

Επεξεργαστές Ειδικού Συνόλου Εντολών

Για την βελτίωση των επιδόσεων των ενσωματωμένων συστημάτων πέρα από τις τεχνικές χρησιμοποίησης επεξεργαστών γενικού σκοπού, στα πλαίσια της σχεδίασης σε επίπεδο συστήματος (System Level Design), αναπτύσσονται επεξεργαστές ειδικού συνόλου εντολών. Οι προγραμματιζόμενοι επεξεργαστές Ειδικού Συνόλου Εντολών παρέχουν την δυνατότητα για πιο οικονομική λύση στην ανάπτυξη υπολογιστικά απαιτητικών ενσωματωμένων συστημάτων. Οι λύσεις ASIC έχουν τη δυνατότητα να επιτυγχάνουν υψηλή ταχύτητα επεξεργασίας σε συνδυασμό με χαμηλή κατανάλωση ισχύος, χωρίς όμως να είναι εφικτός ο επαναπρογραμματισμός τους. Ένα ASIC εκτελεί μόνο τους αλγορίθμους οι οποίοι είχαν προβλεφθεί κατά το σχεδιασμό του. Επίσης ο χρόνος ανάπτυξης τους είναι απαγορευτικός για όποιες μεταγενέστερες μετατροπές. Οι επεξεργαστές γενικού σκοπού έχουν την δυνατότητα προγραμματισμού τους, πλην όμως παρουσιάζουν πολλές φορές χαμηλές επιδόσεις, σε σχέση με τις απαιτήσεις των ενσωματωμένων συστημάτων. Η υποστήριξη ειδικών εντολών επέκτασης από τον επεξεργαστή επιφέρει πολλές βελτιώσεις στα χαρακτηριστικά του επεξεργαστή. Κυρίως μπορεί να αυξήσει την επεξεργαστική του ισχύ. Αυτό μπορεί να συμβεί μειώνοντας την καθυστέρηση των σταδίων διοχέτευσης, τα οποία προκαλούνται από

ασυνέπεια δεδομένων

(data hazard). Επίσης μειώνεται το μέγεθος του κώδικα του προγράμματος, αυτό έχει ως αποτέλεσμα μικρότερες απαιτήσεις σε μνήμη εντολών, αλλά και μικρότερη κατανάλωση ενέργειας, αφού οι προσπελάσεις στην μνήμη είναι αρκετά ενεργοβόρες

Έτσι σχεδιάζονται επεξεργαστές με βάση το λογισμικό που πρόκειται να εκτελέσουν, έχοντας υπόψη τα ιδιαίτερα χαρακτηριστικά των εφαρμογών, ώστε να ικανοποιηθούν οι απαιτήσεις σε χρόνο, ισχύ και επιφάνεια υλικού. Μία συμπληρωματική τεχνική είναι η προσθήκη ειδικών εντολών για την εκτέλεση των πιο απαιτητικών τμημάτων του κώδικα, ή η προσθήκη επαναδιαμορφούμενων συνεπεξεργαστών που μπορούν να χρησιμοποιηθούν συμπληρωματικά. Τέτοιες λύσεις υπάρχουν από την

ARM

(

OptimoDE

), την

MIPS

(

CorExtend

) και σε κατασκευαστές

FPGAs

.

Διαδικασία για την ανάπτυξη ενός επεξεργαστή Ειδικού Συνόλου Εντολών

Για να φτάσουμε από τον κώδικα μιας εφαρμογής στην σχεδίαση ενός επεξεργαστή Ειδικού Συνόλου Εντολών ακολουθούνται τα παρακάτω βήματα:

1. Καθορίζονται οι περιορισμοί που έχουμε σε χρόνο εκτέλεσης, κατανάλωση ισχύος, επιφάνεια υλικού. Τις περισσότερες φορές δεν είναι δυνατό να ικανοποιηθούν ταυτόχρονα όλοι οι στόχοι, έτσι επιλέγουμε τους πιο σημαντικούς δίνοντας κατάλληλες προτεραιότητες.

