure, function and genetics, 23, p. 566-579 (1995).

[27] Martin J, Letellier G, Marin A, Taly JF, de Brevern AG, Gibrat JF. "Protein secondary structure assignment revisited: a detailed analysis of different assignment methods" (2005).

[28] Fixed Point Arithmetic: http://cnx.org/content/m11054/latest/

[29] Virtex-5 Family Overview:
http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf