



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Μέτρηση Απόδοσης Μνήμης και Μηχανισμός Ανάκτησης Δεδομένων σε Υπερυπολογιστή Βασισμένο σε Αναδιατασσόμενη Λογική

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΑΛΑΪΤΖΗ ΚΩΝΣΤΑΝΤΙΝΟΥ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Επιβλέπων :	Καθηγητής Δόλλας Απόστολος
Μέλος Επιτροπής:	Καθηγητής Πνευματικάτος Διονύσιος
Μέλος Επιτροπής:	Αναπληρωτής Καθηγητής Παπαευσταθίου Ιωάννης

Χανιά, Μάρτιος 2015

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή μου, κ. Απόστολο Δόλλα, που ήταν και ο επιβλέπων της παρούσας διπλωματικής, για την εποικοδομητική συνεργασία που είχαμε σε επίπεδο διπλωματικής, για το χρόνο, που αφιέρωσε για να με βοηθήσει και κυρίως για την εμπιστοσύνη, που έδειξε στο πρόσωπό μου. Ευχαριστώ επίσης τον καθηγητή, κύριο Διονύσιο Πνευματικάτο και τον αναπληρωτή καθηγητή, κύριο Ιωάννη Παπαευσταθίου, που δέχτηκαν να αξιολογήσουν τη διπλωματική μου διατριβή.

Το επόμενο άτομο που θα ήθελα να ευχαριστήσω είναι ο Δρ. Ευριπίδης Σωτηριάδης για την υποστήριξή του κατά τη διάρκεια της εκπόνησης της εργασίας, τις πρακτικές συμβουλές του, αλλά και για το γεγονός ότι ήταν πάντα διαθέσιμος, όταν τον χρειαζόμουν. Ευχαριστώ επίσης το Δρ. Γρηγόρη Χρυσό και το μεταπτυχιακό φοιτητή Χρήστο Ρουσόπουλο για το χρόνο που διαθέσαν και τη βοήθειά τους, στο κομμάτι του Convey.

Το μεγαλύτερο ευχαριστώ αξίζει όμως στην οικογένεια μου, που με στηρίζει στις επιλογές μου και βρίσκεται πάντα δίπλα μου, προσφέροντάς μου την ασφάλεια και τη δύναμη να πραγματοποιώ τους στόχους μου. Επομένως η συγκεκριμένη δουλειά είναι αφιερωμένη σε αυτούς, για όσα μου έχουν προσφέρει.

Περίληψη

Ο κλάδος των υβριδικών υπερ-υπολογιστών με χρήση αναδιατασσόμενης λογικής (FPGAs) είναι ένας αναπτυσσόμενος κλάδος των υπολογιστικών συστημάτων. Τα μηχανήματα αυτά όπως το Convey παρέχουν καινούργιες δυνατότητες στους χρήστες και βελτιωμένες επιδόσεις στις FPGAs. Ο κλάδος των συστημάτων αυτών είναι αρκετά σύγχρονος και δεν έχει μελετηθεί εκτενώς.

Η παρούσα διπλωματική εργασία στοχεύει στην εξοικείωση με την καινούρια τεχνολογία που παρέχει το Convey και στη μελέτη και ανάλυση του τρόπου λειτουργίας του υποσυστήματος μνήμης του συνεπεξεργαστή (FPGAs) στο Convey HC-2. Μέσω πειραμάτων που πραγματοποιήθηκαν στα πλαίσια της εργασίας μετρήθηκε η ταχύτητα επικοινωνίας των FPGAs με το σύστημα της μνήμης μέσω της διεπαφής των memory controllers (MCs). Τα πειράματα στοχεύουν κυρίως στη μέτρηση της ταχύτητα προσπέλασης στοιχείων από τη μνήμη. Χρησιμοποιήθηκαν διαφορετικοί τρόποι προσπέλασης των δεδομένων, έτσι ώστε να βρεθεί ο αποδοτικότερος και να προταθεί για μελλοντικές εργασίες στο Convey. Η ακριβής ποσοτική μελέτη του υποσυστήματος μνήμης είναι ιδιαίτερα χρήσιμη για τη διαστασιολόγηση των προβλημάτων κατά τη διαδικασία σχεδίασης αρχιτεκτονικών.

Τα αποτελέσματα των πειραμάτων παρέχουν σημαντικές πληροφορίες και δίνουν κατευθύνσεις για τον τρόπο προσπέλασης μεμονωμένων και μαζικών στοιχείων από τη μνήμη, αλλά και δομών δεδομένων, όπως είναι οι τριγωνικοί πίνακες. Η ανάλυση των αποτελεσμάτων των μετρήσεων αποκάλυψε σε μεγάλο βαθμό τον τρόπο λειτουργίας του συστήματος μνήμης και των MCs, ο οποίος δεν επιδέχεται σημαντικές παρεμβάσεις και είναι σε μεγάλο βαθμό τελειοποιημένος, τουλάχιστον στο εύρος των πειραμάτων που διεξήχθησαν.

Περιεχόμενα

Μέτρηση Απόδοσης Μνήμης και Μηχανισμός Ανάκτησης Δεδομένων σε Υπερυπολογιστή Βασισμένο σε Αναδιατασσόμενη Λογική.....		1
1	Εισαγωγή.....	5
1.1	Υβριδικοί Υπερ-Υπολογιστές.....	5
1.2	Σκοπός - Συνεισφορά.....	6
1.3	Δομή της Διπλωματικής.....	7
2	Σχετική Έρευνα	8
2.1	Stencil	8
2.1.1	Stencil - Convey HC-1.....	10
2.2	Γραμμική Άλγεβρα	11
2.2.1	Γραμμική Άλγεβρα – Convey HC-1	12
2.3	Περιγραφή αλγορίθμου UNAFold του Zuker – Τριγωνικοί Πίνακες.....	13
3	Μοντέλα πρόσβασης μνήμης.....	15
3.1	Διαφυλλωμένη μνήμη	15
3.1.1	Υψηλής Τάξης Διαφύλλωση (high-order interleaving)	16
3.1.2	Χαμηλής Τάξης Διαφύλλωση (N-way low-order interleaving)	17
3.2	Παράγοντας Διαφύλλωσης	17
3.3	Μοντέλο Convey.....	19
3.3.1	Διεπαφή Memory controller	19
3.3.2	Λειτουργικότητα Διεπαφής MC	20
3.3.3	Αιτήματα Μνήμης	20
3.3.4	Αποκρίσεις Μνήμης.....	21
3.3.5	Σύστημα Μνήμης 1024 Bank	21
3.3.6	Binary Διαφύλλωση.....	23
3.3.7	31/31 Διαφύλλωση	23
3.3.8	Εύρος Ζώνης της Μνήμης	24
3.3.9	Προαιρετική Διεπαφή Crossbar MC.....	24

4	Μοντελοποίηση Προβλήματος – Μεθοδολογία Μετρήσεων.....	26
4.1	Χρόνος απόκρισης μνήμης για μεμονωμένες προσπελάσεις	28
4.2	Χρόνος απόκρισης μνήμης για μαζικές προσπελάσεις (burst data).....	30
4.2.1	Μετρήσεις προσπέλασης στοιχείων σε διαδοχικές θέσεις μνήμης.....	31
4.2.2	Μετρήσεις προσπέλασης στοιχείων με «άλματα» (strides) διευθύνσεων	33
4.3	Αποτύπωμα Μνήμης Αλγορίθμου UNAFold του Zuker	38
4.3.1	Profiling UNAFold	40
4.3.2	Απαιτήσεις Μνήμης Αλγορίθμου UNAFold - Προσομοίωση σε Matlab.....	40
4.4	Χρόνος Απόκρισης Μνήμης - Τριγωνικοί Πίνακες	42
4.4.1	Προσπέλαση όλων των στοιχείων του πίνακα.....	43
4.4.2	Προσπέλαση των στοιχείων του κάτω τριγωνικού πίνακα.....	43
4.4.3	Προσπέλαση του πίνακα με υβριδική μέθοδο	44
4.5	Προβλήματα στη χρήση του Convey	46
5	Αποτελέσματα	51
5.1	Αποτελέσματα μετρήσεων χρόνου απόκρισης μνήμης για μεμονωμένες προσπελάσεις	51
5.2	Αποτελέσματα μετρήσεων χρόνου απόκρισης μνήμης για μαζικές προσπελάσεις (burst data)	56
5.2.1	Αποτελέσματα μετρήσεων προσπέλασης στοιχείων σε διαδοχικές θέσεις μνήμης	56
5.2.2	Αποτελέσματα μετρήσεων προσπέλασης στοιχείων με «άλματα» (strides) διευθύνσεων	59
5.3	Τρόπος Λειτουργίας του Συστήματος Μνήμης	60
5.4	Απόκριση Μνήμης – Τριγωνικοί Πίνακες	63
6	Συμπεράσματα και Μελλοντικές Επεκτάσεις	65
6.1	Συμπεράσματα	65
6.2	Μελλοντικές Επεκτάσεις	67
	Βιβλιογραφία.....	68

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1 – Αρχιτεκτονική Convey [3]	6
Σχήμα 2.1 – Αναπαράσταση Stencil [11]	9
Σχήμα 2.2 – Μνήμης για Υπολογισμό του Stencil [11]	9
Σχήμα 2.3 – Stencil Υψηλότερης Τάξης [20]	9
Σχήμα 2.4 – Αναπαράσταση Stencil στους 3 άξονες [20]	10
Σχήμα 2.5 – Διασπορά Στοιχείων στη Μνήμη και Τρόπος Διαβάσματος της Μνήμης [20]....	10
Σχήμα 2.6 – Δευτεροταγής Δομή RNA [26]	13
Σχήμα 2.7 – Δεσμοί Βάσεων [26]	14
Σχήμα 3.1 – Συνδέσεις AE - MC στον συνεπεξεργαστή [4]	19
Σχήμα 3.2 – Block Diagram Διεπαφής MC [4]	20
Σχήμα 3.3 – AE σε MC σήματα αιτημάτων [4]	21
Σχήμα 3.4 – Ιεραρχία Μνήμης	22
Σχήμα 3.5 – Ιεραρχία Μνήμης μόνο με DIMMs.....	22
Σχήμα 3.6 – Binary Διαφύλλωση	23
Σχήμα 3.7 – Binary Διαφύλλωση μόνο με DIMM	23
Σχήμα 3.8 Απεικόνιση σύνδεσης διαύλων σχεδίασης με τους Memory Controllers μέσω του Crossbar [4]	25
Σχήμα 4.1 – Αρχιτεκτονική Convey HC-2 [10]	27
Σχήμα 4.2 – Αναπαράσταση Διαδοχικών Στοιχείων στη Μνήμη (Απόσταση 1 λέξη)	31
Σχήμα 4.3 – Κατανομή Στοιχείων σε Διαδοχικές Θέσεις Μνήμης στις Τράπεζες.....	32
Σχήμα 4.4 – Αναπαράσταση Στοιχείων στη Μνήμη με Απόσταση 64 bytes (8 λέξεις).....	33
Σχήμα 4.5 – Κατανομή Στοιχείων στις Τράπεζες – Απόσταση Στοιχείων 64 bytes (8 λέξεις).....	34
Σχήμα 4.6 – Αναπαράσταση Στοιχείων στη Μνήμη με Απόσταση 512 bytes (64 λέξεις).....	34
Σχήμα 4.7 – Κατανομή Στοιχείων στις Τράπεζες – Απόσταση Στοιχείων 512 bytes (64 λέξεις)	35
Σχήμα 4.8 – Αναπαράσταση Στοιχείων στη Μνήμη με Απόσταση 1024 bytes (128 λέξεις).....	36
Σχήμα 4.9 – Κατανομή Στοιχείων στις Τράπεζες – Απόσταση Στοιχείων 1024 bytes (128 λέξεις).....	36
Σχήμα 4.10 – Αναπαράσταση Στοιχείων στη Μνήμη με Απόσταση 8192 bytes (1024 λέξεις)	37
Σχήμα 4.11 – Κατανομή Στοιχείων στις Τράπεζες – Απόσταση Στοιχείων 8192 bytes (1024 λέξεις).....	38
Σχήμα 4.12 – Στοιχεία Q	41
Σχήμα 4.13 – Στοιχεία Qm	41
Σχήμα 4.14 – Στοιχεία Qprime	41

Σχήμα 4.15 – Μέθοδος Προσπέλασης Όλων των Στοιχείων του Πίνακα.....	43
Σχήμα 4.16 – Μέθοδος Προσπέλασης Convey των Στοιχείων του Πίνακα.....	44
Σχήμα 4.17 – Μέθοδος Προσπέλασης των Στοιχείων του Πίνακα με Υβριδική Μέθοδο	45
Σχήμα 4.18 – Βασική Αρχιτεκτονική Σχεδιάσεων	48
Σχήμα 5.1 – Τρόπος Λειτουργίας 1 DIMM.....	61
Σχήμα 5.2 – Αποκωδικοποίηση Διεύθυνσης από το Σύστημα Μνήμης.....	61
Σχήμα 5.3 – Τρόπος Λειτουργίας Συστήματος Μνήμης (16 DIMMs).....	62

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διαγράμματα 4.1 – Αποτελέσματα Profiling UNAFold	40
Διαγράμματα 4.2 – Απαιτήσεις Qprime, Q, Qm για διάφορα μήκη RNA	41
Διαγράμματα 5.1 – Ενδεικτικά Παραδείγματα Εμφάνισης Μοτίβου Υψηλών Τιμών.....	54
Διαγράμματα 5.2 – Ενδεικτικά Παραδείγματα Μετρήσεων (100 πρώτες επαναλήψεις των πειραμάτων).....	55
Διαγράμματα 5.3 – Αποτελέσματα Μετρήσεων (100 πρώτες επαναλήψεις των πειραμάτων)	58

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 4.1 – Αποτελέσματα Profiling UNAFold.....	40
Πίνακας 5.1 – Μετρήσεις Προσπέλασης Μεμονωμένων Στοιχείων.....	51
Πίνακας 5.2 – Αποτελέσματα Μετρήσεων Προσπέλασης Συνεχόμενων Στοιχείων (1 MC)...	56
Πίνακας 5.3 – Αποτελέσματα Μετρήσεων Προσπέλασης 1.000 Στοιχείων (1, 4, 8, και 16 MC)	57
Πίνακας 5.4 – Αποτελέσματα Μετρήσεων Προσπέλασης Στοιχείων με Άλματα Διευθύνσεων	59
Πίνακας 5.5 – Αποτελέσματα Μετρήσεων Προσπέλασης Στοιχείων – Βέλτιστο «Άλμα» στη Μνήμη	63
Πίνακας 5.6 – Αποτελέσματα Μετρήσεων Προσπέλαση Τριγωνικών Πινάκων	64

1

Εισαγωγή

1.1 Υβριδικοί Υπερ-Υπολογιστές

Τις τελευταίες δεκαετίες, η εξέλιξη των υπολογιστικών συστημάτων έχει οδηγήσει στην εκθετική αύξηση της πυκνότητας της «λογικής», όπως προβλέπεται από το Νόμο του Moore [1]. Τα χαρακτηριστικά των υπολογιστικών συστημάτων, όπως η συχνότητα του ρολογιού έχουν βελτιωθεί σημαντικά. Επίσης ο αριθμός των τρανζίστορ που χρησιμοποιούνται σε ποικίλες καινοτόμες αρχιτεκτονικές έχει αυξηθεί ραγδαία. Αποτέλεσμα αυτών των βελτιώσεων είναι η αύξηση των επιδόσεων των υπολογιστικών συστημάτων.

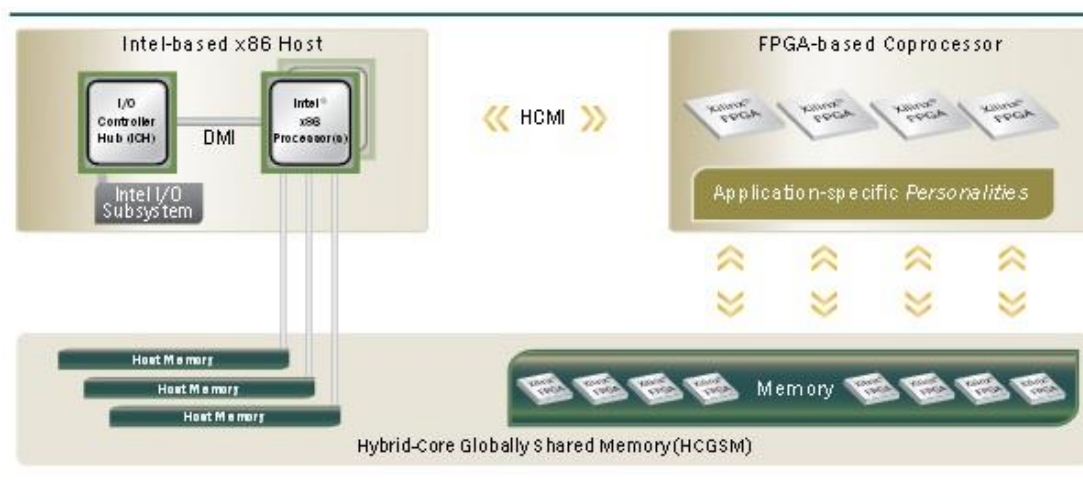
Ωστόσο, τα τελευταία χρόνια η απόδοση μονοπύρηνων μηχανημάτων έχει παραμείνει στάσιμη. Η αύξηση της πολυπλοκότητας του συστήματος και η αύξηση της συχνότητας του ρολογιού, έχουν καταστεί ο περιοριστικός παράγοντας στο σχεδιασμό. Στην πραγματικότητα, οι νόμοι της φυσικής έχουν δημιουργήσει ένα εμπόδιο που είναι σχεδόν αδύνατο να υπερνικηθεί με τους παραδοσιακούς επεξεργαστές και ημιαγωγικές αρχιτεκτονικές.

Σε μια προσπάθεια να παρακάμψουν τους νόμους της φυσικής, κάποιες εταιρείες έχουν κατασκευάσει ετερογενείς υπολογιστικές αρχιτεκτονικές, που χρησιμοποιούν πύλες ημιαγωγών σε πιο αποδοτικές διαμορφώσεις. Συγκεκριμένα, τα στοιχεία των υπολογιστών, όπως μονάδες επεξεργασίας γραφικών γενικής χρήσης και η αναδιατασσόμενη λογική χρησιμοποιούνται για την εκτέλεση λειτουργιών για συγκεκριμένες εφαρμογές απευθείας στο

hardware. Το αποτέλεσμα του συνδυασμού αυξάνει σημαντικά την αποτελεσματικότητα των υπολογιστικών συστημάτων με παράλληλη αύξηση της απόδοσής τους.

Εταιρίες όπως η «Convey computers» και η «Maxeler technologies» έχουν παρουσιάσει υβριδικούς υπερ-υπολογιστές και έχουν δημιουργήσει αρχιτεκτονικές που συνδυάζουν τους επεξεργαστές γενικού σκοπού με επεξεργαστές αναδιατασσόμενης λογικής, έτσι ώστε να κατασκευάσουν υπολογιστικά συστήματα με εξαιρετικές δυνατότητες.

Η «Convey computers» έχει κατασκευάσει αντίστοιχα έναν υβριδικό υπερ-υπολογιστή, του οποίου τα δύο μοντέλα είναι το HC-1 και HC-2(ex) (Σχήμα 1.1). Το υπολογιστικό σύστημα του υβριδικού υπερ-υπολογιστή αποτελείται από έναν επεξεργαστή με το βιομηχανικό πρότυπο της Intel και έναν συνεπεξεργαστή αποτελούμενο από FPGAs. Οι δύο επεξεργαστικές μονάδες έχουν κοινή μνήμη. Ο συνδυασμός αυτός παρέχει ένα σύστημα που προσφέρει βελτιωμένες επιδόσεις, χωρίς να θυσιάζει την ευελιξία και την ευκολία χρήσης του συστήματος [3, 10].



Σχήμα 1.1 – Αρχιτεκτονική Convey [3]

1.2 Σκοπός - Συνεισφορά

Σκοπός της παρούσας διπλωματικής εργασίας ήταν η βαθύτερη εξοικείωση με τη χρήση του υπερ-υπολογιστή Convey HC-2 με έμφαση στη χρήση της εξωτερικής μνήμης. Η αρχιτεκτονική της μνήμης και της διασύνδεσής της με τις FPGAs είναι μία καινοτομία του Convey και έτσι η μελέτη και μέτρηση της ταχύτητας του υποσυστήματος της μνήμης του συνεπεξεργαστή ήταν ο βασικός στόχος. Η συγκεκριμένη μελέτη έγινε με σκοπό την αξιοποίηση των αποτελεσμάτων, έτσι ώστε να προταθεί ένας εναλλακτικός τρόπος προσπέλασης των στοιχείων της μνήμης σε εφαρμογές απαιτητικές σε προσβάσεις στη μνήμη (Memory-Intensive Applications).

Η συνεισφορά της παρούσας εργασίας είναι:

i) Αποκτήθηκε τεχνογνωσία στη χρήση του υπερ-υπολογιστή Convey HC-2. Αποδείχτηκε ότι η σωστή και αποδοτική χρήση του μηχανήματος εξαρτάται από διάφορες παραμέτρους του συστήματος στον επεξεργαστή και στο συνεπεξεργαστή (FPGAs) και σε διαφορετικά επίπεδα της υλοποίησης. Οι παράμετροι αυτές καταγράφηκαν και μπορούν να αξιοποιηθούν σε μελλοντικές σχεδιάσεις.

ii) Τα πειράματα που πραγματοποιήθηκαν έδειξαν τον ακριβή τρόπο λειτουργίας του συστήματος διασύνδεσης της μνήμης με τους memory controllers και τις FPGAs. Από τα αποτελέσματα των πειραμάτων προκύπτει ο αποδοτικότερος τρόπος χρήσης των memory controllers από τις FPGAs. Η ακριβής ποσοτική μελέτη του υποσυστήματος μνήμης είναι ιδιαίτερα χρήσιμη για τη διαστασιολόγηση των προβλημάτων κατά τη διαδικασία σχεδίασης αρχιτεκτονικών.

iii) Τέλος η μέθοδος προσέλασης των στοιχείων της μνήμης που προτάθηκε απέδειξε ότι το σύστημα της μνήμης είναι ιδιαίτερα βελτιστοποιημένο και η απόδοσή του είναι δύσκολο να ξεπεραστεί, τουλάχιστον στο εύρος των πειραμάτων που διεξήχθησαν.

1.3 Δομή της Διπλωματικής

Στο Κεφάλαιο 2 γίνεται περιγραφή μερικών απαιτητικών δομών μνήμης, όπως τα stencils, ο πολλαπλασιασμός πινάκων και οι τριγωνικοί πίνακες (αλγόριθμος zucker) που ήταν το έναυσμα για τη μελέτη της διασύνδεσης της μνήμης με τις FPGAs. Επίσης παρουσιάζονται κάποιες εργασίες που έχουν γίνει για το Convey HC-1.

Στο Κεφάλαιο 3 αναλύονται τρόποι διαφύλλωσης της μνήμης και η σημασία τους για την επικοινωνία επεξεργαστή – μνήμης. Επίσης περιγράφεται ο τρόπος κατασκευής της μνήμης του Convey, η ιεραρχία και η διαφύλλωση και αναλύεται ο τρόπος διασύνδεσης της μνήμης με το συνεπεξεργαστή (FPGA) μέσω των διεπαφών των memory controllers.

Στο Κεφάλαιο 4 περιγράφεται ο τρόπος μοντελοποίησης των πειραμάτων που πραγματοποιήθηκαν στα πλαίσια της μελέτης της μνήμης. Τα πειράματα έγιναν για προσπελάσεις μεμονωμένων στοιχείων, μαζικών στοιχείων και στοιχείων τριγωνικών πινάκων.

Στο Κεφάλαιο 5 παρουσιάζονται και αναλύονται τα αποτελέσματα που προέκυψαν από τα πειράματα. Επίσης γίνεται περιγραφή του τρόπου λειτουργίας του συστήματος μνήμης του Convey σύμφωνα με τα συμπεράσματα των μετρήσεων.

Στο Κεφάλαιο 6 παρουσιάζονται συνοπτικά τα συμπεράσματα που προέκυψαν από την περάτωση της διπλωματικής εργασίας και γίνεται αναφορά σε μελλοντικές εργασίες με τις οποίες μπορεί να επεκταθεί.

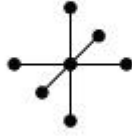
2

Σχετική Έρευνα

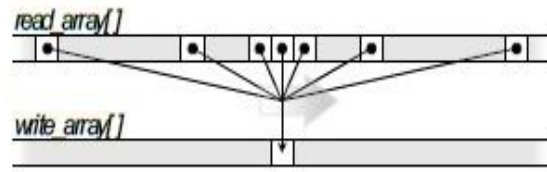
Η σχετική έρευνα που πραγματοποιήθηκε μπορεί να αποτυπωθεί ως έναυσμα για την εργασία που έγινε. Υπάρχουν πολλές απαιτητικές δομές μνήμης σε διάφορους αλγόριθμους όπως είναι τα stencils, ο πολλαπλασιασμός πινάκων από τον τομέα της γραμμικής άλγεβρας και η άντληση δεδομένων από τριγωνικούς πίνακες όπως στον αλγόριθμο UNAFold του Zuker. Όλα αυτά τα παραδείγματα έδωσαν το έναυσμα για τη μελέτη της διασύνδεσης της μνήμης με το συνεπεξεργαστή στο Convey HC-2, έτσι ώστε να γίνει ευκολότερος και αποδοτικότερος ο σχεδιασμός αρχιτεκτονικών για το συγκεκριμένο υπερ-υπολογιστή.

2.1 Stencil

Οι αλγόριθμοι που ασχολούνται με την επίλυση διαφορικών εξισώσεων αποτελούν ένα μεγάλο κομμάτι των επιστημονικών εφαρμογών σε ποικίλους τομείς, όπως ο ηλεκτρομαγνητισμός και η δυναμική των υγρών. Οι εφαρμογές αυτές, τα λεγόμενα stencils, συχνά χρησιμοποιούν επαναληπτικές τεχνικές «finite-difference» που σαρώνουν ένα χωρικό πλέγμα και εκτελούν υπολογισμούς εγγύτερου γείτονα. Σε έναν υπολογισμό stencil κάθε σημείο σε ένα πολυδιάστατο πλέγμα ενημερώνεται με σταθμισμένη συνεισφορά από ένα υποσύνολο των γειτόνων του, τόσο σε χρόνο όσο και σε χώρο. Αναπαραστάσεις ενός γενικού υπολογισμού stencil, καθώς και το μοτίβο πρόσβασης της μνήμης που προκύπτει για τον υπολογισμό του stencil φαίνονται στα Σχήμα 2.1 και Σχήμα 2.2.

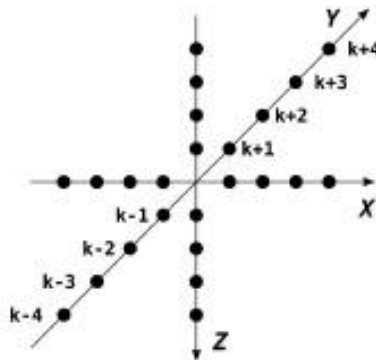


Σχήμα 2.1 – Αναπαράσταση Stencil [11]



Σχήμα 2.2 – Μνήμης για Υπολογισμό του Stencil [11]

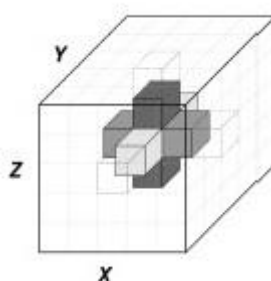
Οι stencil υπολογισμοί εκτελούν μεγάλες σαρώσεις σε δομές δεδομένων, που είναι συνήθως μεγαλύτερες από το μέγεθος της διαθέσιμης κρυφής μνήμης. Επιπλέον τα δεδομένα που επαναχρησιμοποιούνται περιορίζονται στον αριθμό των σημείων του stencil, που συνήθως είναι μικρός. Αυτοί οι υπολογισμοί καλύπτουν ένα χαμηλό ποσοστό της θεωρητικής απόδοσης, δεδομένου ότι τα στοιχεία από την κύρια μνήμη δεν μπορούν να μεταφερθούν αρκετά γρήγορα, με αποτέλεσμα να μην μπορεί να αποφευχθεί η στασιμότητα των υπολογιστικών μονάδων στους σύγχρονους μικροεπεξεργαστές. Η αναδιοργάνωση των stencil υπολογισμών συντελεί στην εκμετάλλευση της ιεραρχίας της μνήμης. Τα stencils επικεντρώνονται κυρίως σε «tiling optimizations» που τείνουν να εκμεταλλευτούν την τοπικότητα της μνήμης εκτελώντας εργασίες μεγέθους μπλοκ (μνήμης) πριν από τη μετάβαση στο επόμενο μπλοκ [9, 11].



Σχήμα 2.3 – Stencil Υψηλότερης Τάξης [20]

Εκτενείς έρευνες έχουν γίνει για την εφαρμογή 2D και 3D stencil σε hardware [12, 13]. Οι Durbano et al. [12] ήταν μεταξύ των πρώτων που πρότειναν μια FPGA εφαρμογή με 3D stencil, ως μέρος ενός πλήρους συστήματος επιτάχυνσης για προσομοίωση πεπερασμένων διαφορών πεδίου-χρόνου. Άλλες εφαρμογές 3D stencil σε FPGA's εκμεταλλεύονται την επαναχρησιμοποίηση των δεδομένων διοχετεύοντας τα δεδομένα μέσω πολλαπλών FIFOs. Ο κύριος περιορισμός αυτών των εφαρμογών είναι ότι επικεντρώνονται σε stencil μονού σημείου (π.χ. $2 \times 3 \times 2$ stencil). Παρ' ότι έχουν μελετηθεί εκτενώς, οι περισσότερες εφαρμογές της πραγματικής ζωής απαιτούν υψηλότερης τάξης stencils.

Τα stencils επίσης χρησιμοποιούνται σε επιστημονικούς τομείς, όπως είναι η ρευστοδυναμική, η γεωμετρική μοντελοποίηση, ο ηλεκτρομαγνητισμός και η επεξεργασία εικόνας. Στους τομείς αυτούς πραγματοποιείται χρήση επαναληπτικών τεχνικών πεπερασμένων διαφορών σάρωσης 2D και 3D πλέγματος. Σε κάθε 3D stencil υπολογισμό του σημείου είναι απαραίτητη η πρόσβαση σε δεδομένα και από τους τρεις άξονες του χώρου, όπως φαίνεται στο Σχήμα 2.4. Ως εκ τούτου ένας 3D stencil υπολογισμός αυξάνει την πολυπλοκότητα, όχι μόνο αυξάνοντας 3 φορές τον αριθμό των υπολογισμών, αλλά και μέσω των αραιών μοτίβων πρόσβασης των δεδομένων που προκύπτουν από τη γραμμικότητα της μνήμης. Παρουσιάζεται στο Σχήμα 2.5 ένα παράδειγμα 3D stencil υπολογισμού όσον αφορά τη διασπορά των στοιχείων στη μνήμη που χρειάζονται για τον υπολογισμό, αλλά και η διαδρομή στη μνήμη που απαιτείται για την ανάκτηση των συγκεκριμένων δεδομένων [12, 13].



Σχήμα 2.4 – Αναπαράσταση Stencil στους 3 άξονες [20]



Σχήμα 2.5 – Διασπορά Στοιχείων στη Μνήμη και Τρόπος Διαβάσματος της Μνήμης [20]

2.1.1 Stencil - Convey HC-1

Οι Jin Z. και Bakos J. [14] στην εργασία τους περιγράφουν μία τεχνική βελτίωσης της απόδοσης του εύρους ζώνης της μνήμης, μέσω ενός παραδείγματος για 3D stencil έξι σημείων. Η μνήμη του Convey αποτελείται από DRAM τράπεζες (Σχήμα 3.1), οι καθυστερήσεις που παρατηρούνται σε τέτοιας μορφής μνήμες δεν είναι πάντα ομοιόμορφες και εξαρτώνται από το μοντέλο πρόσβασης στη μνήμη. Οι memory controllers του Convey προσπαθούν να κρύψουν τις καθυστερήσεις στις προσβάσεις των DRAMs υποστηρίζοντας εκατοντάδες αιτήσεις μνήμης ταυτόχρονα και αναδιατάσσοντας τις αιτήσεις μεταφέροντάς τις, σε αδρανείς τράπεζες, έτσι ώστε να μην καθυστερούν. Η λειτουργία αυτή έχει ως αποτέλεσμα οι αιτήσεις να μην εξυπηρετούνται πάντα με τη σειρά που αποστέλλονται. Η προσπάθεια που γίνεται έχει ως στόχο, η σχεδίαση να έχει σε κάθε κύκλο ρολογιού πρόσβαση

στη μνήμη, που αποτελεί και το όριο της μνήμης. Η διεπαφή που προτείνεται στην εργασία αναδιατάσσει τις αιτήσεις μνήμης ανάμεσα στη σχεδίαση και στους memory controllers. Στη συνέχεια οι memory controllers αναδιατάσσουν ξανά αυτές τις αιτήσεις, αλλά η διεπαφή αποθηκεύει σε buffers τα δεδομένα που στέλνει η μνήμη και τα προωθεί στη σχεδίαση με την αναμενόμενη σειρά που η σχεδίαση τα περιμένει.

Στο παράδειγμα για υπολογισμούς 3D stencil έξι σημείων, η σχεδίαση περιμένει τα στοιχεία που χρειάζεται για κάθε υπολογισμό να έρθουν σε διαδοχικούς κύκλους, αλλά στη μνήμη τα στοιχεία αυτά δεν είναι αποθηκευμένα σε διαδοχικές θέσεις. Χωρίς τη διεπαφή που προτείνεται, τα στοιχεία που δεν είναι σε διαδοχικές θέσεις μνήμης θα διαβαστούν με τυχαία σειρά και το σύστημα της μνήμης καταλήγει σε χαμηλής απόδοσης αποτέλεσμα. Με τη χρήση της διεπαφής που προτείνουν όλες οι αιτήσεις δεδομένων εξυπηρετούνται με τη σειρά που ζητήθηκαν και προσεγγίζουν το επιθυμητό αποτέλεσμα απόδοσης (1 στοιχείο ανά κύκλο ρολογιού).

Μια άλλη μελέτη για stencil υπολογισμούς στο Convey HC-1 πραγματοποιήθηκε από μια ομάδα ερευνητών στο πανεπιστήμιο του Αμβούργου [16]. Στην ερευνητική αυτή μελέτη συγκρίνεται η απόδοση του Convey HC-1 για διάφορους τύπους stencil με την απόδοση άλλων υπολογιστικών συστημάτων όπως είναι οι CPUs και GPUs. Η μεγαλύτερη απόδοση που κατάφεραν να προσεγγίζουν είναι τα 22 Gflops/s στις FPGAs του Convey HC-1, που σε σύγκριση ακόμα και με έναν μονοπύρρηνο επεξεργαστή, που φτάνει τα 4 Gflops/s, δεν είναι αξιοσημείωτη. Στις συγκρίσεις με τις GPUs οι διαφορές ήταν μεγάλες, καθώς διαπίστωσαν ότι είναι δύσκολο να επιτευχθεί μεγάλη παραλληλία στο Convey γιατί τα δεδομένα πρέπει να τοποθετηθούν στη μνήμη χειροκίνητα.

2.2 Γραμμική Άλγεβρα

Η γραμμική άλγεβρα, μέσω των αλγορίθμων και των πράξεων που παρέχει, δίνει λύσεις σε πολλές επιστημονικές εφαρμογές. Με τη ραγδαία εξέλιξη της τεχνολογίας έχει καταστεί εφικτή η επιτάχυνση του hardware σε εφαρμογές της γραμμικής άλγεβρας με τη χρήση αναδιατασσόμενης λογικής. Βασικοί αλγόριθμοι και πράξεις της γραμμικής άλγεβρας βρίσκουν εφαρμογή σε συστήματα με βάση τις FPGAs, όπως το εσωτερικό γινόμενο, ο πολλαπλασιασμός και η παραγοντοποίηση πινάκων.

Ο πολλαπλασιασμός αραιών πινάκων κινητής υποδιαστολής είναι ένα βασικό υπολογιστικό μοντέλο που χρησιμοποιείται από πολλές εφαρμογές. Οι μεγάλες αποστάσεις των θέσεων των δεδομένων σε αραιούς πίνακες μειώνει σημαντικά την απόδοση των εφαρμογών που χρησιμοποιούν πολλαπλασιασμούς πινάκων με αριθμούς κινητής

υποδιαστολής σε υπολογιστές γενικού σκοπού. Οι εφαρμογές αυτές βασίζονται σε μεγάλο βαθμό στην ιεραρχία της κρυφής μνήμης για την επίτευξη υψηλών επιδόσεων. Οι άφθονοι πόροι υλικού που διαθέτουν οι σημερινές FPGAs προσφέρουν νέες ευκαιρίες για τη βελτίωση της απόδοσης τέτοιων προβλημάτων.

Οι FPGAs έχουν χρησιμοποιηθεί σε εφαρμογές ακεραίων και σταθερού προσήμου αριθμούς, όπως στην επεξεργασία σήματος. Ωστόσο με τη ραγδαία εξέλιξη της τεχνολογίας, οι FPGAs διαθέτουν περισσότερα CLBs από τους προκατόχους τους, με αποτέλεσμα να είναι πλέον εφικτή η χρησιμοποίησή τους, σε ένα ευρύτερο φάσμα εφαρμογών συμπεριλαμβανομένων και εκείνων που απαιτούν πράξεις κινητής υποδιαστολής [23]. Κάποιοι ερευνητές υποστηρίζουν ότι οι FPGAs έχουν γίνει άκρως ανταγωνιστικές με τους επεξεργαστές (CPUs) σε απόδοση [22]. Άλλοι ερευνητές έχουν χρησιμοποιήσει FPGAs για αρκετές υπολογιστικά απαιτητικές εφαρμογές, όπως η μοριακή δυναμική και ο πολλαπλασιασμός «πυκνών» πινάκων, και πέτυχαν πολύ υψηλές αποδόσεις [21, 24, 25]. Παράλληλα κάποιες εταιρίες έχουν αρχίσει να χρησιμοποιούν FPGAs για υπολογιστές υψηλών επιδόσεων ή ακόμα και υπερ-υπολογιστές, όπως το Convey και το Maxeler.

2.2.1 Γραμμική Άλγεβρα – Convey HC-1

Ο Bakos J. [8] σύγκρινε την απόδοση του Convey HC-1 για πολλαπλασιασμό πινάκων με μηχανήματα με διαφορετικές αρχιτεκτονικές, όπως ένας server με επεξεργαστή Dual Xeon 5520 και ένας server με κάρτα γραφικών (GPU) Nvidia Tesla S1070. Στην εργασία του παρουσιάστηκαν τα χαμηλής απόδοσης αποτελέσματα του Convey HC-1 για πολλαπλασιασμούς πινάκων με αριθμούς κινητής υποδιαστολής μονής ακρίβειας (Single-precision general matrix-matrix multiply) σε σχέση με τα άλλα δύο μηχανήματα. Επίσης παρουσίασε το πλεονέκτημα του Convey σε εφαρμογές με μεγάλες απαιτήσεις μνήμης (Memory-Intensive Applications), όπως ο αλγόριθμος Smith-Waterman. Κατέληξε στο συμπέρασμα ότι το Convey δεν μπορεί να ανταγωνιστεί τα όρια απόδοσης των άλλων δύο μηχανημάτων. Ωστόσο σε εφαρμογές οι οποίες χρειάζονται διανύσματα των οποίων τα στοιχεία δεν είναι «σωστά» ευθυγραμμισμένα, δηλαδή δεν βρίσκονται σε διαδοχικές θέσεις μνήμης, το Convey επιτυγχάνει υψηλότερες αποδόσεις από τα υπόλοιπα μηχανήματα, καθώς η αρχιτεκτονική του παρέχει μεγαλύτερη ευελιξία στην επικοινωνία με τη μνήμη.

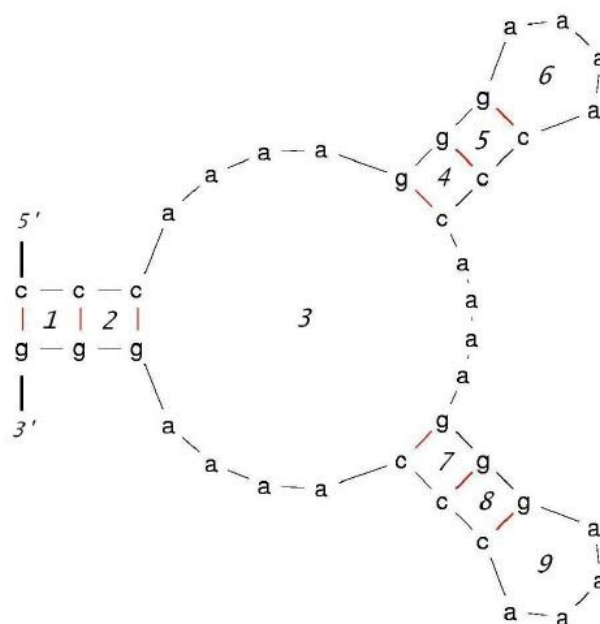
Σε μία άλλη εργασία οι Nagar K. και Bakos J. [18] παρουσίασαν την απόδοση του Convey HC-1 για πολλαπλασιασμούς αραιών πινάκων διπλής ακρίβειας αριθμών κινητής υποδιαστολής (sparse matrix-vector multiplier). Η καινοτομία τους αφορούσε τον τρόπο άντλησης και κυρίως τον τρόπο διαχείρισης των δεδομένων από τη μνήμη του Convey, έτσι ώστε να εκμεταλλεύονται όσο το δυνατόν περισσότερο το εύρος ζώνης που παρέχει το μηχανήμα με τα λιγότερα cache misses. Τα αποτελέσματα της απόδοσης του Convey HC-1 τα

σύγκριναν με μία NVIDIA Tesla S1070 κάρτα γραφικών, την απόδοση της οποίας ξεπέρασαν σε γενικές γραμμές.

2.3 Περιγραφή αλγορίθμου UNAFold του Zuker – Τριγωνικοί

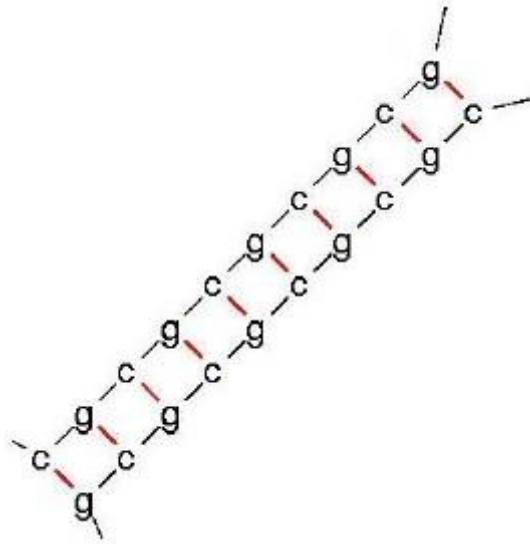
Πίνακες

Ο αλγόριθμος UNAFold του Zuker υπολογίζει τη δευτεροταγή δομή του RNA, που είναι η πρώτη εκτίμηση της αναπαράστασης του μακρομορίου στο χώρο. Η δευτεροταγής δομή του RNA (Σχήμα 2.6) είναι το δεύτερο επίπεδο οργάνωσης του μακρομορίου και περιέχει πληροφορία για την αναδίπλωσή του [2].



Σχήμα 2.6 – Δευτεροταγής Δομή RNA [26]

Για τον υπολογισμό της δομής αυτής του RNA είναι απαραίτητη η σύγκριση των ενεργειών που αναπτύσσονται ανάμεσα στους δεσμούς των βάσεων (Σχήμα 2.7) του μακρομορίου και η επιλογή του δεσμού που αναπτύσσει τη μικρότερη. Οι ενέργειες είναι προκαθορισμένες και έχουν υπολογιστεί από τους βιολόγους. Για την ευκολότερη πρόσβαση στις τιμές των ενεργειών τα δεδομένα είναι οργανωμένα σε πολυδιάστατους πίνακες. Κάθε διάσταση του πίνακα αντιστοιχεί στην τιμή κάποιας βάσης ή ζεύγους που ανήκει στο σχηματισμό που αναφέρεται η εκάστοτε ενέργεια. Αν ο σχηματισμός περιέχει τέσσερις βάσεις, ο αντίστοιχος πίνακας θα είναι τεσσάρων διαστάσεων.



Σχήμα 2.7 – Δεσμοί Βάσεων [26]

Το κύριο χαρακτηριστικό του αλγορίθμου όσον αφορά τη δυσκολία μεταφοράς του σε μία σχεδίαση για αναδιατασσόμενη λογική είναι ο μεγάλος αριθμός προσβάσεων στη μνήμη. Οι προσβάσεις αυτές προκύπτουν από δύο τριγωνικούς πίνακες Q_{prime} , Q και από τον πίνακα Q_m , των οποίων το μέγεθος είναι $N \times N$ όπου N ο αριθμός των βάσεων του μακρομορίου του RNA. Κάθε στοιχείο από αυτούς τους πίνακες αντιστοιχεί στην ενέργεια κάποιας δομής (ένωσης) [17, 26].

3

Μοντέλα πρόσβασης μνήμης

3.1 Διαφυλλωμένη μνήμη

Ο κύριος σκοπός στο σχεδιασμό ενός συστήματος μνήμης είναι η προσφορά επαρκούς χωρητικότητας αποθήκευσης με ταυτόχρονη διατήρηση ενός αποδεκτού επιπέδου μέσης απόδοσης και χαμηλού μέσου κόστους ανά δυαδικό ψηφίο. Αυτό επιτυγχάνεται χρησιμοποιώντας έναν αριθμό από διαφορετικές μονάδες μνήμης με διαφορετικούς λόγους κόστους/απόδοσης και το σχηματισμό μίας ιεραρχίας μνημών. Για την υλοποίηση των κρυφών μνημών και της κύριας μνήμης χρησιμοποιούνται ημιαγωγικές μνήμες, ενώ για την υλοποίηση της βοηθητικής μνήμης χρησιμοποιούνται μνήμες μαγνητικών δίσκων.

Για την υλοποίηση μνήμης υποστήριξης χρησιμοποιούνται μνήμες μαγνητικών ταινιών. Τα κύρια χαρακτηριστικά των ημιαγωγικών μνημών, που χρησιμοποιούνται για την υλοποίηση των κρυφών μνημών και της κύριας μνήμης, είναι ο μικρός χρόνος προσπέλασης. Ο χρόνος αυτός είναι σταθερός ανεξάρτητα από τη διεύθυνση της θέσης μνήμης που προσπελάσσεται. Οι μνήμες αυτές είναι μνήμες ανάγνωσης/εγγραφής και η πληροφορία που είναι αποθηκευμένη σ' αυτές χάνεται όταν κοπεί η τροφοδοσία. Τα κύρια χαρακτηριστικά των μνημών που χρησιμοποιούνται για την υλοποίηση της βοηθητικής μνήμης και της μνήμης υποστήριξης είναι το πολύ μικρό κόστος ανά δυαδικό ψηφίο αποθηκευμένης πληροφορίας και η διατήρηση της αποθηκευμένης πληροφορίας ακόμη και μετά τη διακοπή της τροφοδοσίας. Ο χρόνος προσπέλασης πληροφορίας που είναι αποθηκευμένη σ' αυτές τις μνήμες είναι πολύ μεγαλύτερος από ότι στις ημιαγωγικές μνήμες. Σ' έναν υπολογιστή η

ιεραρχία μνημών οργανώνεται κατά τέτοιο τρόπο ώστε οποιαδήποτε πληροφορία υπάρχει στη μνήμη ενός επιπέδου i να υπάρχει και στη μνήμη οποιουδήποτε άλλου επιπέδου j που βρίσκεται πιο μακριά από την Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) από το επίπεδο i . Στην ιεραρχία μνημών όσο απομακρυνόμαστε από την ΚΜΕ, η χωρητικότητα και ο χρόνος προσπέλασης της μνήμης αυξάνεται, ενώ το κόστος ανά δυαδικό ψηφίο μειώνεται. Η επιτυχία της ιεραρχίας μνημών οφείλεται σε μία ιδιότητα που έχουν τα περισσότερα προγράμματα, που καλείται τοπικότητα των αναφορών. Στα σύγχρονα υπολογιστικά συστήματα η ιεραρχία μνημών αποτελείται συνήθως από ένα ή δύο επίπεδα κρυφής μνήμης, την κύρια μνήμη και τη βοηθητική μνήμη.

Οι κύριοι παράγοντες που επηρεάζουν το ρυθμό μεταφοράς δεδομένων από και προς την κύρια μνήμη είναι η διαμόρφωση της κύριας μνήμης και τα χαρακτηριστικά των ολοκληρωμένων κυκλωμάτων που χρησιμοποιούνται για την υλοποίησή της. Υπάρχουν δύο διαφορετικοί τρόποι διαμόρφωσης της κύριας μνήμης, η υψηλής τάξης διαφύλλωση μνήμης και η χαμηλής τάξης διαφύλλωση μνήμης. Σύμφωνα με την υψηλής τάξης διαφύλλωση μνήμης, διαδοχικές διευθύνσεις αντιστοιχούνται σε διαδοχικές θέσεις του ίδιου τμήματος μνήμης. Σύμφωνα με τη χαμηλής τάξης διαφύλλωση μνήμης, διαδοχικές διευθύνσεις αντιστοιχούνται σε διαδοχικά τμήματα μνήμης. Το μεγάλο πλεονέκτημα της οργάνωσης N-δρόμων χαμηλής τάξης διαφύλλωσης είναι ότι τα περιεχόμενα N θέσεων μνήμης με διαδοχικές διευθύνσεις μπορούν να προσπελαστούν ταυτόχρονα σ' έναν κύκλο μνήμης.

Ένα παράδειγμα, η χωρητικότητα της κυρίας μνήμης είναι $N=2^n$ λέξεις. Τότε η φυσική διεύθυνση μιας θέσης μνήμης συνίσταται από n δυαδικά ψηφία $\alpha_{n-1}\alpha_{n-2}\dots\alpha_1\alpha_0$. Επίσης υποθέστε ότι η κύρια μνήμη αποτελείται από $M=2^m$ τμήματα μνήμης. Επομένως κάθε τμήμα περιέχει $2^n/2^m = 2^{n-m}$ διευθύνσεις. Υπάρχουν δύο τρόποι κατανομής των διευθύνσεων στα M τμήματα της μνήμης, ανάλογα με τη μέθοδο διαφύλλωσης που θα είναι διαμορφωμένη η μνήμη[19].

3.1.1 Υψηλής Τάξης Διαφύλλωση (high-order interleaving)

Σύμφωνα με τη μια μέθοδο οι διευθύνσεις κατανέμονται στα $M=2^m$ τμήματα μνήμης έτσι ώστε το πρώτο τμήμα μνήμης να περιέχει τις διαδοχικές διευθύνσεις από 0 έως και $2^{n-m}-1$, το δεύτερο τμήμα τις διευθύνσεις από 2^{n-m} έως και $2 \cdot 2^{n-m}-1$ κλπ. Γενικά το τμήμα μνήμης i , με $0 \leq i \leq M-1$, περιέχει τις διαδοχικές διευθύνσεις από $i \cdot 2^{n-m}$ μέχρι και $(i+1) \cdot 2^{n-m}-1$. Αυτή η κατανομή διευθύνσεων συνεπάγεται ότι τα m περισσότερα σημαντικά δυαδικά ψηφία της διεύθυνσης χρησιμοποιούνται για την επιλογή του τμήματος μνήμης, ενώ τα υπόλοιπα $n-m$ λιγότερο σημαντικά δυαδικά ψηφία χρησιμοποιούνται για την επιλογή της διεύθυνσης μέσα στο τμήμα μνήμης. Συχνά αυτός ο τρόπος κατανομής των διευθύνσεων μεταξύ των τμημάτων της μνήμης καλείται υψηλής τάξης διαφύλλωση μνήμης (high-order interleaving).

Αυτός ο τρόπος κατανομής των διευθύνσεων μεταξύ των τμημάτων της μνήμης έχει το πλεονέκτημα ότι επιτρέπει εύκολη επέκταση της μνήμης προσθέτοντας ένα ή περισσότερα τμήματα μνήμης έως ότου φτάσει τη μέγιστη τιμή $M-1$. Ωστόσο η τοποθέτηση διαδοχικών διευθύνσεων μνήμης στο ίδιο τμήμα μνήμης μπορεί να δημιουργήσει καθυστερήσεις σε κάποιες περιπτώσεις, όπως στην περίπτωση συστημάτων συνεχούς διοχέτευσης.

3.1.2 Χαμηλής Τάξης Διαφύλλωση (N-way low-order interleaving)

Σύμφωνα με την άλλη μέθοδο κατανομής διευθύνσεων μεταξύ των μονάδων μνήμης οι διευθύνσεις κατανέμονται έτσι ώστε διαδοχικές διευθύνσεις να ανήκουν σε διαδοχικά τμήματα μνήμης. Αυτή η κατανομή διευθύνσεων συνεπάγεται ότι τα m λιγότερο σημαντικά ψηφία της διεύθυνσης επιλέγουν τη μονάδα μνήμης και τα υπόλοιπα $n-m$ δυαδικά ψηφία επιλέγουν τη διεύθυνση μέσα σε κάθε μονάδα μνήμης. Δηλαδή, αν υπάρχουν $N=2^m$ μονάδες μνήμης, τότε οι διευθύνσεις $0, 1, 2, \dots, i, \dots$ κατανέμονται στις N μονάδες μνήμης σύμφωνα με τον κανόνα: η διεύθυνση i ανήκει στη μονάδα μνήμης M_j εάν $i=j(\bmod N)$. Αυτός ο τρόπος οργάνωσης της κύριας μνήμης καλείται N -δρόμων χαμηλής τάξης διαφύλλωση μνήμης (N -way low-order interleaving).

Το μεγάλο πλεονέκτημα της οργάνωσης N -δρόμων χαμηλής τάξης διαφύλλωσης είναι ότι τα περιεχόμενα N θέσεων μνήμης με διαδοχικές διευθύνσεις μπορούν να προσπελαστούν ταυτόχρονα σ' έναν κύκλο μνήμης [19].

3.2 Παράγοντας Διαφύλλωσης

Σκοπός της διαφυλλωμένης μνήμης είναι η εκμετάλλευση του δυναμικού εύρους ζώνης της μνήμης όλων των τσιπ στο σύστημα. Ωστόσο τα περισσότερα συστήματα μνήμης ενεργοποιούν μόνο τα τσιπ που περιλαμβάνουν τις απαραίτητες λέξεις. Αυτή η μέθοδος επηρεάζει την ισχύ του συστήματος της μνήμης οδηγώντας σε διαφορετικές αποφάσεις που εξαρτώνται από τη σχετική σημασία της ισχύος έναντι της απόδοσης.

Ένας τρόπος για να βελτιωθεί το εύρος ζώνης της μνήμης είναι η εκμετάλλευση του δυναμικού παραλληλισμού του εύρους, δηλαδή να έχουμε πολλές μονάδες στο σύστημα μνήμης. Οι μονάδες είναι συχνά εύρους μιας λέξης, με στόχο το εύρος του διαύλου και της μνήμης να μην αλλάζουν, αλλά στέλνοντας διευθύνσεις σε αρκετές μονάδες, να επιτρέπεται σε όλες αυτές να κάνουν ταυτόχρονη ανάγνωση.

Για παράδειγμα στέλνοντας μια διεύθυνση σε τέσσερις μονάδες προκύπτει μια ποινή αστοχίας των 76 κύκλων ρολογιού δίνοντας ένα εύρος ζώνης περίπου 0.4 byte ανά κύκλο ρολογιού. Οι μονάδες αυτές επίσης είναι χρήσιμες για τις εγγραφές στη μνήμη. Για την εξυπηρέτηση συνεχόμενων εγγραφών θα έπρεπε το σύστημα να περιμένει τις προηγούμενες

εγγραφές να τελειώσουν, η χρήση μονάδων επιτρέπει μια εγγραφή ανά κύκλο ρολογιού με την προϋπόθεση οι εγγραφές να μην προορίζονται για την ίδια μονάδα.

Η αντιστοίχιση των διευθύνσεων προς τις μονάδες επηρεάζει τη συμπεριφορά του συστήματος μνήμης. Το παρακάτω παράδειγμα προϋποθέτει ότι οι διευθύνσεις των τεσσάρων μονάδων θα διαφυλλώνονται στο επίπεδο λέξης. Η μονάδα 0 έχει όλες τις λέξεις των οποίων η διεύθυνση modulo 4 είναι 0, η μονάδα 1 έχει όλες τις λέξεις των οποίων η διεύθυνση modulo 4 είναι 1 και ομοίως για τα υπόλοιπα. Η αντιστοίχιση αυτή αναφέρεται ως ο παράγοντας διαφύλλωσης.

Η διαφυλλωμένη μνήμη προσδιορίζεται ως μονάδες μνήμης που είναι διαφυλλωμένες ανά λέξη. Η διαφύλλωση βελτιστοποιεί την ακολουθία των προσπελάσεων μνήμης. Μια αστοχία ανάγνωσης της μνήμης είναι ένα ιδεώδες ταίριασμα με τη διαφυλλωμένη μνήμη ανά λέξη γιατί οι λέξεις σε ένα μπλοκ αναγινώσκονται ακολουθιακά. Οι μνήμες ετεροχρονισμένης εγγραφής πραγματοποιούν εγγραφές καθώς και αναγνώσεις ακολουθιακά λαμβάνοντας ακόμα περισσότερη απόδοση από τη διαφυλλωμένη μνήμη ανά λέξη. Η ποσότητα των μονάδων που θα πρέπει να συμπεριληφθούν για τη διαφύλλωση της μνήμης αποτελεί ένα σημαντικό ερώτημα. Μια μετρική που χρησιμοποιείται σε ανυσματικούς υπολογιστές είναι η ακόλουθη:

$$\text{Αριθμός μονάδων} \geq \text{αριθμός κύκλων ρολογιού για την} \\ \text{προσπέλαση μιας λέξης στη μονάδα}$$

Στόχος του συστήματος μνήμης είναι να μεταφέρει πληροφορία από μια νέα μονάδα, ανά κύκλο ρολογιού για ακολουθιακές προσπελάσεις. Για να διαπιστωθεί, γιατί αυτός ο τύπος ισχύει, περιγράφεται ένα παράδειγμα, στο οποίο υπάρχουν λιγότερες μονάδες από τους κύκλους ρολογιού για τη προσπέλαση μιας λέξης σε μια μονάδα 64bit, όπως 8 μονάδες με χρόνο προσπέλασης 10 κύκλους ρολογιού. Μετά από 10 κύκλους ρολογιού η κεντρική μονάδα επεξεργασίας θα μπορούσε να πάρει μια λέξη από τη μονάδα 0 και μετά η μονάδα 0 θα ξεκινούσε ανάκληση της επόμενης επιθυμητής λέξης καθώς η κεντρική μονάδα επεξεργασίας έλαβε τις επόμενες 7 λέξεις από τις 7 μονάδες. Στον κύκλο ρολογιού 18 η κεντρική μονάδα επεξεργασίας θα ήταν στην πόρτα της μονάδας 0 περιμένοντάς την να φέρει την επόμενη λέξη. Η κεντρική μονάδα επεξεργασίας θα έπρεπε να περιμένει μέχρι τον κύκλο ρολογιού 20 για να εμφανιστεί η λέξη. Έτσι χρειάζονται περισσότερες μονάδες από κύκλους ρολογιού για να προσπελαστεί μια μονάδα ώστε να αποφευχθεί η αναμονή.

Όσο η χωρητικότητα ανά τσιπ μνήμης αυξάνει υπάρχουν λιγότερα τσιπ στο σύστημα μνήμης ίδιου μεγέθους κάνοντας τις πολλαπλές μονάδες πιο ακριβές. Για παράδειγμα μια κύρια μνήμη 512 MB παίρνει 256 τσιπ μνήμης των 4M*4bits οργανωμένα σε 16 μονάδες των 16 τσιπ μνήμης. Παρόλα αυτά χρησιμοποιεί μόνο 16 τσιπ μνήμης (64M*4bits) για 64MB καθιστώντας τη μία μονάδα το όριο. Πολλοί κατασκευαστές επιθυμούν μια επιλογή μικρής

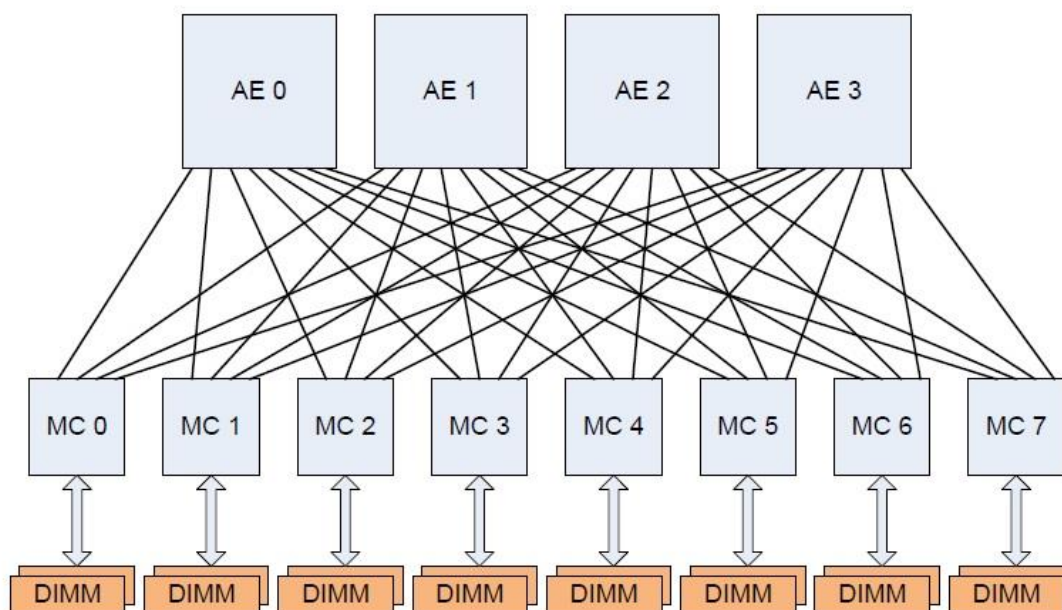
μνήμης στο βασικό μοντέλο. Αυτός ο συρρικνωμένος αριθμός τσιπ είναι το βασικό μειονέκτημα των μονάδων της διαφυλλωμένης μνήμης. Για τη διόρθωση αυτής της αδυναμίας τα τσιπ οργανώνονται σε ευρύτερα μονοπάτια των $16\text{ M} * 16\text{ bits}$. Ένα δεύτερο μειονέκτημα των μονάδων μνήμης είναι η δυσκολία της επέκτασης της κύριας μνήμης. Είτε το σύστημα μνήμης πρέπει να υποστηρίξει πολλαπλές γενιές τσιπ μνήμης, είτε ο ελεγκτής μνήμης να μπορεί να αλλάξει τη διαφύλλωση που βασίζεται στο μέγεθος της φυσικής μνήμης ή και τα δύο [19].

3.3 Μοντέλο Convey

3.3.1 Διεπαφή Memory controller

Η λειτουργία της διεπαφής του memory controller (MC) είναι να παρέχει στις FPGAs (Application Engine) τη δυνατότητα άμεσης πρόσβασης στη μνήμη του συνεπεξεργαστή. Όπως φαίνεται από το Σχήμα 3.1 κάθε μία από τις 4 Application Engine (AE) συνδέεται και με τους 8 MCs μέσω 300MHz διεπαφής DDR. Το Convey παρέχει τη διεπαφή του MC για τις FPGAs. Κάθε ένας από τους 8 MC είναι συνδεδεμένος σε φυσικό μέγεθος με το 1/8 της μνήμης του συνεπεξεργαστή και κάθε memory controller είναι συνδεδεμένος με 2 DIMMs.

Σκοπός της κάθε FPGA είναι να αποκωδικοποιεί τις εικονικές διευθύνσεις μνήμης, έτσι ώστε να στέλνει σε συγκεκριμένους MC τα αιτήματα για διαβάσματα και εγγραφές στη μνήμη [4].



Σχήμα 3.1 – Συνδέσεις AE - MC στον συνεπεξεργαστή [4]

3.3.2 Λειτουργικότητα Διεπαφής MC

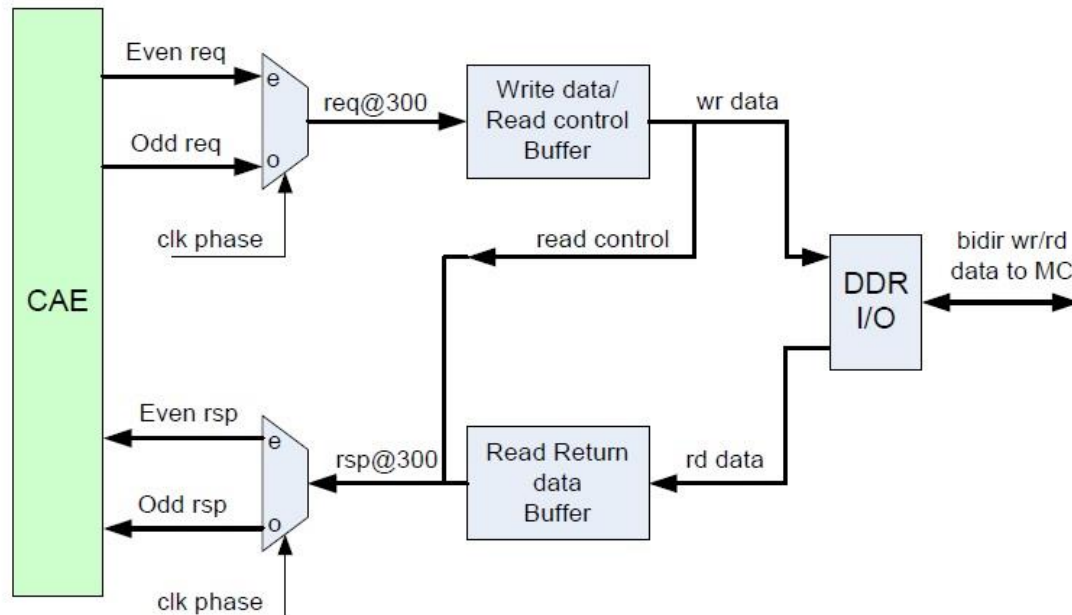
Η διασύνδεση μεταξύ του AE με τον MC τρέχει στα 300MHz, αλλά προκειμένου να διευκολυνθεί ο χρονισμός των FPGA, η διεπαφή των 300MHz μετατρέπεται σε 2 κανάλια μνήμης των 150MHz από και προς τη διεπαφή του AE, όπως φαίνεται στο Σχήμα 3.2.

3.3.2.1 Λειτουργία Εγγραφής

Τα δεδομένα που προορίζονται για εγγραφή στη μνήμη αποθηκεύονται σε μια FIFO(μνήμη) μέχρι να σταλούν για εγγραφή στη μνήμη του συνεπεξεργαστή. Για την εγγραφή δεδομένων δεν υπάρχει καμία «απάντηση» από τη μνήμη στις FPGAs.

3.3.2.2 Λειτουργία Ανάγνωσης

Για την ανάγνωση δεδομένων η διεπαφή του MC χρησιμοποιεί το δίαυλο και της εγγραφής και της ανάγνωσης. Στο δίαυλο εγγραφής αποθηκεύονται οι πληροφορίες ελέγχου που χρειάζεται η AE και στο δίαυλο ανάγνωσης τα δεδομένα ανάγνωσης. Όταν η αίτηση ανάγνωσης φτάσει στον MC τότε όλα τα δεδομένα και από τους δυο διαύλους αποθηκεύονται στον ελεγκτικό μηχανισμό της ανάγνωσης με σειρά προτεραιότητας, ανάλογα με τον τρόπο που ζητήθηκαν [4].

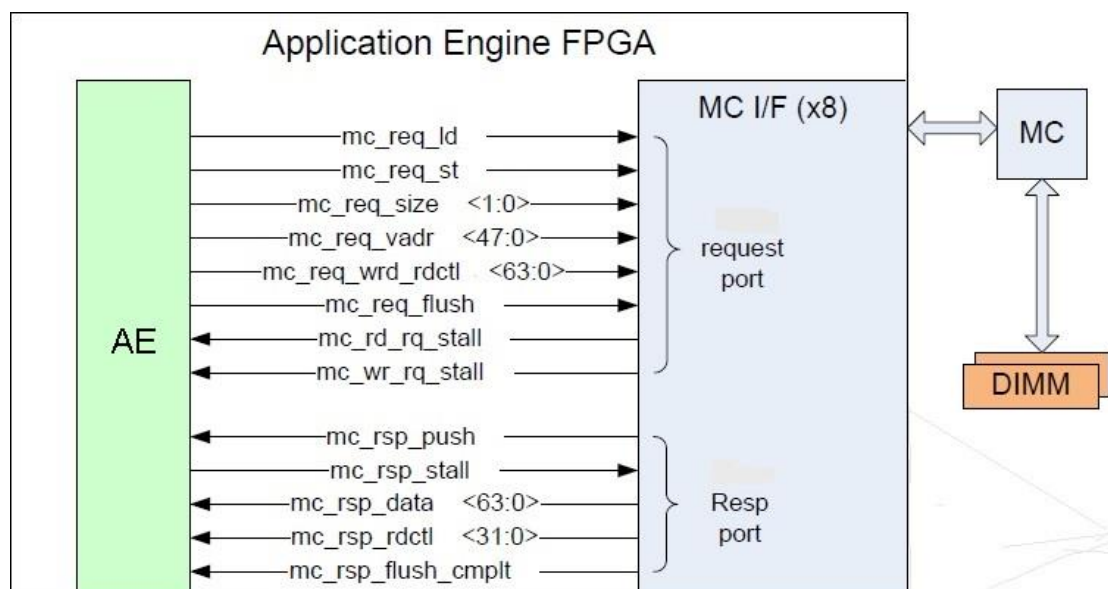


Σχήμα 3.2 – Block Diagram Διεπαφής MC [4]

3.3.3 Αιτήματα Μνήμης

Μία αίτηση χρήσης της μνήμης μπορεί να σταλεί ανά κύκλο ρολογιού (150MHz) εκτός και αν ο MC έχει αίτημα «καθυστερήσης» από τα σήματα `mc_rd_rq_stall` (ανάγνωση)

ή `mc_wr_rq_stall` (εγγραφή). Κάθε φορά που γίνεται αποστολή ενός αιτήματος στο MC πρέπει να ακολουθείται από το κατάλληλο σήμα ενεργοποιημένο, δηλαδή `mc_req_ld` (ανάγνωση) και `mc_req_st` (εγγραφή). Το σήμα `mc_req_size` δείχνει το μέγεθος «λέξης» των δεδομένων που ζητούνται ή γράφονται, δηλαδή ένα byte, μία λέξη (2bytes), διπλή λέξη (4bytes), τετραπλή λέξη (8bytes). Η διεύθυνση που στέλνεται σε κάθε αίτημα είναι 48-bit και πρέπει να στέλνεται στον ίδιο κύκλο που γίνεται το αίτημα ανάγνωσης ή εγγραφής [4].



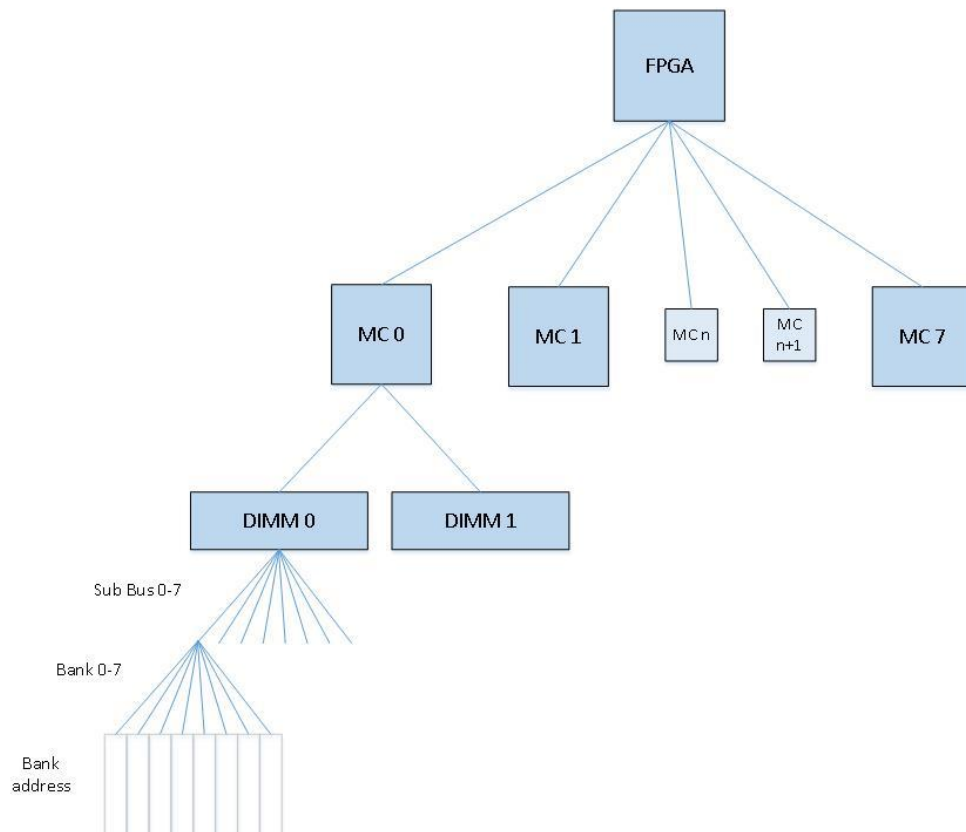
Σχήμα 3.3 – ΑΕ σε MC σήματα αιτημάτων [4]

3.3.4 Αποκρίσεις Μνήμης

Η απάντηση επιστρέφει στην ΑΕ (μόνο για τις αιτήσεις ανάγνωσης) με τη θύρα `mc_rsp_push`. Η θύρα αυτή λειτουργεί με ρολόι στα 150MHz. Τα δεδομένα προωθούνται από το MC σε κάθε κύκλο που το `mc_rsp_push` γίνεται '1'. Η απάντηση από τον MC μπορεί να ανασταλεί από την ΑΕ με το σήμα `mc_rsp_stall`. Από τη στιγμή που το σήμα `stall` γίνει '1' ο MC μπορεί να στείλει άλλες 5 επιπλέον απαντήσεις τις οποίες έχει δρομολογήσει πριν το σήμα `stall` ενεργοποιηθεί. Γι' αυτό το λόγο θεωρείται σκόπιμο το σήμα `stall` να είναι συνδεδεμένο με ένα σήμα «almost full» μιας FIFO, ώστε να μη χάνονται δεδομένα [4].

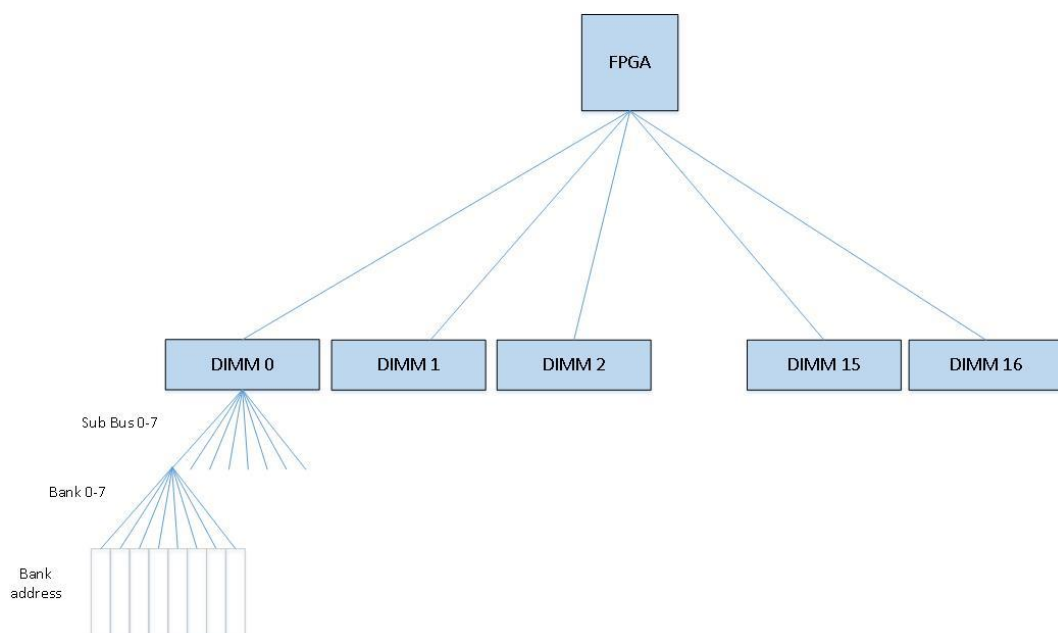
3.3.5 Σύστημα Μνήμης 1024 Bank

Το σύστημα μνήμης της Convey χρησιμοποιεί Scatter/Gather DIMMs και έχει 1024 τράπεζες μνήμης (banks). Οι τράπεζες είναι μοιρασμένες στους οκτώ memory controllers. Κάθε memory controller έχει δύο διαύλους (64-bits) τα DIMMs, και κάθε διάυλος (DIMM) έχει πρόσβαση σε οκτώ sub busses. Τέλος κάθε sub bus έχει οκτώ τράπεζες. Οι 1024 τράπεζες είναι αποτέλεσμα του $8 \text{ MCs} * 2 \text{ DIMMs/MC} * 8 \text{ sub bus/DIMM} * 8 \text{ bank/sub bus}$. Το Σχήμα 3.4 απεικονίζει το σύστημα ιεράρχησης της μνήμης του συνεπεξεργαστή [4].



Σχήμα 3.4 – Ιεραρχία Μνήμης

Εναλλακτικά η ιεραρχία της μνήμης μπορεί να παρουσιαστεί, όπως στο Σχήμα 3.5, χωρίς τους MCs μόνο με τα DIMMs αναπαριστώντας έτσι όλους τους πραγματικούς διαύλους επικοινωνίας FPGA-μνήμης. Η αναπαράσταση αυτή της ιεραρχίας της μνήμης χρησιμοποιείται στην παρούσα διπλωματική και βοηθάει στην κατανόηση των πειραμάτων που επιλέχθηκαν να πραγματοποιηθούν.

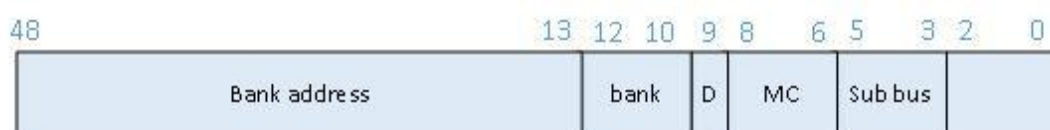


Σχήμα 3.5 – Ιεραρχία Μνήμης μόνο με DIMMs

Ο συνεπεξεργαστής του Convey υποστηρίζει δύο λειτουργίες διαφύλλωσης, τη δυαδική διαφύλλωση και την 31-31 διαφύλλωση. Η επιλογή της λειτουργίας διαφύλλωσης πραγματοποιείται κατά την εκκίνηση του προγράμματος [4].

3.3.6 Binary Διαφύλλωση

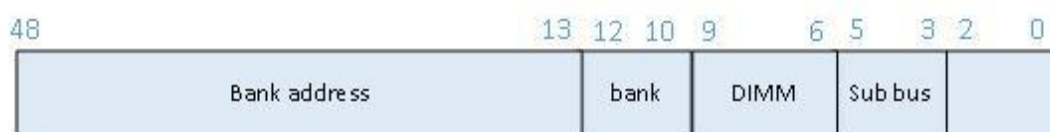
Όταν έχει επιλεχθεί η binary διαφύλλωση τα 1024 banks είναι προσβάσιμα με την ακόλουθη εικονική ανάθεση διευθύνσεων.



Σχήμα 3.6 – Binary Διαφύλλωση

Το 9 bit λειτουργεί για την επιλογή του ενός από τα δύο DIMMs στον εκάστοτε memory controller που χρησιμοποιείται [4].

Εναλλακτικά ακολουθώντας την ιεραρχία από το Σχήμα 3.5, η binary διαφύλλωση μπορεί να παρουσιαστεί όπως στο Σχήμα 3.7.



Σχήμα 3.7 – Binary Διαφύλλωση μόνο με DIMM

3.3.7 31/31 Διαφύλλωση

Το μοντέλο 31/31 διαφύλλωσης έχει διαμορφωθεί ώστε να πληρεί τις παρακάτω προϋποθέσεις:

- Παρέχει το μεγαλύτερο δυνατό εύρος ζώνης για άλματα στην πρόσβαση μνήμης.
- Κρατάει κάθε γραμμή μνήμης (64-bytes) σε ένα memory controller.
- Διατηρείται το μοτίβο διαφύλλωσης σε όλες τις προσπελάσεις των τμημάτων της εικονικής μνήμης. Αυτό βοηθάει στο να επιτυγχάνονται μεγάλα «άλματα» στα τμήματα μνήμης, που γίνονται λίγες προσπελάσεις.
- Όλες οι εικονικές διευθύνσεις μνήμης αντιστοιχούν σε μοναδικές φυσικές διευθύνσεις μνήμης.

Το σύστημα χρησιμοποιεί μια προσέγγιση δύο επιπέδων ιεραρχικής διαφύλλωσης. Τα 1024 banks χωρίζονται σε 32 ομάδες των 32 banks το καθένα. Το πρώτο επίπεδο διαφύλλωσης επιλέγει μία από τις 31 ομάδες των banks. Το δεύτερο επίπεδο διαφύλλωσης επιλέγει ένα από τα 31 banks από μια ομάδα. Αξίζει να σημειωθεί ότι από τις 32 ομάδες των

banks η μία δεν χρησιμοποιείται. Αντίστοιχα και ένα bank από κάθε ομάδα banks δεν χρησιμοποιείται. Ο αριθμός των 31 banks και 31 ομάδων των banks χρησιμοποιείται για τη μεγιστοποίηση του εύρους ζώνης της μνήμης για όσα περισσότερα άλματα μνήμης είναι δυνατόν. Βάσει αυτής της σύμβασης παρατηρείται 6% σπατάλη της μνήμης και κατ' επέκταση μείωση κατά 6% του μέγιστου εύρους ζώνης που μπορεί να αξιοποιηθεί [4].

3.3.8 Εύρος Ζώνης της Μνήμης

Το υποσύστημα της μνήμης στο συνεπεξεργαστή έχει δυνατότητα 80GB/s εύρος ζώνης μνήμης από τους AEs στη μνήμη του συνεπεξεργαστή. Προκειμένου να αξιοποιηθεί πλήρως το εύρος ζώνης της μνήμης πρέπει να ισχύουν τα παρακάτω:

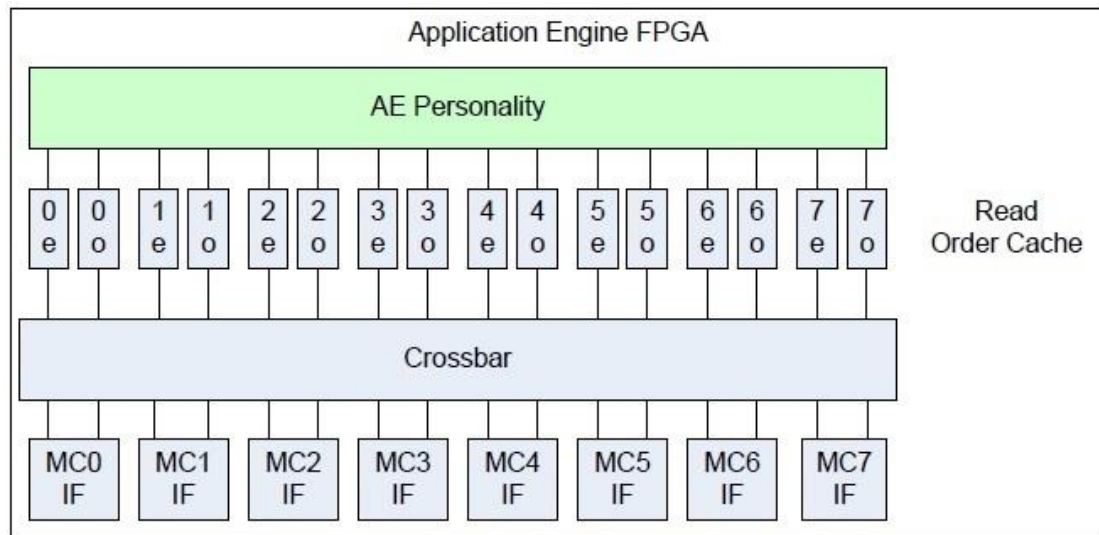
- a) Πρέπει να υποβάλλονται αιτήσεις από κάθε AE σε κάθε MC ανά κύκλο ρολογιού, εκτός και αν υπάρχει σήμα stall σε κάποιον MC.
- b) Ένα από τα παρεχόμενα μοντέλα διαφύλλωσης πρέπει να χρησιμοποιείται με τέτοιο τρόπο ώστε να διανέμει με βέλτιστο τρόπο τα αιτήματα μνήμης σε MCs, DIMMs και banks.

Κάθε διεπαφή AE-προς-MC παρέχει 2.5GB/s εύρος ζώνης. Όταν πολλαπλασιάζονται οι 4 AEs με τους 8 MCs, προκύπτουν οι 32 δίαυλοι που έχουν τη δυνατότητα να παρέχουν 80GB/s εύρος ζώνης. Ομοίως κάθε ένα από τα 16 DIMMs έχει τη δυνατότητα να μεταδίδει με 5GB/s, έτσι τα DIMMs ουσιαστικά επιτυγχάνουν να παρέχουν τα 80GB/s εύρος ζώνης. Τα μοντέλα διαφύλλωσης που περιγράφονται παραπάνω στοχεύουν στο να διανέμουν τα αιτήματα σε όλα τα MCs και DIMMs, έτσι ώστε να επιτευχθεί η καλύτερη δυνατή απόδοση.

Η απόδοση της μνήμης θα περιορίζεται εάν η εκάστοτε σχεδίαση δεν χρησιμοποιεί όλα τα διαθέσιμα AEs, MCs και DIMMs. Για παράδειγμα, αν μία σχεδίαση κάνει χρήση μόνο του MC_0 σε κάθε AE, τότε η διεπαφή AE-MC θα υποστηρίζει 10GB/s εύρος ζώνης. Ωστόσο αν όλα τα αιτήματα προορίζονται για το ίδιο DIMM το εύρος ζώνης περιορίζεται σε 5GB/s [4].

3.3.9 Προαιρετική Διεπαφή Crossbar MC

Η προαιρετική, αλλά συγχρόνως πολύ χρήσιμη επιλογή του crossbar, που παρουσιάζεται στο Σχήμα 3.8, χρησιμοποιείται για τη σύνδεση κάθε δίαυλου της σχεδίασης με κάθε μία διεπαφή των MC. Το crossbar επιτρέπει στη σχεδίαση να διατηρεί μια αφηρημένη άποψη για τη μνήμη, δεδομένου ότι την αποκωδικοποίηση των διευθύνσεων μνήμης των αιτήσεων/αποκρίσεων, τη χειρίζεται το crossbar. Το crossbar υποστηρίζει δυαδική διαφύλλωση.



Σχήμα 3.8 Απεικόνιση σύνδεσης διαύλων σχεδίασης με τους Memory Controllers μέσω του Crossbar [4]

Συνεπώς σε κάθε σχεδίαση υπάρχει ουσιαστικά η δυνατότητα χρήσης 16 DIMMs (8 MCs * 2 DIMMs) όπως φαίνεται στο Σχήμα 3.5 και όχι 8 όπως φαίνεται στο Σχήμα 3.4, γιατί ο κάθε MC έχει δύο διαύλους επικοινωνίας με τη μνήμη [4].

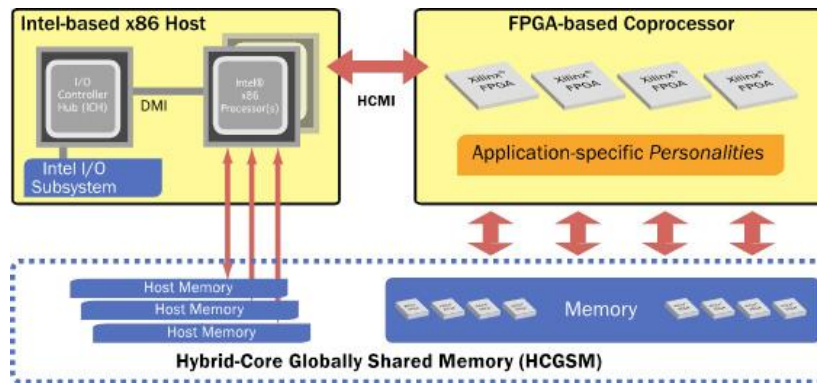
4

Μοντελοποίηση Προβλήματος – Μεθοδολογία

Μετρήσεων

Η ασχολία με έναν υβριδικό υπερ-υπολογιστή νέας και σχετικά άγνωστης τεχνολογίας, όπως είναι το Convey και το Maxeler εγείρει ποικίλα ερωτήματα για τον τρόπο λειτουργίας του. Τα μηχανήματα αυτά παρέχουν περισσότερες δυνατότητες και πόρους από τις απλές FPGAs, αλλά και τους συμβατικούς υπολογιστές, με αποτέλεσμα η υπολογιστική τους ισχύ να αυξάνεται σημαντικά. Η μελέτη υποσυστημάτων των μηχανημάτων αυτών, όπως είναι η διασύνδεση Μνήμης-FPGAs συνεισφέρει στην αποδοτικότερη χρήση των μηχανημάτων. Στόχος της παρούσας διπλωματικής εργασίας είναι να μελετηθεί ο τρόπος λειτουργίας της μνήμης του συνεπεξεργαστή και να βρεθεί ένας αξιόπιστος τρόπος μέτρησης της ταχύτητας με την οποία μπορεί η μνήμη του συνεπεξεργαστή να τροφοδοτεί και να τροφοδοτείται με δεδομένα από τις FPGAs στο Convey HC-2.

Η βελτιστοποίηση της ταχύτητας μεταφοράς δεδομένων είναι σημαντική, καθώς σε πολλές εφαρμογές, ο χρόνος εκτέλεσης των αλγορίθμων είναι μικρότερος από το χρόνο που χρειάζονται τα καλώδια διασύνδεσης (I/O), να τροφοδοτούν τις FPGAs με δεδομένα, καθυστερώντας τη συνολική διαδικασία. Το Convey HC-2 χρησιμοποιεί μια καινούρια αρχιτεκτονική (Σχήμα 4.1), ένας από του στόχους της οποίας είναι να βελτιώσει τον τρόπο και την ταχύτητα διασύνδεσης των FPGAs με τη μνήμη [4, 10].



Σχήμα 4.1 – Αρχιτεκτονική Convey HC-2 [10]

Για την καλύτερη κατανόηση των πειραμάτων κρίνεται σκόπιμη η παρουσίαση δύο σχημάτων (Σχήμα 3.5 και Σχήμα 3.7) που προέκυψαν από ορισμένες συμβάσεις, που ακολουθούνται στην παρούσα διπλωματική. Το πρώτο αφορά στην ιεραρχία της μνήμης και το δεύτερο στη διαφύλλωση της μνήμης που έχουν αναλυθεί στην Ενότητα 3.3. Στην παρούσα διπλωματική όταν αναφέρεται ο όρος MCs, θα εννοούνται οι memory controllers που χρησιμοποιεί η σχεδίαση πάνω από το crossbar (εικονικοί), ενώ όταν αναφέρεται ο όρος DIMMs, θα εννοούνται οι memory controllers κάτω από το crossbar, δηλαδή οι πραγματικοί (Σχήμα 3.8).

Οι μετρήσεις που πραγματοποιήθηκαν για τη μελέτη της ταχύτητας της μνήμης χωρίστηκαν σε τρεις μεγάλες ομάδες πειραμάτων. Το κριτήριο με το οποίο διαχωρίστηκαν τα πειράματα είναι ο τύπος δεδομένων που αιτείται να προσπελάσει κάθε φορά η σχεδίαση. Παρακάτω παρουσιάζονται συνοπτικά όλα τα πειράματα που πραγματοποιήθηκαν στα πλαίσια της διπλωματικής εργασίας.

Πειράματα:

- a) Χρόνος απόκρισης μνήμης για μεμονωμένες προσπελάσεις
 1. Μετρήσεις με τη χρήση 1 MC
 2. Μετρήσεις με τη χρήση 4 MCs
 3. Μετρήσεις με τη χρήση 8 MCs
 4. Μετρήσεις με τη χρήση 16 MCs
- b) Χρόνος απόκρισης μνήμης για μαζικές προσπελάσεις (burst data)
 1. Μετρήσεις προσπέλασης στοιχείων σε διαδοχικές θέσεις μνήμης
 - i. Μετρήσεις για διαφορετικό όγκο δεδομένων (8KB, 80KB, 800KB)
 - ii. Μετρήσεις με τη χρήση 1 MC
 - iii. Μετρήσεις με τη χρήση 4 MCs
 - iv. Μετρήσεις με τη χρήση 8 MCs
 - v. Μετρήσεις με τη χρήση 16 MCs
 2. Μετρήσεις προσπέλασης στοιχείων με «άλματα» (strides) διευθύνσεων

- c) Χρόνος απόκρισης μνήμης για προσπέλαση τριγωνικών πινάκων
 - 1. Μελέτη του αποτυπώματος μνήμης του αλγόριθμου UNAFold του Zuker
 - 2. Μετρήσεις προσπέλασης τριγωνικών πινάκων
 - i. Διάβασμα όλων των στοιχείων του πίνακα
 - ii. Διάβασμα του κάτω τριγωνικού πίνακα με τη μέθοδο του Convey
 - iii. Διάβασμα του κάτω τριγωνικού πίνακα με υβριδική μέθοδο

4.1 Χρόνος απόκρισης μνήμης για μεμονωμένες προσπελάσεις

Ως χρόνος απόκρισης (latency) της μνήμης ορίζεται ο χρόνος, από τη στιγμή αιτήματος μίας λέξης (μονάδα μνήμης) από τη μνήμη, μέχρι τη στιγμή που αυτή η λέξη θα είναι διαθέσιμη προς αξιοποίηση από τη μονάδα επεξεργασίας που πραγματοποιήσει το αίτημα.

Η συγκεκριμένη ομάδα μετρήσεων διέπεται από συγκεκριμένους κανόνες. Αρχικά αποθηκεύονται τιμές σε προκαθορισμένες διευθύνσεις μνήμης και στη συνέχεια στέλνονται αιτήματα ανάγνωσης για την προσπέλαση αυτών των διευθύνσεων. Επιλέχθηκε, μετά από πειράματα, κάθε ανάγνωση διεύθυνσης να πραγματοποιείται 1.000 φορές επαναληπτικά. Ο αριθμός αυτός παρέχει ένα ασφαλές μέσο όρο χρόνου προσπέλασης ενός δεδομένου από τη μνήμη. Μεταξύ των αναγνώσεων αφήνεται ένα διάστημα «ηρεμίας» του συστήματος που αντιστοιχεί σε 2.000 κύκλους ρολογιού. Στόχος της επιλογής αυτής είναι το νούμερο να είναι αρκετά μεγάλο, έτσι ώστε το σύστημα μνήμης-FPGA να έχει προλάβει να επανέλθει σε κατάσταση «ηρεμίας».

Ο μετρούμενος χρόνος έχει ως έναρξη τη στιγμή που η σχεδίαση στείλει ένα αίτημα για ανάγνωση μιας διεύθυνσης και σταματάει όταν η μνήμη στείλει, μέσω του συστήματος των MCs, το περιεχόμενο της διεύθυνσης που ζητήθηκε. Η μέτρηση του χρόνου γίνεται σε κύκλους ρολογιού. Στη συνέχεια υπολογίζεται ο μέσος όρος όλων των χρόνων που προέκυψαν από τις 1.000 επαναλήψεις της ίδιας μέτρησης.

Οι πρώτες μετρήσεις είχαν ως στόχο να βρεθεί ένα σημείο αναφοράς, ώστε με γνώμονα αυτό το σημείο να επιλεγούν και οι επόμενοι τύποι πειραμάτων. Το πρώτο σημαντικό χαρακτηριστικό της μνήμης που μελετήθηκε είναι ο χρόνος απόκρισής της για αιτήσεις μεμονωμένων αιτημάτων των FPGAs.

Τα πρώτα πειράματα έγιναν με τη χρήση ενός MC. Στη συγκεκριμένη κατηγορία πειραμάτων πραγματοποιήθηκαν τέσσερις διαφορετικές μετρήσεις αλλάζοντας σε κάθε μέτρηση τη διεύθυνση του στοιχείου που ζητείται, έτσι ώστε ο κάθε τύπος μέτρησης να ζητάει στοιχεία που η προσπέλασή τους γίνεται από διαφορετική δομή της μνήμης, σύμφωνα με τη διαφύλλωση της μνήμης όπως περιγράφεται στο Σχήμα 3.7.

Στη συνέχεια πραγματοποιήθηκαν πειράματα με τη χρήση 4, 8 και 16 MCs από τη σχεδίαση ταυτόχρονα. Για κάθε πείραμα αυτής της κατηγορίας πραγματοποιήθηκαν, επίσης, τέσσερις διαφορετικές μετρήσεις. Στις μετρήσεις αυτές η μόνη διαφορά είναι η απόσταση των διευθύνσεων των στοιχείων που ζητούνται από τη σχεδίαση. Παραδείγματος χάρη, για δύο πειράματα με 8 MCs γίνεται προσπέλαση 8 στοιχείων σε κάθε μέτρηση (ένα στοιχείο από κάθε MC). Στο ένα πείραμα τα στοιχεία που ζητούνται είναι σε διαδοχικές θέσεις (συνεχόμενα) στη μνήμη (απέχουν το ένα από το άλλο 8 bytes απόσταση), ενώ στο άλλο τα στοιχεία απέχουν το ένα από το άλλο 64 bytes (μη διαδοχικές θέσεις μνήμης).

Η διάκριση αυτή των μετρήσεων γίνεται με σκοπό να μελετηθεί αν και κατά πόσο η απόσταση των διευθύνσεων των στοιχείων που ζητείται να προσπελαστούν επηρεάζει την ταχύτητα με την οποία ανταποκρίνεται το σύστημα της μνήμης. Επιλέχθηκαν τέσσερις διαφορετικές αποστάσεις διευθύνσεων όσες είναι και οι διαφορετικές δομές της μνήμης (Σχήμα 3.5 Σχήμα 3.4), έτσι ώστε σε κάθε μέτρηση να αλλάζει μόνο ένα πεδίο της διεύθυνσης των στοιχείων. Παραδείγματος χάρη, όταν αλλάζει η διεύθυνση των στοιχείων κατά 1024 bytes, τα στοιχεία αυτά θα βρίσκονται σε διαφορετική «τράπεζα», αλλά τα υπόλοιπα χαρακτηριστικά της διεύθυνσης θα παραμένουν ίδια (DIMM, sub bus, bank address), όπως φαίνεται από το Σχήμα 3.7.

Οι πέντε διαφορετικοί τύποι μετρήσεων έχουν ως εξής:

1. Απόσταση διευθύνσεων 8 bytes.

Σε αυτήν την περίπτωση τα στοιχεία βρίσκονται σε διαδοχικές θέσεις στη μνήμη, γιατί κάθε «λέξη» στη μνήμη του Convey HC-2 έχει μέγεθος 8 bytes (block size). Τα στοιχεία αυτά θα εξυπηρετούνται από το σύστημα της μνήμης από το ίδιο DIMM, αλλά από διαφορετικό sub bus και συνεπώς θα βρίσκονται και σε διαφορετική τράπεζα (Σχήμα 3.5, Σχήμα 3.7).

2. Απόσταση διευθύνσεων 64 bytes

Τα στοιχεία σε αυτές τις μετρήσεις θα έχουν απόσταση 8 «λέξεων» ($8 \text{ λέξεις} * 8 \text{ bytes}$) μεταξύ τους. Συνεπώς σύμφωνα με τη διαφύλλωση της μνήμης τα στοιχεία θα εξυπηρετηθούν από διαφορετικούς διαύλους της μνήμης το καθένα, διαφορετικά DIMMs.

3. Απόσταση διευθύνσεων 1024 bytes

Στην περίπτωση αυτή τα στοιχεία απέχουν κατά 128 «λέξεις» το ένα από το άλλο. Τα στοιχεία θα εξυπηρετηθούν από το ίδιο DIMM, το ίδιο sub bus, αλλά θα βρίσκονται σε διαφορετική τράπεζα.

4. Απόσταση διευθύνσεων 8192 bytes

Στην τελευταία περίπτωση τα στοιχεία θα έχουν απόσταση μεταξύ τους 1024 «λέξεις». Ο τρόπος με τον οποίο θα εξυπηρετηθούν είναι το σύστημα της μνήμης

να χρησιμοποιήσει το ίδιο DIMM, το ίδιο sub bus και τα στοιχεία να βρίσκονται και στην ίδια τράπεζα, αλλά να έχουν διαφορετική διεύθυνση (bank address).

4.2 Χρόνος απόκρισης μνήμης για μαζικές προσπελάσεις

(burst data)

Μετά τα πειράματα για τη μέτρηση του χρόνου απόκρισης της μνήμης για μεμονωμένα στοιχεία ακολούθησαν πειράματα για τη μέτρηση του χρόνου απόκρισης της μνήμης για μαζικές (διαδοχικές) προσπελάσεις στη μνήμη. Οι περισσότερες εφαρμογές χρειάζονται πολλά δεδομένα σε κάθε κύκλο για τον υπολογισμό αλγορίθμων και συνεπώς ένα ζητούμενο μέγεθος που πρέπει να μελετηθεί είναι ο χρόνος που χρειάζεται η μνήμη να ανταποκριθεί σε μαζικά αιτήματα ανάγνωσης δεδομένων από τις FPGAs.

Η συγκεκριμένη ομάδα μετρήσεων διέπεται από ορισμένους γενικούς κανόνες. Αρχικά αποθηκεύονται τιμές σε προκαθορισμένες διευθύνσεις μνήμης και στη συνέχεια στέλνονται αιτήματα ανάγνωσης για την προσπέλαση αυτών των διευθύνσεων. Επιλέχθηκε μετά από πειράματα κάθε ομάδα προσπελάσεων να πραγματοποιεί 1.000 διαδοχικές προσπελάσεις σε κάθε μέτρηση. Το νούμερο αυτό παρέχει έναν αρκετά μεγάλο αριθμό προσπελάσεων ώστε να εξασφαλίζεται ένα ασφαλές συμπέρασμα από τους χρόνους που θα μετρηθούν. Ανάμεσα σε κάθε ομάδα αναγνώσεων αφήνεται ένα διάστημα «ηρεμίας» του συστήματος 2.000 κύκλων ρολογιού. Στόχος της επιλογής αυτής είναι το νούμερο να είναι αρκετά μεγάλο έτσι ώστε το σύστημα μνήμης-FPGA να έχει προλάβει να επανέλθει σε κατάσταση «ηρεμίας». Πραγματοποιήθηκε επανάληψη των μετρήσεων 1.000 φορές για διαφορετικές ομάδες στοιχείων.

Η έναρξη του μετρούμενου χρόνου γίνεται τη στιγμή που η σχεδίαση στέλνει το πρώτο αίτημα ανάγνωσης. Στη συνέχεια η σχεδίαση περιμένει έως ότου η μνήμη στείλει μέσω των DIMMs το περιεχόμενο όλων των διευθύνσεων που ζητήθηκαν και σταματάει τη μέτρηση μόλις σταλεί το τελευταίο στοιχείο. Η μέτρηση του χρόνου γίνεται σε κύκλους ρολογιού. Στη συνέχεια υπολογίζεται ο μέσος όρος όλων των χρόνων που προέκυψαν από τις 1.000 επαναλήψεις της ίδιας μέτρησης.

Οι ομάδες πειραμάτων σε αυτή την ενότητα χωρίζονται ανάλογα με την απόσταση των διευθύνσεων των στοιχείων που ζητούνται από τη μνήμη. Η FPGA αιτείται στοιχεία από τη μνήμη για προσπέλαση με δύο τρόπους. Στον πρώτο τρόπο η σχεδίαση αιτείται στοιχεία που βρίσκονται σε διαδοχικές θέσεις στη μνήμη και στο δεύτερο τρόπο στοιχεία που είναι σε μη διαδοχικές θέσεις και το σύστημα χρειάζεται να κάνει «άλματα» (strides) στη μνήμη.

4.2.1 Μετρήσεις προσπέλασης στοιχείων σε διαδοχικές θέσεις μνήμης

Τα πρώτα πειράματα έγιναν με τη χρήση ενός MC. Οι μετρήσεις αυτής της ομάδας πειραμάτων έγιναν για στοιχεία που βρίσκονται σε διαδοχικές θέσεις στη μνήμη, δηλαδή η απόσταση μεταξύ των στοιχείων είναι μία «λέξη» ή 8 bytes. Στο Σχήμα 4.2 απεικονίζεται ο τρόπος με τον οποίο είναι τοποθετημένα τα στοιχεία με διαφορά 1 «λέξης» στη μνήμη.

	j						
i	1	2	3	4	5	6	7
	8	9	10	11	->		
		->	N-1	N	N+1	->	

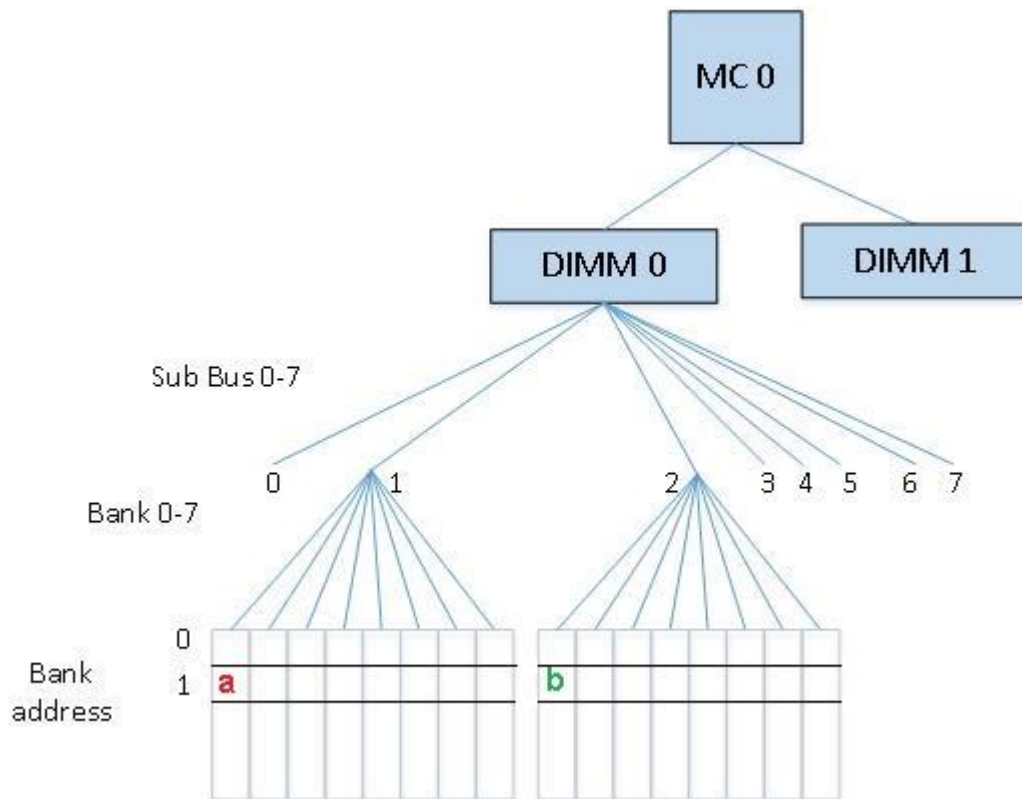
Σχήμα 4.2 – Αναπαράσταση Διαδοχικών Στοιχείων στη Μνήμη (Απόσταση 1 λέξη)

Για την κατανόηση του τρόπου με τον οποίο τα στοιχεία είναι κατανομημένα σε τράπεζες αλλά και για το μονοπάτι που ακολουθεί το σύστημα της μνήμης για την προσπέλαση των στοιχείων αυτών, παρουσιάζεται ένα παράδειγμα. Για δύο στοιχεία σε διαδοχικές θέσεις μνήμης αναπαριστώνται οι διευθύνσεις τους (σε δεκαδικό σύστημα) η (a) 8200 (bytes) και η (b) 8208 (bytes) και απεικονίζονται γραφικά στο Σχήμα 4.3 όταν τα δύο στοιχεία βρίσκονται σε διαφορετική τράπεζα. Επίσης φαίνεται ότι το μονοπάτι που θα ακολουθηθεί για την προσπέλαση των στοιχείων διαφοροποιείται στο επίπεδο του sub bus.

(a) 8200 σε δυαδικό [...0001][000][0][000][001][000] (bytes)

(b) 8208 σε δυαδικό [...0001][000][0][000][010][000] (bytes)

Ο διαχωρισμός στα νούμερα σε δυαδική μορφή γίνεται με βάση τη διαφύλλωση της μνήμης (Σχήμα 3.6).



Σχήμα 4.3 – Κατανομή Στοιχείων σε Διαδοχικές Θέσεις Μνήμης στις Τράπεζες

Στην περίπτωση των μετρήσεων με ένα MC για διαδοχικά στοιχεία στη μνήμη πραγματοποιήθηκαν πειράματα με διαφορετικό πλήθος προσπελάσεων στη μνήμη. Εκτός από τη προσπέλαση 1.000 στοιχείων που περιγράφονται στους γενικούς κανόνες της ομάδας πειραμάτων, πραγματοποιήθηκαν μετρήσεις για 10.000 και για 100.000 προσπελάσεις διαδοχικών στοιχείων στη μνήμη. Οι μετρήσεις αυτές πραγματοποιήθηκαν για να διαπιστωθεί αν αλλάζει η συμπεριφορά του συστήματος μνήμης από την αλλαγή του όγκου των δεδομένων που αιτούνται προσπέλασης στο χρόνο.

Στη συνέχεια έγιναν πειράματα με τη χρήση περισσότερων MCs για την προσπέλαση 1.000 διαδοχικών στοιχείων ο καθένας. Χρησιμοποιήθηκαν 4, 8 και 16 MCs από τη FPGA ταυτόχρονα, ώστε να μελετηθεί η ανταπόκριση του συστήματος μνήμης σε μεγαλύτερο όγκο αιτημάτων στον ίδιο χρόνο. Σε κάθε κύκλο ρολογιού στις μετρήσεις με ένα MC η σχεδίαση έκανε αποστολή ενός αιτήματος για «διάβασμα», ενώ σε αυτές τις μετρήσεις η σχεδίαση στέλνει 4, 8 και 16 αιτήματα αντίστοιχα ανά κύκλο ρολογιού. Με αυτόν τον τρόπο η σχεδίαση αυξάνει τον αριθμό των αιτημάτων που στέλνει στο χρόνο, γιατί ενώ με ένα MC έστελνε 1.000 αιτήσεις προσπέλασης σε 1.000 κύκλους, τώρα στον ίδιο χρόνο στέλνει 4.000, 8.000 και 16.000 αιτήσεις αντίστοιχα.

4.2.2 Μετρήσεις προσπέλασης στοιχείων με «άλματα» (strides) διευθύνσεων

Σε πολλές εφαρμογές τα στοιχεία που απαιτούνται για υπολογισμούς δεν βρίσκονται σε διαδοχικές θέσεις μνήμης, γι' αυτό το λόγο πραγματοποιήθηκαν μετρήσεις για την προσπέλαση τέτοιων στοιχείων.

Μετρήσεις για την απόδοση του Convey HC-1 για δεδομένα που απαιτούν «άλματα» στη μνήμη αναφέρονται στην ερευνητική εργασία του Bakos [8]. Στην παρούσα διπλωματική εργασία μελετήθηκε ο τρόπος που ανταποκρίνεται το σύστημα της μνήμης σε αιτήματα της FPGA για στοιχεία που απαιτούν τεσσάρων ειδών «άλματα» στη μνήμη.

1. Απόσταση διευθύνσεων 64 bytes

Στις μετρήσεις αυτές τα στοιχεία που η σχεδίαση ζητάει από τη μνήμη να προσπελάσει απέχουν μεταξύ τους 64 bytes, δηλαδή 8 «λέξεις». Στο Σχήμα 4.4 απεικονίζεται ένα παράδειγμα του τρόπου με τον οποίο είναι τοποθετημένα τα στοιχεία που ζητούνται στη μνήμη και η σειρά με την οποία η σχεδίαση στέλνει τις αιτήσεις.

		j											
i	1			<-	8	->			2			<-	8
	->			3			<-	8	->			4	
		<-	8	->			5			<-	8	->	
		6			<-	8	->			7	->		

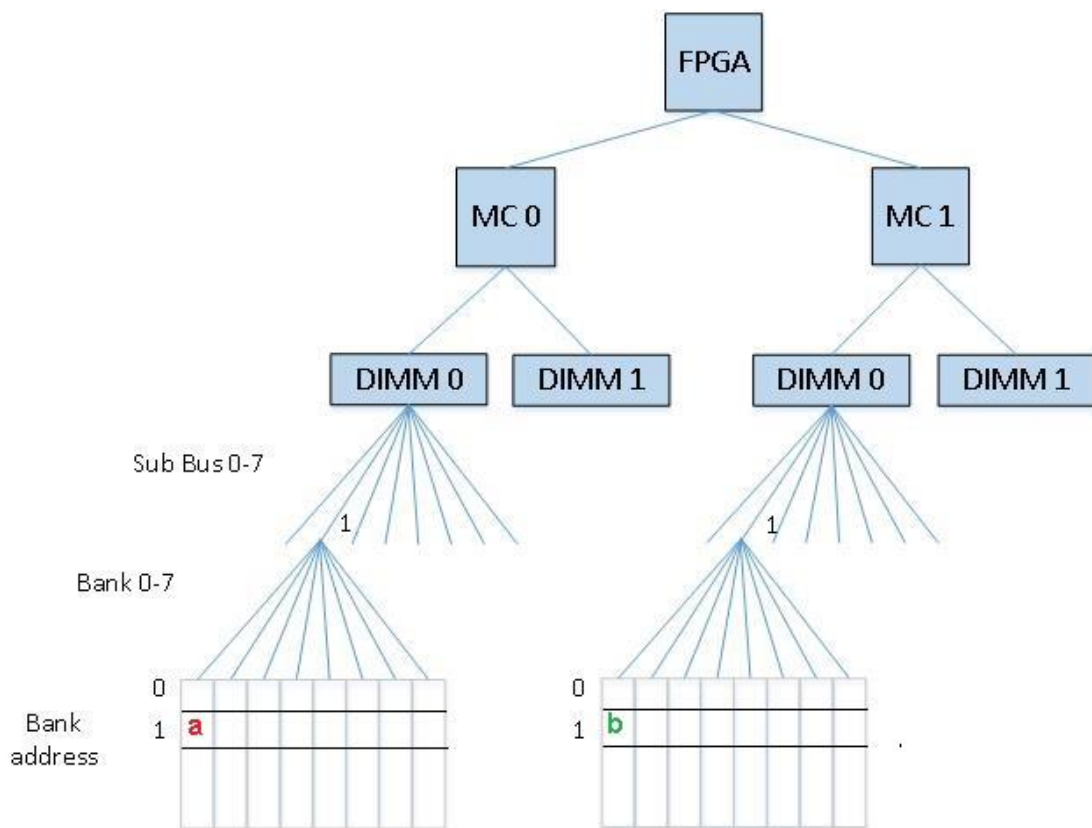
Σχήμα 4.4 – Αναπαράσταση Στοιχείων στη Μνήμη με Απόσταση 64 bytes (8 λέξεις)

Τα στοιχεία των οποίων οι διευθύνσεις απέχουν 64 bytes εξυπηρετούνται από το σύστημα μνήμης από διαφορετικό MC το καθένα. Για να γίνει κατανοητό, πώς είναι τα στοιχεία κατανομημένα στις τράπεζες της μνήμης, αλλά και τα μονοπάτια που θα ακολουθηθούν στη μνήμη για την προσπέλασή τους, ακολουθεί ένα παράδειγμα. Για δύο διευθύνσεις (σε δεκαδικό σύστημα) την (a) 8200 (bytes) και την (b) 8264 (bytes) απεικονίζεται γραφικά στο Σχήμα 4.5 σε ποια τράπεζα μνήμης βρίσκεται το καθένα. Επίσης φαίνεται ότι το μονοπάτι που θα ακολουθηθεί για την προσπέλαση των στοιχείων διαφοροποιείται στο επίπεδο του MC.