2. Αναλύεται ο κώδικας με σκοπό τα συχνά επαναλαμβανόμενα τμήματα να υλοποιηθούν με ειδικές εντολές ή επιταχυντή υλικού. Στο στάδιο της ανάλυσης του κώδικα από την γλώσσα υψηλού επιπέδου C, εξετάζουμε θέματα όπως την παραλληλία της εφαρμογής, συχνότητα εκτέλεσης τμημάτων του κώδικα, εύρος μεταβλητών, τύπους δεδομένων. Σε αυτές τις αναλύσεις στηρίζονται πολλά εργαλεία παραγωγής κώδικα

HDL

από κώδικα

C

, όπως ο

C

to

Hardware

Compiler

της

ALTERA

. Πολλές φορές παράγεται ένα ενδιάμεσο στάδιο μοντελοποίησης

TLM

(

Transaction

Level

Modeling

) που μας επιτρέπει την αναπαράσταση του συστήματος σε επίπεδο επικοινωνίας των επεξεργαστικών μονάδων με σκοπό την σχεδίαση του συστήματος από επίπεδο

System

Level

Design

.

3. Ανάλυση των πιθανών αρχιτεκτονικών λύσεων από πλευράς επίτευξης των στόχων. Ανάλογα με τα συμπεράσματα του προηγούμενου σταδίου καθορίζονται τα αρχιτεκτονικά χαρακτηριστικά του επεξεργαστή όπως ο αριθμός των πολλαπλασιαστών, ο αριθμός των σταδίων διοχέτευσης, ο αριθμός των παράλληλων επεξεργαστικών μονάδων, η προσθήκη ή όχι επιταχυντή υλικού.

4. Αναπτύσσονται οι ειδικές εντολές και λειτουργικές μονάδες. Από τον κώδικα της εφαρμογής εξάγεται το CDFG (Control-Data Flow Graph) διάγραμμα, έτσι έχουμε εικόνα για τις εξαρτήσεις ελέγχου και δεδομένων, τους τύπους δεδομένων, τις τοπικές και καθολικές μεταβλητές, το μέγεθος του βασικού μπλοκ, ο αριθμός των λειτουργιών πολλαπλασιασμού και συσσώρευσης, στοιχεία που μας βοηθούν να εντοπίσουμε τα τμήματα του κώδικα που μπορούν να υλοποιηθούν με ειδικές εντολές ή με επιταχυντή υλικού.

5. Γίνεται η σύνθεση και η δοκιμή του υλικού του επεξεργαστή. Εδώ είτε επιλέγουμε εντολές από μια βιβλιοθήκη είτε γίνεται η σύνθεσή τους από την αρχή. Η επιλογή εξαρτάται από τους στόχους που έχουν τεθεί στο στάδιο 1, έχοντας υπόψη την επίτευξη κώδικα με τον μικρότερο αριθμό εντολών, την μεγαλύτερη επαναχρησιμοποίηση των ειδικών εντολών και με μικρό κύκλο εκτέλεσης. Εδώ μπορεί να εξεταστεί η περίπτωση ειδικών εντολών πολλαπλών ορισμάτων εξόδου, όπως και πολλαπλών κύκλων εκτέλεσης. Η δυνατότητα επέκτασης ενός επεξεργαστή σε FPGA όπως ο NIOS II μας επιτρέπει την εισαγωγή ειδικών εντολών απλού κύκλου, πολλαπλού κύκλου ή με διοχέτευση, υλοποίηση του πολλαπλασιαστή σε υλικό ή σε λογισμικό.

Γίνεται η σύνθεση και η δοκιμή του λογισμικού. Εδώ με βάση τις νέες εντολές που έχουμε προσθέσει ή τα τμήματα κώδικα που εξυπηρετείται από υλικό, ο νέος κώδικας μεταγλωττίζεται ξανά για το σύνολο εντολών που παρήχθησαν. Τα εργαλεία ανάπτυξης λογισμικού του επεξεργαστή πρέπει να μας επιτρέπουν την εισαγωγή νέων εντολών. Πολλές τεχνολογίες FPGAs υποστηρίζουν soft instruction επεξεργαστές, όπου μία ή περισσότερες soft εντολές μπορούν να υλοποιηθούν χρησιμοποιώντας επαναπροσδιοριζόμενους πόρους, σε ένα μονό chip. Επεξεργαστές όπως ο Xilinx Microblaze και ο Altera Nios είναι διαθέσιμοι από τους FPGA κατασκευαστές. Συχνά αυτοί οι επεξεργαστές υποστηρίζουν προσαρμογή (customization) των πόρων και ειδικές εντολές (custom instructions). Οι ειδικές εντολές έχουν δύο σημαντικά πλεονεκτήματα. Πρώτον, μειώνουν τον χρόνο για το instruction fetch και decode, δεδομένου ότι μια ειδική εντολή αντικαθιστά πολλές κανονικές εντολές. Δεύτερον, επιπλέον πόροι μπορεί να ανατεθούν σε μια ειδική εντολή για να βελτιωθεί η απόδοση.