Ετήσια Τεχνική Έκθεση

Έτος 2012

ΘΑΛΗΣ – Πολυτεχνείο Κρήτης

Πλατφόρμα προηγμένων μαθηματικών μεθόδων και λογισμικού για την επίλυση προβλημάτων πολλαπλών πεδίων (multiphysics, multidomain) σε σύγχρονες υπολογιστικές αρχιτεκτονικές: Εφαρμογή σε προβλήματα Περιβαλλοντικής Μηχανικής και Ιατρικής (MATENVMED- MIS 379416)

Δράση 3.2

ΥΛΟΠΟΙΗΣΗ ΣΕ FPGAs ΚΑΙ ΠΟΛΥΠΥΡΗΝΑ ΣΥΣΤΗΜΑΤΑ (MULTICORES / MANYCORES)

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Πίνακας Περιεχομένων

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

1. Σκοπός

Ο σκοπός της Δράσης 3.2 είναι η αποδοτική υλοποίηση των αριθμητικών μεθόδων για ασυνεχή προβλήματα πολλαπλών πεδίων σε αρχιτεκτονικές επεξεργασίας γραφικών (Graphics Processor Units, GPU) καθώς και σε επαναδιατασσόμενες (reconfigurable) αρχιτεκτονικές (πχ. Field Programmable Gate Arrays, FPGAs).

Στόχος της μελέτης θα είναι να εκτιμηθεί η καταλληλόλητα του κάθε πακέτου λογισμικού (συνολικά ή κατά τμήματά του) για εκτέλεση σε καθεμιά από τις πολυπύρηνες πλατφόρμες που θα επιλεγούν ως πειραματικές πλατφόρμες στο έργο, ή εναλλακτικά για επιτάχυνση με χρήση εξειδικευμένου κατά περίπτωση αναδιατασσόμενου υλικού.

Με βάση τα αποτελέσματα της μελέτης θα μεταφερθούν και απεικονιστούν τα πακέτα λογισμικού – είτε καθολικά, είτε κατά τμήματα - στα κατάλληλα κατά περίπτωση πολυπύρηνα συστήματα. Επίσης θα κατασκευαστεί εξειδικευμένο υλικό για την επιτάχυνση συγκεκριμένων υπολογιστικών πυρήνων των εφαρμογών και το απαραίτητο λογισμικό για την επικοινωνία με τον υπολογιστή ξενιστή (Host computer).

Συγκεκριμένα για την υλοποίηση σε FPGA, ενδέχεται να χρησιμοποιηθεί λογισμικό CAD που θα αναπτυχθεί στο Πανεπιστήμιο Θεσσαλίας. Το λογισμικό αυτό μετατρέπει εφαρμογές που έχουν αναπτυχθεί σε OpenCL σε επιταχυντές υλικού (hardware accelerators). Με αυτόν τον τρόπο επιταχύνει κατά πολύ την ανάπτυξη ενός συστήματος FPGA χωρίς να χρειάζονται γνώσεις της αρχιτεκτονικής του συστήματος.

Ως ενδεικτικές πολυπύρηνες πλατφόρμες προτείνονται οι επεξεργαστές γενικού σκοπού της Intel ή της AMD καθώς και GPUs της NVIDIA ή της ATI. Ως ενδεικτική πλατφόρμα αναδιατασσόμενης λογικής προτείνονται συστήματα τα οποία περιλαμβάνουν Xilinx FPGA και μικροεπεξεργαστή σε κάρτα PCI-Express και μπορούν να τοποθετηθούν σε σύστημα αρχιτεκτονικής x86 ή συμβατό.

Σημειώνουμε ότι για τις ανάγκες της προτεινόμενης έρευνας είναι επιβεβλημένη η προμήθεια υπολογιστικών συστημάτων FPGA/GPU αρχιτεκτονικής.

2. Εισαγωγή

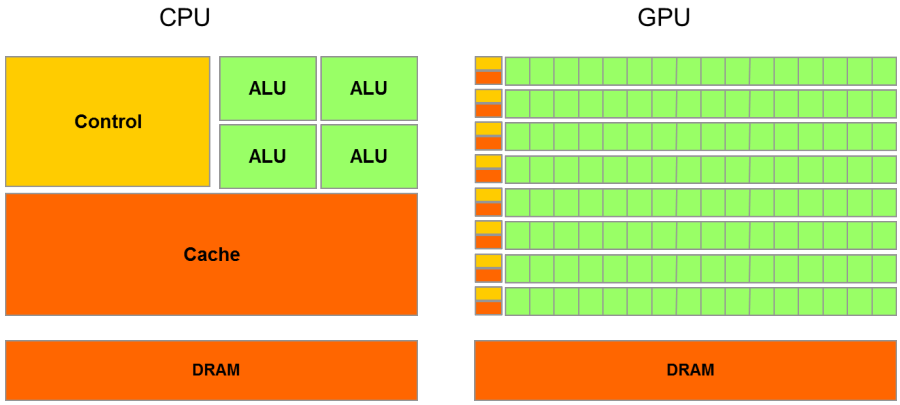
Σε αυτό το κεφάλαιο θα κάνουμε μία ανασκόπηση των βασικών επεξεργαστικών μονάδων που θα χρησιμοποιηθούν στην Δράση 3.2 του MATENVMED.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

2.1. Τεχνολογία Επεξεργασίας Γραφικών (Graphics Processor Unit, GPU)

Η GPU είναι ένας σύγχρονος πολυεπεξεργαστής (multicore) με υψηλό βαθμό παραλληλίας [4]. Μία σύγχρονη GPU διαθέτει μία ενοποιημένη αρχιτεκτονική για γραφικά και υπολογιστική που χρησιμοποιείται και ως προγραμματιζόμενος επεξεργαστής. Οι αρχιτεκτονικές GPU βασίζονται σε παράλληλες διατάξεις πολλών και σχετικά απλών προγραμματιζόμενων επεξεργαστών (Streaming Processors, SPs).

Σε σύγκριση με τις πολυπύρηνες CPUs (multicore CPUs), οι GPUs έχουν πολύ μεγαλύτερο αριθμό πυρήνων (cores) με διαφορετικό στυλ αρχιτεκτονικής που εστιάζει στην αποδοτική εκτέλεση πολλών νημάτων (threads). Με την χρήση πολλών απλούστερων πυρήνων και με την βελτιστοποίηση της συμπεριφοράς της παραλληλίας δεδομένων μεταξύ ομάδων νημάτων, μεγαλύτερο μέρος των τρανζίστορ του κάθε chip είναι αφιερωμένο σε υπολογισμούς, και μικρότερο σε μνήμες και σε υλικό ελέγχου

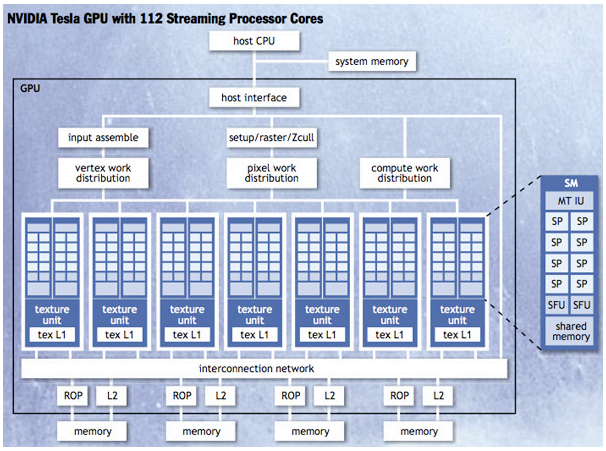


Εικόνα 1. Βασικές αρχές της αρχιτεκτονικής της CPU και της GPU

(Control) όπως φαίνεται στην Εικόνα 1.

Όπως φαίνεται στην Εικόνα 2, μία σύγχρονη GPU περιέχει πολλούς πυρήνες επεξεργαστών. Ο επεξεργαστής της Εικόνα 2 αποτελείται από 112 επεξεργαστές συνεχούς ροής (Streaming Processors, SPs) οι οποίοι είναι οργανωμένοι ως 14 πολυνηματικοί πολυεπεξεργαστές συνεχούς ροής (Streaming Multiprocessors, SMs). Κάθε SP εκτελεί σε κάθε χρονική στιγμή και ένα νήμα (thread) της γλώσσας CUDA που θα περιγραφεί στο επόμενο κεφάλαιο. Κάθε SM έχει οκτώ πυρήνες SP, δύο μονάδες ειδικών μαθηματικών συναρτήσεων (Special Function Units, SFUs), κρυφές μνήμες cache εντολών και σταθερών, μία Multithreaded Μονάδα εντολών και μία κοινόχρηστη μνήμη η οποία μπορεί να προσπελασθεί από όλα τα SPs που ανήκουν στο δεδομένο SM.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Εικόνα 2. Βασική αρχιτεκτονική της NVIDIA Tesla GPU.

Αυτή είναι η βασική αρχιτεκτονική Tesla που υλοποιείται στην GeForce 8800 GPU της NVIDIA. Στην ενοποιημένη αυτή αρχιτεκτονική \εκτελούνται τα παραδοσιακά προγράμματα γραφικών για σκίαση κορυφών, γεωμετρικών σχημάτων και pixels καθώς και προγράμματα γενικότερης υπολογιστικής μέσω της γλώσσας προγραμματισμού CUDA αλλά και της OpenCL.

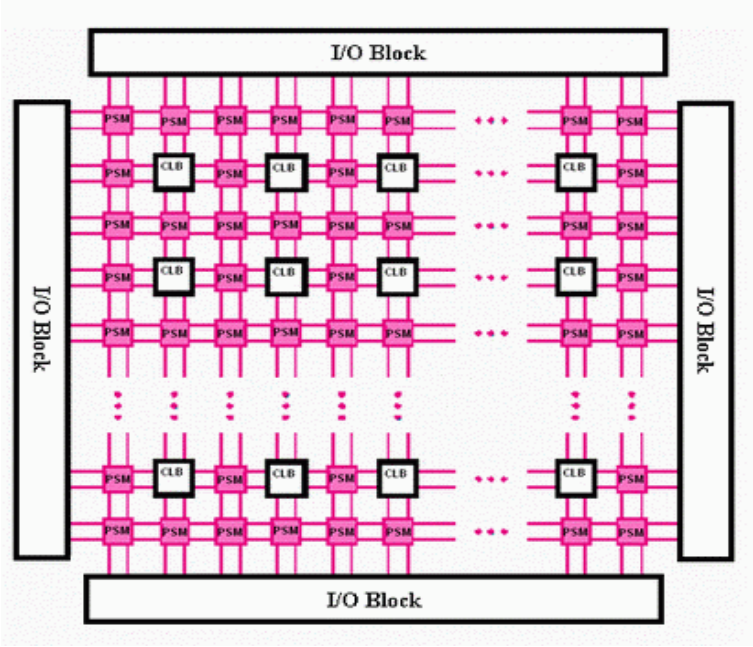
Η Εικόνα 2 δείχνει 7 ομάδες από SMs όπου δύο SMs μοιράζονται μία μονάδα επεξεργασίας υφής (texture unit) και μία μνήμη L1 cache για δεδομένα (data). Το texture unit παραδίδει φιλτραρισμένα αποτελέσματα στον SM με δεδομένο ένα σύνολο συντεταγμένων ενός χάρτη υφής (texture map). Η μνήμη L1 cache χρησιμεύει για να μειώσει τον αριθμό των προσπελάσεων από τα SPs στην κύρια μνήμη (DRAM). Ο αριθμός των επεξεργαστών και ο αριθμός των μνημών μπορούν να προσαρμοσθούν ώστε να σχεδιάζονται συστήματα GPU για διαφορετική απόδοση και διαφορετικό κόστος.

Ένα βασικό χαρακτηριστικό της αρχιτεκτονικής GPU είναι ότι όλα τα SPs ενός SM εκτελούν την ίδια εντολή κάθε χρονική στιγμή, πιθανώς με διαφορετικά δεδομένα. Υλοποιεί δηλ. η GPU ένα σύστημα Single Instruction Multiple Data (SIMD). Αυτός ο περιορισμός βοηθάει στην μείωση του instruction bandwidth και ουσιαστικά κάνει δυνατή την ταυτόχρονη εκτέλεση μεγάλου αριθμού νημάτων (threads) σε κάθε κύκλο μηχανής.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

2.2. Τεχνολογία Επαναδιατασσόμενης Λογικής (Reconfigurable Logic, FPGAs)

Τα ολοκληρωμένα κυκλώματα επαναδιατασσόμενης λογικής (reconfigurable logic) στα οποία ανήκουν και οι FPGAs (Field Programmable Logic Arrays), περιλαμβάνουν συστοιχίες από υπολογιστικά κυκλώματα (CLBs) η λειτουργικότητα των οποίων μπορεί να μεταβληθεί πολλές φορές κατά τον χρόνο ζωής τους [1]. Ουσιαστικά αυτά τα υπολογιστικά κυκλώματα είναι Lookup Tables (LUTs) με N εισόδους (συνήθως το N είναι μεταξύ 4 και 6) και τα οποία μπορούν να υλοποιήσουν οποιαδήποτε συνδυαστική boolean συνάρτηση με N εισόδους. Αυτά τα υπολογιστικά κυκλώματα προγραμματίζονται μέσω configuration bits και συνδέονται μεταξύ τους μέσω πόρων διασύνδεσης (interconnects) που επίσης μπορούν να προγραμματιστούν μέσω configuration bits (Εικόνα 3). Με αυτόν τον τρόπο οποιοσδήποτε αλγόριθμος μπορεί να απεικονισθεί στις συστοιχίες λογικών κυκλωμάτων με το να υπολογίσουμε πρώτα την λογική boolean συνάρτηση του αλγορίθμου, να την απεικονίσουμε στο λογικό κύκλωμα και μετά να διασυνδέσουμε τα λογικά κυκλώματα μεταξύ τους.



Εικόνα 3. Η αρχιτεκτονική της FPGA.

Αν και η βασική αρχιτεκτονική της Εικόνα 3 παραμένει σχεδόν αναλλοίωτη σε διαδοχικές γενιές κυκλωμάτων επαναδιατασσόμενης λογικής,

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

οι σύγχρονες FPGA περιλαμβάνουν επιπλέον διατάξεις που έχουν πολύ πιο συγκεκριμένη λειτουργικότητα. Τέτοιες διατάξεις είναι οι εξής:

- Εσωτερικές μνήμες SRAM. Κάθε FPGA περιέχει έναν μεγάλο αριθμό από μνήμες οι οποίες μπορούν να προσπελασθούν σε έναν κύκλο μηχανής. Οι μνήμες προσφέρουν πολύ μεγάλο bandwidth στην εφαρμογή γιατί γλυτώνουν το σύστημα από χρονοβόρες και κοστοβόρες προσπελάσεις σε εξωτερικές μνήμες DRAM.
- Μονάδες για Digital Signal Processing (DSP). Οι μονάδες αυτές αποτελούνται από κυκλώματα πολλαπλασιαστών και προσθετών (multiply-add) που είναι αφιερωμένα στο να επιτελούν πολύ γρήγορες πράξεις σε εφαρμογές όπως DSP filtering.
- Ολόκληροι επεξεργαστές και περιφερειακά. Οι σύγχρονες FPGAs έχουν φτάσει σε τέτοιο βαθμό ολοκλήρωσης ώστε είναι δυνατόν να περιέχουν ολόκληρα συστήματα όπως επεξεργαστές ARM και ακόμα και GPUs. Για παράδειγμα, η Zynq FPGA από την Xilinx και την ARM συνδυάζει σε ένα τσιπ έναν διπύρηνο επεξεργαστή ARM Cortex-A9 με τα παραδοσιακά υπολογιστικά κυκλώματα (CLBs) [2]. Αυτή η FPGA μπορεί να τρέξει οποιοδήποτε λειτουργικό σύστημα το οποίο έχει τρέξει στον ARM (πχ Linux) και χρησιμοποιεί τα CLBs κυρίως για την δημιουργία επιταχυντών υλικού (hardware accelerators).

2.3. Ετερογενή Συστήματα (Heterogeneous Systems)

Στην σύγχρονη υπολογιστική, ο όρος Ετερογενή Συστήματα (Heterogeneous Systems) αναφέρεται σε υπολογιστικές πλατφόρμες οι οποίες αποτελούνται από ανόμοιους επεξεργαστές. Για παράδειγμα μπορεί να αποτελούνται από συμβατικές πολυπύρηνες CPUs, από GPUs και τώρα τελευταία ακόμα και από FPGAs. Η χρήση διαφορετικών αρχιτεκτονικών σε μία ετερογενή πλατφόρμα έχει επεκταθεί τα τελευταία χρόνια κυρίως λόγω της ανάγκης για αύξηση της ταχύτητας αλλά και της ενεργειακής απόδοσης ενός τέτοιου συστήματος.

Με το να χρησιμοποιούμε ακριβώς τον επεξεργαστή που χρειαζόμαστε για να επιλύσουμε ένα πρόβλημα μπορούμε να επιτύχουμε μείωση της κατανάλωσης ισχύος και ενέργειας στο ποσό που απαιτείται ακριβώς για να επιλυθεί το πρόβλημα. Από την άλλη μεριά όμως, η χρήση διαφορετικών αρχιτεκτονικών δυσκολεύει την ανάπτυξη λογισμικού για ένα τέτοιο σύστημα, μιας και η μηχανικοί λογισμικού θα πρέπει να αναπτύσσουν εφαρμογές χρησιμοποιώντας διαφορετικά εργαλεία για κάθε επεξεργαστή.

Για να επιλυθεί το πρόβλημα του προγραμματισμού εφαρμογών σε ετερογενή συστήματα, η βιομηχανία υπολογιστών δημιούργησε προγραμματιστικά μοντέλα τα οποία μπορούν να χρησιμοποιηθούν για τον προγραμματισμό διαφορετικών αρχιτεκτονικών. Για παράδειγμα, η OpenCL επεκτείνει την γλώσσα C και ορίζει ένα Application Programming Interface (API) το οποίο επιτρέπει σε προγράμματα να τρέχουν σε έναν Host επεξεργαστή (στην CPU) και να δημιουργούν παράλληλους πυρήνες (kernels). Αυτοί οι πυρήνες μπορούν να στέλνονται για εκτέλεση είτε σε μία

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

συμβατική CPU, είτε σε GPUs είτε ακόμα και σε FPGAs. Τα προγράμματα σε OpenCL μπορούν να μεταγλωττιστούν κατά την διάρκεια της εκτέλεσης του προγράμματος (run-time compilation) γεγονός που κάνει τις εφαρμογές σε OpenCL φορητές (portable) για διαφορετικές αρχιτεκτονικές.

3. Μεθοδολογία

3.1. Προγραμματισμός εφαρμογών σε GPU (CUDA, OpenCL)

Το μοντέλο προγραμματισμού της CUDA (ή OpenCL) επεκτείνει τις γλώσσες προγραμματισμού C και C++ ώστε να εκμεταλλεύονται τους υψηλότερους βαθμούς παραλληλίας που υπάρχουν στις GPUs. Από την αρχική κυκλοφορία της CUDA το 2007, πολλοί προγραμματιστές ανέπτυξαν προγράμματα σε CUDA/OpenCL για μια ευρεία γκάμα εφαρμογών μεταξύ των οποίων multimedia, επεξεργασία σεισμικών δεδομένων, ταξινόμηση, αναζήτηση, υπολογιστική χημεία κοκ.

Ο προγραμματιστής CUDA γράφει ένα σειριακό πρόγραμμα που καλεί παράλληλους πυρήνες (kernels) οι οποίοι μπορεί να είναι από απλές συναρτήσεις μέχρι πλήρη προγράμματα. Ένας πυρήνας εκτελείται παράλληλα από ένα σύνολο παράλληλων νημάτων (threads) έτσι ώστε κάθε νήμα να εκτελείται σε ένα SP σε κάθε χρονική στιγμή. Ο προγραμματιστής οργανώνει αυτά τα νήματα σε μία ιεραρχία μπλοκ νημάτων και πλέγματα από μπλόκς. Ένα τέτοιο μπλοκ νημάτων (thread block) είναι ένα σύνολο από ταυτόχρονα νήματα που μπορούν να συνεργαστούν μέσω συγχρονισμού barrier και μέσω κοινής πρόσβασης σε ένα χώρο μνήμης ιδιωτικό για το μπλοκ. Ένα πλέγμα (grid) είναι ένα σύνολο από μπλοκ νημάτων που το καθένα (μπλοκ) μπορεί να εκτελείται ανεξάρτητα και για αυτόν τον λόγο μπορούν να εκτελούνται παράλληλα.

Όταν καλεί έναν πυρήνα, ο προγραμματιστής καθορίζει τον αριθμό των νημάτων ανά μπλοκ και των αριθμό των μπλοκ που αποτελούν το πλέγμα. Κάθε νήμα λαμβάνει έναν μοναδικό αριθμό ταυτότητας νήματος (thread ID) *threadIdx* μέσα στο μπλοκ νημάτων που ανήκει με αρίθμηση 0, 1, 2, … ,blockDim-1, και κάθε μπλοκ νημάτων λαμβάνει έναν μοναδικό αριθμό ταυτότητας μπλοκ (block ID) *blockIdx* μέσα στο πλέγμα που ανήκει. Για ευκολία, τα μπλοκ νημάτων μπορούν να έχουν μέχρι και 3 διαστάσεις και προσπελαύνονται μέσω των πεδίων .x, .y και .z.

Όλα τα παραπάνω μπορούν να φανούν με ένα απλό παράδειγμα στο οποίο θέλουμε να προσθέσουμε δύο πίνακες μίας διάστασης (Εικόνα 4). Οι πίνακες είναι κινητής υποδιαστολής απλής ακρίβειας.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Στο δεξί μέρος της Εικόνα 4 ο κώδικας CUDA είναι ο πυρήνας που εκτελεί κάθε νήμα (thread). Όταν ο προγραμματιστής καλεί τον πυρήνα vadd από τον κώδικα του Host, το σύστημα χρόνου εκτέλεσης (run time system) της CUDA δημιουργεί έναν αριθμό νημάτων και κάθε νήμα εκτελεί την δική του εκδοχή του πυρήνα. Σε κάθε νήμα αντιστοιχεί ένα διαφορετικό idx και κάθε νήμα επεξεργάζεται και διαφορετικό κομμάτι των πινάκων a, b, και c. Η εκτέλεση ενός thread ανατίθεται σε ένα συγκεκριμένο SP στην GPU.

```
void add(int* a,
        int* b,
        int* c) {
for (int idx=0;idx<sizeof(a); idx++)
   c[idx] = a[idx] + b[idx];
}
```

```
__kernel void vadd(
        __global int* a,
        __global int* b,
        __global int* c) {
   idx = threadIDx.x +
        blockDim.x *blockIdx.x;
   c[idx] = a[idx] + b[idx];
}
```

Εικόνα 4. Κώδικας πρόσθεσης δύο διανυσμάτων σε C (αριστερά) και σε CUDA (δεξιά)

3.2. Προγραμματισμός εφαρμογών σε FPGAs

Προγραμματισμός με HDL

Η ανάπτυξη κυκλωμάτων σε FPGAs είναι μια επίπονη και χρονοβόρα διαδικασία επειδή απαιτεί από τον σχεδιαστή γνώσεις λεπτομερείς γνώσεις υλικού (hardware) και αρχιτεκτονικής. Ο σχεδιαστής θα πρέπει να αναλύσει τις προδιαγραφές του προβλήματος που πρέπει να αναλύσει, να χωρίσει το πρόβλημα σε μικρότερα προβλήματα και να υλοποιήσει το κάθε ένα από αυτά με την χρήση γλωσσών περιγραφής υλικού (Hardware Description Languages, HDL) όπως η Verilog και η VHDL. Οι γλώσσες αυτές περιγράφουν την αρχιτεκτονική του συστήματος με αρκετά μεγάλη λεπτομέρεια σε επίπεδο κύκλων ρολογιού και ως εκ τούτου είναι ακατάλληλες για χρήση από προγραμματιστές λογισμικού που προτιμούν να μην εισέρχονται σε τόσο μεγάλες λεπτομέρειες και σε τόσο χαμηλό επίπεδο κοντά στο hardware. Η υλοποίηση ενός αλγορίθμου σε FPGA απαιτεί τόσο αυτόν κάθε αυτό το σχεδιασμό του αλγορίθμου σε κάποια γλώσσα περιγραφής υλικού όσο και καλή γνώση του τρόπου με τον οποίο τα δεδομένα της εφαρμογής προσπελαύνονται.

Η διαδικασία σχεδίασης ενός αλγορίθμου σε υλικό διαφέρει κατά πολύ από την ανάπτυξη εφαρμογών λογισμικού. Κατά την πλειοψηφία θεωρείται πιο δύσκολη και περισσότερο στρυφνή εφόσον δεν συγχωρεί πολλές παραβλέψεις που κάνουν οι προγραμματιστές λογισμικού. Μία πολύ σημαντική διαφορά μεταξύ λογισμικού και υλικού είναι πως το δεύτερο είναι εν γένει παράλληλο. Έτσι για να γίνει ακόμα καλύτερη εκμετάλλευση των

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

δυνατοτήτων ενός προσαρμοσμένου hardware accelerator χρειάζεται η υλοποίηση να χαρακτηρίζεται από παραλληλία πράξεων.