(a) 8200 σε δυαδικό [...0001][000][0][000][001][000] (bytes)

(b) 8264 σε δυαδικό [...0001][000][0][001][001][000] (bytes)

Ο διαχωρισμός στα νούμερα σε δυαδική μορφή γίνεται με βάση τη διαφύλλωση της μνήμης (Σχήμα 3.6).



Σχήμα 4.5 – Κατανομή Στοιχείων στις Τράπεζες – Απόσταση Στοιχείων 64 bytes (8 λέξεις)

2. Απόσταση διευθύνσεων 512 bytes

Στις μετρήσεις αυτές οι διευθύνσεις των στοιχείων απέχουν 512 bytes, δηλαδή 64 «λέξεις». Στο Σχήμα 4.6 απεικονίζεται μια αναπαράσταση του τρόπου με τον οποίο είναι τοποθετημένα τα στοιχεία που ζητούνται στη μνήμη και η σειρά με την οποία η σχεδίαση στέλνει τις αιτήσεις.

		j											
i	1			<-	64	->			2			<-	64
	->			3			<-	64	->			4	
		<-	64	->			5			<-	64	->	
		6			<-	64	->			7	->		

Σχήμα 4.6 – Αναπαράσταση Στοιχείων στη Μνήμη με Απόσταση 512 bytes (64 λέξεις)

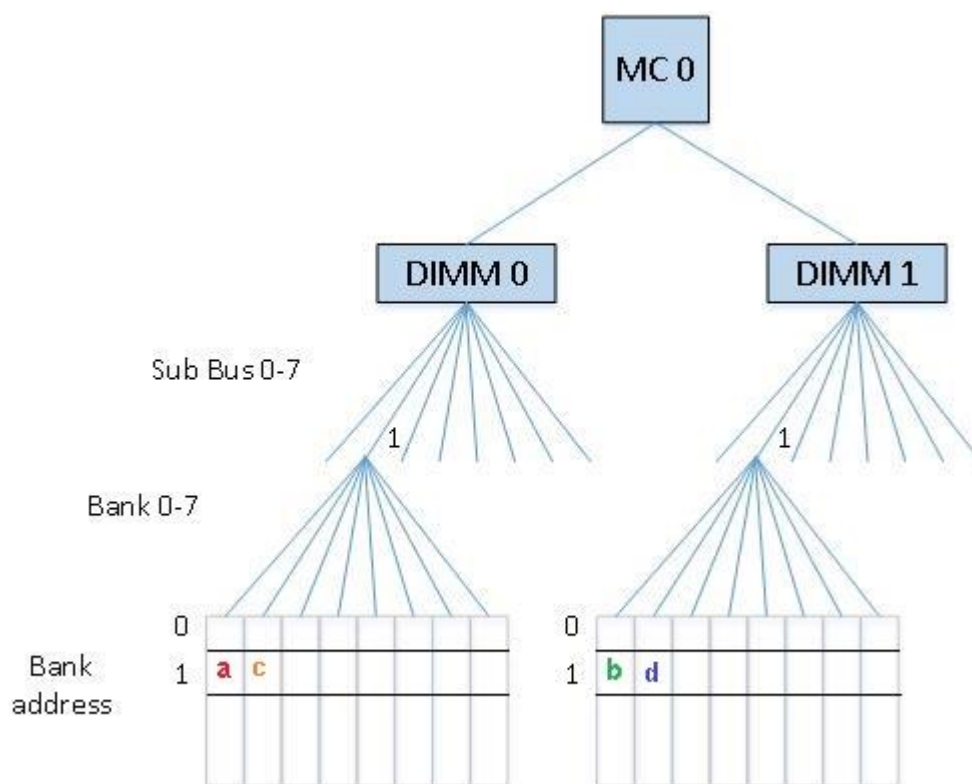
Τα στοιχεία που οι διευθύνσεις τους έχουν απόσταση 512 bytes εξυπηρετούνται από το σύστημα μνήμης από δύο διαδοχικά DIMMs. Ακολούθως δίνεται ένα παράδειγμα που περιγράφει τον τρόπο που κατανέμονται τα στοιχεία στις τράπεζες της μνήμης και το

μονοπάτι που οδηγεί την προσπέλαση του κάθε στοιχείου. Για δύο διευθύνσεις (σε δεκαδικό σύστημα) την (a) 8200 (bytes) και την (b) 8712 (bytes) απεικονίζεται γραφικά στο Σχήμα 4.7 σε ποια τράπεζα μνήμης βρίσκεται το καθένα. Επίσης φαίνεται ότι το μονοπάτι που θα ακολουθηθεί για την προσπέλαση των στοιχείων διαφοροποιείται στο επίπεδο του DIMM.

(a) 8200 σε δυαδικό [...0001][000][0][000][001][000] (bytes)

(b) 8712 σε δυαδικό [...0001][000][1][000][001][000] (bytes)

Ο διαχωρισμός στα νούμερα σε δυαδική μορφή γίνεται με βάση τη διαφύλλωση της μνήμης (Σχήμα 3.6).



Σχήμα 4.7 – Κατανομή Στοιχείων στις Τράπεζες – Απόσταση Στοιχείων 512 bytes (64 λέξεις)

3. Απόσταση διευθύνσεων 1024 bytes

Οι μετρήσεις που ακολουθούν είναι για διευθύνσεις στοιχείων που απέχουν 1024 bytes, δηλαδή 128 «λέξεις». Στο Σχήμα 4.8 απεικονίζεται ο τρόπος που είναι τοποθετημένα τα στοιχεία που ζητούνται στη μνήμη και η σειρά με την οποία η σχεδίαση στέλνει τις αιτήσεις.

		j											
i	1			<-	128	->			2			<-	128
	->			3			<-	128	->			4	
		<-	128	->			5			<-	128	->	
		6			<-	128	->			7	->		

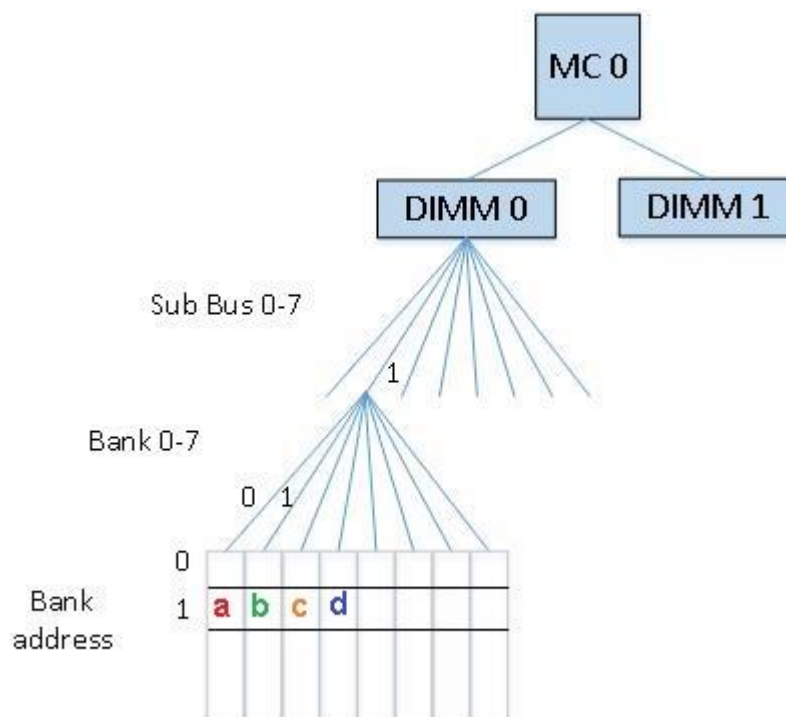
Σχήμα 4.8 – Αναπαράσταση Στοιχείων στη Μνήμη με Απόσταση 1024 bytes (128 λέξεις)

Τα στοιχεία που οι διευθύνσεις τους απέχουν 1024 bytes βρίσκονται σε διαφορετική τράπεζα μνήμης το καθένα. Αμέσως μετά παρουσιάζεται ένα παράδειγμα για τον τρόπο κατανομής των στοιχείων στις τράπεζες της μνήμης και απεικονίζεται το μονοπάτι που οδηγεί την προσπέλαση του κάθε στοιχείου. Για δύο διευθύνσεις (σε δεκαδικό σύστημα) την (a) 8200 (bytes) και την (b) 9224 (bytes) απεικονίζεται γραφικά στο Σχήμα 4.9 σε ποια τράπεζα μνήμης βρίσκεται το καθένα. Επίσης φαίνεται ότι το μονοπάτι που θα ακολουθηθεί για την προσπέλαση των στοιχείων διαφοροποιείται στο επίπεδο του bank.

(a) 8200 σε δυαδικό [...0001][000][0][000][001][000] (bytes)

(b) 9224 σε δυαδικό [...0001][001][0][000][001][000] (bytes)

Ο διαχωρισμός στα νούμερα σε δυαδική μορφή γίνεται με βάση τη διαφύλλωση της μνήμης (Σχήμα 3.6).



Σχήμα 4.9 – Κατανομή Στοιχείων στις Τράπεζες – Απόσταση Στοιχείων 1024 bytes (128 λέξεις)

4. Απόσταση διευθύνσεων 8192 bytes

Στις μετρήσεις αυτές οι διευθύνσεις των στοιχείων απέχουν 8192 bytes, δηλαδή 1024 «λέξεις». Στο Σχήμα 4.10 απεικονίζεται μια αναπαράσταση του τρόπου που είναι τοποθετημένα τα στοιχεία που ζητούνται από τη μνήμη και η σειρά με την οποία η σχεδίαση στέλνει τις αιτήσεις.

		j											
i	1			<-	1024	->			2			<-	1024
	->			3			<-	1024	->			4	
		<-	1024	->			5			<-	1024	->	
		6			<-	1024	->			7	->		

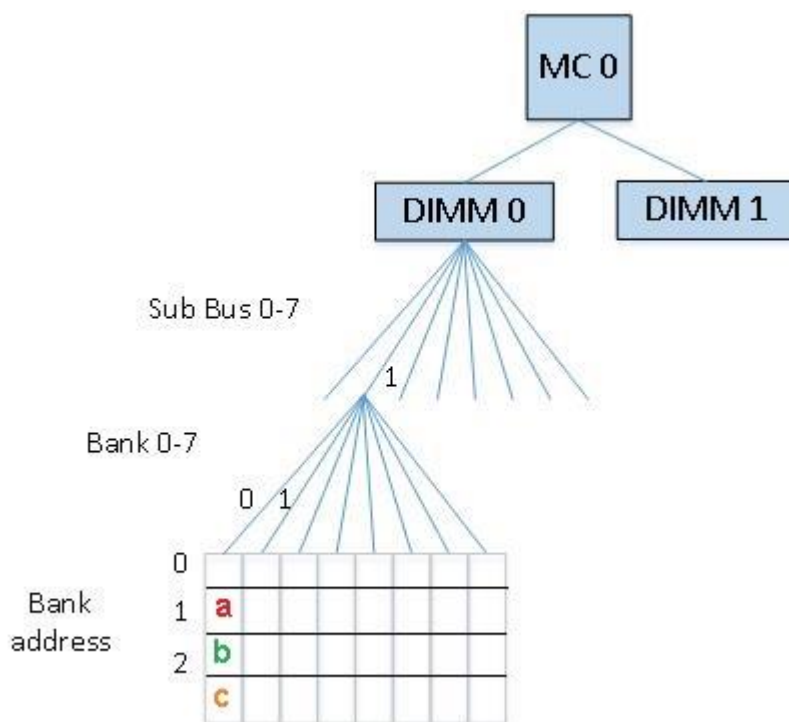
Σχήμα 4.10 – Αναπαράσταση Στοιχείων στη Μνήμη με Απόσταση 8192 bytes (1024 λέξεις)

Τα στοιχεία που οι διευθύνσεις έχουν απόσταση μεταξύ τους 8192 bytes βρίσκονται στην ίδια τράπεζα. Ακολουθώς παρουσιάζεται ένα παράδειγμα που περιγράφει τον τρόπο που κατανέμονται τα στοιχεία στην τράπεζα της μνήμης και το μονοπάτι που οδηγεί την προσπέλαση του κάθε στοιχείου. Για δύο διευθύνσεις (σε δεκαδικό σύστημα) την (a) 8200 (bytes) και την (b) 16392 (bytes) απεικονίζεται γραφικά στο Σχήμα 4.11 σε ποια τράπεζα μνήμης βρίσκονται. Επίσης φαίνεται ότι το μονοπάτι που θα ακολουθηθεί για την προσπέλαση των στοιχείων διαφοροποιείται μόνο στο πεδίο του bank address.

(a) 8200 σε δυαδικό [...0001][000][0][000][001][000] (bytes)

(b) 16392 σε δυαδικό [...0010][000][0][000][001][000] (bytes)

Ο διαχωρισμός στα νούμερα σε δυαδική μορφή γίνεται με βάση τη διαφύλλωση της μνήμης (Σχήμα 3.6).



Σχήμα 4.11 – Κατανομή Στοιχείων στις Τράπεζες – Απόσταση Στοιχείων 8192 bytes (1024 λέξεις)

4.3 Αποτύπωμα Μνήμης Αλγορίθμου UNAFold του Zuker

Ο αλγόριθμος του Zuker και συγκεκριμένα ο UNAFold (Unified Nucleic Acid Folding and hybridization package) υπολογίζει τη δευτεροταγή δομή του RNA. Ο υπολογισμός αυτός γίνεται με τη εύρεση της ελάχιστης ενέργειας που μπορεί να αναπτύξουν οι δεσμοί των βάσεων στο μακρομόριο. Από πειραματικά δεδομένα υπολογίζεται η συνεισφορά κάθε δεσμού που μπορεί να αναπτυχθεί στο μακρομόριο έτσι ώστε να εξαχθεί η βέλτιστη δομή, δηλαδή η δομή με την ελάχιστη ενέργεια. Τα πειραματικά δεδομένα προέρχονται από μετρήσεις της ενέργειας πραγματικών μακρομορίων RNA.

Τα δεδομένα είναι οργανωμένα σε μορφή πολυδιάστατων πινάκων, έτσι ώστε να είναι ευκολότερη η πρόσβαση στις τιμές τους. Κάθε διάσταση του πίνακα αντιστοιχεί στην τιμή κάποιας βάσης ή ζεύγους που ανήκει στο σχηματισμό που αναφέρεται η εκάστοτε ενέργεια, συνεπώς αν ο σχηματισμός περιέχει τέσσερις βάσεις ο αντίστοιχος πίνακας θα είναι τεσσάρων διαστάσεων. Ο υπολογισμός της βέλτιστης δομής του μακρομορίου γίνεται σταδιακά, δηλαδή πρώτα υπολογίζεται η ελάχιστη ενέργεια μικρότερων τμημάτων του μακρομορίου (υποακολουθίες) και στη συνέχεια η συνολική ελάχιστη ενέργεια, που οδηγεί στη βέλτιστη δομή.

Ο αλγόριθμος χρησιμοποιεί τρεις άνω τριγωνικούς πίνακες διαστάσεων $N \times N$, $Q_{\text{prime}}(\cdot, \cdot)$, $Q_{\text{m}}(\cdot, \cdot)$ και $Q(\cdot, \cdot)$ όπου N ο αριθμός των βάσεων του μακρομορίου του RNA.

Κάθε στοιχείο από αυτούς τους τρεις πίνακες αντιστοιχεί στην ενέργεια κάποιας δομής, δηλαδή του δεσμού ανάμεσα στις βάσεις. Ο Qprime πίνακας είναι εκείνος ο οποίος περιέχει την πραγματική βέλτιστη τιμή ενώ οι πίνακες Qm και Q περιέχουν προσωρινές τιμές κάποιου υποσυνόλου των περιπτώσεων της βέλτιστης δομής. Στη συνέχεια παρουσιάζονται οι αναδρομικές σχέσεις για τον υπολογισμό των τιμών των πινάκων.

Στον πίνακα Qprime αποθηκεύονται οι βέλτιστες τιμές για το σύνολο των υποακολουθιών, σύμφωνα με την παρακάτω εξίσωση:

$$W(i,j) = \min \begin{cases} V(i,j) \\ L(i,j) \\ b + c + d3(i+1,j,i) + V(i+1,j) \\ b + c + d5(i,j-1,j) + V(i,j-1) \\ b + W(i+1,j) \\ b + W(i,j-1) \\ 2b + c + d3(i+1,j,i) + d5(i,j-1,j) + V(i+1,j-1) \end{cases}$$

Στον πίνακα Qm αποθηκεύονται οι βέλτιστες τιμές για τις υποακολουθίες οι οποίες προκύπτουν από τη συνένωση δύο άλλων βέλτιστων υποακολουθιών σε σειρά από τον πίνακα W.

$$L(i,j) = \min_k W(i,k) + W(k+1,j), \quad \forall k \in [i+4, j-4]$$

Στον πίνακα Q αποθηκεύονται οι βέλτιστες τιμές για τις υποακολουθίες των οποίων οι δύο βάσεις στα άκρα τους σχηματίζουν ζεύγος.

$$V(i,j) = \min \begin{cases} eh(i,j) \\ es(i,j) + V(i+1,j-1) \\ ebl(i,j,k) + V(k,j-1), \quad \forall k \in [i+2, j-4] \\ ebr(i,j,k) + V(i+1,k), \quad \forall k \in [i+5, j-1] \\ ei(i,j,k,l) + V(k,l), \quad \forall k,l: i+1 < k, l < j-1, l-k > 3 \\ a + c + L(i+1,j-1) \\ a + c + L(i+2,j-1) + d5(i,j,i+1) + b \\ a + c + L(i+1,j-2) + d3(i,j,j-1) + b \\ a + c + L(i+2,j-2) + d5(i,j,i+1) + d3(i,j,j-1) + 2b \end{cases}$$

Όπου

- a: Offset
- b: Free base penalty
- c: Helix penalty

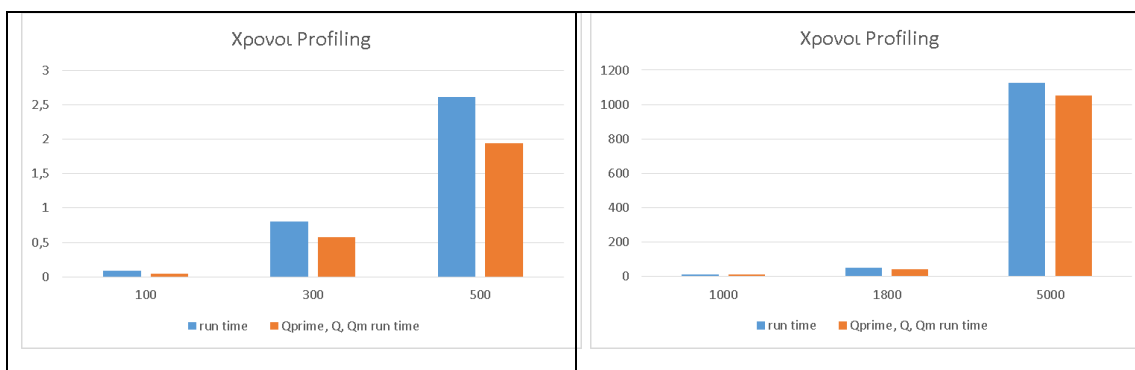
4.3.1 Profiling UNAFold

Για τον εντοπισμό των σημείων του κώδικα του αλγόριθμου UNAFold, στα οποία ο επεξεργαστής ξοδεύει τον περισσότερο χρόνο (hotspots), χρησιμοποιήθηκε ο profiler gprof, ενώ χρησιμοποιήθηκαν και διαφορετικά μήκη δειγμάτων RNA. Η αλληλεξάρτηση των τριών εξισώσεων (Qprime, Q, Qm), όπως παρουσιάζεται στον Πίνακα 4.1 και στα Διαγράμματα 4.1, φαίνεται ότι επηρεάζει σε μεγάλο ποσοστό το συνολικό χρόνο εκτέλεσης που χρειάζεται ο αλγόριθμος. Όσο μεγαλώνει το μακρομόριο του RNA, αυξάνεται και το ποσοστό του χρόνου που απαιτούν οι τρεις εξισώσεις σε σχέση με το συνολικό χρόνο.

Πίνακας 4.1 – Αποτελέσματα Profiling UNAFold

RNA (αριθμός βάσεων)	Χρόνος εκτέλεσης αλγορίθμου (sec)	Ποσοστό χρόνου εκτέλεσης Qprime, Q, Qm (%)
100	0,09	55,4
300	0,8	72,5
500	2,61	74,5
1000	12,63	78,9
1800	51,45	83,2
5000	1128,06	93,4

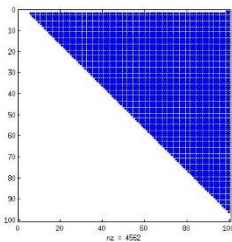
Διαγράμματα 4.1 – Αποτελέσματα Profiling UNAFold



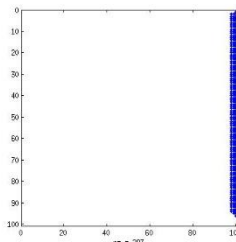
4.3.2 Απαιτήσεις Μνήμης Αλγόριθμου UNAFold - Προσομοίωση σε Matlab

Από τα αποτελέσματα του profiling διαπιστώθηκε ότι οι τρεις εξισώσεις απαιτούν το συντριπτικό ποσοστό του συνολικού χρόνου εκτέλεσης του αλγόριθμου UNAFold. Προσομοιώθηκαν σε Matlab τα στοιχεία των πινάκων που απαιτούν οι εξισώσεις Qprime, Q, Qm για τον υπολογισμό της ελάχιστης ενέργειας. Τα στοιχεία των πινάκων είναι της μορφής που φαίνεται στο Σχήμα 4.12, στο Σχήμα 4.13 και στο Σχήμα 4.14. Αποδεικνύεται ότι η

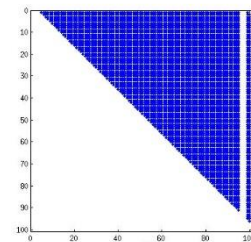
πλειονότητα των στοιχείων που χρειάζονται για τον υπολογισμό του αλγορίθμου αντλούνται από τους πίνακες Qprime και Q, που είναι άνω τριγωνικοί πίνακες.



Σχήμα 4.14 – Στοιχεία Qprime



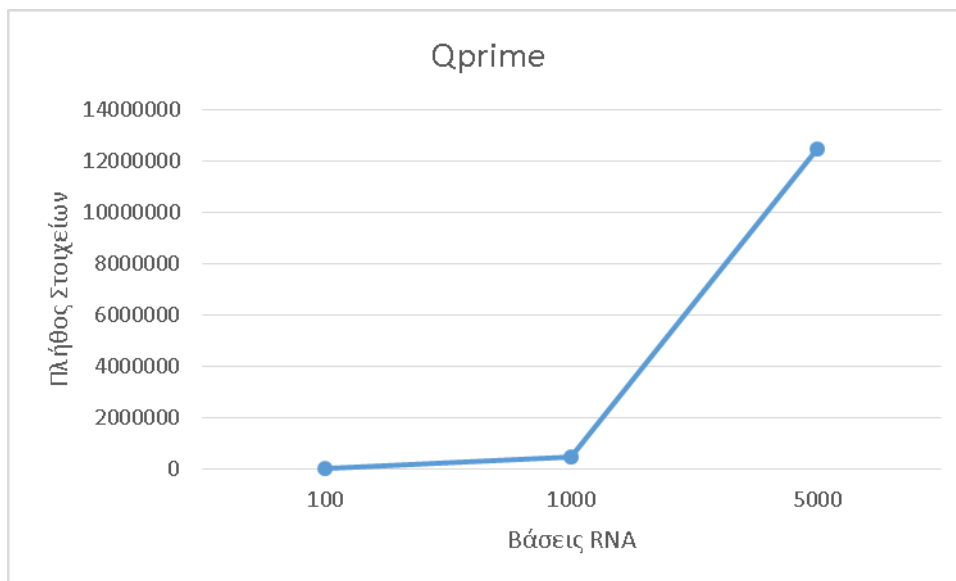
Σχήμα 4.13 – Στοιχεία Qm

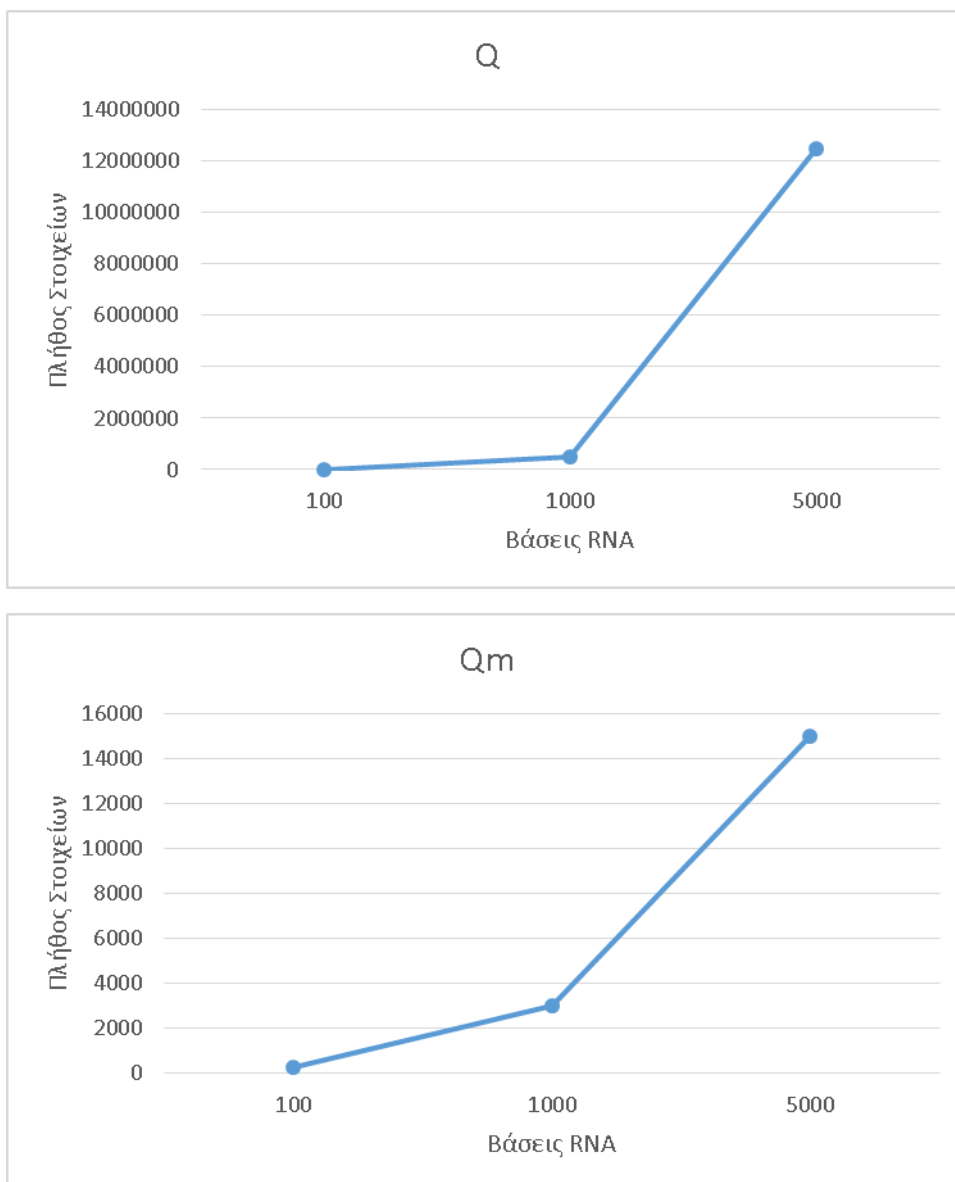


Σχήμα 4.12 – Στοιχεία Q

Στα Διαγράμματα 4.2 απεικονίζεται το πλήθος των στοιχείων, που χρειάζεται ο αλγόριθμος για τον υπολογισμό των Qprime, Q και Qm, για διαφορετικά μεγέθη μακρομορίου. Φαίνεται ότι όσο αυξάνεται το μέγεθος του RNA, ο ρυθμός αύξησης των απαιτούμενων στοιχείων, αυξάνεται εκθετικά. Τα στοιχεία αυτά βρίσκονται στα άνω τριγωνικά τμήματα αυτών των πινάκων, άρα το ζητούμενο που προκύπτει είναι πώς θα διαβαστεί αποδοτικά ένας τριγωνικός πίνακας.

Διαγράμματα 4.2 – Απαιτήσεις Qprime, Q, Qm για διάφορα μήκη RNA





4.4 Χρόνος Απόκρισης Μνήμης - Τριγωνικοί Πίνακες

Η ανάλυση του αλγόριθμου UNAFold του Zuker ήταν η αφορμή για τη μελέτη του χρόνου απόκρισης της μνήμης για άντληση δεδομένων από τριγωνικούς πίνακες. Για τον υπολογισμό του αλγορίθμου απαιτούνται πολλαπλές προσβάσεις με μεγάλους τριγωνικούς πίνακες. Στην προηγούμενη ενότητα αναλύθηκε το αποτύπωμα της μνήμης που χρειάζεται ο αλγόριθμος ώστε να βρεθεί ο αποδοτικότερος τρόπος προσπέλασης των δεδομένων που απαιτούνται για τον υπολογισμό του.

Πραγματοποιήθηκαν τρία διαφορετικά πειράματα για προσπέλαση τριγωνικών πινάκων για να διαπιστωθεί πώς ανταποκρίνεται η μνήμη στον καθένα.

- Προσπέλαση όλων των στοιχείων του πίνακα (4.4.1)

- b) Προσπέλαση των στοιχείων του κάτω τριγωνικού πίνακα (4.4.2)
c) Προσπέλαση του πίνακα με υβριδική μέθοδο (4.4.3)

Ωστόσο στα πειράματα τίθεται περιορισμός, που προέρχεται από το ίδιο το Convey HC-2, που δεν επιτρέπει δέσμευση μνήμης παραπάνω από 2GB στο συνεπεξεργαστή για κάθε πίνακα. Τα 2GB προκύπτουν ως όριο, γιατί η μνήμη του συνεπεξεργαστή έχει συνολικό μέγεθος 32GB τα οποία είναι διαμοιρασμένα στα 16 DIMMs.

4.4.1 Προσπέλαση όλων των στοιχείων του πίνακα

Στις μετρήσεις αυτές ζητείται η προσπέλαση όλων των στοιχείων του πίνακα, και χρησιμοποιούνται από τη σχεδίαση τα στοιχεία μόνο του κάτω τριγωνικού. Το διάβασμα των στοιχείων γίνεται όπως αναλύθηκε στην Ενότητα 4.2.1, δηλαδή διαβάζονται όλες οι διαδοχικές διευθύνσεις του πίνακα ανά κύκλο ρολογιού. Στο Σχήμα 4.15 αποτυπώνεται ένα παράδειγμα της μεθόδου προσπέλασης των στοιχείων ολόκληρου του πίνακα, στη μνήμη. Φαίνεται επίσης η σειρά προσπέλασης των στοιχείων στη μνήμη.

[illegible]

Σχήμα 4.15 – Μέθοδος Προσπέλασης Όλων των Στοιχείων του Πίνακα

4.4.2 Προσπέλαση των στοιχείων του κάτω τριγωνικού πίνακα

Στο πείραμα αυτό γίνεται προσπέλαση μόνο των διευθύνσεων των στοιχείων του κάτω τριγωνικού πίνακα. Αυτή η μέθοδος προσομοιώνει τον τρόπο με τον οποίο λειτουργεί το σύστημα της μνήμης του Convey HC-2. Όταν η σχεδίαση φτάνει στη διαγώνιο του πίνακα μεταβαίνει στην αρχή της επόμενης σειράς μέχρι να διαβαστεί το τελευταίο στοιχείο του πίνακα. Ένα ενδεικτικό παράδειγμα της μεθόδου παρουσιάζεται στο Σχήμα 4.16.

i	j															
	1	→	<-	63	->		2		<-	63	->		3			
	4	5	→	<-	63	->		6		<-	63	->		7		
	8	9	10	→	<-	63	->		11		<-	63	->		12	
	13	14	15	16	→											
								k-2	k-1	k	→	<-	63	->		k+1
	k+2	k+11	→				
	k+12															

Σχήμα 4.17 – Μέθοδος Προσπέλασης των Στοιχείων του Πίνακα με Υβριδική Μέθοδο

4.5 Προβλήματα στη χρήση του Convey

Το Convey αποτελεί ένα νέο μηχάνημα που συνδυάζει πολλές τεχνολογίες μεταξύ τους. Η κατανόηση του τρόπου λειτουργίας του καθιστάτε δύσκολη υπόθεση καθώς δεν υπάρχουν επαρκείς οδηγίες ούτε στα manual της Convey αλλά ούτε και από άλλους χρήστες που έχουν αντιμετωπίσει παρόμοια προβλήματα.

Κατά τη διάρκεια της παρούσας διπλωματικής το manual της Convey για το κομμάτι των FPGAs αναβαθμίστηκε δύο φορές. Κάθε καινούρια έκδοση περιείχε περισσότερες πληροφορίες κυρίως για τον τρόπο επικοινωνίας των FPGAs με τις άλλες διεπαφές που παρέχει το σύστημα. Χρειάστηκαν πολλές πειραματικές δοκιμές ώστε να κατανοηθεί ο σωστός τρόπος χρήσης των σημάτων που ενώνουν τις διεπαφές με τις FPGAs, όπως το crossbar, οι MCs και άλλες, . Κάποια συμπεράσματα που προέκυψαν από τις δοκιμές αυτές συμπεριλαμβάνονταν στις αλλαγές των αναβαθμισμένων εκδόσεων του manual.

Δύο ήταν τα βασικά χαρακτηριστικά του συστήματος που χρειάστηκαν ενδελεχή μελέτη, ο τρόπος λειτουργίας της μνήμης και ο τρόπος επικοινωνίας FPGA – μνήμης. Για τον τρόπο λειτουργίας της μνήμης μελετήθηκε η διαφύλλωση και η ιεραρχία της έτσι ώστε να κατανοηθεί η δομή της. Για τον τρόπο επικοινωνίας FPGA – μνήμης αναλύθηκε κυρίως ο τρόπος λειτουργίας των MCs. Η λειτουργία των MCs περιείχε σημεία που δεν αναλύονταν στο manual και ο σωστός τρόπος που έπρεπε να ενεργοποιούνται τα σήματα ελέγχου τους βρέθηκε ύστερα από πολλές δοκιμές.

Αναλυτικότερα θα παρουσιαστούν παρακάτω κάποια συγκεκριμένα προβλήματα που προέκυψαν και επιλύθηκαν κατά τη διάρκεια χρήσης του Convey. Τα προβλήματα κατηγοριοποιήθηκαν ανάλογα με το επίπεδο της υλοποίησης στο οποίο εμφανίστηκαν. Η υλοποίηση έγινε με τη χρήση δύο απομακρυσμένων μηχανημάτων, ένας server για το simulation (kition.mhl.tuc.gr) και το Convey HC-2ex (pella.mhl.tuc.gr)

➤ Κώδικας C:

Στη δέσμευση χώρου, το σύστημα έχει όριο τα 2 GB. Όταν όμως ο χρήστης δεσμεύει πίνακες από 3 GB και πάνω το σύστημα δεν εμφανίζει κάποιο λάθος ούτε στο compile ούτε στη δημιουργία του bit file. Όταν γινόταν η εκτέλεση της σχεδίασης στην pella έβγαζε segmentation fault χωρίς καμία περαιτέρω πληροφορία για το είδος του λάθους. Μετά από πολλές δοκιμές ανακαλύφτηκε το λάθος, το όριο του μεγέθους δέσμευσης μεταβλητών δεν αναφέρεται πουθενά στο manual. Το μέγεθος αυτό εξαρτάται από το κάθε μηχάνημα γιατί σχετίζεται με το μέγεθος της φυσικής μνήμης. Η pella που έγιναν όλα τα πειράματα διαθέτει 64 GBs μνήμη και ο μέγιστος χώρος δέσμευσης για έναν πίνακα είναι 2 GB.

➤ Κώδικας Assembly:

Στην assembly αντιμετωπίστηκε ένα λάθος με τη σύνταξη των εντολών, που στο tutorial ήταν δηλωμένες λανθασμένα. Η assembly που χρησιμοποιείται από το σύστημα του Convey έχει ανάποδα ορισμένους τους καταχωρητές προορισμού σε σχέση με τις εντολές της assembly για mips, κάτι το οποίο δεν εμφανιζόταν ως λάθος στο compile. Το λάθος εντοπίστηκε στο επίπεδο του simulation όταν τα ορίσματα από τη C στη Verilog (Επεξεργαστής – Συνεπεξεργαστής) είχαν λανθασμένες τιμές.

Παράδειγμα :

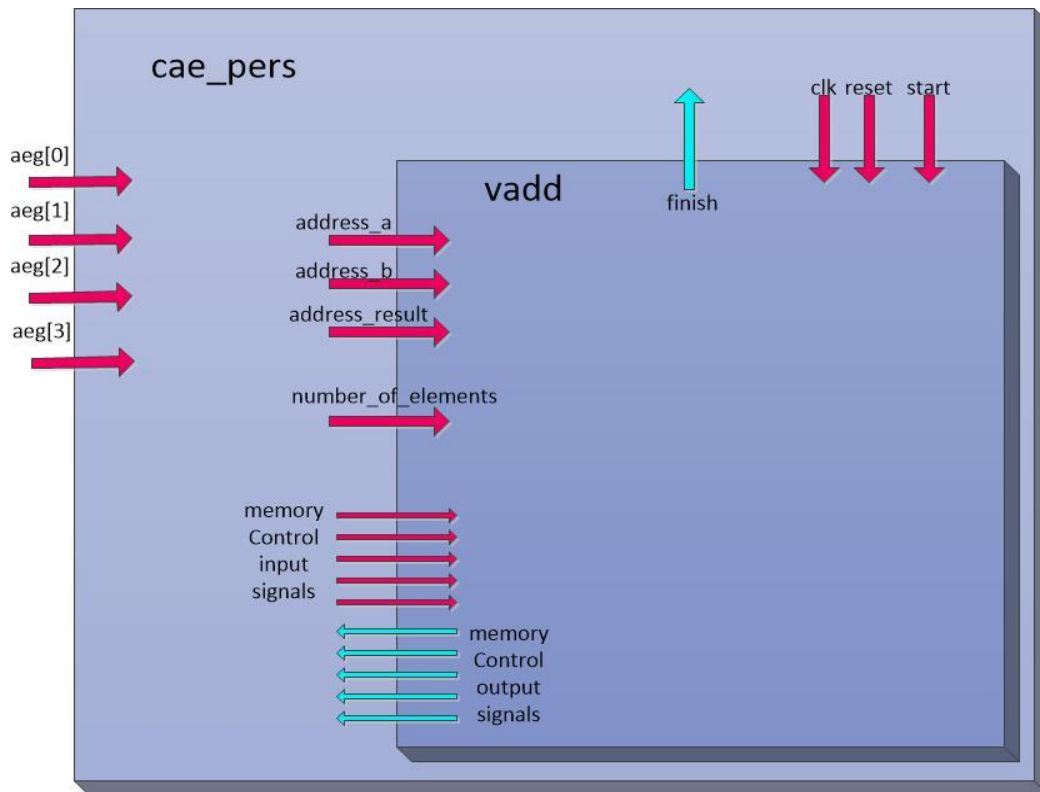
Mips: add \$d, \$s, \$t

Convey: add \$t, \$s, \$d

Δεύτερο πρόβλημα με την assembly διαπιστώθηκε κατά τη διάρκεια των μετρήσεων. Υπήρχε η λανθασμένη εντύπωση ότι για τις μετρήσεις των πειραμάτων η σχεδίαση φορτωνόταν και έτρεχε μόνο στη μία FPGA, αλλά μετά από κάποιες τελευταίες μετρήσεις διαπιστώθηκε ότι φορτώνονταν με την ίδια σχεδίαση όλες οι FPGAs (και οι τέσσερις) με τα ίδια δεδομένα και σε όλα τα πειράματα είχαν καταγραφεί και μελετηθεί λανθασμένες τιμές. Ενώ ο στόχος ήταν να μετρηθεί η ταχύτητα απόκρισης της μνήμης με τη χρήση μίας FPGA με τον λάθος κώδικα assembly και οι τέσσερις FPGAs ζητούσαν ταυτόχρονα να διαβάζουν τα ίδια δεδομένα, κάτι το οποίο πρόσθετε μεγάλη καθυστέρηση στα αποτελέσματα των μετρήσεων. Όταν εντοπίστηκε το λάθος επαναλήφθηκαν όλα τα πειράματα με το σωστό κώδικα assembly και οι τιμές των μετρήσεων ήταν αισθητά μειωμένες από τις αρχικές.

➤ Κώδικας Verilog – Vhdl:

Η βασική αρχιτεκτονική του συστήματος παρουσιάζεται στο Σχήμα 4.18.



Σχήμα 4.18 – Βασική Αρχιτεκτονική Σχεδιάσεων

Cae_pers

Το Cae_pers αποτελεί το top level της κάθε σχεδίασης στο Convey, στο αρχείο αυτό γίνονται port map όλα τα σήματα για τις διαθέσιμες διεπαφές που μπορεί να χρησιμοποιήσουν οι FPGAs για να επικοινωνήσουν με το υπόλοιπο σύστημα του Convey. Από αυτό το αρχείο πρέπει να επιλεγτούν μόνο τα απαραίτητα σήματα που θα χρειαστεί η κάθε σχεδίαση και τα υπόλοιπα να μηδενιστούν. Επίσης σε αυτό το αρχείο γίνονται port map όλα τα σήματα ελέγχου των 16 διαύλων επικοινωνίας με τη μνήμη. Η παράληψη αρχικοποίησης ή η λανθασμένη σύνδεση ορισμένων σημάτων δεν εμφανίζει λάθη κατά το compile, αλλά στο simulation φαινόταν η σχεδίαση να μην λειτουργεί σωστά.

Επίσης ο καταχωρητής «4» διαπιστώθηκε ότι είναι δεσμευμένος από το σύστημα της Convey, όταν χρησιμοποιείται από η σχεδίαση η διεπαφή του crossbar. Η ανάθεση αυτή δεν αναφέρεται σε κανένα manual, ούτε προκύπτει κάποιο σφάλμα στο compile, αλλά εμφανίζονταν λανθασμένες τιμές στους καταχωρητές στο simulation που δεν μπορούσαν να εξηγηθούν.

Vadd

Το Vadd περιέχει την εκάστοτε σχεδίαση σε Verilog ή VHDL και από αυτό το module γίνεται κυρίως η διαχείριση των σημάτων για τον έλεγχο των διαύλων που χρησιμοποιεί η FPGA για να στέλνει αιτήματα και να λαμβάνει αποκρίσεις προς και

από τη μνήμη. Αρχικά δοκιμάστηκαν όλα τα σήματα ελέγχου των διαύλων έτσι ώστε να φανεί η χρησιμότητα του καθενός και ο σωστός τρόπος διαχείρισης του, για τις αναγνώσεις και τις εγγραφές που θα χρειάζονταν τα πειράματα της παρούσας διπλωματικής.

Η κατανόηση του τρόπου διαχείρισης των σημάτων αυτών ήταν μια διαδικασία χρονοβόρα που κατέληξε να δημιουργηθεί ένα abstract αρχείο vadd το οποίο διέθετε μόνο την απαραίτητη πληροφορία για τη σωστή διαχείριση των διαύλων της μνήμης και την αποθήκευση των χρήσιμων δεδομένων. Το abstract αρχείο αυτό όπως και το cae pers χρησιμοποιήθηκαν από άλλες διπλωματικές εργασίες ως βάση για την επικοινωνία των σχεδιάσεων με τη μνήμη του Convey.

Τα σημαντικότερα προβλήματα σε αυτό το επίπεδο δεν εμφανίζονταν κατά το compile ούτε κατά τη δημιουργία του bit file ούτε κατά την εκτέλεση.

1. Η σωστή αρχικοποίηση όλων των σημάτων

Η μη αρχικοποίηση ακόμα και ενός μετρητή δημιουργούσε σοβαρό πρόβλημα κατά τη διάρκεια εκτέλεσης του bit file στο Convey. Τις περισσότερες φορές είχε ως αποτέλεσμα το σύστημα να βγαίνει εκτός δικτύου και να χρειάζεται να εισέλθει ο διαχειριστής στο data center του noc για να γίνει reboot η pella, μια διαδικασία που ήταν πολύ χρονοβόρα.

2. Η υπέρβαση των ορίων των δεσμευμένων πινάκων της μνήμης

Ο έλεγχος των ορίων των πινάκων που έχουν δεσμευτεί από τον κώδικα της C είναι απαραίτητος στη σχεδίαση και υπέρβαση των ορίων προκαλεί «κρέμασμα» του συστήματος του Convey και στη συνέχεια χρειάζεται reboot από τον διαχειριστή ή από το data center.

3. Ο σωστός τερματισμός της λειτουργίας των FPGAs

Από τα πρώτα προβλήματα που εντοπίστηκαν ήταν ότι δεν γινόταν σωστός τερματισμός της λειτουργίας των FPGAs κατά τη διάρκεια επιστροφής του προγράμματος στον επεξεργαστή για την ολοκλήρωση της εκτέλεσης. Αποτέλεσμα αυτής της δυσλειτουργίας ήταν να επιστρέφει το πρόγραμμα στον επεξεργαστή και ο συνεπεξεργαστής να συνεχίζει να βρίσκεται σε λειτουργία, κάτι που αντίκειται στην ορθή λειτουργία του Convey. Γι' αυτό το λόγο το σύστημα δεν ολοκλήρωνε την εκτέλεση του προγράμματος και έβγαине εκτός δικτύου και χρειαζόταν ο διαχειριστής να κάνει reboot από το data center.

➤ Bit file

Η δημιουργία ενός bit file είναι μια πολύ χρονοβόρα διαδικασία αφού απαιτείται τουλάχιστον ένα χρονικό διάστημα δύο ωρών για την ολοκλήρωσή του. Επίσης

διαπιστώθηκε ότι η δημιουργία ενός bit file γεμίζει έναν ειδικό φάκελο του simulation server (kition) και από τη στιγμή που γεμίζει αυτός ο φάκελος κάθε νέα λειτουργία που εκτελείται στο server, είτε compile είτε simulation είτε bit file, προκαλεί σφάλμα στο σύστημα, και χρειάζεται άδειασμα του συγκεκριμένου φακέλου από τον διαχειριστή. Ο ρυθμός εμφάνισης του προβλήματος ήταν κάθε 3-4 μέρες, αν το Convey χρησιμοποιούνταν από 2-3 άτομα και οι εργασίες που έτρεχαν τη στιγμή που γέμιζε ήταν άχρηστες και έπρεπε να δρομολογηθούν ξανά. Εναλλακτικά βρέθηκε ένας τρόπος μείωσης του ρυθμού εμφάνισης του προβλήματος διαγράφοντας κάθε χρήστης κάποια συγκεκριμένα αρχεία που του επέτρεπε το σύστημα της Convey μετά την ολοκλήρωση ενός bit file.

Πρόβλημα με το makefile του συστήματος του Convey το οποίο ήταν αρχικοποιημένο για μηχανήμα HC-1 ενώ το μηχανήμα στο οποίο εκτελούνταν το bit file ήταν HC-2ex. Αποτέλεσμα αυτού του λάθους ήταν να μην εκτελείται σωστά η σχεδίαση, αλλά χωρίς να «κρεμάει» το μηχανήμα.

➤ Run:

Οι αρχικές οδηγίες που υπήρχαν στο manual για την εκτέλεση μίας σχεδίασης στο Convey ήταν να «καθαρίζονται» οι FPGAs από προηγούμενα personalities με την εντολή «opt/convey/sbin/mpcache -f» και στη συνέχεια να εκτελείται η εντολή run. Ο τρόπος αυτός εκτελούσε τις σχεδιάσεις σωστά αλλά πρόσθετε στα αποτελέσματα μεγάλες καθυστερήσεις. Στη συνέχεια όταν δοκιμάστηκαν και κάποιες επιπρόσθετες εντολές από ένα άλλο manual «opt/convey/sbin/mpcache -a -S και -L -S» που φόρτωναν στις FPGAs τη σχεδίαση τότε οι μετρήσεις ήταν πιο ακριβείς. Γι' αυτό το λόγο επανεκτελέστηκαν όλα τα πειράματα από την αρχή.

➤ Προβλήματα σύνδεσης :

Μέσω VPN σύνδεσης προέκυπταν χαμηλές ταχύτητες κυρίως στο server του simulation κάτι που καθιστούσε τον έλεγχο των σχεδιάσεων χρονοβόρα διαδικασία καθώς η εμφάνιση του γραφικού περιβάλλοντος του kition ήταν πολύ αργή, όπως και η αντιγραφή αρχείων από το kition στην pella. Το πρόβλημα λύθηκε όταν χρησιμοποιήθηκε ενδιάμεσος server (μηχανογραφικού κέντρου) με ip του πολυτεχνείου μέσω του οποίου γινόταν η σύνδεση στο kition και στην pella.

5

Αποτελέσματα

5.1 Αποτελέσματα μετρήσεων χρόνου απόκρισης μνήμης για μεμονωμένες προσπελάσεις

Αρχικά τα πρώτα πειράματα πραγματοποιήθηκαν για να μελετηθεί ο χρόνος απόκρισης της μνήμης για μεμονωμένες προσπελάσεις στοιχείων. Χρησιμοποιώντας και εναλλάσσοντας όλες τις δομές της μνήμης (sub bus, DIMM, bank, bank address) ζητείται ένα στοιχείο από κάθε MC και υπολογίζεται ο χρόνος, σε κύκλους ρολογιού, που χρειάζεται για να γίνει διαθέσιμο το στοιχείο στην FPGA. Τα αποτελέσματα των πειραμάτων παρουσιάζονται στον Πίνακα 5.1.

Πίνακας 5.1 – Μετρήσεις Προσπέλασης Μεμονωμένων Στοιχείων

Μέσος Χρόνος Προσπέλασης Μεμονωμένων Στοιχείων (κύκλοι ρολογιού)				
Απόσταση διευθύνσεων (bytes)	64 (MC)	8 (sub bus)	1024 (bank)	8192 (bank addr)
1 MC	115,4	-	-	-
4 MCs	119,8	118,4	122,3	157,9
8 MCs	122,3	122,3	131,7	203,6
16 MCs	122,4	124,6	153,1	303,1

Στα αποτελέσματα του Πίνακα 5.1 φαίνεται ο μέσος όρος 1.000 μεμονωμένων αιτημάτων προσπέλασης της μνήμης, μετρημένα σε κύκλους ρολογιού. Όλα τα πειράματα

έγιναν με τη χρήση μίας FPGA. Η πρώτη στήλη δείχνει τον αριθμό των memory controllers (MCs) που χρησιμοποιεί η σχεδίαση του κάθε πειράματος. Η απόσταση των διευθύνσεων των στοιχείων είναι ενδεικτική για τις μετρήσεις με 1 MC. Το μέγεθος αυτό υποδηλώνει τη διαφορά σε bytes (ή θέσεις μνήμης) των διευθύνσεων των στοιχείων που ζητούνται από τη μνήμη να προσπελαστούν.

Από τον Πίνακα 5.1 και συγκεκριμένα από τα αποτελέσματα των μετρήσεων για 1 MC, προκύπτει ότι ο χρόνος προσπέλασης ενός στοιχείου είναι 115-116 κύκλοι ρολογιού. Ο χρόνος αυτός αντιστοιχεί στο χρόνο που χρειάζεται ένας DIMM για να διανύσει μία διαδρομή μέσω της δομής της μνήμης και να προσπελάσει το στοιχείο που του ζητήθηκε. Επίσης από τον Πίνακα 5.1 και κυρίως από τα αποτελέσματα των μετρήσεων για τα πειράματα με τη χρήση 16 MCs, προκύπτει ότι όσο αυξάνεται ο αριθμός των MCs που χρησιμοποιεί η σχεδίαση αυξάνεται σταδιακά και ο χρόνος απόκρισης της μνήμης, συναρτήσει πάντα της θέσης των στοιχείων που ζητούνται.

Για τα πειράματα που έγιναν με τη χρήση 4, 8 και 16 MCs το Convey HC-2 χρησιμοποιεί διαφορετικό αριθμό πραγματικών διαύλων μνήμης-FPGA (DIMMs) για να εξυπηρετήσει τα αιτήματα που στέλνει η σχεδίαση, ανάλογα με τις θέσεις των στοιχείων στη μνήμη. Τα στοιχεία των οποίων οι διευθύνσεις διαφέρουν κατά 8 bytes (διαδοχικές θέσεις μνήμης), εξυπηρετούνται από διαφορετικό sub bus, αλλά όχι πάντα από διαφορετικό DIMM. Παραδείγματος χάρη, στις μετρήσεις με χρήση 16 MCs και απόσταση διευθύνσεων στοιχείων 8 bytes, το Convey HC-2 χρησιμοποιεί για την προσπέλασή τους, 2 διαφορετικά DIMMs, παρόλο που η σχεδίαση στέλνει αιτήματα σε 16 MCs. Το γεγονός αυτό συμβαίνει γιατί όπως έχει αναφερθεί στην εισαγωγή του Κεφαλαίου 4, όταν στις σχεδιάσεις χρησιμοποιείται η επιλογή του crossbar (Ενότητα 0) οι FPGAs δεν έχουν άμεση επαφή με τους πραγματικούς memory controllers, αλλά με αντίστοιχο αριθμό εικονικών.

Συνεχίζοντας την ανάλυση των μετρήσεων τα στοιχεία των οποίων οι διευθύνσεις έχουν διαφορά 64 bytes βρίσκονται σε διαφορετικό DIMM. Για παράδειγμα, συγκρίνοντας τα πειράματα με 8 και 16 MCs για στοιχεία με διαφορές στις διευθύνσεις τους κατά 64 bytes διαπιστώνεται ότι στην πρώτη περίπτωση η σχεδίαση χρησιμοποιεί 8 MCs και στη δεύτερη περίπτωση 16 MCs. Το σύστημα της μνήμης για να προσπελάσει τα στοιχεία που ζητούνται θα χρησιμοποιήσει 8 DIMMs και 16 DIMMs, αντίστοιχα.

Τα στοιχεία με διαφορές της τάξης των 1024 bytes βρίσκονται σε διαφορετική τράπεζα και τα στοιχεία με διαφορά διευθύνσεων τα 8192 bytes βρίσκονται στην ίδια τράπεζα. Η προσπέλαση των στοιχείων και στους δύο αυτούς τύπους πειραμάτων θα πραγματοποιείται πάντα από το ίδιο DIMM.

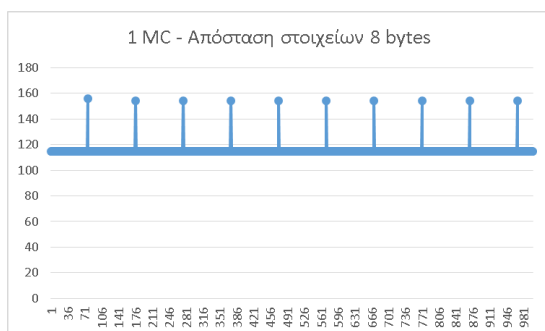
Ένα πρώτο συμπέρασμα είναι ότι οι διαφορές που προκύπτουν στους χρόνους εξυπηρέτησης των αιτημάτων επηρεάζονται τόσο από το σύνολο των DIMMs που

χρησιμοποιεί το σύστημα μνήμης για την προσπέλαση των στοιχείων όσο και από το πλήθος και τη διαφορά των διευθύνσεων των στοιχείων που ζητούνται την ίδια ή κοντινή χρονική στιγμή, ανεξάρτητα από τον αριθμό των MCs που χρησιμοποιεί η σχεδίαση.

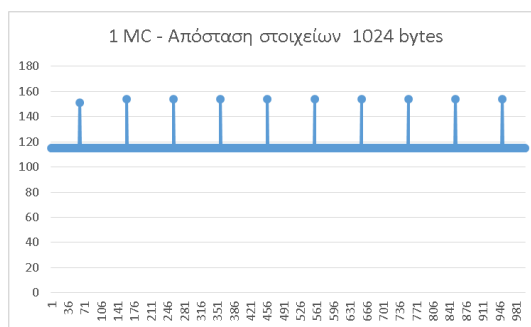
Για παράδειγμα, όταν χρησιμοποιεί η σχεδίαση 16 MCs αλλά τα 16 στοιχεία που ζητούνται βρίσκονται στην ίδια τράπεζα (διαφορά διευθύνσεων 8192 bytes) θα εξυπηρετηθούν όλες οι αιτήσεις από το ίδιο DIMM και απαιτούνται κατά μέσο όρο 300 κύκλοι ρολογιού. Ενώ όταν η σχεδίαση ζητάει από 16 MCs στοιχεία (διαφορά διευθύνσεων 64 bytes) που εξυπηρετούνται από 16 διαφορετικά DIMMs ο αριθμός των κύκλων του ρολογιού μειώνεται αισθητά στους 120 κατά μέσο όρο. Επίσης όταν ζητούνται από τους 16 MCs στοιχεία με διαφορές στις διευθύνσεις 1024 bytes που και αυτά εξυπηρετούνται από 1 DIMM ο χρόνος που χρειάζεται το σύστημα για να τα διαβάσει είναι 130 κύκλοι ρολογιού, πολύ λιγότεροι από τους 300 κύκλους. Συνεπώς φαίνεται ότι και η απόσταση των διευθύνσεων των στοιχείων επηρεάζει την απόδοση ανεξάρτητα από το πόσα DIMMs χρησιμοποιεί το σύστημα της μνήμης.

Από την ανάλυση των αποτελεσμάτων των 1.000 επαναλήψεων που πραγματοποιήθηκαν παρατηρήθηκαν δύο ενδιαφέροντα μοτίβα. Στα Διαγράμματα 5.1g και 5.1h παραλείπεται η πρώτη επανάληψη του πειράματος γιατί εμφανίζεται ένα δεύτερο ενδιαφέρον μοτίβο στις πρώτες επαναλήψεις που θα αναλυθεί στα Διαγράμματα 5.2. Το πρώτο που παρατηρήθηκε είναι ότι στις 1.000 επαναλήψεις των πειραμάτων οι μετρήσεις εμφάνισαν ένα επαναλαμβανόμενο μοτίβο υψηλών τιμών (Διαγράμματα 5.1). Ο ρυθμός με τον οποίο επαναλαμβάνεται αυτό το μοτίβο δεν είναι πάντα σταθερός, όπως επίσης και οι τιμές του διαφέρουν. Η ένταση του μοτίβο εξαρτάται από τον αριθμό των MCs που χρησιμοποιεί η σχεδίαση και τον αριθμό των DIMMs που χρησιμοποιεί το σύστημα της μνήμης για την προσπέλαση των στοιχείων. Τα αίτια που προκαλούν την εμφάνιση του μοτίβο των υψηλών τιμών δεν είναι σαφή. Εικάζεται πως οι καθυστερήσεις (stalls) που προκαλεί το σύστημα της μνήμης και οι καθυστερήσεις που προκαλούνται από συγκρούσεις αιτημάτων προσπέλασης από διαφορετικούς MCs αποτελούν αίτια της εμφάνισης του μοτίβο υψηλών τιμών. Μερικά ενδεικτικά παραδείγματα παρουσιάζονται στα Διαγράμματα 5.1. Φαίνεται από τα διαγράμματα ότι όσους λιγότερους MCs χρησιμοποιεί η σχεδίαση, το μοτίβο των υψηλών τιμών έχει μικρότερο ρυθμό εμφάνισης και πιο σταθερές τιμές. Όσο αυξάνεται ο αριθμός των MCs που ενεργοποιεί η σχεδίαση τόσο πυκνώνει η εμφάνιση υψηλών τιμών, εκτός ίσως από το Διαγράμμα 5.1f που το μοτίβο παρουσιάζει πιο αργό ρυθμό εμφάνισης και με σταθερές αλλά υψηλές τιμές.

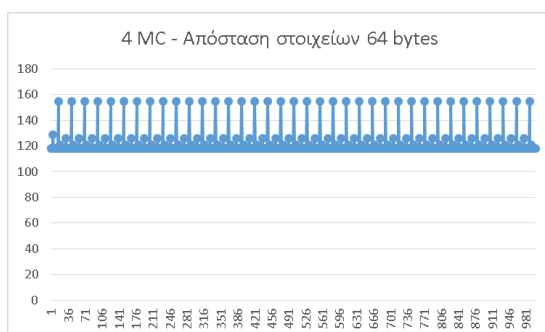
Διαγράμματα 5.1 – Ενδεικτικά Παραδείγματα Εμφάνισης Μοτίβου Υψηλών Τιμών



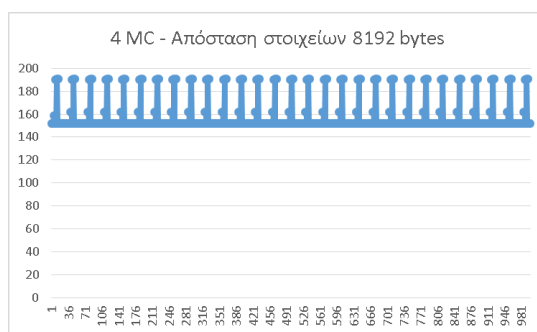
(a)



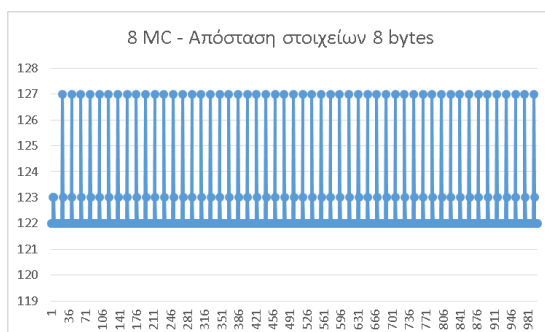
(b)



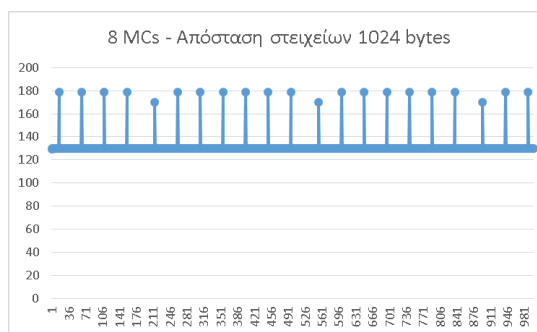
(c)



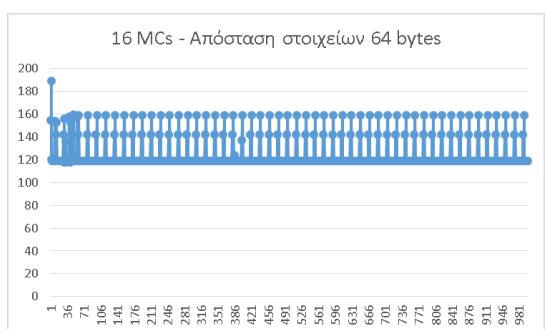
(d)



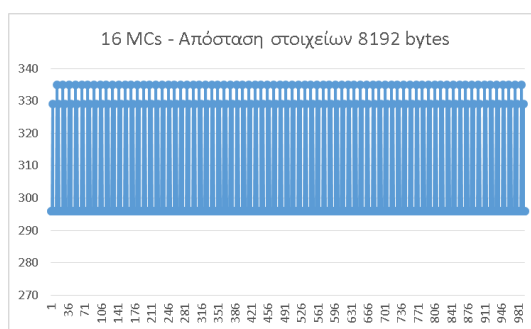
(e)



(f)



(g)

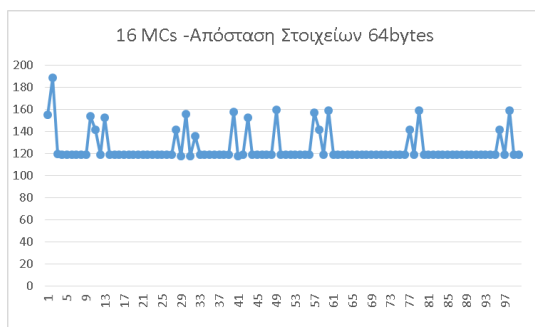


(h)

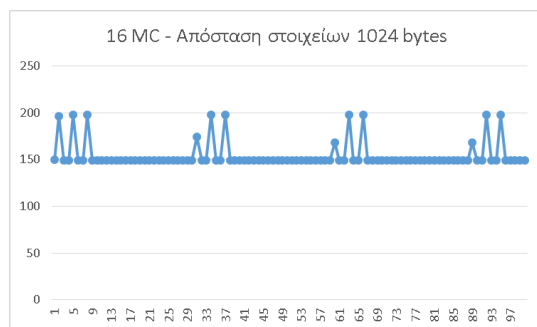
Το δεύτερο ενδιαφέρον μοτίβο που παρατηρήθηκε εμφανίζεται κυρίως στις μετρήσεις των πειραμάτων με τη χρήση 16 MCs και μόνο στις πρώτες επαναλήψεις των πειραμάτων

αυτών. Παρατηρείται πρώτον από τα Διαγράμματα 5.2, ότι όσο μεγαλώνει η απόσταση των διευθύνσεων των στοιχείων που αιτεί η σχεδίαση (Διαγράμματα 5.2e, 5.2f) τόσο περισσότερο χρόνο χρειάζεται το σύστημα της μνήμης για να εξυπηρετήσει τις πρώτες αιτήσεις της σχεδίασης. Δεύτερον, φαίνεται από τα Διαγράμματα 5.2, ότι μετά από τις πρώτες μετρήσεις του ίδιου τύπου πειράματος, το σύστημα παρουσιάζει κάποιου είδους προσαρμοστικότητα στον τρόπο προσπέλασης των στοιχείων και ο χρόνος εξυπηρέτησης των αιτήσεων της σχεδίασης σταθεροποιείται. Μία αιτία εμφάνισης του μοτίβο εκτιμάται πως προέρχεται από τον τρόπο που φορτώνεται το bit file στις FPGAs του Convey HC-2 και από τις αρχικές τιμές που έχει το σύστημα των DIMMs.

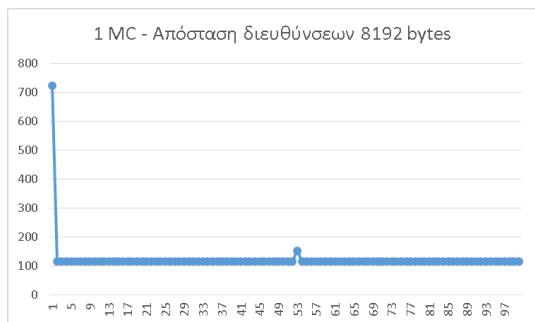
Διαγράμματα 5.2 – Ενδεικτικά Παραδείγματα Μετρήσεων (100 πρώτες επαναλήψεις των πειραμάτων)



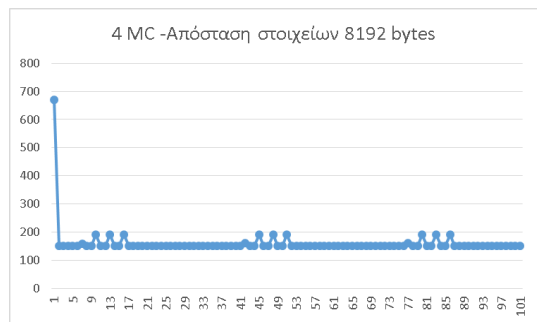
(a)



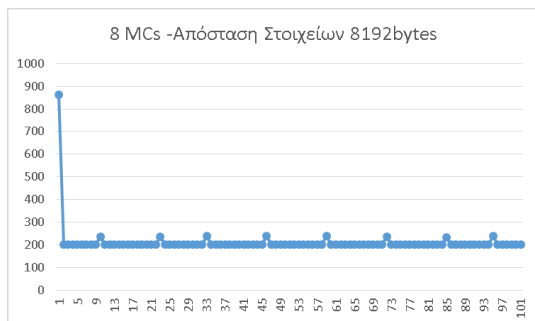
(b)



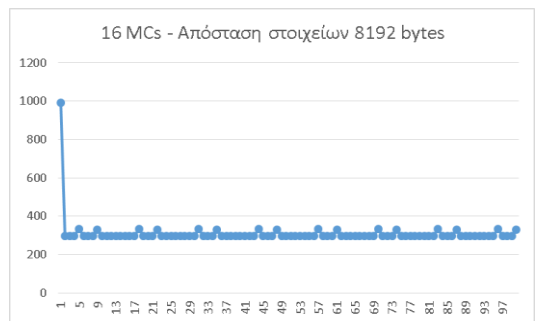
(c)



(d)



(e)



(f)

5.2 Αποτελέσματα μετρήσεων χρόνου απόκρισης μνήμης για μαζικές προσπελάσεις (burst data)

Μετά τα πειράματα για τη μέτρηση του χρόνου απόκρισης της μνήμης για μεμονωμένα στοιχεία ακολούθησαν πειράματα για τη μέτρηση του χρόνου απόκρισης της μνήμης για μαζικές διαδοχικές προσπελάσεις στη μνήμη.

Σ' αυτόν τον τύπο μετρήσεων η σχεδίαση στέλνει μεγάλους αριθμούς αιτήσεων (κυρίως 1.000 αιτήσεις) προσπέλασης διαφορετικών στοιχείων από τη μνήμη, μία αίτηση ανά κύκλο ρολογιού. Στη συνέχεια υπολογίζεται συνολικά ο χρόνος που χρειάζεται η μνήμη να εξυπηρετήσει όλα τα διαφορετικά αιτήματα. Σε όλα τα πειράματα που ακολουθούν έγινε χρήση μίας FPGA.

Οι ομάδες πειραμάτων σε αυτή την ενότητα χωρίζονται ανάλογα με την απόσταση των διευθύνσεων των στοιχείων που ζητούνται από τη μνήμη. Η FPGA αιτεί στοιχεία από τη μνήμη για προσπέλαση με δύο τρόπους. Στον πρώτο τρόπο η σχεδίαση αιτεί στοιχεία που βρίσκονται σε διαδοχικές θέσεις στη μνήμη και στο δεύτερο τρόπο για στοιχεία σε μη διαδοχικές θέσεις με αποτέλεσμα το σύστημα της μνήμης να κάνει «άλματα» (strides).

Αξίζει να σημειωθεί ότι το Convey HC-2 δεν επιτρέπει δέσμευση χώρου στη μνήμη του συνεπεξεργαστή μεγαλύτερο από 2GB. Τα 2GB είναι ο συνολικός χώρος σε φυσικές διευθύνσεις μνήμης που έχει διαμοιραστεί στις τράπεζες του κάθε DIMM στο σύστημα της μνήμης.

5.2.1 Αποτελέσματα μετρήσεων προσπέλασης στοιχείων σε διαδοχικές θέσεις μνήμης

Τα πρώτα πειράματα έγιναν για στοιχεία σε διαδοχικές θέσεις στη μνήμη, δηλαδή κάθε στοιχείο έχει διαφορά στη διεύθυνση με το επόμενο κατά 8 bytes (1 λέξη). Στις μετρήσεις αυτές έγινε χρήση 1 MC από τη σχεδίαση της FPGA και σε κάθε μέτρηση άλλαζε ο όγκος των στοιχείων που ζητούσε η σχεδίαση. Οι τρεις τύποι μετρήσεων που υλοποιήθηκαν ήταν για προσπέλαση 1.000, 10.000 και 100.000 στοιχείων. Τα πειράματα επαναλήφθηκαν 1.000 φορές για διαφορετικά στοιχεία και στον Πίνακα 5.2 παρουσιάζεται ο μέσος όρος του χρόνου απόκρισης του συστήματος μνήμης σε κύκλους ρολογιού.

Πίνακας 5.2 – Αποτελέσματα Μετρήσεων Προσπέλασης Συνεχόμενων Στοιχείων (1 MC)

Προσπέλαση Στοιχείων - Διαδοχικές Θέσεις Μνήμης			
Δεδομένα	8KB (1.000 λέξεις)	80KB (10.000 λέξεις)	800KB (100.000 λέξεις)
Κύκλοι Ρολογιού	1.185	10.233	100.204

Από τα αποτελέσματα των πειραμάτων του Πίνακα 5.2 προκύπτει ότι όταν ζητούνται στοιχεία από τη μνήμη με μαζικό (burst) τρόπο, το σύστημα της μνήμης τα εξυπηρετεί με μεγαλύτερη ταχύτητα σε σύγκριση με τα αποτελέσματα των πειραμάτων για μεμονωμένα στοιχεία (Ενότητα 5.1). Από τα αποτελέσματα των μετρήσεων γίνεται εμφανές ότι ανεξάρτητα από τον όγκο των δεδομένων που ζητάει η σχεδίαση, η ταχύτητα με την οποία εξυπηρετούνται οι αιτήσεις είναι ανάλογη των αιτημάτων με μία καθυστέρηση της τάξης των 100-300 κύκλων ρολογιού.

$$\frac{\text{ζητούμενα στοιχεία}}{\text{κύκλοι ρολογιού}} \cong 1$$

Τα αποτελέσματα του Πίνακα 5.2 παρείχαν κάποιες πρώτες ενδείξεις για τον τρόπο λειτουργίας του συστήματος μνήμης του Convey HC-2 και έδωσαν το έναυσμα για περαιτέρω μελέτη του τρόπου εξυπηρέτησης μαζικών αιτημάτων από τη μνήμη.

Στη συνέχεια, στην επόμενη ομάδα πειραμάτων έγινε χρήση περισσότερων MCs, αναφορικά 4, 8 και 16 MCs, έτσι ώστε να στέλνονται περισσότερες αιτήσεις προσπέλασης στη μνήμη ταυτόχρονα. Κάθε MC, σε αυτόν τον τύπο πειραμάτων, στέλνει 1.000 αιτήματα προσπέλασης διαδοχικών στοιχείων στη μνήμη. Με αυτόν τον τρόπο σε κάθε πείραμα αυξάνεται ο αριθμός των στοιχείων που πρέπει να εξυπηρετήσει ταυτόχρονα το σύστημα μνήμης.

Στον Πίνακα 5.3 παρουσιάζονται τα αποτελέσματα των μετρήσεων για σχεδιάσεις που χρησιμοποιούν 1, 4, 8 και 16 MCs και στέλνουν 1.000 αιτήματα προσπέλασης στοιχείων της μνήμης από κάθε MC. Ο αριθμός επαναλήψεων των πειραμάτων είναι 1.000 φορές για διαφορετικά στοιχεία και στον Πίνακα 5.3 παρουσιάζεται ο μέσος όρος του χρόνου απόκρισης του συστήματος μνήμης σε κύκλους ρολογιού.

Πίνακας 5.3 – Αποτελέσματα Μετρήσεων Προσπέλασης 1.000 Στοιχείων (1, 4, 8, και 16 MC)

1.000 Στοιχεία/MC				
MC	1	4	8	16
Συνολικά Στοιχεία	1.000	4.000	8.000	16.000
Κύκλοι Ρολογιού	1.186	1.232	1.300	1.936
Στοιχεία/κύκλο ρολογιού	1,1	3,2	6,2	8,3

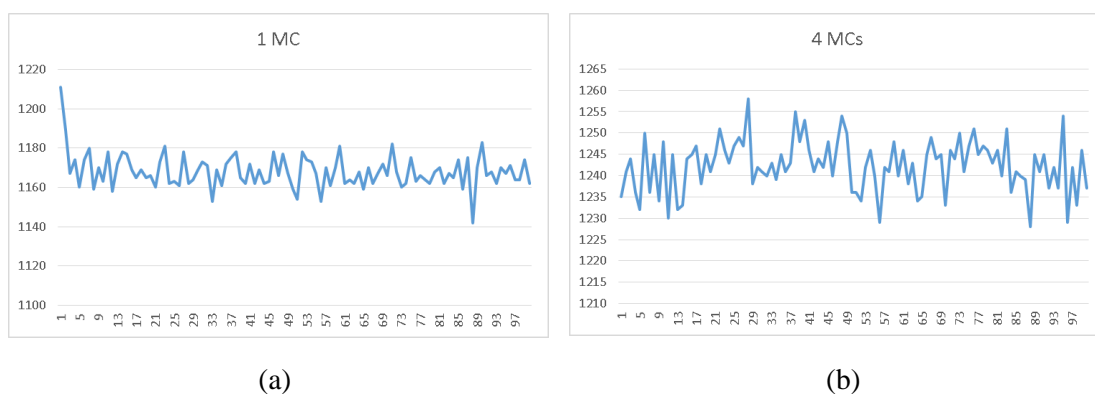
Παρατηρείται στον Πίνακα 5.3 ότι ο χρόνος εξυπηρέτησης των αιτημάτων αυξάνεται όσο αυξάνεται ο αριθμός των MCs που στέλνουν ταυτόχρονα αιτήματα στη μνήμη, όμως παράλληλα αυξάνεται πολύ ο αριθμός των στοιχείων που γίνονται διαθέσιμα στη σχεδίαση.

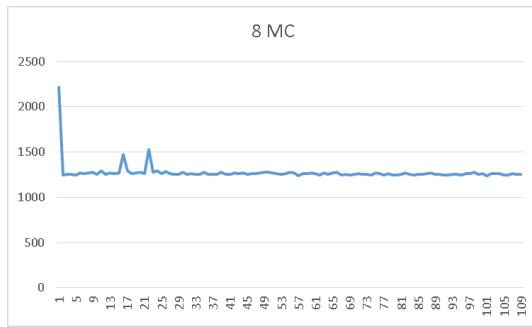
Από τις μετρήσεις του Πίνακα 5.2 με τη χρήση 1 MC αναμενόταν να μπορεί το σύστημα μνήμης να εξυπηρετεί και τους 4, 8 και 16 MCs με την ταχύτητα που εξυπηρετεί τον 1 MC, αφού το σύστημα του Convey HC-2 παρέχει στις FPGAs 16 ανεξάρτητους διαύλους

επικοινωνίας με τη μνήμη. Κάτι τέτοιο φαίνεται στον Πίνακα 5.3 πως δεν ισχύει αφού ο χρόνος εξυπηρέτησης αυξάνεται όσο αυξάνεται ο αριθμός των MCs που χρησιμοποιεί η σχεδίαση. Η βέλτιστη αναλογία στοιχείων προς κύκλους ρολογιού επιτυγχάνεται με τη χρήση 16 MCs και συνεπώς προκύπτει ότι η μία FPGA θα έχει περίπου 8 στοιχεία διαθέσιμα σε κάθε κύκλο ρολογιού για επεξεργασία, ενώ αναμενόταν η σχεδίαση με 16 MCs να έχει διαθέσιμα 16 στοιχεία ανά κύκλο ρολογιού. Δεύτερη καλύτερη αναλογία επιτυγχάνεται στο πείραμα με τη χρήση 8 MCs, που ανά κύκλο ρολογιού η FPGA θα έχει διαθέσιμα περίπου 6 στοιχεία. Τα αποτελέσματα αυτά προκύπτουν από το γεγονός ότι οι MCs που ενεργοποιεί η σχεδίαση δεν αντιστοιχούν ένας προς έναν με τα DIMMs του συστήματος μνήμης και άρα ένας MC εξυπηρετείται από πολλά DIMMs. Πιο συγκεκριμένα για 1.000 στοιχεία θα χρειαστεί και τα 16 DIMMs πολλές φορές το καθένα. Με αυτό τον τρόπο δημιουργούνται συγκρούσεις, όταν δηλαδή πολλοί MCs χρειάζονται το ίδιο DIMM για να προσπελάσει τα στοιχεία που ζητάνε.

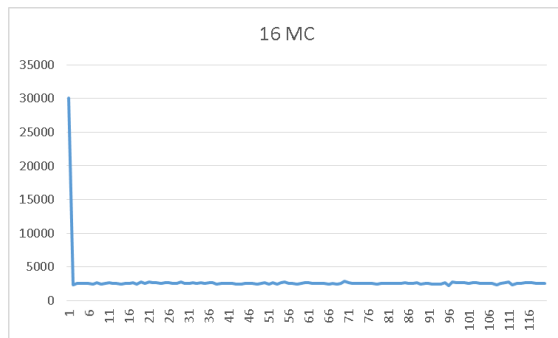
Από την περαιτέρω ανάλυση των αποτελεσμάτων των μετρήσεων προέκυψε το συμπέρασμα ότι το σύστημα μνήμης παρουσιάζει κάποιου είδους προσαρμοστικότητα στο χρόνο εξυπηρέτησης των στοιχείων για τις μετρήσεις με 16 MCs, όταν γίνεται επανάληψη του ίδιου μοτίβο προσπελάσεων, όπως φάνηκε και στα Διαγράμματα 5.2. Από τα Διαγράμματα 5.3 και κυρίως για τα πειράματα με 8 και 16 MCs φαίνεται ότι το σύστημα μνήμης καθυστερεί να προσπελάσει τα στοιχεία στις πρώτες επαναλήψεις των πειραμάτων, ενώ στη συνέχεια οι χρόνοι απόκρισής του σταθεροποιούνται και κυμαίνονται στο ίδιο εύρος τιμών. Από αυτό το γεγονός μπορεί να υποστηριχθεί ότι το σύστημα μνήμης προσαρμόζεται στις απαιτήσεις της σχεδίασης όταν επαναλαμβάνεται το ίδιο μοτίβο προσπελάσεων.

Διαγράμματα 5.3 – Αποτελέσματα Μετρήσεων (100 πρώτες επαναλήψεις των πειραμάτων)





(c)



(d)

5.2.2 Αποτελέσματα μετρήσεων προσπέλασης στοιχείων με «άλματα» (strides)

διευθύνσεων

Στον Πίνακα 5.3 παρουσιάζονται τα αποτελέσματα των μετρήσεων για την προσπέλαση στοιχείων των οποίων οι θέσεις (διευθύνσεις) είναι μη διαδοχικές στη μνήμη. Στις μετρήσεις αυτές έγινε χρήση 1 MC από τη σχεδίαση της FPGA. Στα πειράματα πραγματοποιούνται 1.000 διαδοχικές προσπελάσεις στοιχείων από τη μνήμη. Για τα στοιχεία έχουν επιλεγεί τέσσερις διαφορετικές αποστάσεις (άλματα) διευθύνσεων. Τα πειράματα επαναλήφθηκαν 1.000 φορές για διαφορετικά στοιχεία και στον Πίνακα 5.4 παρουσιάζεται ο μέσος όρος του χρόνου απόκρισης του συστήματος μνήμης σε κύκλους ρολογιού.

Πίνακας 5.4 – Αποτελέσματα Μετρήσεων Προσπέλασης Στοιχείων με Άλματα Διευθύνσεων

1 MC - Προσπέλαση 1.000 στοιχείων					
Δομή μνήμης	sub bus	MC	DIMM	bank	bank address
Απόσταση διευθύνσεων (bytes)	8	64	512	1024	8192
Κύκλοι ρολογιού	1.185	1.173	2.372	6.503	46.278

Ένα πρώτο συμπέρασμα που προκύπτει από τον Πίνακα 5.3 είναι ότι όσο μεγαλύτερη είναι η απόσταση μεταξύ των διευθύνσεων των διαδοχικών στοιχείων που ζητάει η σχεδίαση τόσο περισσότερο χρόνο χρειάζεται η μνήμη να εξυπηρετήσει αυτά τα αιτήματα. Επίσης, από τον Πίνακα 5.4 διαπιστώνεται ότι την μεγαλύτερη απόδοση το σύστημα την επιτυγχάνει για στοιχεία που απέχουν το ένα από το άλλο 8 ή 64 bytes, κάτι το οποίο παρατηρείται και από τον Bakos [8]. Για αυτό το μέγεθος «αλμάτων» το σύστημα της μνήμης επιτυγχάνει να τροφοδοτεί την FPGA με τον επιθυμητό αποδοτικότερο τρόπο, δηλαδή κάθε κύκλο ρολογιού με ένα στοιχείο. Για αποστάσεις διευθύνσεων 512 bytes η απόδοση του συστήματος της μνήμης μειώνεται στο μισό περίπου, δηλαδή η FPGA τροφοδοτείται με ένα στοιχείο ανά δύο κύκλους ρολογιού. Για αποστάσεις 1024 και 8192 bytes η απόκριση του συστήματος είναι πολύ πιο αργή από το επιθυμητό.

Το σύστημα της μνήμης είναι δομημένο ώστε να λειτουργεί με «έξυπνο» τρόπο προκειμένου να εξυπηρετεί με μεγάλη ταχύτητα πολλαπλά αιτήματα. Από το συνδυασμό και την ανάλυση των αποτελεσμάτων των Ενοτήτων 5.1 και 5.2 αρχίζει να φαίνεται ο τρόπος λειτουργίας του συστήματος της μνήμης του Convey HC-2.

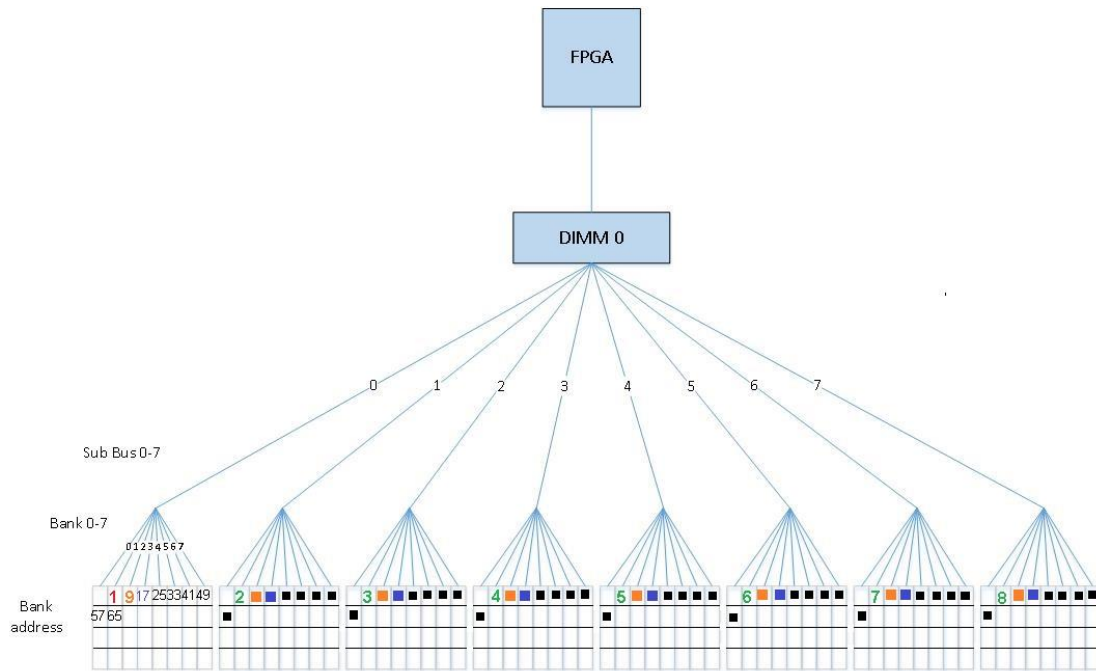
5.3 Τρόπος Λειτουργίας του Συστήματος Μνήμης

Το σύστημα της μνήμης προκειμένου να εξυπηρετήσει με ταχύτητα τα ζητούμενα στοιχεία αλλά και αυτά που θεωρεί πιθανό να ζητηθούν σε μελλοντικό χρόνο προσπαθεί και καταφέρνει να αξιοποιεί όλους τους πόρους που διαθέτει συνέχεια. Όταν μία σχεδίαση αιτεί ένα στοιχείο είναι πολύ πιθανό να ζητήσει και τα στοιχεία που βρίσκονται στις επόμενες θέσεις μνήμης από αυτό, δηλαδή σε επόμενα αιτήματα προς τη μνήμη. Το σύστημα των DIMMs μπορεί να θεωρηθεί ότι λειτουργεί ως μιας μορφής «κρυφή μνήμη».

Για κάθε MC της σχεδίασης το σύστημα των DIMMs διαθέτει μία FIFO (FIFO_{MC}), που πιθανόν έχει μέγεθος 1024 θέσεων. Κάθε φορά που ένας MC από τη σχεδίαση στέλνει ένα αίτημα στη μνήμη δεν λειτουργεί μόνο το DIMM που θα εξυπηρετήσει το συγκεκριμένο στοιχείο αλλά και τα 16 DIMMs. Τα 16 DIMMs γεμίζουν κάθε FIFO_{MC} με στοιχεία που βρίσκονται στις επόμενες διαδοχικές θέσεις μνήμης από το ζητούμενο στοιχείο, δηλαδή που απέχουν κατά 8 bytes (1 λέξη). Με αυτόν τον τρόπο το σύστημα επιτυγχάνει αντί να χρειάζεται 100 κύκλους ρολογιού για να φέρει το κάθε στοιχείο στη σχεδίαση να «πληρώνει» μία φορά τους 100 κύκλους στην αρχή και να έχει τα επόμενα στοιχεία έτοιμα, εφόσον ζητηθούν από τη σχεδίαση.

Με τη μέθοδο αυτή το σύστημα των DIMMs επιτυγχάνει την αποθήκευση στη FIFO_{MC} ενός μεγάλου αριθμού στοιχείων (μάλλον 1024) από τη μνήμη σε περίπου 100-200 κύκλους ρολογιού. Οι κύκλοι ρολογιού που αναφέρονται είναι υπολογισμένοι κατά προσέγγιση, καθώς εμφανίζονται καθυστερήσεις (stalls) στους memory controllers με αποτέλεσμα να μην υπάρχει ακρίβεια στα αποτελέσματα των μετρήσεων.

Όταν η σχεδίαση ζητάει το πρώτο στοιχείο, τα 16 DIMMs αντλούν από τη μνήμη τα 128 στοιχεία που βρίσκονται στις επόμενες θέσεις μνήμης από αυτό που ζητήθηκε και αποθηκεύονται στη FIFO_{MC}. Με αυτό τον τρόπο το κάθε DIMM αντλεί από τη μνήμη τα 8 από τα 128 στοιχεία (128 στοιχεία / 16 DIMMs) που αντιστοιχούν στις διευθύνσεις που αυτό εξυπηρετεί, δηλαδή τα στοιχεία που διαφέρουν μόνο στο πεδίο του sub bus. Στη συνέχεια το κάθε DIMM αλλάζοντας την παράμετρο της τράπεζας στη διεύθυνση κατά μία, αντλεί τα επόμενα 8 στοιχεία που διαφέρουν μόνο στο sub bus και τα αποθηκεύει στη FIFO_{MC}. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να γεμίσει η FIFO_{MC}, ενώ εκτιμάται ότι η FIFO_{MC} έχει μέγεθος 1024 θέσεων.



Σχήμα 5.1 – Τρόπος Λειτουργίας 1 DIMM

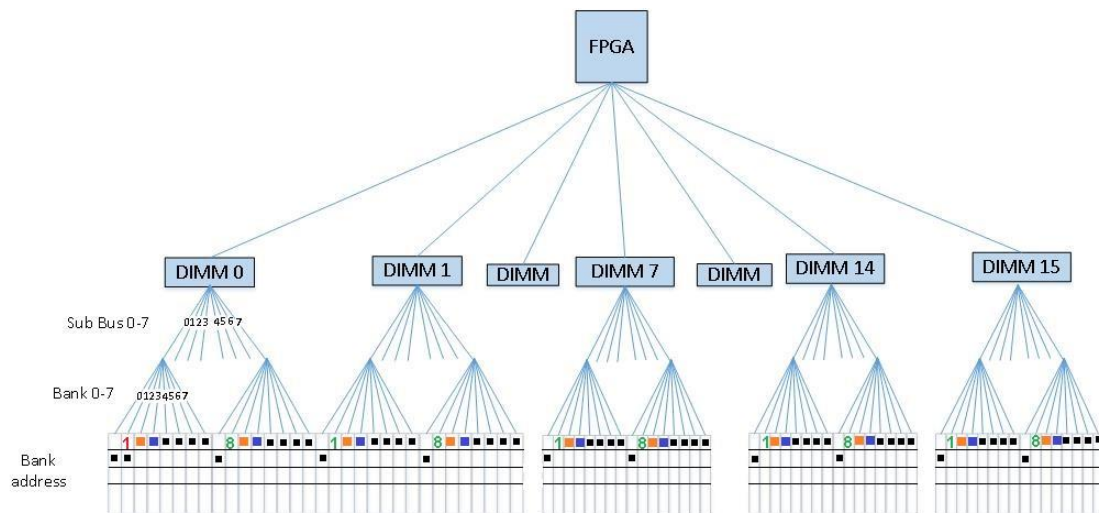
Για παράδειγμα, στο Σχήμα 5.1 παρουσιάζεται η εκτίμηση λειτουργίας ενός DIMM όταν ζητήσει η FPGA από τη μνήμη να προσπελάσει το στοιχείο με διεύθυνση 1024. Η αποκωδικοποίηση της διεύθυνσης 1024 στο Σχήμα 5.2 γίνεται με βάση το Σχήμα 3.7.

bank addr	bank	DIMM	sub bus	
[...000]	[001]	[0000]	[000]	[000]

Σχήμα 5.2 – Αποκωδικοποίηση Διεύθυνσης από το Σύστημα Μνήμης

Συνεπώς για την προσπέλαση του στοιχείου στη θέση μνήμης 1024 θα χρησιμοποιηθεί το πρώτο DIMM (DIMM 0), το πρώτο sub bus του DIMM (sub bus 0) και το στοιχείο θα βρίσκεται στην πρώτη θέση (bank addr 0) της δεύτερης τράπεζας μνήμης (bank 1). Στο Σχήμα 5.1 με κόκκινο χρώμα παρουσιάζεται το στοιχείο στη θέση 1024 το οποίο ζητάει η σχεδίαση. Στη συνέχεια απαριθμούνται τα επόμενα στοιχεία που θα προσπελάσει το σύστημα μνήμης και τα οποία θα αποθηκευτούν στη FIFO_{MC}. Με πράσινο χρώμα απεικονίζονται τα πρώτα στοιχεία που θα προσπελαστούν, με πορτοκαλί τα επόμενα, με μπλε τα αμέσως επόμενα και ούτω καθ' εξής.

Η διαδικασία που παρουσιάζεται στο Σχήμα 5.1 επαναλαμβάνεται την ίδια χρονική στιγμή και στα 16 DIMMs, έτσι ώστε να γεμίσει η FIFO_{MC} με τα στοιχεία που βρίσκονται στις επόμενες θέσεις μνήμης από αυτό που η σχεδίαση ζήτησε να προσπελαστεί. Στη συνέχεια στο Σχήμα 5.3 παρουσιάζεται ένα παράδειγμα της εκτίμηση της λειτουργίας όλου του συστήματος της μνήμης και για τα 16 DIMMs.



Σχήμα 5.3 – Τρόπος Λειτουργίας Συστήματος Μνήμης (16 DIMMs)

Στο Σχήμα 5.3 παρουσιάζεται η εκτίμηση λειτουργίας ολόκληρου του συστήματος μνήμης και των 16 DIMMs όταν ζητήσει η FPGA από τη μνήμη να προσπελάσει το στοιχείο με διεύθυνση 1024. Με κόκκινο χρώμα απεικονίζεται το στοιχείο που έχει ζητήσει η σχεδίαση, με πράσινο τα πρώτα στοιχεία που θα αποθηκεύσουν τα 16 DIMMs στη FIFO_{MC}, με μπλε τα επόμενα στοιχεία, με πορτοκαλί τα αμέσως επόμενα και ούτω καθ' εξής, μέχρι να γεμίσει η FIFO_{MC}.

Ο τρόπος λειτουργίας του συστήματος της μνήμης μπορεί να δικαιολογήσει τα αποτελέσματα των πειραμάτων στην Ενότητα 5.2. Για τα αποτελέσματα του Πίνακα 5.2 επιτυγχάνεται, ανεξάρτητα από τον όγκο των δεδομένων, η μνήμη να τροφοδοτεί την FPGA με νέο δεδομένο κάθε κύκλο ρολογιού. Αναμενόταν μετά τα αποτελέσματα για μεμονωμένες προσπελάσεις τα αποτελέσματα για μαζικές προσπελάσεις να είναι πολλαπλάσια της 1:1 αναλογίας που επιτυγχάνεται τελικά. Το γεγονός είναι ότι το σύστημα της μνήμης έχει διαθέσιμα στη FIFO_{MC} τα πρώτα 1.000 στοιχεία σε χρόνο πολύ μικρότερο των 1.000 κύκλων ρολογιού που ζητάει η σχεδίαση. Εκτιμάται ότι ο χρόνος αυτός κυμαίνεται από 150-200 κύκλους ρολογιού.

Συνεχίζοντας, από τη στιγμή που στη σχεδίαση οι MCs είναι εικονικοί και δεν αντιστοιχούν στα πραγματικά DIMMs αναμενόταν ότι η χρήση μεγαλύτερου αριθμού MCs την ίδια χρονική στιγμή θα αύξανε ανάλογα και το χρόνο προσπέλασης των στοιχείων από τη μνήμη, κάτι όμως που δεν συνέβη στην πραγματικότητα. Από τα αποτελέσματα του Πίνακα 5.3 φάνηκε ότι ακόμα και για σχεδιάσεις με χρήση 8 MCs η μνήμη ανταποκρίνεται στα αιτήματα της σχεδίασης σε ικανοποιητικό χρόνο. Η δυνατότητα αυτή προκύπτει από το γεγονός ότι το σύστημα έχει προσπελάσει τα 1.000 στοιχεία που θα ζητηθούν από τον κάθε MC σε 150-200 κύκλους ρολογιού. Άρα για τα 8.000 στοιχεία που αιτούνται οι 8 MCs (1.000 στοιχεία / 1 MC) το σύστημα της μνήμης ανταποκρίνεται σε χρόνο περίπου ίσο με 1.200-

1.600 κύκλους ρολογιού (8 * 150-200), κάτι που γίνεται εμφανές και από τα πειραματικά αποτελέσματα.

Μετά την ανάλυση του τρόπου λειτουργίας του συστήματος των DIMMs, έγιναν πειράματα για την εύρεση ενός τρόπου εκμετάλλευσης της λειτουργικότητας του συστήματος από μια σχεδίαση με σκοπό την προσπέλαση μεγάλων μπλοκ μνήμης με ταχύτερο τρόπο. Σε συνέχεια των πειραμάτων στην Ενότητα 5.2.2 μελετήθηκε το όριο της μνήμης όταν της ζητείται να προσπελάσει στοιχεία με διαδοχικά «άλματα» στις θέσεις μνήμης. Σκοπός ήταν η εύρεση του καταλληλότερου «άλματος» έτσι ώστε να διατηρείται η απόδοση του ενός στοιχείου προς έναν κύκλο ρολογιού. Αποτέλεσμα αυτών των πειραμάτων ήταν ο Πίνακας 5.4 να τροποποιηθεί όπως φαίνεται στον Πίνακα 5.5.

Πίνακας 5.5 – Αποτελέσματα Μετρήσεων Προσπέλασης Στοιχείων – Βέλτιστο «Άλμα» στη Μνήμη

1 MC - Προσπέλαση 1.000 στοιχείων						
Δομή μνήμης	sub bus	MC		DIMM	bank	bank address
Απόσταση διευθύνσεων (bytes)	8	64	504	512	1024	8192
Κύκλοι ρολογιού	1185	1173	1190	2372	6503	46278

Από τα αποτελέσματα των μετρήσεων, όπως φαίνονται στον Πίνακα 5.5, προέκυψε ότι ο πιο αποδοτικός τρόπος προσπέλασης μεγάλων τμημάτων της μνήμης είναι όταν η σχεδίαση ζητάει στοιχεία με διαφορά διευθύνσεων 504 bytes (63λέξεις). Όταν η σχεδίαση ζητάει ανά κύκλο ρολογιού στοιχεία με απόσταση 63 «λέξεων» το σύστημα των DIMMs εξυπηρετεί αυτές τις αιτήσεις με την ίδια ταχύτητα όπως και για στοιχεία σε διαδοχικές θέσεις μνήμης (απόσταση 1 λέξη ή 8 bytes). Παραδείγματος χάρη αν η σχεδίαση στέλνει 1.000 αιτήσεις στα DIMMs, με τη διαφορά των στοιχείων στις 63 «λέξεις» (504 bytes), το σύστημα θα τις εξυπηρετήσει σε 1190 κύκλους ρολογιού, όπως και στην περίπτωση που τα στοιχεία απείχαν κατά 1 «λέξη» (8 bytes) ή 8 «λέξεις» (64 bytes). Με αυτό τον τρόπο όμως η σχεδίαση καταφέρνει να φτάσει σε 1190 κύκλους ρολογιού βαθύτερα στη μνήμη κατά 63.000 θέσεις ή 504Kbytes από την πρώτη θέση.

5.4 Απόκριση Μνήμης – Τριγωνικοί Πίνακες

Για να μελετηθεί το σύστημα μνήμης σε πιο απαιτητικές δομές μνήμης, έγινε μέτρηση της ταχύτητας άντλησης δεδομένων από τριγωνικούς πίνακες. Οι τρόποι άντλησης των δεδομένων επιλέχτηκαν σύμφωνα με τα αποτελέσματα των μετρήσεων της Ενότητας 5.2 και αφού ήταν γνωστή η λειτουργία του συστήματος μνήμης (Ενότητα 5.3). Για το λόγο αυτό πραγματοποιήθηκαν τρεις διαφορετικοί τύποι πειραμάτων.

- Προσπέλαση όλων των στοιχείων του πίνακα
- Προσπέλαση των στοιχείων του κάτω τριγωνικού πίνακα

- a. Binary Διαφύλλωση
- b. 31/31 Διαφύλλωση
- c) Προσπέλαση του πίνακα με υβριδικό τρόπο

Τα αποτελέσματα των μετρήσεων για πίνακες 10.000×10.000 και 5.000×5.000 στοιχεία παρουσιάζονται στον Πίνακα 5.6.

Πίνακας 5.6 – Αποτελέσματα Μετρήσεων Προσπέλαση Τριγωνικών Πινάκων

Τριγωνικοί Πίνακες (κύκλοι ρολογιού)			
Τρόπος Διαβάσματος	A	B	C
Διαφύλλωση	binary	binary	binary
12.500.000 Στοιχεία	25.000.000	12.500.519	12.745.650
50.000.000 Στοιχεία	100.000.000	50.014.178	51.001.819

Από τον Πίνακα 5.6 προκύπτει ότι για τη προσπέλαση ενός τριγωνικού πίνακα ο αποδοτικότερος τρόπος είναι το απευθείας διάβασμα των στοιχείων του κάτω τριγωνικού πίνακα. Ο πιο αργός τρόπος είναι η προσπέλαση όλων των στοιχείων του πίνακα, όπως ήταν αναμενόμενο. Ο υβριδικός τρόπος που προτάθηκε από την παρούσα διπλωματική για τη προσπέλαση των στοιχείων του κάτω τριγωνικού πίνακα προκύπτει λιγότερο αποδοτικός για τις συγκεκριμένες δομές και πιο συγκεκριμένα σε ποσοστό 2%, ενώ αναμενόταν να είναι τουλάχιστον το ίδιο αποδοτικός με το σύστημα του Convey.

Από τις μετρήσεις αυτές εξάγεται το συμπέρασμα ότι το σύστημα της μνήμης είναι πολύ αποδοτικό και με τους δύο τρόπους διαφύλλωσης της μνήμης. Το σύστημα μνήμης είναι κατασκευασμένο έτσι ώστε να προσαρμόζεται γρήγορα στις απαιτήσεις κάθε σχεδίασης με τον αποδοτικότερο τρόπο. Διαπιστώνεται ότι το σύστημα των DIMMs έχει την ικανότητα να αντιμετωπίζει «άλματα» στις διευθύνσεις της μνήμης αποτελεσματικά, ώστε να ανακτήσει τα ζητούμενα στοιχεία πιο γρήγορα. Ειδικότερα όταν τα «άλματα» είναι λίγα σε αριθμό και μεταξύ των «αλμάτων» παρεμβάλλονται προσπελάσεις διαδοχικών θέσεων μνήμης όπως στην περίπτωση που μελετάται, δηλαδή του τριγωνικού πίνακα. Όταν μια σχεδίαση ζητάει στοιχεία με μικρό αριθμό μεγάλων «αλμάτων» (> 8192 bytes) διευθύνσεων ή με πολλά μικρά «άλματα» (< 1024 bytes), το σύστημα μνήμης του Convey HC-2 παρουσιάζει μεγάλη αξιοπιστία στην ταχύτητα προσπέλασης των στοιχείων αυτών. Επίσης αν συνδυαστούν οι μετρήσεις των τριγωνικών πινάκων με τη λειτουργία της μνήμης, όπως αναλύθηκε στην Ενότητα 5.3, προκύπτει ότι όταν η σχεδίαση χρειάζεται το 50% των στοιχείων ενός πίνακα πρέπει να χρησιμοποιείται η δεύτερη μέθοδος που μελετήθηκε, δηλαδή η απευθείας προσπέλαση αυτών των στοιχείων χωρίς επέμβαση στον τρόπο προσπέλασης των στοιχείων από το σύστημα μνήμης του Convey. Το συμπέρασμα αυτό ισχύει μόνο εφόσον τα στοιχεία ζητούνται ανά κύκλο ρολογιού.

6

Συμπεράσματα και Μελλοντικές Επεκτάσεις

6.1 Συμπεράσματα

Στη διπλωματική εργασία μελετήθηκε ο τρόπος και η ταχύτητα ανάκτησης δεδομένων από τη μνήμη σε έναν υβριδικό υπερ-υπολογιστή, το Convey HC-2. Τα συμπεράσματα που αντλούνται από την εργασία συνοψίζονται παρακάτω. Επιπλέον γίνεται αναφορά σε κάποια γενικά συμπεράσματα που αφορούν τις αρχές που διέπουν τη λειτουργία των memory controllers και προέκυψαν από την ανάλυση των αποτελεσμάτων.

- Το σύστημα της μνήμης πρέπει να τροφοδοτείται συνεχώς με αιτήματα από τις FPGAs, δηλαδή σε κάθε κύκλο ρολογιού είναι απαραίτητο η σχεδίαση να ζητάει δεδομένα ώστε το σύστημα DIMM-Μνήμης να είναι πιο αποδοτικό.
- Τα δεδομένα που ζητάει η σχεδίαση είναι προτιμότερο να βρίσκονται σε γειτονικές θέσεις στη μνήμη, δηλαδή οι διευθύνσεις τους να μην απέχουν παραπάνω από 504 bytes ή 63 λέξεις. Με αυτόν τον τρόπο και για μεγάλο αριθμό αιτήσεων (μεγαλύτερο από 1000) επιτυγχάνεται ο λόγος στοιχεία / κύκλοι ρολογιού να ισούται με 1.
- Για το διάβασμα τριγωνικών πινάκων προέκυψε ότι αποδοτικότερος τρόπος ανάκτησης των δεδομένων είναι το απευθείας διάβασμα των στοιχείων με το σύστημα προσπέλασης της μνήμης του Convey.
- Όταν χρησιμοποιείται μία FPGA είναι αποδοτικότερο η σχεδίαση να χρησιμοποιεί και τους 16 MCs ταυτόχρονα. Η διαπίστωση αυτή οδηγεί στο συμπέρασμα ότι η

ταυτόχρονη χρήση MCs από μία σχεδίαση εξαρτάται από τον αριθμό των FPGAs που ενεργοποιούνται κάθε φορά.

- Ο τύπος διαφύλλωσης που θα χρησιμοποιείται εξαρτάται από την κατανομή των δεδομένων στη μνήμη.
 - a) Η binary διαφύλλωση θα χρησιμοποιείται για δεδομένα συνεχόμενα στη μνήμη γιατί με αυτόν τον τρόπο επιτυγχάνεται το μεγαλύτερο εύρος ζώνης στη μεταφορά δεδομένων.
 - b) Η 31/31 διαφύλλωση μειώνει το διαθέσιμο εύρος ζώνης, αλλά είναι ο βέλτιστος τρόπος όταν πρέπει να γίνονται περιορισμένα «άλματα» στα δεδομένα της μνήμης.

Γενικά Συμπεράσματα

- Η εκμάθηση και χρήση της πλατφόρμας και των εργαλείων του Convey είναι χρονοβόρα διαδικασία και έχει μεγάλο βαθμό δυσκολίας.
- Η χρήση του Crossbar στις σχεδιάσεις καθίσταται σχεδόν απαραίτητη. Ειδικά, κάθε σχεδίαση θα πρέπει να αποκωδικοποιεί τις διευθύνσεις που στέλνει στη μνήμη, για να χρησιμοποιεί κάθε φορά το σωστό DIMM.
- Η λειτουργία του συστήματος των DIMMs παρουσιάζει χαρακτηριστικά που θυμίζουν τη λειτουργία της κρυφής μνήμης.
- Ο τρόπος αξιοποίησης όλων των DIMMs από το μηχανήμα είναι σχεδόν ανεξάρτητος από τις απαιτήσεις των σχεδιάσεων, με σκοπό την αποδοτικότερη λειτουργία του συστήματος I/O των FPGAs.
- Η αξιοπιστία του τρόπου διαχείρισης της μνήμης από τα DIMMs, επιτυγχάνεται σε μεγάλο βαθμό από την αρχιτεκτονική του Convey.

6.2 Μελλοντικές Επεκτάσεις

Η μελέτη που πραγματοποιήθηκε στη διπλωματική εργασία ανοίγει τον ορίζοντα για πολλές μελλοντικές επεκτάσεις.

Ένας ευρύτερος τομέας μελλοντικών επεκτάσεων είναι η περαιτέρω μελέτη του συστήματος μνήμης. Μία πρώτη διερεύνηση μπορεί να πραγματοποιηθεί με χρήση περισσότερων FPGAs, έτσι ώστε να οριοθετηθεί η αποδοτικότητα της χρήσης του μηχανήματος. Λεπτομερής διερεύνηση μπορεί να πραγματοποιηθεί και προς την κατεύθυνση προβλημάτων που μπορεί να παρεμποδίζουν την αποδοτική λειτουργία του συστήματος. Τέτοια αποτελούν τυχόν συγκρούσεις δεδομένων κατά τη μεταφορά τους από τα DIMMs, όπως και αστοχίες των DIMMs στη μνήμη. Η υιοθέτηση ενός συστήματος επίλυσης των προβλημάτων μπορεί να απασχολήσει μελλοντικές εργασίες. Ο τρόπος λειτουργίας της 31/31 διαφύλλωσης που αποτελεί πατέντα της «Convey computers» δεν είναι σαφής. Η ανάπτυξη ενός μοντέλου που αποσαφηνίζει τον τρόπο λειτουργίας της διαφύλλωσης αυτής θα οδηγήσει στην αποδοτικότερη διαχείριση της μνήμης. Επιπλέον σε μελλοντικές εργασίες μπορεί να μελετηθεί ο τρόπος και να μετρηθεί η ταχύτητα με την οποία γράφουν οι FPGAs στοιχεία στη μνήμη.

Επιπρόσθετα, με αφετηρία τα αποτελέσματα της εργασίας μπορεί να πραγματοποιηθούν μετρήσεις για τη συμπεριφορά του συστήματος των DIMMs σε διαφορετικές δομές δεδομένων, όπως είναι τα stencils. Βελτιστοποίηση στην ακρίβεια της αρχιτεκτονικής της σχεδίασης ενός αλγορίθμου μπορεί να επιτευχθεί αξιοποιώντας τα αποτελέσματα των μετρήσεων της εργασίας. Τα αποτελέσματα των μετρήσεων που προέκυψαν δίνουν κατευθύνσεις για τον αποδοτικό τρόπο ανάκτησης δεδομένων από το σύστημα μνήμης για μελλοντικές σχεδιάσεις πάνω στο Convey.

Βιβλιογραφία

- [1] http://en.wikipedia.org/wiki/Moore%27s_law
- [2] http://en.wikipedia.org/wiki/Biomolecular_structure#Secondary_structure
- [3] http://www.conveycomputer.com/files/4113/5394/7097/Convey_HC-2_Architectual_Overview.pdf
- [4] Convey, “Convey Personality Development Kit Reference Manual”, Version 5.2, April 2012
- [5] Convey, “Convey Reference Manual”, Version 2.00, September 2009.
- [6] Convey, “Convey Programmers Guide”, Version 1.8, November 2010.
- [7] Convey, “Convey Spat Users Guide”, Version 1.0, June 2009.
- [8] Bakos J. D.: “High-performance heterogeneous computing with the Convey HC-1”. Computing in Science & Engineering, 2010.
- [9] Berger M. and Olinger J., “Adaptive mesh refinement for hyperbolic partial differential equations,” Journal of Computational Physics, vol. 53, 1984, pp. 484–512.
- [10] Brewer T.M.: “Instruction set innovations for the Convey HC-1 computer” - IEEE micro, 2010.
- [11] Datta K., Murphy M., Volkov V., Williams S., Carter J., Oliner L., Patterson D., Shalf J., and Yelick K.: “Stencil Computation Optimization and Auto-tuning on State-of-the-Art Multicore Architectures”, SC '08 Proceedings of the 2008 ACM/IEEE conference on Supercomputing, 2008.
- [12] Durbano J. P., Ortiz F. E., Humphrey J. R., Curt P. F., and Prather D. W.: “FPGA-Based Acceleration of the 3D Finite-Difference Time-Domain Method,” in FCCM '04: Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines: IEEE Computer Society, 2004, pp. 156–163.
- [13] He C., Zhao W. and Lu M.: “Time Domain Numerical Simulation for Transient Waves on Reconfigurable Coprocessor Platform,” in FCCM'05: Proceedings of the 13th Annual IEEE Symposium on Field-Programmable Custom Computing Machines. IEEE Computer Society, 2005, pp. 127–136.

- [14] Jin Z. and Bakos J. D.: “Memory Access Scheduling on the Convey HC-1”. FCCM, 2013.
- [15] Karl Savio and Pimenta Pereira, “Characterization of FPGA-based High Performance Computers”, thesis in Electrical Engineering, 2011.
- [16] Kunkel, Julian M., and Petra Nerge. “System performance comparison of stencil operations with the Convey HC-1”. Research Group Scientific Computing, University of Hamburg, Tech. Rep, 2010, pp. 11-16.
- [17] Markham N. R. and Zuker M.: “Unafold: Software for nucleic acid folding and hybridization”. In Jonathan M. Keith, editor, Bioinformatics, Volume II. Structure, Functions and Applications, number 453 in Methods in Molecular Biology, chapter 1, Humana Press, 2008, pp. 3–31.
- [18] Nagar K. K. and Bakos J. D.: “A sparse matrix personality for the convey HC-1”. In: Field-Programmable Custom Computing Machines (FCCM), 2011 IEEE 19th Annual International Symposium on. IEEE, 2011. pp. 1-8.
- [19] Patterson D. A. and Hennessy J. L.: Computer Organization & Design, The Hardware/ Software Interface, Morgan Kaufmann, 1994, pp. 466-611.
- [20] SHAFIQ, Muhammad, et al.: “Exploiting memory customization in FPGA for 3D stencil computations”. In: Field-Programmable Technology, 2009. FPT 2009. International Conference on. IEEE, 2009. p. 38-45.
- [21] Scrofano R. and Prasanna V. K.: “Computing Lennard-Jones Potentials and Forces with Reconfigurable Hardware”. In International Conference on Engineering of Reconfigurable Systems and Algorithms, June 2004.
- [22] Underwood K. D. and Hemmert K. S.: “Closing the Gap: CPU and FPGA Trends in Sustainable Floating-Point BLAS Performance”. In Proceedings of 2004 IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM’04), April 2004.
- [23] Zhuo L. and Prasanna V. K.: “Sparse Matrix-Vector Multiplication on FPGAs”. FPGA '05 In Proceedings of ACM/SIGDA 13th symposium on Field-programmable gate arrays, 2005, pp. 63-74.
- [24] Zhuo L. and Prasanna V. K.: “Scalable and Modular Algorithms for Floating-Point Matrix Multiplication on FPGAs”. In Proceedings of The 18th International Parallel & Distributed Processing Symposium, April 2004.
- [25] Zhuo L. and Prasanna V. K.: “High Performance Linear Algebra Operations on Reconfigurable Systems”. SC '05 In Proceedings of ACM/IEEE conference on Supercomputing, 2005, pp. 2.
- [26] Zuker M and Stiegler P.: “Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information”. Nucleic Acids Res. Jan 10; 9(1) , 1981, pp. 48-133.