Η σημασία της γνώσης σχετικά με τον τρόπο με τον οποίο τα δεδομένα μιας εφαρμογής προσπελαύνονται γίνεται εμφανής αν σκεφτεί κανείς πως η μεταφορά και διαχείριση τμημάτων μνήμης παίζει πολύ μεγάλο ρόλο στην απόδοση ενός αλγορίθμου. Στην περίπτωση των hardware accelerators υπό την μορφή FPGA το πρόβλημα γίνεται μεγαλύτερο επειδή η διαθέσιμη μνήμη είναι προδιαγεγραμμένη και σε σύγκριση με ένα τυπικό υπολογιστή κατά πολύ μικρότερη. Έτσι σε περιπτώσεις στις οποίες δεν γίνεται εκμετάλλευση του μοτίβου προσπέλασης δεδομένων, ένας αλγόριθμος μπορεί φαινομενικά να απαιτεί περισσότερη μνήμη από όση είναι διαθέσιμη σε μία δεδομένη FPGA και έτσι να μην είναι δυνατή η σύνθεσή του. Αν όμως αποδειχθεί με μαθηματικό τρόπο πως από η πραγματική απαίτηση μνήμης είναι μικρότερη από τη φαινομενική τότε γίνεται εφικτή η σύνθεση του σε μία FPGA.

Πέρα από το παραπάνω πρακτικό μέρος η γνώση που σχετίζεται με το μοτίβο προσπέλασης μνήμης μπορεί να χρησιμοποιηθεί για περαιτέρω βελτίωση της απόδοσης ενός αλγορίθμου. Εφόσον αναγνωρισθούν τα στοιχεία ενός πίνακας που διαβάζονται από κάποιο κομμάτι κώδικα υλοποιημένο ως hardware accelerator τότε μπορούμε να μειώσουμε το κόστος μεταφοράς μνήμης από τον επεξεργαστή στον hardware accelerator με το να μεταφέρουμε μόνο τα χρήσιμα στοιχεία της μνήμης που αντιστοιχεί στον δεδομένο πίνακα. Αντίστοιχη διαδικασία χρησιμοποιείται για την μεταφορά των αποτελεσμάτων από τον hardware accelerator πίσω στον επεξεργαστή, με το να αναγνωρίζονται και να μεταφέρονται μόνο κομμάτια μνήμης που αντιστοιχούν σε αποτελέσματα που παρήγαγε ο hardware accelerator κατά την εκτέλεσή του.

Προγραμματισμός με C/OpenCL – High Level Synthesis tools

Για να γίνουν οι FPGAs ελκυστικές στο επικρατούσα υπολογιστική (mainstream computing) ή και στην υπολογιστική υψηλών επιδόσεων (high performance computing) θα πρέπει να χρησιμοποιήσουν για τον προγραμματισμό τους μοντέλα πλησιέστερα στα μοντέλα ανάπτυξης λογισμικού. Αυτό γίνεται επιτακτική ανάγκη λόγω της δυνατότητας των FPGAs να ενσωματώνονται σε συστήματα που αποτελούνται από CPU και GPUs και να προσφέρουν υψηλότερη απόδοση και ενεργειακή αποδοτικότητα.

Για λόγους αύξησης της αποδοτικότητας των σχεδιαστών αλλά κυρίως λόγω της επιθυμίας να επεκταθεί ο αριθμός των σχεδιαστών FPGA και σε προγραμματιστές λογισμικού, έχει δημιουργηθεί μια μεγάλη ώθηση τα τελευταία χρόνια σε εργαλεία σχεδίασης υψηλού επιπέδου (High Level Synthesis Tools). Τα εργαλεία αυτά δίνουν την ευκαιρία στον χρήστη να εκφράσει το κύκλωμα που θέλει να υλοποιήσει σε μια FPGA σε κάποια γλώσσα προγραμματισμού υψηλού επιπέδου όπως C, C++, OpenCL και να αναλάβουν την μετάφραση του προγράμματος αυτού σε γλώσσα HDL, Verilog

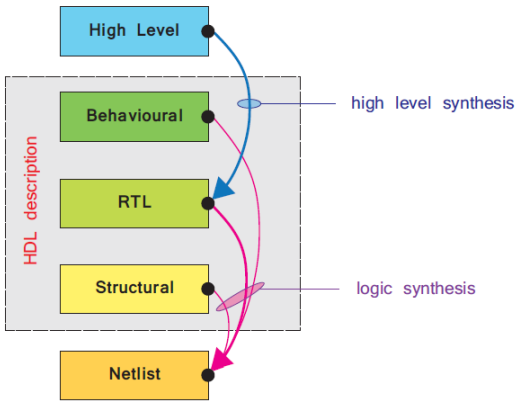
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ή VHDL. Εκτός από τον ίδιο τον αλγόριθμο σε κάποια γλώσσα σαν την C, ο σχεδιαστής θα πρέπει να καθορίσει και διάφορους περιορισμούς στο τελικό του κύκλωμα όπως χρόνο απόκρισης (latency), throughput, συχνότητα ρολογιού, αλλά και το μέγεθος του τελικού κυκλώματος σε αριθμό βασικών συστατικών στοιχείων. Αυτοί οι περιορισμοί καθοδηγούν ένα High Level Synthesis Tool στο να δημιουργήσει το κατάλληλο τελικό κύκλωμα σε HDL (Εικόνα 5)

Στην συνέχεια, μια σειρά από εργαλεία CAD μετατρέπουν το HDL κύκλωμα σε λογικές πύλες (gate level netlist) και από εκεί το τοποθετούν στα υπάρχοντα υπολογιστικά κυκλώματα (CLBs) αφού το διασυνδέσουν μέσω των διαύλων διασύνδεσης (interconnects). Το τελικό αποτέλεσμα είναι ένα configuration file το οποίο περιέχει όλα τα configuration bits που διαμορφώνουν την λειτουργικότητα των CLBs και των διασυνδέσεων.

Βασιζόμενοι σε όλες τις προηγούμενες παρατηρήσεις, είναι σκοπός μας να εξετάσουμε την δυνατότητα χρήσης γλωσσών όπως η OpenCL για την απεικόνιση αλγορίθμων σε FPGAs. Πηγαίνοντας ένα βήμα παραπέρα, θα χρησιμοποιήσουμε τον ίδιο κώδικα OpenCL ή C για εκτέλεση σε όλες τις υπολογιστικές πλατφόρμες που θα χρησιμοποιηθούν για το project. Αυτό θα γίνει δυνατόν μέσω λογισμικού CAD που αναπτύσσεται στο Πανεπιστήμιο Θεσσαλίας και που υλοποιεί τεχνικές High Level Synthesis για την δημιουργία επιταχυντών υλικού από προγράμματα OpenCL [3]. Στα πλαίσια του project, το SOpenCL θα βελτιωθεί ώστε να παράγει βέλτιστο υλικό για τους συγκεκριμένους αλγόριθμους που θα παραχθούν στα πλαίσια του project.

Εκτός του SOpenCL θα δοκιμασθούν και εμπορικά εργαλεία High Level Synthesis όπως το Vivado HLS το οποίο αποτελεί και το στάνταρντ εργαλείο για Xilinx FPGAs. Το Vivado HLS δέχεται σαν είσοδο ένα πρόγραμμα γραμμένο σε C ή C++ καθώς και οδηγίες από τον προγραμματιστή μέσω #pragmas που υποδεικνύουν στο εργαλείο τις βελτιστοποιήσεις στην απόδοση ή στο μέγεθος του παραγόμενου υλικού που θα πρέπει να γίνουν.



Εικόνα 5. High Level Synthesis και Logic Synthesis

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

4. Βιβλιογραφία

[1] Katherine Compton and Scott Hauck. Reconfigurable computing: a survey of systems and software. *ACM Computing Surveys.* Vol. 34, no. 2. June 2002.

[2] L.H. Crockett, R.A. Elliot. The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC. 2014. Strathclyde Academic Media. Available at www.zynqbook.com.

[3] Muhsen Owaida, Nikolaos Bellas, Christos Antonopoulos, Konstantis Daloukas, Charalambos Antoniadis. Massively Parallel Programing Models Used as Hardware Description Languages: The OpenCL Case. *International Conference on Computer-Aided Design (ICCAD),* November 6-10, 2011, San Jose, CA

[4] David A. Patterson and John L. Hennessy, Computer Organization and Design - The Hardware / Software Interface, (Revised 4th Edition). The Morgan Kaufmann Series in Computer Architecture and Design. Academic Press, 2012

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης