ΚΕΦΑΛΑΙΟ 18

ΥΠΟΛΟΓΙΣΤΕΣ ΠΟΛΛΑΠΛΩΝ ΠΥΡΗΝΩΝ

18.1 Ζητήματα Απόδοσης του Υλικού

Αύξηση του Παραλληλισμού

Κατανάλωση Ισχύος

18.2 Ζητήματα Απόδοσης του Λογισμικού

Λογισμικό σε Πολλαπλούς Πυρήνες

Παράδειγμα Εφαρμογής: Το Λογισμικό Παιχνιδιών της Valve

18.3 Οργάνωση Πολλαπλών Πυρήνων

18.4 Οργάνωση Πολλαπλών Πυρήνων της Intel

Ο Επεξεργαστής Intel Core Duo

Ο Επεξεργαστής Intel Core i7

18.5 Ο Επεξεργαστής ARM11 MPCore

Διαχείριση Διακοπών

Συνοχή της Κρυφής Μνήμης

18.6 Προτεινόμενη Μελέτη και Διαδικτυακοί Τόποι

18.7 Όροι-κλειδιά, Ερωτήσεις Επανάληψης και Προβλήματα

ΒΑΣΙΚΑ ΣΗΜΕΙΑ

- Ένας υπολογιστής πολλαπλών πυρήνων ή chip πολυεπεξεργαστών, περιέχει δύο ή περισσότερους επεξεργαστές πάνω στο ίδιο chip.
- Η χρήση ολοένα και πιο πολύπλοκων chips ενός επεξεργαστή έφτασε σε ένα όριο, λόγω των ζητημάτων απόδοσης του υλικού, συμπεριλαμβανομένων και των ορίων που αφορούν τον παραλληλισμό σε επίπεδο εντολής και την κατανάλωση ισχύος.
- Από την άλλη, η αρχιτεκτονική πολλαπλών πυρήνων αποτελεί μία πρόκληση για τις εταιρίες που αναπτύσσουν λογισμικό, προκειμένου να εκμεταλλευτούν τη δυνατότητα της πολλαπλής νημάτωσης σε πολλαπολούς πυρήνες.
- Οι βασικές μεταβλητές μίας οργάνωσης πολλαπλών πυρήνων είναι το πλήθος των επεξεργαστών πάνω στο στο chip, το πλήθος των επιπέδων της κρυφής μνήμης, και ο βαθμός στον οποίο αυτή μοιράζεται.
- Ένα άλλο σχεδιαστικό ζήτημα ενός συστήματος πολλαπλών πυρήνων, είναι αν οι διαφορετικοί πυρήνες θα είναι υπερβαθμωτοί ή αν θα υλοποιούν ταυτόχρονη πολυνημάτωση.

Ένας υπολογιστής **πολλαπλών πυρήνων**, ο οποίος είναι γνωστός και ως **chip πολυεπεξεργαστών**, συνδυάζει δύο ή περισσότερους επεξεργαστές (οι οποίοι ονομάζονται πυρήνες) σε ένα κομμάτι πυριτίου. Τυπικά, κάθε πυρήνας αποτελείται από όλα τα στοιχεία ενός ανεξάρτητου επεξεργαστή, όπως είναι οι καταχωρητές, η αριθμητική και λογική μονάδα, το υλικό διασωλήνωσης, και η μονάδα ελέγχου, και επιπλέον, κρυφές μνήμες εντολών και δεδομένων Επιπέδου 1. Επιπλέον των πολλαπλών πυρήνων, τα σύγχρονα chips πολλαπλών πυρήνων συμπεριλαμβάνουν και κρυφή μνήμη Επιπέδου 2, μερικές φορές και Επιπέδου 3.

Αυτό το κεφάλαιο παρουσιάζει μία γενική θεώρηση των συστημάτων πολλαπλών πυρήνων. Θα ξεκινήσουμε εξετάζοντας τους παράγοντες που καθορίζουν την απόδοση του υλικού και οδήγησαν στην ανάπτυξη των υπολογιστών πολλαπλών πυρήνων, αλλά και τις προκλήσεις που αφορούν την ανάπτυξη λογισμικού κατάλληλου για να εκμεταλλευτεί την ισχύ ενός συστήματος πολλαπλών πυρήνων. Στη συνέχεια, θα εξετάσουμε την οργάνωση των πολλαπλών πυρήνων. Τέλος, θα παρουσιάσουμε δύο παραδείγματα προϊόντων πολλαπλών πυρήνων, της Intel και της ARM.

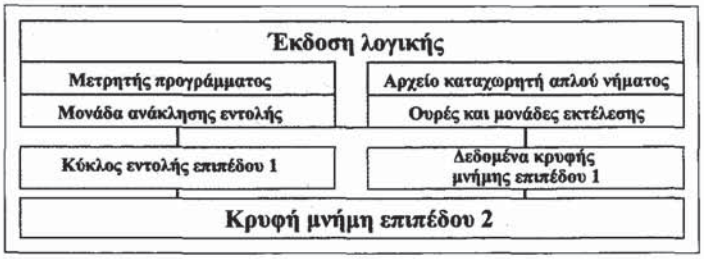
18.1 ΖΗΤΗΜΑΤΑ ΑΠΟΔΟΣΗΣ ΤΟΥ ΥΛΙΚΟΥ

Όπως παρουσιάσαμε στο Κεφάλαιο 2, τα συστήματα πολλαπλών επεξεργαστών εμφάνισαν εκθετική αύξηση στην αποδοτικότητα της εκτέλεσης των εντολών, μέσα σε διάστημα δεκαετιών. Το Σχήμα 2.12 δείχνει ότι αυτή η αύξηση οφείλεται μερικώς σε βελτιώσεις της οργάνωσης του επεξεργαστή πάνω στο chip και μερικώς στην αύξηση της συχνότητας του ρολογιού.

Αύξηση του Παραλληλισμού

Οι αλλαγές στην οργάνωση της σχεδίασης του επεξεργαστή επικεντρώθηκαν κυρίως στην αύξηση του παραλληλισμού σε επίπεδο εντολής, έτσι ώστε να ολοκληρώνεται όσο το δυνατόν περισσότερη εργασία σε έναν κύκλο ρολογιού. Χρονολογικά, αυτές οι αλλαγές συμπεριλαμβάνουν τα ακόλουθα (Σχήμα 18.1):

- **Διασωλήνωση:** Οι εντολές εκτελούνται μέσα από μία δομή διασωλήνωσης, έτσι ώστε ενώ μία



(α) Υπερβαθμωτός

(β) Ταυτόχρονης πολυνημάτωσης

(γ) Πολλαπλών πυρήνων

Σχήμα 18.1: Εναλλακτικές μορφές οργάνωσης ενός chip

εντολή εκτελείται σε ένα στάδιο, μία άλλη να εκτελείται σε ένα άλλο στάδιο.

- **Υπερβαθμωτή οργάνωση:** Κατασκευάστηκαν πολλαπλές διασωληνώσεις με τη βοήθεια αντιγράφων των πόρων εκτέλεσης. Το γεγονός αυτό επέτρεψε την παράλληλη εκτέλεση εντολών σε παράλληλες διασωληνώσεις, με την προϋπόθεση ότι δεν υπάρχουν κίνδυνοι.
- **Ταυτόχρονη πολυνημάτωση:** Αντιγράφηκαν οι τράπεζες καταχωρητών, ώστε τα πολλαπλά νήματα να έχουν τη δυνατότητα να μοιράζονται τους πόρους της διασωλήνωσης.

Για κάθε μία από αυτές τις καινοτομίες, οι σχεδιαστές προσπάθησαν να αυξήσουν την απόδοση του συστήματος, αυξάνοντας την πολυπλοκότητα. Στην περίπτωση της διασωλήνωσης, οι απλές δομές διασωλήνωσης τριών σταδίων αντικαταστάθηκαν αρχικά από δομές 5 σταδίων, και έπειτα από δομές πολύ περισσότερων σταδίων, ενώ ορισμένες υλοποιήσεις έχουν φτάσει να περιέχουν ακόμη και περισσότερα

από 12 στάδια. Υπάρχει ένα πρακτικό όριο όσον αφορά το πόσο μακριά μπορεί να φτάσει αυτή η τάση, επειδή όταν υπάρχουν περισσότερα στάδια, υπάρχει ανάγκη για περισσότερα λογικά κυκλώματα, περισσότερες διασυνδέσεις, και περισσότερα σήματα ελέγχου. Με την υπερβαθμωτή οργάνωση, είναι δυνατόν να αυξηθεί η απόδοση αν αυξηθεί το πλήθος των παράλληλων διασωληνώσεων. Ξανά, υπάρχουν μειωμένα ωφέλη όσο αυξάνεται το πλήθος των διασωληνώσεων. Αυτό συμβαίνει γιατί απαιτούνται περισσότερα λογικά κυκλώματα για τη διαχείριση των κινδύνων και για την τοποθέτηση των πόρων εκτέλεσης των εντολών. Τελικά, η εκτέλεση ενός απλού νήματος μπορεί να φτάσει σε ένα σημείο όπου οι κίνδυνοι και οι εξαρτήσεις δεδομένων αποτρέπουν την πλήρη χρήση των διαθέσιμων διασωληνώσεων. Το ίδιο σημείο μειωμένων κερδών θα συναντήσουμε και στην ταυτόχρονη πολυνημάτωση, καθώς η πολυπλοκότητα της διαχείρισης πολλαπλών νημάτων μέσα σε ένα σύνολο διασωληνώσεων περιορίζει το πλήθος των νημάτων και διασωληνώσεων που είναι δυνατόν να χρησιμοποιηθούν αποτελεσματικά.

Το Σχήμα 18.2, το οποίο υπάρχει στην αναφορά [OLUK05], είναι διδακτικό σε αυτό το σημείο. Το γράφημα που βρίσκεται στο πάνω μέρος δείχνει την εκθετική αύξηση της απόδοσης των επεξεργαστών της Intel μέσα στην πορεία των χρόνων.[1] Το μεσαίο γράφημα δημιουργήθηκε συνδυάζοντας τα δημοσιευμένα σχήματα της ίδιας της Intel και τις συχνότητες ρολογιού των επεξεργαστών, ώστε να δοθεί ένα μέτρο του βαθμού στον οποίο η βελτίωση της απόδοσης οφείλεται στην αυξημένη εκμετάλλευση της παραλληλοποίησης σε επίπεδο εντολών. Υπάρχει μία επίπεδη περιοχή της καμπύλης στα τέλη της δεκαετίας του '80, πριν από την εκτεταμμένη εκμετάλλευση του παραλληλισμού. Η περίοδος αυτή ακολουθείται από μία υπερβολική αύξηση, καθώς οι σχεδιαστές είχαν τη δυνατότητα να εκμεταλλευτούν ολοένα και περισσότερο τη διασωλήνωση, τις υπερβαθμωτές τεχνικές, και την ταυτόχρονη πολυνημάτωση. Ωστόσο, ξεκινώντας περίπου από το 2000, εμφανίζεται μία ακόμη επίπεδη περιοχή στην καμπύλη, καθώς φτάσαμε στα όρια της αποτελεσματικής εκμετάλλευσης του παραλληλισμού σε επίπεδο εντολής.

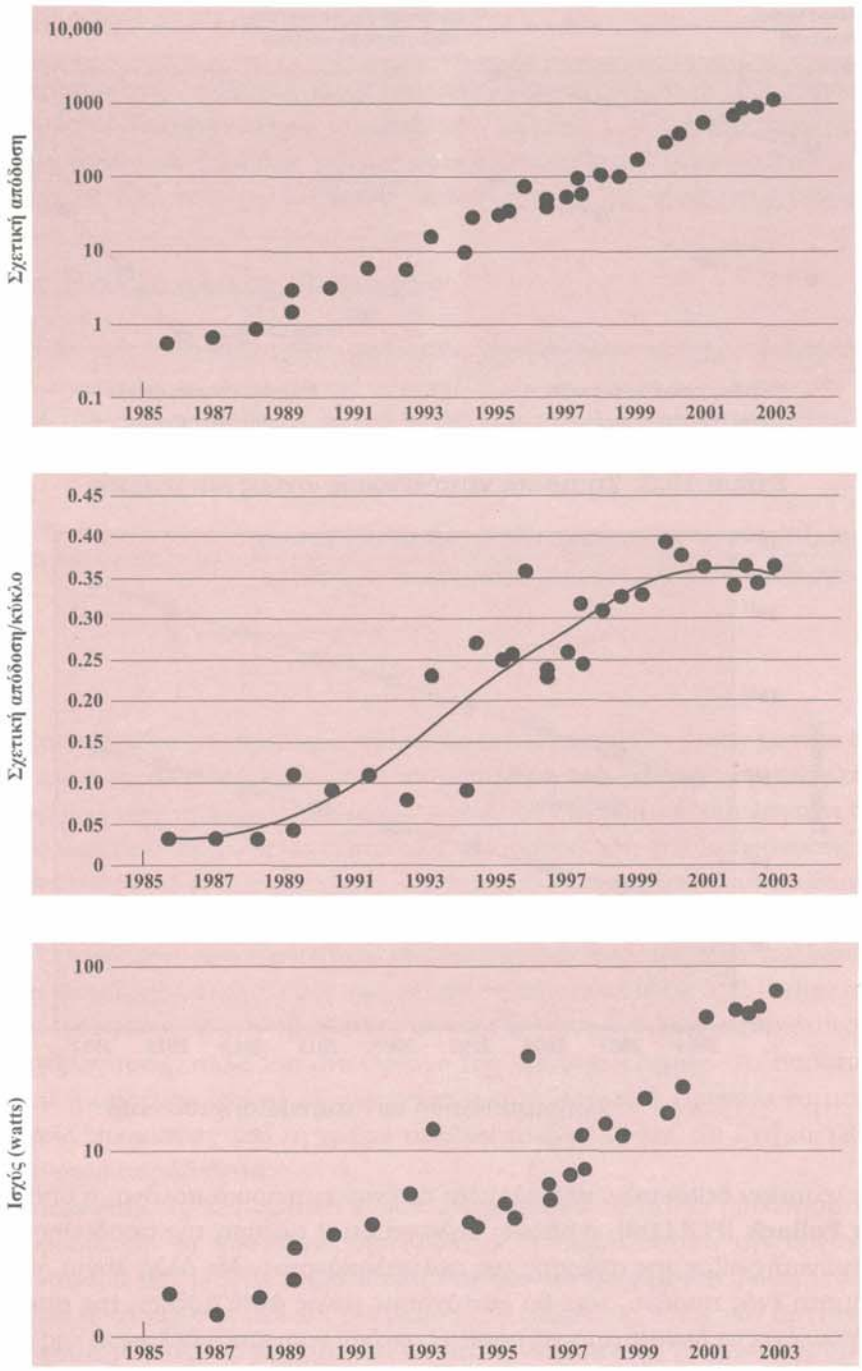
Υπάρχει ένα αντίστοιχο σύνολο προβλημάτων, το οποίο επικεντρώνεται στη σχεδίαση και υλοποίηση του chip του υπολογιστή. Η αύξηση της πολυπλοκότητας για να αντιμετωπιστούν όλα τα ζητήματα λογικής τα οποία σχετίζονταν με τις πολύ μεγάλες διασωληνώσεις, τις πολλαπλές υπερβαθμωτές διασωληνώσεις, και τις πολλαπλές τράπεζες καταχωρητών ταυτόχρονων πολλαπλών νημάτων, σημαίνει ότι η επιπλέον περιοχή του chip χρησίμευε για την υλοποίηση λογικών σημάτων συντονισμού και μεταφοράς. Αυτό το γεγονός ηύξανε τη δυσκολία της σχεδίασης, κατασκευής, και αποσφαλμάτωσης των chips. Η αυξημένη δυσκολία η οποία υπάρχει στη λογική σχεδίασης του επεξεργαστή, είναι ένας από τους λόγους για τους οποίους ένα ολοένα και αυξανόμενο μέρος της επιφάνειας του chip του επεξεργαστή είναι αφιερωμένο στην απλούστερη λογική της μνήμης. Τα ζητήματα κατανάλωσης ισχύος τα οποία παρατίθενται αμέσως μετά, παρέχουν έναν ακόμη λόγο.

Κατανάλωση Ισχύος

Για να διατηρηθεί η τάση της αυξανόμενης απόδοσης καθώς αυξάνεται το πλήθος των transistors ανά chip, οι σχεδιαστές κατέφυγαν σε πιο εκλεπτυσμένες μορφές σχεδίασης των επεξεργαστών (διασωλήνωση, υπερβαθμωτοί επεξεργαστές, ταυτόχρονη πολυνημάτωση) και σε υψηλότερες συχνότητες ρολογιού. Δυστυχώς, οι απαιτήσεις σε ισχύ αυξήθηκαν εκθετικά, καθώς αυξανόταν η πυκνότητα του chip και η συχνότητα του ρολογιού. Αυτό το γεγονός απεικονίζεται στο κατώτερο μέρος του Σχήματος 18.2.

Ένας τρόπος να ελεγχθεί η πυκνότητα της ισχύος, είναι να χρησιμοποιηθεί μεγαλύτερο μέρος της επιφάνειας του chip για την κρυφή μνήμη. Τα transistors της μνήμης είναι μικρότερα και έχουν πυκνότητα ισχύος με τάξη μεγέθους μικρότερη από εκείνη των λογικών κυκλωμάτων του επεξεργαστή (δείτε το Σχήμα 18.3α). Όπως δείχνει το Σχήμα 18.3β ([BORK03]), το ποσοστό της επιφάνειας του chip το οποίο είναι αφιερωμένο στη μνήμη έχει ξεπεράσει το 50%, καθώς έχει αυξηθεί η πυκνότητα των transistors του chip.

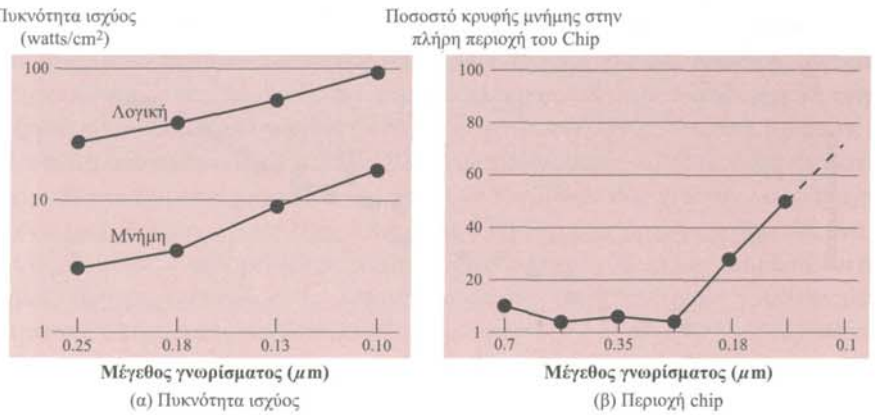
[1] Τα δεδομένα βασίζονται σε δημοσιευμένα σχήματα της ίδιας της Intel, κανονικοποιημένα για διάφορα μοντέλα.



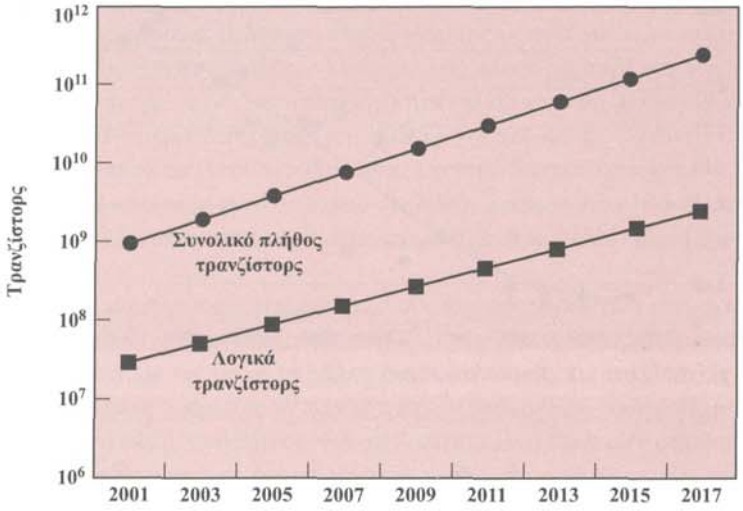
Σχήμα 18.2: Ορισμένες τάσεις που αναπτύχθηκαν στο υλικό της Intel

Το Σχήμα 18.4 ([BORK07]), δείχνει που οδηγεί η τάση της κατανάλωσης ισχύος. Περίπου το 2015, αναμένουμε να δούμε chips μικροεπεξεργαστών με περίπου 100 δισεκατομμύρια transistors σε μία επιφάνεια πυριτίου 300 mm^2. Υποθέτοντας ότι ένα ποσοστό της επιφάνειας περίπου 50%-60% είναι αφιερωμένο στη μνήμη, το chip θα υποστηρίζει κρυφή μνήμη ίση περίπου με 100MB και θα αφήνει περίπου 1 δισεκατομμύριο transistors διαθέσιμα για την υλοποίηση λογικών κυκλωμάτων.

Ένα βασικό ζήτημα σχετίζεται με τον τρόπο που θα χρησιμοποιηθούν όλα αυτά τα λογικά κυκλώματα. Όπως συζητήθηκε προηγουμένως σε αυτή την ενότητα, υπάρχουν όρια στην αποτελεσματική χρήση των τεχνικών όπως οι υπερβαθμωτοί επεξεργαστές και η ταυτόχρονη πολυνημάτωση. Γενικά,



Σχήμα 18.3: Ζητήματα κατανάλωσης ισχύος και μνήμης



Σχήμα 18.4: Χρησιμοποίηση των transistors του chip

η εμπειρία των τελευταίων δεκαετιών, περικλείεται σε έναν εμπειρικό κανόνα, ο οποίος είναι γνωστός ως **κανόνας του Pollack** [POLL99], ο οποίος δηλώνει ότι η αύξηση της απόδοσης είναι περίπου ανάλογη της τετραγωνικής ρίζας της αύξησης της πολυπλοκότητας. Με άλλα λόγια, αν διπλασιάσουμε τα λογικά κυκλώματα ενός πυρήνα, τότε θα επιτύχουμε μόλις 40% αύξηση της απόδοσης. Η χρήση πολλαπλών πυρήνων έχει τη δυνατότητα να παρέχει σχεδόν γραμμική βελτίωση της απόδοσης, καθώς αυξάνεται το πλήθος των πυρήνων.

Τα ζητήματα κατανάλωσης της ισχύος παρέχουν ένα ακόμη κίνητρο για τη μετακίνηση προς μία οργάνωση πολλαπλών πυρήνων. Επειδή το chip περιέχει μία μεγάλη ποσότητα κρυφής μνήμης, μοιάζει απίθανο όλη αυτή η μνήμη να χρησιμοποιηθεί από ένα νήμα. Ακόμη και στην ταυτόχρονη πολυνημάτωση, δεν είναι δυνατή η εκμετάλλευση μίας γιγαντιαίας κρυφής μνήμης, δεδομένου ότι η νημάτωση γίνεται σε περιορισμένο επίπεδο. Από την άλλη, ένα πλήθος σχετικά ανεξάρτητων νημάτων ή διεργασιών έχει τη δυνατότητα να εκμεταλλευτεί πλήρως την κρυφή μνήμη.

18.2 ΖΗΤΗΜΑΤΑ ΑΠΟΔΟΣΗΣ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ

Μία λεπτομερής εξέταση της απόδοσης του λογισμικού που σχετίζεται με την οργάνωση πολλαπλών πυρήνων, είναι εκτός των σκοπών αυτού του βιβλίου. Σε αυτή την ενότητα, αρχικά παρουσιάζουμε μία γενική θεώρηση αυτών των θεμάτων, και στη συνέχεια εξετάζουμε ένα παράδειγμα που αφορά μία εφαρμογή σχεδιασμένη ώστε να εκμεταλλεύεται τις δυνατότητες της οργάνωσης πολλαπλών πυρήνων.

Λογισμικό σε Πολλαπλούς Πυρήνες

Τα εν δυνάμει ωφέλη της απόδοσης μίας οργάνωσης πολλαπλών πυρήνων εξαρτώνται από τη δυνατότητα της αποτελεσματικής εκμετάλλευσης των παράλληλων πόρων οι οποίοι είναι διαθέσιμοι στην εφαρμογή. Αρχικά, θα επικεντρωθούμε σε μία απλή εφαρμογή, η οποία εκτελείται σε ένα σύστημα πολλαπλών πυρήνων. Από το Κεφάλαιο 2, θυμηθείτε ότι ο νόμος του Amdahl δηλώνει ότι:

$$\text{Αύξηση ταχύτητας} = \frac{\text{χρόνος εκτέλεσης προγράμματος σε έναν επεξεργαστή}}{\text{χρόνος εκτέλεσης προγράμματος σε } N \text{ επεξεργαστές}} = \frac{1}{(1-f) + \frac{f}{N}}$$

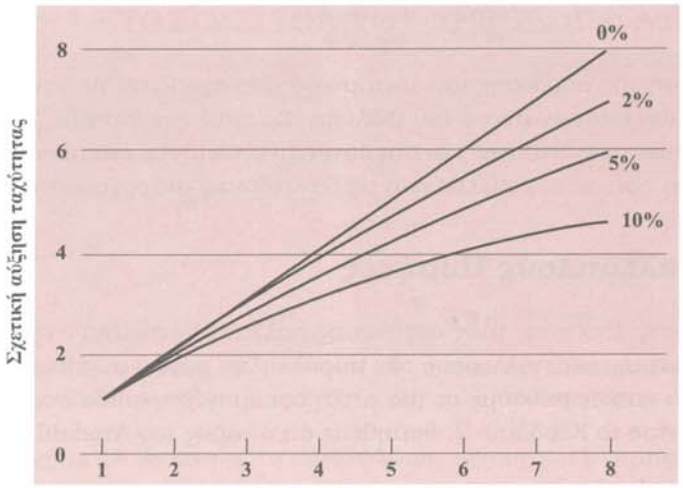
Αυτός ο νόμος υποθέτει ένα πρόγραμμα, στο οποίο ένα ποσοστό $(1-f)$ του χρόνου εκτέλεσης αφορά κώδικα, ο οποίος είναι ακολουθιακός και ένα ποσοστό f αφορά κώδικα, ο οποίος είναι απεριόριστα παραλληλοποιήσιμος, χωρίς επιβάρυνση η οποία οφείλεται στη χρονοδρομολόγηση των διεργασιών.

Ο νόμος αυτός φαίνεται να καθιστά ελκυστική την προοπτική μίας οργάνωσης πολλαπλών πυρήνων. Ωστόσο, όπως δείχνει το Σχήμα 18.5α, ακόμη και ένα μικρό κομμάτι κώδικα του οποίου οι εντολές εκτελούνται η μία μετά την άλλη, έχει σημαντική επίδραση. Αν το ποσοστό αυτού του κώδικα είναι μόνον 10% ($f = 0.9$), η εκτέλεση του προγράμματος σε ένα σύστημα πολλαπλών πυρήνων με 8 επεξεργαστές θα δώσει κέρδη απόδοσης κατά έναν παράγοντα μόλις 4.7. Επιπλέον, το λογισμικό εμφανίζει συχνά επιβαρύνσεις οι οποίες οφείλονται στην επικοινωνία και στην κατανομή της εργασίας σε πολλαπλούς επεξεργαστές, αλλά και στη συνοχή της κρυφής μνήμης. Τα παραπάνω, οδηγούν σε μία καμπύλη όπου η απόδοση φτάνει σε μία μέγιστη τιμή και μετά φαίνεται να μειώνεται λόγω της επιβάρυνσης η οποία προκύπτει από τη χρήση πολλών επεξεργαστών. Το Σχήμα 18.5β ([MCDO05]), αποτελεί ένα κατανοητό παράδειγμα.

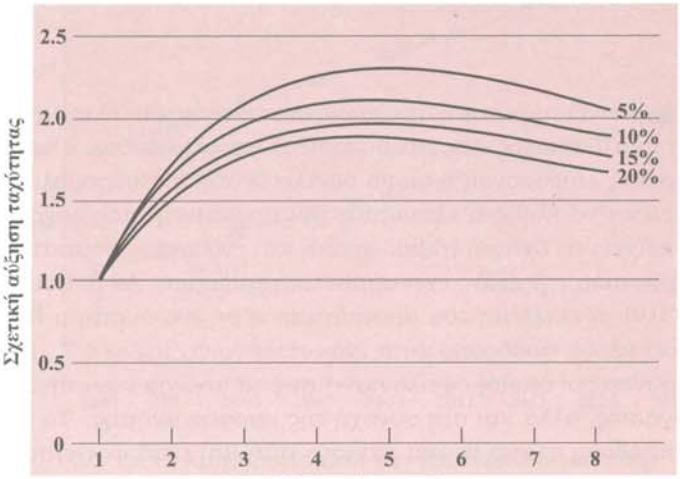
Ωστόσο, οι κατασκευαστές λογισμικού έχουν αντιμετωπίσει αυτό το πρόβλημα και υπάρχει ένα μεγάλο πλήθος εφαρμογών οι οποίες εκμεταλλεύονται αποτελεσματικά την οργάνωση πολλαπλών πυρήνων. Στην αναφορά [MCDO05] παρατίθεται ένα σύνολο εφαρμογών βάσεων δεδομένων, στο οποίο δόθηκε μεγάλη προσοχή στη μείωση του σειριακού τμήματος, εντός των αρχιτεκτονικών υλικού, των λειτουργικών συστημάτων, του ενδιάμεσου επιπέδου λογισμικού και του λογισμικού εφαρμογών βάσεων δεδομένων. Το Σχήμα 18.6 δείχνει το αποτέλεσμα. Όπως φαίνεται στο παράδειγμα, τα συστήματα διαχείρισης και οι εφαρμογές βάσεων δεδομένων είναι μία περιοχή στην οποία είναι δυνατή η αποτελεσματική χρήση των συστημάτων πολλαπλών πυρήνων. Πολλές μορφές εξυπηρετών έχουν επίσης τη δυνατότητα να χρησιμοποιούν αποτελεσματικά την παράλληλη οργάνωση πολλαπλών πυρήνων, επειδή οι εξυπηρέτες τυπικά διαχειρίζονται παράλληλα πολλές σχετικά ανεξάρτητες μεταξύ τους συναλλαγές.

Επιπλέον του λογισμικού γενικού σκοπού του εξυπηρέτη, ένα πλήθος κατηγοριών εφαρμογών ωφελούνται άμεσα από τη δυνατότητα αύξησης της ρυθμαπόδοσης, καθώς αυξάνεται το πλήθος των πυρήνων. Στην αναφορά [MCDO06] παρατίθενται τα ακόλουθα παραδείγματα:

- **Αμιγείς εφαρμογές πολλαπλών νημάτων:** Οι εφαρμογές πολλαπλών νημάτων χαρακτηρίζονται από την ύπαρξη μικρού πλήθους διεργασιών με υψηλό βαθμό νημάτωσης. Παραδείγματα



(α) Επιτάχυνση με 0%, 2%, 5%, και 10% σειριακά τμήματα



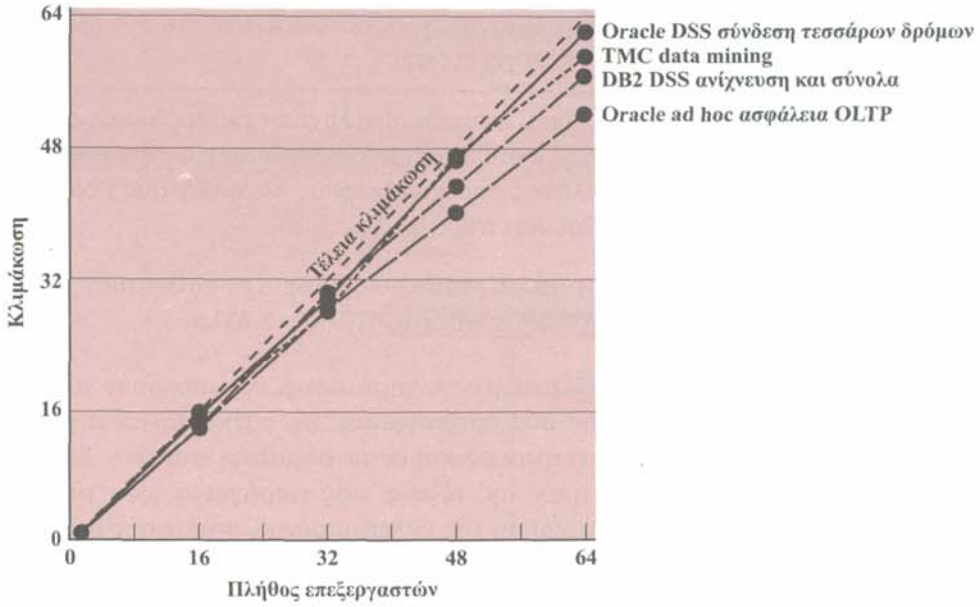
(β) Αύξηση ταχύτητας με επιβαρύνσεις

Σχήμα 18.5: Επίδραση των πολλαπλών πυρήνων στην απόδοση

αποτελούν οι εφαρμογές Lotus Domino και Seibel CRM (Customer Relationship Manager).

- **Εφαρμογές πολλαπλών διεργασιών:** Οι εφαρμογές πολλαπλών διεργασιών χαρακτηρίζονται από την παρουσία πολλών διεργασιών απλής νημάτωσης. Παραδείγματα αποτελούν οι εφαρμογές Oracle, SAP, και PeopleSoft.

- **Εφαρμογές Java:** Οι εφαρμογές Java περικλείουν τη νημάτωση με έναν ουσιαστικό τρόπο. Η γλώσσα Java, όχι μόνον διευκολύνει σημαντικά τις εφαρμογές πολλαπλών νημάτων, αλλά και η Ιδεατή Μηχανή της Java (JVM-Java Virtual Machine), είναι μία διεργασία πολλαπλής νημάτωσης, η οποία παρέχει στις εφαρμογές της γλώσσας χρονοδρομολόγηση και διαχείριση μνήμης. Οι εφαρμογές της γλώσσας Java οι οποίες έχουν άμεσα ωφέλη από τους πόρους πολλαπλών πυρήνων, συμπεριλαμβάνουν διάφορους εξυπηρέτες εφαρμογών όπως εκείνος της Sun, ο WebLogic της BEA, ο Websphere της IBM, και ο Tomcat, ο οποίος είναι ένας εξυπηρέτης εφαρμογών ανοικτού λογισμικού. Όλες οι εφαρμογές οι οποίες χρησιμοποιούν την πλατφόρμα



Σχήμα 18.6: Κλιμάκωση των φορτίων εργασίας μίας βάσης δεδομένων στο υλικό πολλαπλών επεξεργαστών

εξυπηρέτη εφαρμογών Java 2 Enterprise Edition, είναι δυνατόν να ωφεληθούν άμεσα από την τεχνολογία πολλαπλών πυρήνων.

- **Εφαρμογές πολλαπλών στιγμιοτύπων:** Ακόμη και αν μία εφαρμογή δεν έχει τη δυνατότητα να χρησιμοποιήσει και να εκμεταλλευτεί την ύπαρξη των πολλαπλών νημάτων, εξακολουθεί να έχει τη δυνατότητα να αποκομίσει ωφέλη από την αρχιτεκτονική πολλαπλών πυρήνων, εκτελώντας παράλληλα, πολλαπλά στιγμιότυπα της εφαρμογής. Αν τα στιγμιότυπα αυτά απαιτούν κάποιο βαθμό απομόνωσης, είναι δυνατόν να χρησιμοποιηθεί η τεχνολογία συγκάλυψης (για το υλικό του λειτουργικού συστήματος), ώστε να παρέχει σε κάθε στιγμιότυπο το δικό του ξεχωριστό και ασφαλές περιβάλλον.

Παράδειγμα Εφαρμογής: Το Λογισμικό Παιχνιδιών της Valve

Η Valve είναι μία εταιρία ψυχαγωγίας και τεχνολογίας, η οποία έχει αναπτύξει ένα πλήθος από δημοφιλή παιχνίδια, όπως και τη μηχανή Source, η οποία είναι μία από τις πιο δημοφιλείς μηχανές παιχνιδιών. Η Source, είναι μία μηχανή γραφικών την οποία χρησιμοποίησε η εταιρία για τα παιχνίδια της, αλλά έχει δοθεί άδεια χρήσης της και σε άλλες εταιρίες υλοποίησης παιχνιδιών.

Τα πιο πρόσφατα χρόνια, η Valve προγραμμάτισε εκ νέου το λογισμικό της μηχανής Source, προκειμένου να χρησιμοποιήσει πολλαπλή νημάτωση και να εκμεταλλευτεί την ισχύ των chips πολλαπλών επεξεργαστών της Intel και της AMD [REIM06]. Ο νέος κώδικας παρέχει πιο ισχυρή υποστήριξη στα παιχνίδια της Valve, όπως είναι το Half Life 2.

Από την πλευρά της Valve, οι επιλογές που αφορούν τις λεπτομέρειες της νημάτωσης ορίζονται ως ακολούθως:

- **Μη εκλεπτυσμένη νημάτωση:** Οι ξεχωριστές μονάδες οι οποίες ονομάζονται συστήματα, εκχωρούνται σε διαφορετικούς επεξεργαστές. Στη μηχανή Source, αυτό σημαίνει την τοποθέτηση λειτουργιών αναπαράστασης (rendering) σε έναν επεξεργαστή, τεχνητής νοημοσύνης σε έναν άλλο, φυσικής σε άλλο, κ.ο.κ. Αυτή η εκχώρηση είναι απλή. Ουσιαστικά, κάθε βασική μονάδα

διαθέτει απλή νημάτωση και ο βασικός συντονισμός συμπεριλαμβάνει το συγχρονισμό όλων των νημάτων με ένα νήμα σχεδίασης χρονικών γεγονότων.

- **Εκλεπτυσμένη νημάτωση:** Πολλές ίδιες ή παρόμοιες εργασίες μοιράζονται σε πολλούς επεξεργαστές. Για παράδειγμα, ένας βρόχος ο οποίος επαναλαμβάνεται για έναν πίνακα δεδομένων, είναι δυνατόν να διαχωριστεί σε ένα πλήθος από μικρότερους παράλληλους βρόχους σε ξεχωριστά νήματα, τα οποία χρονοδρομολογούνται παράλληλα.

- **Υβριδική νημάτωση:** Η υβριδική νημάτωση συμπεριλαμβάνει την επιλεκτική χρήση της εκλεπτυσμένης νημάτωσης για μερικά συστήματα και της απλής για άλλα.

Η Valve διαπίστωσε ότι μέσω της μη εκλεπτυσμένης νημάτωσης, θα μπορούσε να επιτύχει μέχρι και τη διπλάσια απόδοση χρησιμοποιώντας δύο επεξεργαστές, σε σύγκριση με την απόδοση ενός μόνον επεξεργαστή. Ωστόσο, αυτό το κέρδος ήταν εφικτό σε πειραματικό επίπεδο. Στην πραγματική περιοχή των παιχνιδιών, το κέρδος αυτό ήταν της τάξεως ενός παράγοντα ίσου με 1.2. Επίσης, η εταιρία ανακάλυψε ότι η αποτελεσματική χρήση της εκλεπτισμένης νημάτωσης δεν ήταν απλή. Ο χρόνος που απαιτείται για κάθε μονάδα εργασίας πιθανόν να ποικίλει, και η διαχείριση του χρονισμού των αποτελεσμάτων αλλά και των συνεπειών απαιτούσε πολύπλοκο προγραμματισμό.

Η Valve διαπίστωσε ότι μία υβριδική προσέγγιση ήταν η πιο ελπιδοφόρα και θα απέδιδε στο μέγιστο βαθμό μόλις υπήρχε διαθεσιμότητα συστημάτων πολλαπλών πυρήνων αποτελούμενων από 8 ή 16 επεξεργαστές. Η εταιρία καθόρισε συστήματα τα οποία λειτουργούσαν αποτελεσματικά όταν εκχωρούνταν μονίμως σε έναν επεξεργαστή. Ένα παράδειγμα είναι η μίξη ήχου, η οποία συμπεριλαμβάνει μικρή αλληλεπίδραση του χρήστη, δεν περιορίζεται από τη διάταξη πλαισίων των παραθύρων, και λειτουργεί με το δικό της σύνολο δεδομένων. Άλλες μονάδες, όπως είναι η αναπαράσταση σκηνών, είναι δυνατόν να οργανωθούν σε ένα πλήθος από νήματα, ώστε η μονάδα να έχει τη δυνατότητα να εκτελέσει τις εργασίες σε έναν επεξεργαστή, αλλά να επιτύχει μεγαλύτερη απόδοση όταν διασπάται σε ολοένα και περισσότερους επεξεργαστές.

Το Σχήμα 18.7 απεικονίζει τη δομή νημάτων της μονάδας αναπαράστασης. Σε αυτή την ιεραρχική δομή, τα νήματα υψηλότερου επιπέδου, παράγουν νήματα χαμηλότερου επιπέδου, όταν αυτό απαιτείται. Η μονάδα αναπαράστασης βασίζεται σε ένα κρίσιμο τμήμα της μηχανής Source, την παγκόσμια λίστα, η οποία είναι μία αναπαράσταση σε μορφή βάσης δεδομένων των ιδεατών στοιχείων που υπάρχουν στον κόσμο των παιχνιδιών. Η πρώτη εργασία είναι να προσδιοριστούν οι περιοχές του κόσμου οι οποίες θα πρέπει να αναπαρασταθούν. Η επόμενη εργασία είναι να καθοριστεί ποια αντικείμενα βρίσκονται στη σκηνή, όπως αυτά φαίνονται από διάφορες οπτικές γωνίες. Στη συνέχεια έρχεται η εντατική εργασία του επεξεργαστή. Η μονάδα αναπαράστασης θα πρέπει να εργαστεί στην αναπαράσταση κάθε αντικειμένου από πολλές οπτικές γωνίες, όπως εκείνη του παίκτη, των οθονών, και από την οπτική των αντανακλάσεων στο νερό.

Ορισμένα από τα βασικά χαρακτηριστικά της στρατηγικής νημάτωσης, οι οποίες χρησιμοποιούνται από τη μονάδα αναπαράστασης, παρατίθενται στην αναφορά [LEON07]:

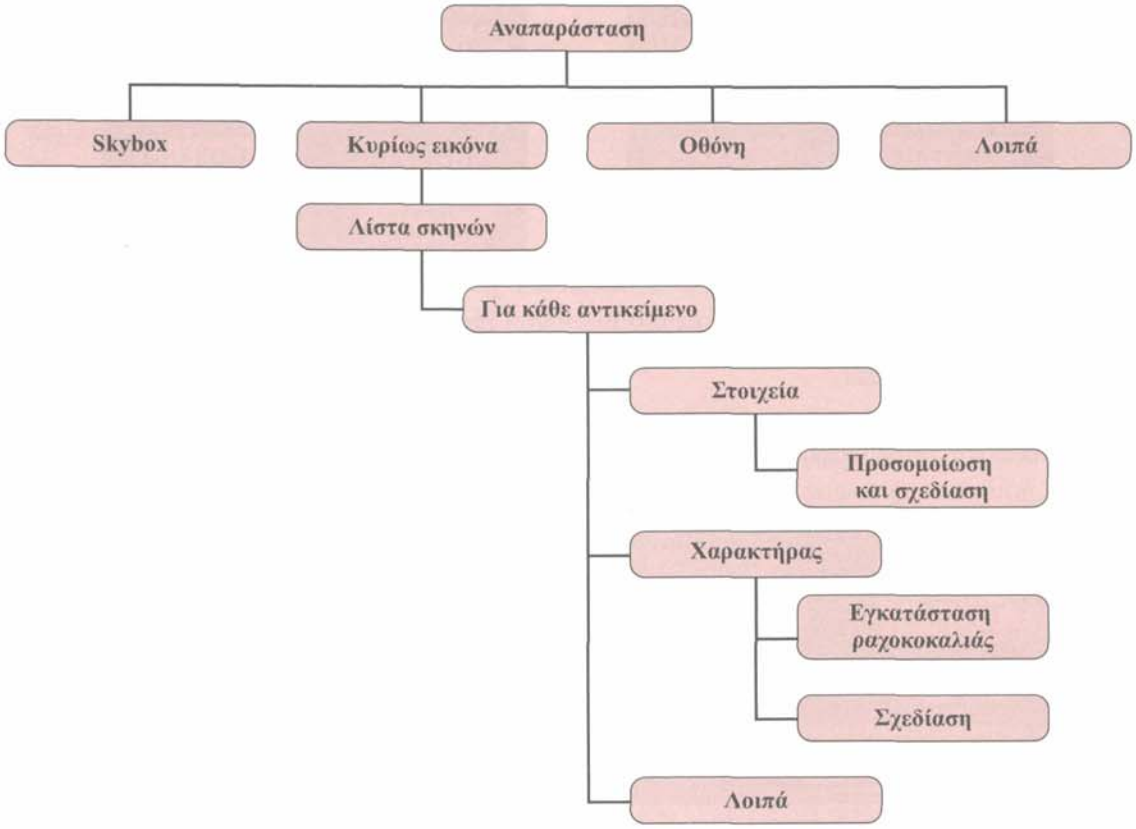
- Κατασκευή λιστών παράλληλης αναπαράστασης σκηνών (π.χ. ο κόσμος και η αντανάκλασή του στο νερό).

- Προσομοιώσεις επικάλυψης γραφικών.

- Υπολογισμός μετασχηματισμού χαρακτήρων, παράλληλα, για κάθε χαρακτήρα και για κάθε σκηνή.

- Παράλληλη εκτέλεση πολλαπλών νημάτων.

Οι σχεδιαστές ανακάλυψαν ότι το απλό κλείδωμα για ένα νήμα των σημαντικών βάσεων δεδομένων, όπως είναι η παγκόσμια λίστα, ήταν πολύ αναποτελεσματικό. Για περισσότερο από το 95% του



Σχήμα 18.7: Υβριδική νημάτωση για τη μονάδα αναπαράστασης

χρόνου, ένα νήμα προσπαθεί να διαβάσει από ένα σύνολο δεδομένων και μόνον 5% του χρόνου το πολύ δαπανάται για εγγραφές σε ένα σύνολο δεδομένων. Επομένως, είναι αποτελεσματικός ένας μηχανισμός ταυτοχρονισμού, ο οποίος είναι γνωστός ως απλός εγγραφέας, πολλαπλοί αναγνώστες.

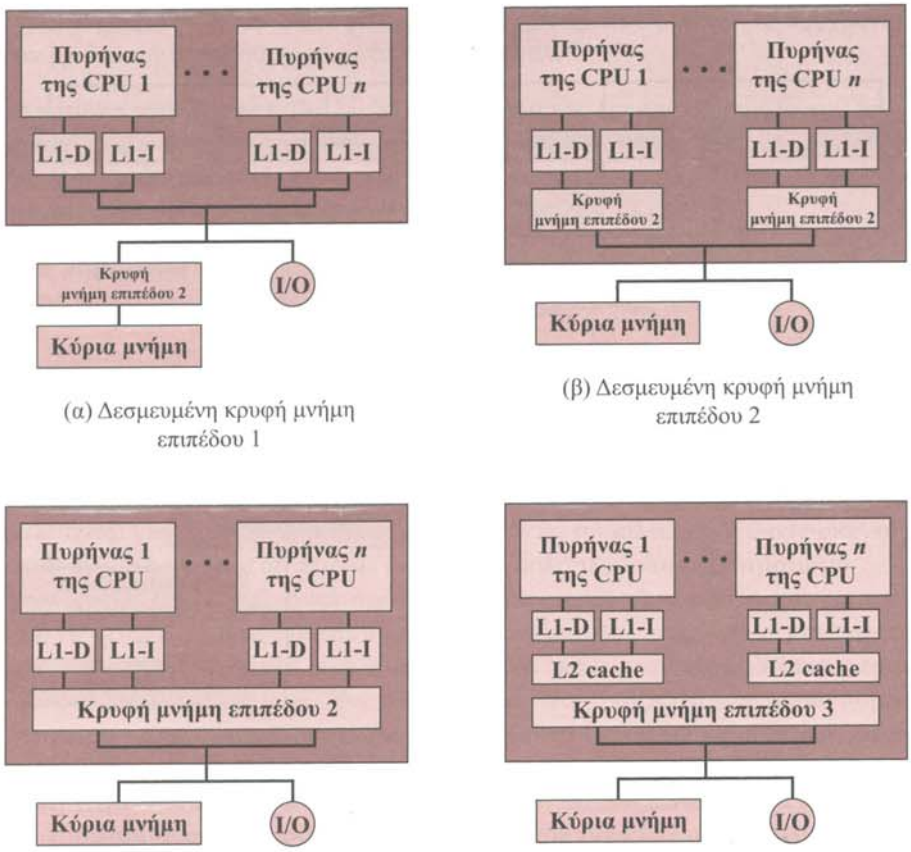
18.3 ΟΡΓΑΝΩΣΗ ΠΟΛΛΑΠΛΩΝ ΠΥΡΗΝΩΝ

Σε ένα υψηλό επίπεδο σχεδίασης, οι βασικές μεταβλητές μίας μορφής οργάνωσης πολλαπλών πυρήνων είναι οι ακόλουθες:

- *Το πλήθος των επεξεργαστών πάνω στο chip*
- Το πλήθος των επιπέδων της κρυφής μνήμης
- Η ποσότητα της κοινόχρηστης κρυφής μνήμης

Το Σχήμα 18.8 απεικονίζει τέσσερις γενικές μορφές οργάνωσης συστημάτων πολλαπλών πυρήνων. Το Σχήμα 18.8α είναι μία μορφή οργάνωσης την οποία συναντούμε σε ορισμένα από τα πρώτα chips πολλαπλών πυρήνων, και εξακολουθεί να συναντάται σε ενσωματωμένα chips. Σε αυτή τη μορφή οργάνωσης, πάνω στο chip υπάρχει κρυφή μνήμη μόνον Επιπέδου 1 και ο κάθε πυρήνας έχει τη δική του κρυφή μνήμη. Σχεδόν χωρίς παραλλαγή, οι κρυφές μνήμες διαιρούνται σε μνήμες εντολών και δεδομένων. Ένα παράδειγμα αυτής της μορφής οργάνωσης είναι ο επεξεργαστής ARM11 MPCore.

Η οργάνωση του Σχήματος 18.8β είναι επίσης μία μορφή στην οποία δεν υπάρχει κοινόχρηστη κρυφή μνήμη πάνω στο chip. Ωστόσο, υπάρχει αρκετός χώρος στο chip, ώστε να επιτραπεί η ύπαρξη κρυφής μνήμης Επιπέδου 2. Ένα παράδειγμα αυτής της οργάνωσης αποτελεί ο επεξεργαστής AMD



(α) Δεσμευμένη κρυφή μνήμη επιπέδου 1

(β) Δεσμευμένη κρυφή μνήμη επιπέδου 2

(γ) Κοινόχρηστη κρυφή μνήμη επιπέδου 2

(δ) Κοινόχρηστη κρυφή μνήμη επιπέδου 3

Σχήμα 18.8: Εναλλακτικές μορφές οργάνωσης πολλαπλών πυρήνων

Opteron. Το Σχήμα 18.8γ απεικονίζει μία παρόμοια κατανομή του χώρου του chip στη μνήμη, όπου όμως χρησιμοποιείται μία κοινόχρηστη κρυφή μνήμη Επιπέδου 2. Ο επεξεργαστής Intel Core Duo παρουσιάζει αυτή την οργάνωση. Τέλος, καθώς η ποσότητα της κρυφής μνήμης που είναι διαθέσιμη πάνω στο chip συνεχίζει να αυξάνεται, τα ζητήματα απόδοσης υπαγορεύουν το διαμερισμό της κρυφής μνήμης και τη δημιουργία ενός τρίτου επιπέδου (κρυφή μνήμη Επιπέδου 3). Η μνήμη αυτή είναι κοινόχρηστη, ενώ οι κρυφές μνήμες Επιπέδου 1 και 2 είναι δεσμευμένες σε κάθε επεξεργαστή. Ένα παράδειγμα αυτής της μορφής οργάνωσης, αποτελεί ο επεξεργαστής Intel Core i7.

Η χρησιμοποίηση μίας κοινής κρυφής μνήμης Επιπέδου 2 στο chip παρουσιάζει αρκετά πλεονεκτήματα σε σύγκριση με την ύπαρξη αποκλειστικών κρυφών μνημών:

1. Η δομική παρεμβολή, μειώνει συνολικά τους ρυθμούς αστοχίας. Με άλλα λόγια, αν ένα νήμα ενός πυρήνα προσπελαύνει μία θέση της κύριας μνήμης, το πλαίσιο το οποίο περιέχει την αναφερόμενη θέση προσκομίζεται στην κοινόχρηστη κρυφή μνήμη. Αν ένα νήμα ενός άλλου πυρήνα ζητήσει σύντομα να προσπελάσει το ίδιο μπλοκ μνήμης, οι θέσεις μνήμης θα βρίσκονται ήδη διαθέσιμες στην κοινόχρηστη μνήμη πάνω στο chip.

2. Ένα πλεονέκτημα το οποία σχετίζεται με τα παραπάνω, είναι ότι τα δεδομένα τα οποία μοιράζονται πολλοί πυρήνες, δεν είναι ανάγκη να υπάρχουν πολλές φορές στο κοινόχρηστο επίπεδο κρυφής μνήμης.

3. Με τους κατάλληλους αλγορίθμους αντικατάστασης πλαισίων, η ποσότητα της κοινόχρηστης κρυφής μνήμης, η οποία κατανέμεται σε κάθε πυρήνα, είναι δυναμική, έτσι ώστε τα νήματα τα οποία έχουν μικρότερη τοπικότητα να έχουν τη δυνατότητα να χρησιμοποιούν περισσότερη κρυφή μνήμη.

4. Η επικοινωνία ανάμεσα στους επεξεργαστές είναι εύκολο να υλοποιηθεί μέσα από θέσεις της κοινόχρηστης μνήμης.

5. Η χρήση μίας κοινής κρυφής μνήμης Επιπέδου 2 περιορίζει το πρόβλημα της συνοχής της κρυφής μνήμης μόνον στο Επίπεδο 1, κάτι το οποίο μπορεί να δώσει ένα ακόμη πλεονέκτημα όσον αφορά την απόδοση.

Ένα πιθανό πλεονέκτημα που προκύπτει από την ύπαρξη στο chip μόνον δεσμευμένης κρυφής μνήμης Επιπέδου 2, είναι ότι κάθε πυρήνας απολαμβάνει ταχύτερη προσπέλαση στη δική του κρυφή μνήμη Επιπέδου 2. Αυτό αποτελεί πλεονέκτημα για νήματα τα οποία εμφανίζουν ισχυρή τοπικότητα.

Καθώς τόσο η ποσότητα της διαθέσιμης μνήμης όσο και το πλήθος των πυρήνων αυξάνονται, η χρήση μίας κοινής κρυφής μνήμης Επιπέδου 3 σε συνδυασμό είτε με μία κοινή κρυφή μνήμη Επιπέδου 2, είτε με δεσμευμένες ανά πυρήνα κρυφές μνήμες Επιπέδου 2, φαίνεται πιθανό να δίνει καλύτερη απόδοση σε σύγκριση με μία απλή κοινή κρυφή μνήμη Επιπέδου 2 μεγάλης χωρητικότητας.

Μία ακόμη απόφαση σχεδίασης σε ένα σύστημα πολλαπλών πυρήνων, είναι αν οι διαφορετικοί πυρήνες θα είναι υπερβαθμωτοί ή αν θα υλοποιούν ταυτόχρονη υπερνημάτωση. Για παράδειγμα, ο επεξεργαστής Intel Core Duo χρησιμοποιεί υπερβαθμωτούς πυρήνες, ενώ ο επεξεργαστής Intel Core 7i χρησιμοποιεί πυρήνες ταυτόχρονης υπερνημάτωσης. Η ταυτόχρονη υπερνημάτωση έχει ως αποτέλεσμα την αύξηση του πλήθους των νημάτων σε επίπεδο υλικού, τα οποία υποστηρίζονται από το σύστημα πολλαπλών πυρήνων. Επομένως, στο επίπεδο εφαρμογής, ένα σύστημα πολλαπλών πυρήνων αποτελούμενο από 4 πυρήνες ταυτόχρονης πολυνημάτωσης όπου κάθε πυρήνας υποστηρίζει 4 νήματα, μοιάζει το ίδιο με ένα σύστημα πολλαπλών πυρήνων, το οποίο περιέχει 16 πυρήνες. Καθώς το λογισμικό αναπτύσσεται ώστε να εκμεταλλεύεται όσο το δυνατόν πιο πλήρως τους παράλληλους πόρους, η προσέγγιση της ταυτόχρονης πολυνημάτωσης φαίνεται πιο ελκυστική από την υπερβαθμωτή προσέγγιση.

18.4 ΟΡΓΑΝΩΣΗ ΠΟΛΛΑΠΛΩΝ ΠΥΡΗΝΩΝ ΤΗΣ INTEL

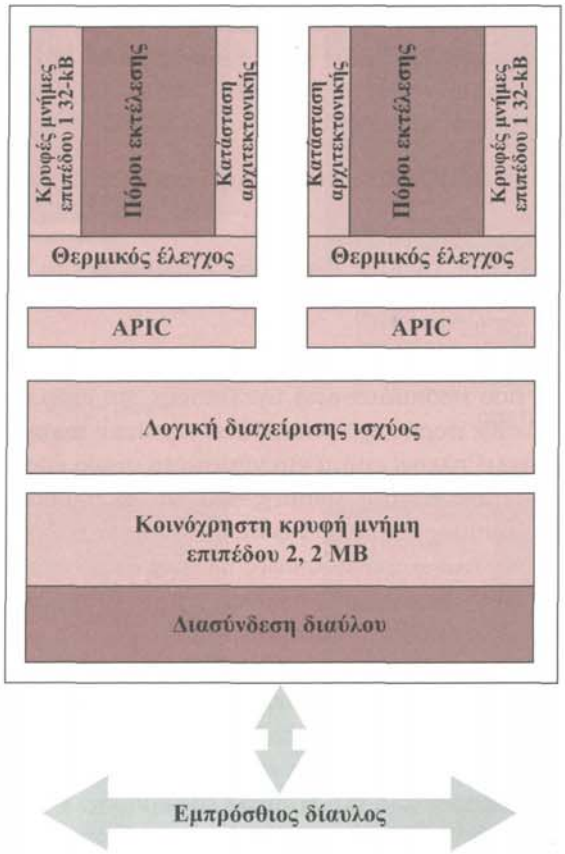
Η Intel έχει εισάγει ένα πλήθος από προϊόντα πολλαπλών πυρήνων τα τελευταία χρόνια. Σε αυτή την ενότητα, εξετάζουμε δύο παραδείγματα: τους επεξεργαστές Core Duo και Core i7.

Ο Επεξεργαστής Intel Core Duo

Ο επεξεργαστής Core Duo, ο οποίος εισήχθη το 2006, αποτελείται από δύο υπερβαθμωτούς επεξεργαστές x86, με κοινόχρηστη κρυφή μνήμη Επιπέδου 2 (Σχήμα 18.8γ).

Η γενική δομή του επεξεργαστή Core Duo απεικονίζεται στο Σχήμα 18.9. Ας εξετάσουμε τα βασικά στοιχεία, ξεκινώντας από το πάνω μέρος του σχήματος. Όπως συνηθίζεται στα συστήματα πολλαπλών πυρήνων, κάθε πυρήνας έχει τη δική του δεσμευμένη **κρυφή μνήμη Επιπέδου 1**. Σε αυτή την περίπτωση, κάθε πυρήνας διαθέτει μία κρυφή μνήμη εντολών και μία κρυφή μνήμη δεδομένων, μεγέθους 32 KB.

Κάθε πυρήνας διαθέτει μία ανεξάρτητη **μονάδα θερμικού ελέγχου**. Με δεδομένη την υψηλή πυκνότητα των transistors των σύγχρονων chips, η θερμική διαχείριση είναι μία σημαντική δυνατότητα, ειδικά για τους φορητούς υπολογιστές. Η μονάδα θερμικού ελέγχου του επεξεργαστή Core Duo είναι σχεδιασμένη ώστε να διαχειρίζεται τη διάχυση της θερμότητας μέσα στο chip, ώστε να βελτιώνει την απόδοση του επεξεργαστή εντός των θερμικών περιορισμών. Επίσης, η θερμική διαχείριση βελτιώνει



Σχήμα 18.9: Διάγραμμα μπλοκ του επεξεργαστή Intel Core Duo

την εργονομία χρησιμοποιώντας ένα σύστημα ψύξης με μικρότερο ακουστικό θόρυβο του ανεμιστήρα. Ουσιαστικά, η μονάδα θερμικής διαχείρισης χρησιμοποιεί ψηφιακούς αισθητήρες, για υψηλής ακρίβειας μετρήσεις της θερμοκρασίας της επιφάνειας πυριτίου. Κάθε πυρήνας ορίζεται ως μία ανεξάρτητη θερμική ζώνη. Η μέγιστη θερμοκρασία κάθε ζώνης αναφέρεται ξεχωριστά μέσω δεσμευμένων καταχωρητών, οι οποίοι δειγματοληπτούνται με τη βοήθεια λογισμικού. Αν η θερμοκρασία μέσα σε έναν πυρήνα ξεπεράσει μία τιμή κατωφλίου, η μονάδα θερμικού ελέγχου μειώνει το ρυθμό ρολογιού του πυρήνα, ώστε να μειωθεί η παραγωγή θερμότητας.

Το επόμενο σημαντικό στοιχείο της οργάνωσης του επεξεργαστή Core Duo, είναι ο **Εξελιγμένος Προγραμματιζόμενος Ελεγκτής Διακοπών** (Advanced Programmable Interrupt Controller- APIC). Ο APIC εκτελεί ένα πλήθος λειτουργιών, συμπεριλαμβανομένων των παρακάτω:

1. Ο APIC παρέχει διακοπές ανάμεσα στους επεξεργαστές, κάτι το οποίο επιτρέπει σε κάθε διεργασία να διακόψει οποιονδήποτε άλλο επεξεργαστή ή σύνολο επεξεργαστών. Στην περίπτωση του επεξεργαστή Core Duo, ένα νήμα που βρίσκεται στον ένα πυρήνα, μπορεί να παράγει μία διακοπή, η οποία γίνεται αποδεκτή από τον τοπικό APIC, μεταφέρεται στον APIC του άλλου πυρήνα και μεταδίδεται στον άλλο πυρήνα ως διακοπή.

2. Ο APIC δέχεται διακοπές Ε/Ε και τις δρομολογεί προς τον κατάλληλο πυρήνα.

3. Κάθε APIC περιέχει ένα χρονόμετρο, το οποίο ορίζεται από το λειτουργικό σύστημα, ώστε να παράγονται διακοπές στον τοπικό πυρήνα αυτού του APIC.

Πίνακας 18.1: Λανθάνων χρόνος της κρυφής μνήμης (σε κύκλους ρολογιού)

Επεξεργαστής	Συχνότητα Ρολογιού	Κρυφή Μνήμη Επιπέδου 1	Κρυφή Μνήμη Επιπέδου 2	Κρυφή Μνήμη Επιπέδου 3
Core 2 Quad	2.66 GHz	3 κύκλοι	15 κύκλοι	-
Core i7	2.66 GHz	4 κύκλοι	11 κύκλοι	39 κύκλοι

Η **λογική διαχείρισης ισχύος** είναι υπεύθυνη για τη μείωση της κατανάλωσης ισχύος, όταν αυτό είναι εφικτό, αυξάνοντας επομένως τη διάρκεια ζωής της μπαταρίας των φορητών υπολογιστών. Ουσιαστικά, η λογική διαχείρισης ισχύος παρακολουθεί τις θερμικές συνθήκες και τη δραστηριότητα της κεντρικής μονάδας επεξεργασίας και ρυθμίζει κατάλληλα τα επίπεδα τάσης αλλά και την κατανάλωση ισχύος. Η λογική διαχείρισης ισχύος συμπεριλαμβάνει μία εξελιγμένη δυνατότητα διέλευσης του ρεύματος, επιτρέποντας την υλοποίηση μίας ιδιαίτερα εκλεπτισμένης λογικής ελέγχου, η οποία ενεργοποιεί τα υποσυστήματα λογικής κάθε επεξεργαστή, μόνον αν, και όταν αυτό είναι αναγκαίο. Επιπλέον, πολλοί δίαυλοι και παρατάξεις είναι διαχωρισμένες, ώστε τα δεδομένα τα οποία είναι αναγκαία σε ορισμένες καταστάσεις λειτουργίες, να τοποθετούνται σε κατάσταση χαμηλής ισχύος, όταν δεν είναι αναγκαία.

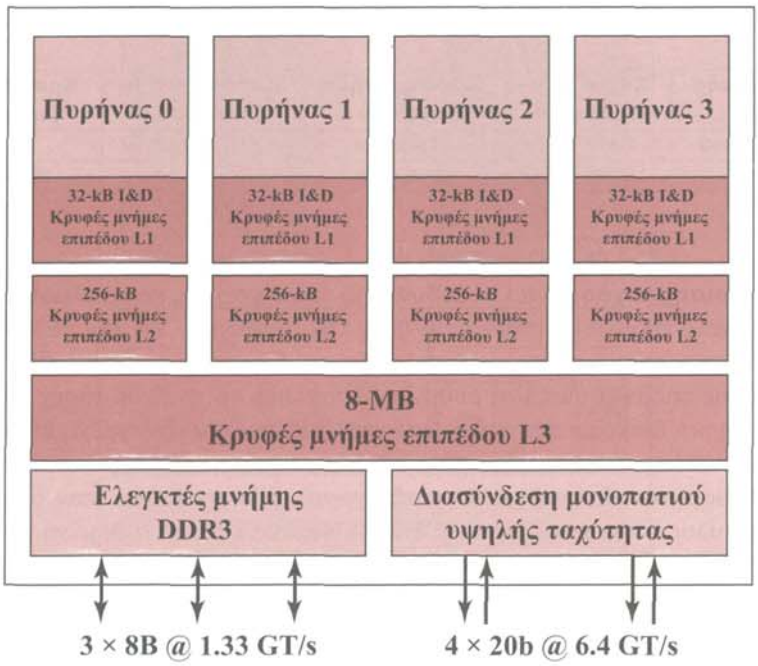
Το chip του επεξεργαστή Core Duo συμπεριλαμβάνει μία κοινόχρηστη **κρυφή μνήμη Επιπέδου 2** μεγέθους 2MB. Τα λογικά κυκλώματα της κρυφής μνήμης επιτρέπουν τη δυναμική κατανομή του χώρου της, βάσει των τρεχουσών αναγκών του πυρήνα, έτσι ώστε να υπάρχει η δυνατότητα εκχώρησης σε έναν πυρήνα έως και του 100% της κρυφής μνήμης Επιπέδου 2. Η κρυφή μνήμη περιέχει λογικά κυκλώματα υποστήριξης του πρωτοκόλλου MESI, για τις συνδεδεμένες κρυφές μνήμες Επιπέδου 1. Το βασικό σημείο το οποίο θα πρέπει να εξεταστεί είναι πότε γίνεται εγγραφή της κρυφής μνήμης στο Επίπεδο 1. Μία γραμμή της κρυφής μνήμης εισέρχεται στην κατάσταση M (Τροποποιημένη), όταν ένας επεξεργαστής γράφει σε αυτή. Αν η γραμμή δεν βρίσκεται σε κατάσταση E (Αποκλειστική) ή M πριν από την εγγραφή, τότε η κρυφή μνήμη στέλνει μία αίτηση Ανάγνωσης Για Ιδιοκτησία (Read For Ownership-RFO), η οποία εξασφαλίζει ότι η γραμμή υπάρχει στην κρυφή μνήμη Επιπέδου 1, ενώ βρίσκεται σε κατάσταση I (Μη Έγκυρη) στην άλλη κρυφή μνήμη Επιπέδου 1. Ο επεξεργαστής Intel Core Duo επεκτείνει αυτό το πρωτόκολλο, ώστε να λαμβάνει υπόψη την περίπτωση όπου πολλαπλά chips Core Duo οργανώνονται σε ένα σύστημα συμμετρικών πολυεπεξεργαστών. Ο ελεγκτής της κρυφής μνήμης Επιπέδου 2 επιτρέπει στο σύστημα να κάνει διάκριση ανάμεσα σε μία κατάσταση όπου τα δεδομένα μοιράζονται ανάμεσα σε δύο τοπικούς πυρήνες, αλλά όχι με τον υπόλοιπο κόσμο, και σε μία κατάσταση, όπου τα δεδομένα μοιράζονται σε μία ή περισσότερες κρυφές μνήμες πάνω στη επιφάνεια του chip, αλλά και σε έναν πράκτορα που βρίσκεται στον εξωτερικό δίαυλο (μπορεί να είναι ένας άλλος επεξεργαστής). Όταν ένας πυρήνας εκδίδει μία αίτηση RFO, τότε, αν η γραμμή μοιράζεται μόνον από την άλλη τοπική κρυφή μνήμη εντός του chip, αυτή η αίτηση διευθετείται εσωτερικά και με μεγάλη ταχύτητα, χωρίς να είναι αναγκαίο να μεταβούμε στον εξωτερικό δίαυλο. Μόνο στην περίπτωση όπου η γραμμή μοιράζεται με άλλον πράκτορα στον εξωτερικό δίαυλο, είναι ανάγκη να εκδοθεί η αίτηση RFO εξωτερικά.

Η **διεπαφή διαύλου** συνδέεται με τον εξωτερικό δίαυλο, γνωστό και ως Εμπρόσθιο Δίαυλο, ο οποίος συνδέεται με την κύρια μνήμη, τους ελεγκτές E/E, και άλλα chips επεξεργαστών.

Ο Επεξεργαστής Intel Core i7

Ο επεξεργαστής Intel Core i7, ο οποίος εισήχθη το Νοέμβριο του 2008, περιλαμβάνει τέσσερις επεξεργαστές ταυτόχρονης πολυνημάτωσης, καθένας εκ των οποίων περιέχει μία δεσμευμένη κρυφή μνήμη Επιπέδου 2 και μία κοινόχρηστη κρυφή μνήμη Επιπέδου 3 (Σχήμα 18.8δ).

Η γενικότερη δομή του επεξεργαστή απεικονίζεται στο Σχήμα 18.10. Κάθε πυρήνας έχει τη δική



Σχήμα 18.10: Διάγραμμα μπλοκ του επεξεργαστή Intel Core i7

του **δεσμευμένη κρυφή μνήμη Επιπέδου 2** και οι τέσσερις πυρήνες μοιράζονται μία **κρυφή μνήμη Επιπέδου 3**, μεγέθους 8 MB. Ένας μηχανισμός, τον οποίο χρησιμοποιεί η Intel για να καταστήσει τις κρυφές της μνήμες πιο αποτελεσματικές, είναι η προανάκληση, όπου το υλικό εξετάζει τα υποδείγματα προσπέλασης της μνήμης και προσπαθεί να γεμίσει τις κρυφές μνήμες με δεδομένα τα οποία υποθέτει ότι θα ζητηθούν σύντομα. Παρουσιάζει ενδιαφέρον να συγκρίνουμε την απόδοση αυτής της μορφής οργάνωσης με τρία επίπεδα κρυφής μνήμης πάνω στο chip, με την οργάνωση δύο επιπέδων της Intel. Ο Πίνακας 18.1 απεικονίζει το λανθάνοντα χρόνο προσπέλασης της κρυφής μνήμης, με όρους των κύκλων ρολογιού, για δύο συστήματα πολλαπλών πυρήνων της Intel τα οποία χρησιμοποιούν την ίδια συχνότητα ρολογιού. Ο επεξεργαστής Core 2 Quad έχει μία κοινόχρηστη κρυφή μνήμη Επιπέδου 2, παρόμοια με εκείνη του Core Duo. Ο Core i7 βελτιώνει την απόδοση της κρυφής μνήμης Επιπέδου 2, χρησιμοποιώντας δεσμευμένες μνήμες και παρέχει σχετικά γρήγορη προσπέλαση της κρυφής μνήμης Επιπέδου 3.

Το chip του επεξεργαστή Core i7 υποστηρίζει δύο μορφές εξωτερικών επικοινωνιών προς άλλα chips. Ο **ελεγκτής μνήμης DDR** φέρνει στο chip τον ελεκτή για την κύρια μνήμη DDR.[2] Η διεπαφή υποστηρίζει τρία κανάλια εύρους 8 bytes, δίνοντας συνολικό εύρος διαύλου ίσο με 192 bits και συνολικό ρυθμό δεδομένων 32GB/s. Με τον ελεγκτή μνήμης πάνω στο chip, παραλείπεται ο εμπρόσθιος δίαυλος.

Η **διασύνδεση γρήγορου μονοπατιού** (QuickPath Interconnect-QPI), αποτελεί μία προδιαγραφή ηλεκτρικής διασύνδεσης από σημείο σε σημείο, με συνοχή της κρυφής μνήμης, την οποία χρησιμοποιούν οι επεξεργαστές και τα chips της Intel. Η QPI επιτρέπει τις επικοινωνίες με υψηλή ταχύτητα ανάμεσα στα συνδεδεμένα chips επεξεργαστών. Η διασύνδεση αυτή λειτουργεί με ταχύτητα 6.4GT/s (μεταφορές ανά δευτερόλεπτο). Με 16 bits ανά μεταφορά, η οποία σημαίνει 12.8 GB/s και δεδομένου ότι η QPI συμπεριλαμβάνει δεσμευμένα μονοπάτια δύο κατευθύνσεων, το συνολικό εύρος ζώνης είναι 25.6 GB/s.

[2] Η σύγχρονη μνήμη DDR RAM περιγράφηκε στο Κεφάλαιο 5.

Πίνακας 18.2: Οι επιλογές διάταξης του επεξεργαστή ARM11 MPCore

Χαρακτηριστικό	Εύρος Επιλογών	Προεπιλεγμένη Τιμή
Επεξεργαστές	1 ως 4	4
Μέγεθος κρυφής μνήμης εντολών ανά επεξεργαστή	16KB, 32KB, ή 64KB	32KB
Μέγεθος κρυφής μνήμης δεδομένων ανά επεξεργαστή	16KB, 32KB, ή 64KB	32KB
Θύρες αφέντη	1 ή 2	2
Εύρος διαύλου διακοπών	0 ως 224 με προσθήκες 32 ακροδεκτών	32 ακροδέκτες
Συνεπεξεργαστής αριθμών κινητής υποδιαστολής ανά επεξεργαστή (VFP)	Συμπεριλαμβάνεται ή όχι	Συμπεριλαμβάνεται

18.5 Ο ΕΠΕΞΕΡΓΑΣΤΗΣ ARM11 MPCORE

Ο επεξεργαστής ARM11 MPCore είναι ένα προϊόν πολλαπλών πυρήνων, το οποίο είναι βασισμένο στην οικογένεια επεξεργαστών ARM11 MPCore. Ο επεξεργαστής ARM11 MPCore είναι δυνατόν να διαμορφωθεί σε μία διάταξη αποτελούμενη από 4 επεξεργαστές, καθένας εκ των οποίων διαθέτει τη δική του κρυφή μνήμη εντολών και δεδομένων Επιπέδου 1. Ο Πίνακας 18.2 παραθέτει τις επιλογές διάταξης του συστήματος, όπου συμπεριλαμβάνονται οι προεπιλεγμένες τιμές.

Το Σχήμα 18.11 παρουσιάζει ένα λογικό διάγραμμα του επεξεργαστή ARM11 MPCore. Τα βασικά στοιχεία του συστήματος, είναι τα εξής:

- **Κατανεμημένος ελεγκτής διακοπών (Distributed Interrupt Controller-DIC):** Διαχειρίζεται την ανίχνευση των διακοπών, αλλά και και τις προτεραιότητές τους. Ο κατανεμημένος ελεγκτής διακοπών διανέμει τις διακοπές ανάμεσα στους επεξεργαστές.

- **Χρονόμετρο:** Κάθε επεξεργαστής διαθέτει το δικό του χρονόμετρο το οποίο έχει τη δυνατότητα να παράγει διακοπές.

- **Κέρβερος (Watchdog):** Εκδίδει προειδοποιήσεις στην περίπτωση βλαβών του λογισμικού. Αν είναι ενεργοποιημένος ένας κέρβερος, τότε λαμβάνει μία προκαθορισμένη τιμή και μετράει αντίστροφα ως το 0. Η τιμή μηδενίζεται περιοδικά. Αν η τιμή του κέρβερου μηδενιστεί, εκδίδεται η προειδοποίηση.

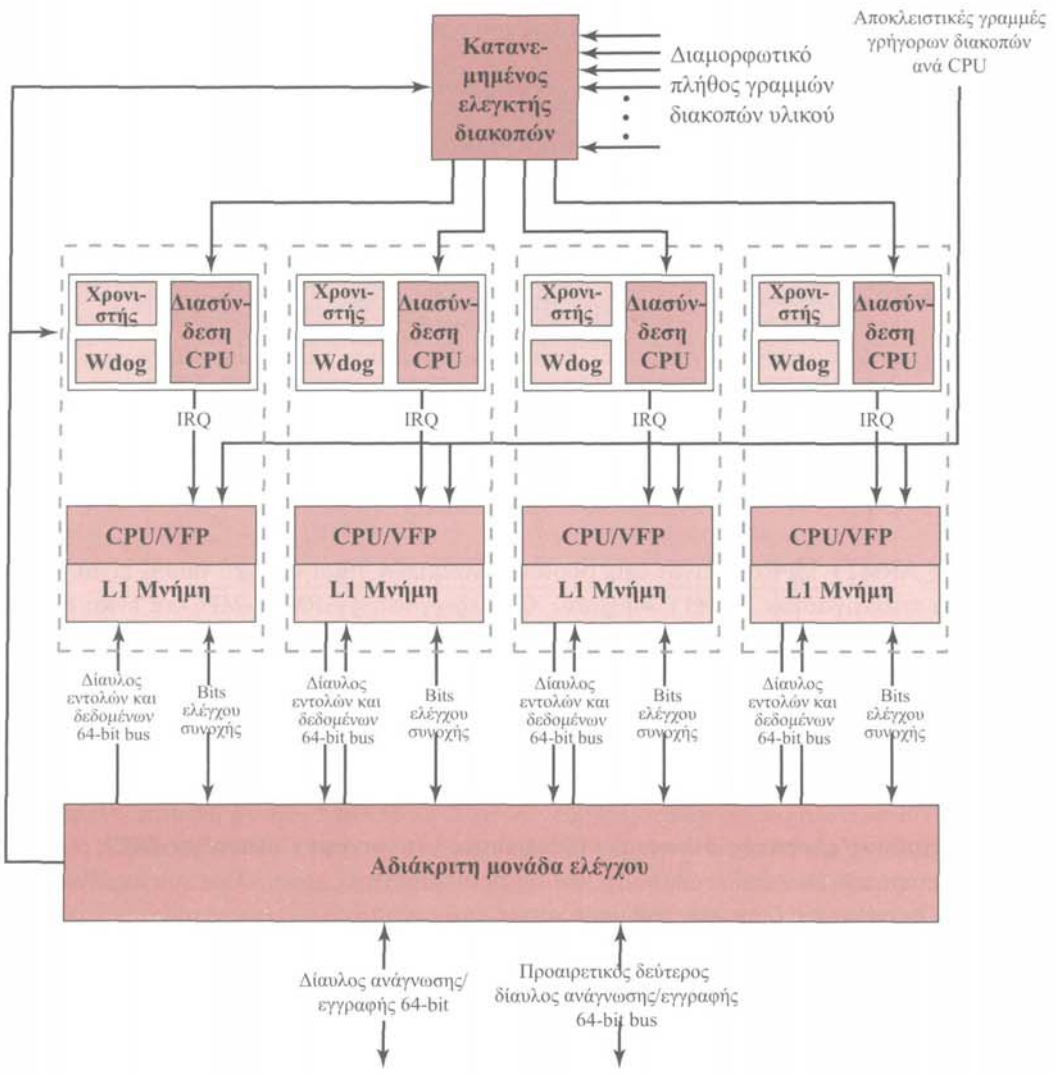
- **Διεπαφή επεξεργαστή:** Διαχειρίζεται τις επιβεβαιώσεις διακοπών, την απόκρυψή τους, αλλά και τις επιβεβαιώσεις ολοκλήρωσής τους.

- **Επεξεργαστής:** Ένας απλός επεξεργαστής ARM11. Οι ξεχωριστοί επεξεργαστές αναφέρονται και **επεξεργαστές ARM11**.

- **Μονάδα διανυσμάτων κινητής υποδιαστολής (Vector Floating Point-VFP):** Ένας συνεπεξεργαστής, ο οποίος υλοποιεί πράξεις κινητής υποδιαστολής στο υλικό.

- **Κρυφή μνήμη Επιπέδου 1:** Κάθε επεξεργαστής διαθέτει τη δική του δεσμευμένη κρυφή μνήμη δεδομένων και εντολών Επιπέδου 1.

- **Αδιάκριτη μονάδα ελέγχου (Snoop Control Unit):** Υπεύθυνη για τη διατήρηση της συνοχής ανάμεσα στις κρυφές μνήμες Επιπέδου 1.



Σχήμα 18.11: Διάγραμμα μπλοκ του του επεξεργαστή ARM11 MPCore

Διαχείριση Διακοπών

Ο κατανεμημένος ελεγκτής διακοπών συλλέγει διακοπές από ένα μεγάλο πλήθος πόρων και παρέχει:

- Απόκρυψη των διακοπών
- Ανάθεση προτεραιοτήτων στις διακοπές
- Διανομή των διακοπών στους επεξεργαστές προορισμούς
- Παρακολούθηση της κατάστασης των διακοπών
- Παραγωγή διακοπών με τη βοήθεια λογισμικού

Ο DIC είναι μία απλή λειτουργική μονάδα, η οποία είναι τοποθετημένη στο σύστημα παραπλεύρως των επεξεργαστών. Με τον τρόπο αυτό, επιτρέπεται στο πλήθος των υποστηριζόμενων από το σύστημα διακοπών να είναι ανεξάρτητες από τη σχεδίαση των επεξεργαστών MP11. Ο DIC απεικονίζεται στη

μνήμη. Με άλλα λόγια, οι καταχωρητές ελέγχου του DIC ορίζονται σε σχέση με μία διεύθυνση βάσης της κύριας μνήμης. Ο DIC προσπελαύνεται από τους επεξεργαστές με χρήση μίας ατομικής διεπαφής, μέσα από την αδιάκριτη μονάδα ελέγχου.

Ο DIC είναι σχεδιασμένος ώστε να εξυπηρετεί δύο λειτουργικές απαιτήσεις:

- Να παρέχει ένα μέσο δρομολόγησης μίας αίτησης διακοπών σε έναν ή πολλούς επεξεργαστές, όπως απαιτείται.

- Να παρέχει ένα μέσο επικοινωνίας ανάμεσα στους επεξεργαστές, έτσι ώστε ένα νήμα που εκτελείται σε έναν επεξεργαστή να έχει τη δυνατότητα να προκαλέσει κάποια δραστηριότητα νήματος σε έναν άλλο επεξεργαστή.

Ως παράδειγμα, το οποίο χρησιμοποιεί και τις δύο απαιτήσεις, θεωρείστε μία εφαρμογή πολλαπλών νημάτων, με νήματα τα οποία εκτελούνται σε πολλαπλούς επεξεργαστές. Έστω ότι η εφαρμογή κατανέμει κάποια ιδεατή μνήμη. Για να διατηρηθεί η συνοχή, το λειτουργικό σύστημα θα πρέπει να ενημερώνει τους πίνακες μετάφρασης σε όλους τους επεξεργαστές. Το λειτουργικό σύστημα θα μπορούσε να ενημερώσει τους πίνακες του επεξεργαστή όπου έλαβε χώρα η εκχώρηση ιδεατής μνήμης και στη συνέχεια, να εκδόσει μία διακοπή σε όλους τους άλλους επεξεργαστές οι οποίοι εκτελούν αυτή την εφαρμογή. Στη συνέχεια, οι άλλοι επεξεργαστές θα μπορούσαν να χρησιμοποιήσουν το αναγνωριστικό αυτής της διακοπής για να αποφασίσουν ότι είναι ανάγκη να ενημερώσουν τους δικούς τους πίνακες μετάφρασης.

Ο κατανεμημένος ελεγκτής διακοπών έχει τη δυνατότητα να δρομολογεί διακοπές προς έναν ή περισσότερους επεξεργαστές με τους ακόλουθους τρόπους:

- Μία διακοπή είναι δυνατόν να κατευθυνθεί προς έναν επεξεργαστή μόνον.

- Μία διακοπή είναι δυνατόν να κατευθυνθεί σε μία συγκεκριμένη ομάδα επεξεργαστών. Ο πολυεπεξεργαστής MPCore "βλέπει" τον πρώτο επεξεργαστή για να αποδεχτεί τη διακοπή, ως τον ευρισκόμενο στην καλύτερη θέση για να τη διαχειριστεί.

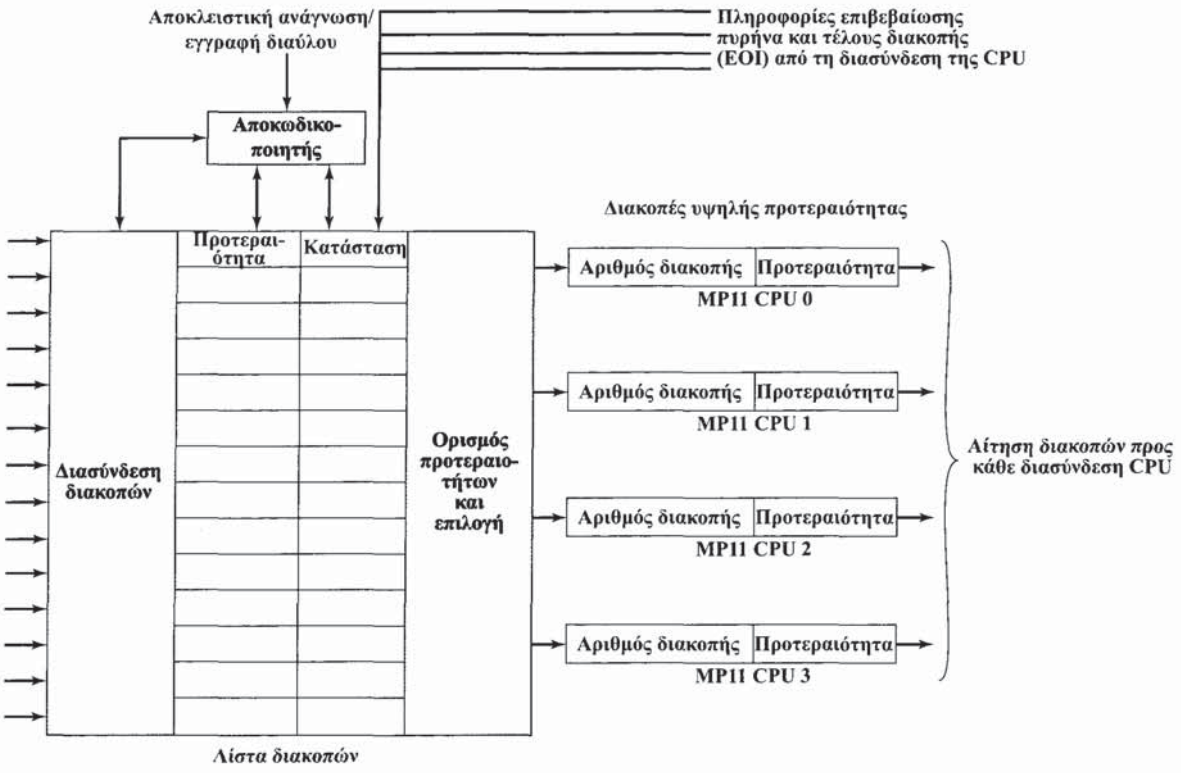
- Μία διακοπή είναι δυνατόν να κατευθυνθεί προς κάθε επεξεργαστή.

Από την πλευρά του λογισμικού το οποίο εκτελείται σε έναν επεξεργαστή, το λειτουργικό σύστημα μπορεί να παράγει μία διακοπή προς όλους τους επεξεργαστές εκτός εκείνου στον οποίο εκτελείται, ή σε συγκεκριμένους επεξεργαστές. Για την υλοποίηση της επικοινωνίας ανάμεσα στα νήματα τα οποία εκτελούνται σε διαφορετικούς επεξεργαστές, ο μηχανισμός διακοπών τυπικά συνδυάζεται με την ύπαρξη κοινόχρηστης μνήμης, ώστε να εξυπηρετείται το πέρασμα μηνυμάτων (Message Passing). Επομένως, όταν ένα νήμα διακόπτεται από μία διακοπή επικοινωνίας μεταξύ επεξεργαστών, διαβάζει από το κατάλληλο μπλοκ της κοινόχρηστης μνήμης, ώστε να εξάγει ένα μήνυμα από το νήμα το οποίο προκάλεσε τη διακοπή. Συνολικά, είναι διαθέσιμο ένα πλήθος από 16 αναγνωριστικά διακοπών ανά επεξεργαστή, προκειμένου να υλοποιείται η επικοινωνία μεταξύ των επεξεργαστών.

Από την πλευρά ενός επεξεργαστή MP11, μία διακοπή είναι δυνατόν να είναι:

- **Ανενεργή:** Μία ανενεργή διακοπή δεν έχει βεβαιωθεί, ή, έχει υποστεί πλήρη επεξεργασία από εκείνο τον επεξεργστή μέσα σε ένα περιβάλλον πολυεπεξεργασίας, όμως είναι πιθανό να παραμένει είτε σε εκκρεμότητα είτε ενεργή σε ορισμένους επεξεργαστές οι οποίοι αποτελούν προορισμούς της, με αποτέλεσμα να μην έχει αφαιρεθεί από την πηγή της.

- **Εκκρεμής:** Μία εκκρεμής διακοπή έχει βεβαιωθεί και η επεξεργασία της δεν έχει ξεκινήσει σε εκείνο τον επεξεργαστή.



Σχήμα 18.12: Διάγραμμα μπλοκ του διανομέα διακοπών

- **Ενεργής:** Μία ενεργή διακοπή είναι εκείνη η οποία έχει ξεκινήσει σε εκείνο τον επεξεργαστή, αλλά η επεξεργασία της δεν έχει ολοκληρωθεί. Μία ενεργή διακοπή μπορεί να απορριφθεί πριν ολοκληρωθεί η επεξεργασία της, αν μία νέα διακοπή με υψηλότερη προτεραιότητα διακόψει την επεξεργασία της σε κάποιον επεξεργαστή MP11.

Οι διακοπές προέρχονται από τις παρακάτω πηγές:

- **Διακοπές μεταξύ των επεξεργαστών:** Κάθε επεξεργαστής έχει τις δικές του διεκοπές (ID0-ID15), οι οποίες ενεργοποιούνται αποκλειστικά από λογισμικό. Η προτεραιότητα των διακοπών αυτών εξαρτάται από τον επεξεργαστή που λαμβάνει τη διακοπή και όχι από εκείνον που τη στέλνει.

- **Διακοπές που προέρχονται από το χρονόμετρο ή τον κέρβερο κάθε επεξεργαστή:** Πρόκειται για τις διακοπές με αριθμούς αναγνωριστικών 29 και 30.

- **Κληρονομική Γραμμή FIQ:** Σε κατάσταση κληρονομικότητας, ο ακροδέκτης FIQ, στη βάση κάθε επεξεργαστή, διαπερνά τα λογικά κυκλώματα του Διανομέα Διακοπών και οδηγεί άμεσα τις αιτήσεις διακοπών στον επεξεργαστή.

- **Διακοπές υλικού:** Οι διακοπές υλικού πυροδοτούνται από προγραμματιζόμενα γεγονότα, πάνω σε συσχετιζόμενες γραμμές εισόδου διακοπών. Οι επεξεργαστές έχουν τη δυνατότητα να υποστηρίξουν μέχρι και 224 γραμμές εισόδου διακοπών. Οι διακοπές υλικού ξεκινούν από τον αριθμό αναγνωριστικού 32.

Το Σχήμα 18.12 απεικονίζει ένα λογικό διάγραμμα του DIC. Ο DIC μπορεί να διαμορφωθεί κατά τέτοιο τρόπο, ώστε να υποστηρίζει από 0- 255 εισόδους διακοπών υλικού. Ο DIC διατηρεί μία λίστα

διακοπών η οποία δείχνει την προτεραιότητα και την κατάστασή τους. Ο Διανομέας Διακοπών μεταδίδει προς κάθε διεπαφή επεξεργαστή την υψηλότερης προτεραιότητας εκκρεμή διακοπή για αυτή τη διεπαφή. Επίσης, λαμβάνει πίσω τις πληροφορίες επιβεβαίωσης της διακοπής, και έχει τη δυνατότητα να αλλάξει την κατάσταση της αντίστοιχης διακοπής. Επίσης, η διεπαφή του επεξεργαστή μεταδίδει πληροφορίες τερματισμού διακοπής (End of Interrupt-EOI), οι οποίες επιτρέπουν στο Διανομέα Διακοπών να ενημερώσει την κατάσταση αυτής της διακοπής, αλλάζοντάς την από Ενεργή σε Ανενεργή.

Συνοχή της Κρυφής Μνήμης

Η αδιάκριτη μονάδα ελέγχου (SCU) του επεξεργαστή ARM11 MPCore είναι σχεδιασμένη ώστε να επιλύει τα περισσότερα από τα κλασικά προβλήματα συμφόρησης τα οποία σχετίζονται με την προσπέλαση σε κοινόχρηστα δεδομένα και με τον περιορισμό της κλιμάκωσης τον οποίο εισάγει η κυκλοφορία που συνδέεται με τη συνοχή ανάμεσα στις κρυφές μνήμες.

Η τεχνική διατήρησης συνοχής της κρυφής μνήμης Επιπέδου 1 βασίζεται στο πρωτόκολλο MESI το οποίο περιγράφηκε στο Κεφάλαιο 17. Η μονάδα SCU παρακολουθεί τα κοινόχρηστα δεδομένα των λειτουργιών, ώστε να βελτιστοποιείται η αλλαγή καταστάσεων του πρωτοκόλλου MESI. Η μονάδα SCU εισάγει τρεις μορφές βελτιστοποίησης: άμεση παρεμβολή δεδομένων, RAMs διπλότυπων ετικετών, και γραμμές μετακόμισης.

Η **άμεση παρεμβολή δεδομένων** επιτρέπει την αντιγραφή καθαρών δεδομένων από την κρυφή μνήμη Επιπέδου 1 ενός επεξεργαστή στην κρυφή μνήμη Επιπέδου 1 ενός άλλου, χωρίς να προσπελαύνεται η εξωτερική μνήμη. Με τον τρόπο αυτό, οι λειτουργίες ανάγνωσης μετά από ανάγνωση ανάγονται στην κρυφή μνήμη Επιπέδου 2 και όχι στην κρυφή μνήμη Επιπέδου 1. Επομένως, μία αστοχία σε τοπική κρυφή μνήμη Επιπέδου 1 θα επιλυθεί σε μία τοπική κρυφή μνήμη Επιπέδου 1 και όχι μέσω προσπέλασης σε μία κοινόχρηστη κρυφή μνήμη Επιπέδου 2.

Θυμηθείτε ότι η θέση της κύριας μνήμης κάθε γραμμής η οποία βρίσκεται στην κρυφή μνήμη, προσδιορίζεται από μία ετικέτα που αντιστοιχεί στη γραμμή αυτή. Οι ετικέτες είναι δυνατόν να υλοποιηθούν σε ένα ξεχωριστό μπλοκ της μνήμης RAM, με μήκος ίδιο με το πλήθος των γραμμών της κρυφής μνήμης. Στη μονάδα SCU, **οι RAMs διπλότυπων ετικετών** αποτελούν αντίγραφα των RAMs ετικετών της κρυφής μνήμης Επιπέδου 1, οι οποίες χρησιμοποιούνται από τη μονάδα SCU για να ελέγξει τη διαθεσιμότητα των δεδομένων πριν στείλει εντολές διατήρησης της συνοχής στους αντίστοιχους επεξεργαστές. Οι εντολές αυτές στέλνονται μόνον στους επεξεργαστές οι οποίοι θα πρέπει να ενημερώσουν τις κρυφές μνήμες τους με δεδομένα τέτοια ώστε να διατηρείται η συνοχή. Με τον τρόπο αυτό μειώνεται η κατανάλωση ισχύος και η επίδραση στην απόδοση, οι οποίες προκύπτουν λόγω της αδιάκριτης εξέτασης και χειρισμού της κρυφής μνήμης κάθε επεξεργαστή, κάθε φορά που ενημερώνεται η μνήμη. Διαθέτοντας δεδομένα ετικετών τοπικά, επιτρέπουμε στη μονάδα SCU να περιορίσει τους χειρισμούς της κρυφής μνήμης, μόνον στους επεξεργαστές οι οποίοι έχουν κοινές μεταξύ τους γραμμές.

Οι **γραμμές μετακόμισης**, επιτρέπουν τη μετακίνηση βρώμικων δεδομένων από έναν επεξεργαστή σε έναν άλλο, χωρίς να γίνεται εγγραφή στην κρυφή μνήμη Επιπέδου 2, αλλά και την ανάγνωση των δεδομένων από την εξωτερική μνήμη. Η λειτουργία περιγράφεται ως ακολούθως: Σε ένα τυπικό πρωτόκολλο MESI, ένας επεξεργαστής έχει μία γραμμή σε κατάσταση M (Τροποποιημένη) και ένας άλλος προσπαθεί να διαβάσει αυτή τη γραμμή. Τότε, εκτελούνται οι ακόλουθες λειτουργίες:

1. Τα περιεχόμενα της γραμμής μεταφέρονται από την τροποποιημένη γραμμή στον επεξεργαστή ο οποίος εκκίνησε τη λειτουργία ανάγνωσης.

2. Τα περιεχόμενα της γραμμής διαβάζονται και γράφονται πίσω στην κύρια μνήμη.

3. Σε αμφότερες τις κρυφές μνήμες, η γραμμή τοποθετείται στην κατάσταση S (Κοινόχρηστη).

Η μονάδα SCU του επεξεργαστή MPCore διαχειρίζεται αυτή την κατάσταση με διαφορετικό τρόπο. Πιο συγκεκριμένα, παρακολουθεί το σύστημα για να εξετάσει αν υπάρχει γραμμή μετακόμισης. Αν ένας επεξεργαστής έχει μία γραμμή σε κατάσταση M και ένας άλλος τη διαβάσει και έπειτα γράψει σε αυτή, η μονάδα υποθέτει ότι αυτή η θέση θα υποστεί την ίδια λειτουργία στο μέλλον. Όταν αυτή η λειτουργία ξεκινήσει εκ νέου, η μονάδα SCU θα μετακινήσει αυτομάτως τη γραμμή της κρυφής μνήμης σε κατάσταση I (μη έγκυρη) αντί να δαπανήσει ενέργεια μετακινώντας την πρώτα σε κατάσταση S (Κοινόχρηστη). Αυτή η βελτίωση αναγκάζει επίσης τον επεξεργαστή να μεταφέρει τη γραμμή της κρυφής μνήμης, άμεσα στον άλλο επεξεργαστή, χωρίς να παρεμβάλλονται ενδιάμεσες λειτουργίες της εξωτερικής μνήμης.

18.6 ΠΡΟΤΕΙΝΟΜΕΝΗ ΜΕΛΕΤΗ

Δύο βιβλία τα οποία παρέχουν καλή κάλυψη των ζητημάτων που τέθηκαν σε αυτό το κεφάλαιο, είναι τα [OLUK07] και [JERR05]. Στα άρθρα [GOCH06] και [MEND06] περιγράφεται ο επεξεργαστής Intel Core Duo. Στην αναφορά [FOG08b] περιγράφεται λεπτομερώς η αρχιτεκτονική της διασωλήνωσης του Core Duo.

Στην αναφορά [ARM08b] καλύπτεται ικανοποιητικά η διασωλήνωση του επεξεργαστή ARM Cortex-A8. Τα [HIRA07] και [GOOD05] παρέχουν καλές γενικές θεωρήσεις πάνω στα ζητήματα αυτού του κεφαλαίου.

ARM08b ARM Limited. *ARM11 MPCore Processor Technical Reference Manual.* ARM DDI 0360E, 2008. www.arm.com

FOG08b Fog, A. *The Microarchitecture of Intel and AMD CPUs.* Copenhagen University College of Engineering, 2008. http://www.agner.org/optimize/

GOCH06 Gochman, S., et al. "Introduction to Intel Core Duo Processor Architecture." *Intel Technology Journal,* May 2006.

GOOD05 Goodacre, J., and Sloss, A. "Parallelism and the ARM Instruction Set Architecture." *Computer,* July 2005.

HIRA07 Hirata, K., and Goodacre, J. "ARM MPCore: The Streamlined and Scalable ARM11 processor core." *Proceedings, 2007 Conference on Asia South Pacific Design Automation,* 2007.

JERR05 Jerraya, A., and Wolf,W., eds. *Multiprocessor Systems-on-Chips.* San Francisco: Morgan Kaufmann, 2005.

MEND06 Mendelson, A., et al. "CMP Implementation in Systems Based on the Intel Core Duo Processor." *Intel Technology Journal,* May 2006.

OLUK07 Olukotun, K.; Hammond, L.; and Laudon, J *Chip Multiprocessor Architecture: Techniques to Improve Throughput and Latency.* San Rafael, CA: Morgan & Claypool, 2007.

Προτεινόμενοι Διαδικτυακοί Τόποι:

- **Multicore Association:** Οργάνωση κατασκευαστών η οποία προωθεί την ανάπτυξη και χρήση. της τεχνολογίας πολλαπλών πυρήνων.

18.7 ΒΑΣΙΚΟΙ ΟΡΟΙ, ΕΡΩΤΗΣΕΙΣ ΕΠΑΝΑΛΗΨΗΣ ΚΑΙ ΠΡΟΒΛΗΜΑΤΑ

Βασικοί Όροι

chip πολυεπεξεργαστών	πολλαπλοί πυρήνες	υπερβαθμωτός επεξεργαστής
νόμος του Amdahl	ταυτόχρονη πολυνημάτωση	

Ερωτήσεις Επανάληψης

18.1. Να συνοψίσετε τις διαφορές ανάμεσα στην απλή διασωλήνωση εντολών, την υπερβαθμωτή και την ταυτόχρονη υπερνημάτωση.

18.2. Να δώσετε διάφορες αιτίες για τις οποίες οι σχεδιαστές αποφάσισαν να μετακινηθούν προς την οργάνωση πολλαπλών πυρήνων αντί να αυξήσουν τον παραλληλισμό μέσα σε έναν επεξεργαστή.

18.3. Γιατί υπάρχει η τάση προς την αυξανόμενη εκχώρηση μέρους της επιφάνειας του chip στην κρυφή μνήμη;

18.4. Να παραθέσετε ορισμένα παραδείγματα εφαρμογών οι οποίες ωφελούνται άμεσα από τη δυνατότητα αύξησης της ρυθμαπόδοσης όταν αυξάνεται το πλήθος των πυρήνων.

18.5. Στο ανώτερο επίπεδο, ποιές είναι οι βασικές μεταβλητές της σχεδίασης μίας μορφής οργάνωσης πολλαπλών πυρήνων;

18.6. Να παραθέστε ορισμένα πλεονεκτήματα της κοινόχρηστης μεταξύ των πυρήνων κρυφής μνήμης Επιπέδου 2, σε σύγκριση με την ύπαρξη κοινής κρυφής μνήμης Επιπέδου 2 σε κάθε πυρήνα.

Προβλήματα

18.1 Θεωρείστε το παρακάτω πρόβλημα. Ένας σχεδιαστής έχει στη διάθεσή του ένα chip και αποφασίζει τι ποσοστό της επιφάνειάς του θα δώσει στην κρυφή μνήμη (Επιπέδου 1, 2, και 3). Το υπόλοιπο του chip θα χρησιμοποιηθεί για έναν πολύπλοκο υπερβαθμωτό επεξεργαστή ή για έναν πυρήνα ταυτόχρονης πολυνημάτωσης, ή για πολλαπλούς πυρήνες κάπως πιο απλής μορφής. Ορίζονται οι παρακάτω παράμετροι:

n = μέγιστο πλήθος των πυρήνων που μπορεί να περιέχει ένα chip

k = πραγματικό πλήθος των πυρήνων($1 \le k \le n$, όπου ο αριθμός $r = n/k$ είναι ακέραιος)

$perf(r)$ = κέρδος ακολουθιακής απόδοσης το οποίο προκύπτει από τη χρήση των πόρων που ισοδυναμούν σε r πυρήνες, ώστε να σχηματιστεί ένας απλός πυρήνας, όπου $perf(1) = 1$.

f = ποσοστό του λογισμικού, το οποίο είναι παραλληλοποιήσιμο σε πολλούς πυρήνες.

Επομένως, αν κατασκευάσουμε ένα chip με n πυρήνες, αναμένουμε ότι κάθε πυρήνας θα δίνει απόδοση 1 και ότι οι n πυρήνες θα έχουν τη δυνατότητα να εκμεταλλευτούν τον παραλληλισμό μέχρι το βαθμό n παράλληλων νημάτων. Ομοίως, αν το chip έχει k πυρήνες, τότε κάθε πυρήνας θα πρέπει να παρουσιάσει απόδοση $perf(r)$ και το chip θα έχει τη δυνατότητα εκμετάλλευσης του παραλληλισμού μέχρι το βαθμό k παράλληλων νημάτων. Μπορούμε να τροποποιήσουμε το νόμο του Amdahl (Εξίσωση 18.1), ώστε να απεικονίζει την κατάσταση αυτή ως ακολούθως:

$$\text{Αύξηση ταχύτητας} = \frac{1}{\frac{1-f}{perf(r)} + \frac{f \times r}{perf(r) \times n}}$$

α. Να δικαιολογήσετε την παραπάνω τροποποίηση του νόμου του Amdahl

β. Χρησιμοποιώντας τον κανόνα του Pollack, θέτουμε $perfr = \sqrt{r}$. Θέτουμε $n = 16$. Επιθυμούμε να κατασκευάσουμε ένα διάγραμμα της αύξησης της ταχύτητας ως συνάρτηση του r, για $f = 0.5$, $f = 0.9$, $f = 0.975$, $f = 0.99$, $f = 0.999$. Τα αποτελέσματα είναι διαθέσιμα σε ένα έγγραφο που βρίσκεται στο δικτυακό τόπο του βιβλίου (multicore-performance.pdf).

γ. Να επαναλάβετε το ερώτημα (β) για $n = 256$.

18.2 Το τεχνικό φυλλάδιο του επεξεργαστή ARM11 MPCore αναφέρει ότι ο Κατανεμημένος Ελεγκτής Διακοπών απεικονίζεται στη μνήμη. Με άλλα λόγια, οι πυρήνες χρησιμοποιούν Ε/Ε με απεικόνιση στη μνήμη, προκειμένου να επικοινωνήσουν με τον DIC. Από το Κεφάλαιο 7, θυμηθείτε ότι όταν χρησιμοποιείται Ε/Ε με απεικόνιση στη μνήμη, υπάρχει ένας μόνο χώρος διευθύνσεων για τις θέσεις μνήμης και τις μονάδες Ε/Ε. Ο επεξεργαστής διαχειρίζεται τους καταχωρητές κατάστασης και δεδομένων των μονάδων Ε/Ε ως θέσεις μνήμης και χρησιμοποιεί τις ίδιες εντολές μηχανής για να προσπελάσει τόσο τη μνήμη, όσο και τις μονάδες Ε/Ε. Βάσει αυτών των πληροφοριών, ποιό μονοπάτι στο λογικό διάγραμμα του Σχήματος 18.11 χρησιμοποιείται ώστε οι πυρήνες να επικοινωνήσουν με τον DIC;

ΠΑΡΑΡΤΗΜΑ Α
ΕΡΓΑΣΙΕΣ ΓΙΑ ΤΗ ΔΙΔΑΣΚΑΛΙΑ ΤΗΣ ΟΡΓΑΝΩΣΗΣ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Α.1 Αλληλεπιδραστικές Προσομοιώσεις

Α.2 Ερευνητικές Εργασίες

Α.3 Εργασίες Προσομοίωσης
Το Εργαλείο SimpleScalar
Το Εργαλείο SMPCache

Α.4 Εργασίες σε Γλώσσα Συμβολομετάφρασης

Α.5 Αναθέσεις Εργασιών Μελέτης/Συγγραφής Αναφοράς

Α.6 Γραπτές Εργασίες

Α.7 Ερωτήσεις Εξέτασης

Πολλοί καθηγητές πιστεύουν ότι οι ερευνητικές εργασίες ή οι εργασίες υλοποίησης είναι πολύ σημαντικές στην κατανόηση των εννοιών της οργάνωσης και αρχιτεκτονικής των επεξεργαστών. Χωρίς εργασίες, είναι πιθανόν δύσκολο για τους φοιτητές να κατανοήσουν ορισμένες από τις βασικές έννοιες αλλά και αλληλεπιδράσεις μεταξύ των στοιχείων. Οι εργασίες ενισχύουν τις έννοιες οι οποίες εισάγονται σε αυτό το βιβλίο, δίνουν στους φοιτητές τη δυνατότητα να εκτιμήσουν καλύτερα τις εσωτερικές εργασίες των επεξεργαστών και των υπολογιστικών συστημάτων, τους δίνουν κίνητρα και εμπιστοσύνη στον εαυτό τους, ότι έχουν καταφέρει να κατανοήσουν το υλικό.

Σε αυτό το βιβλίο, προσπάθησα να παρουσιάσω τις έννοιες της οργάνωσης και αρχιτεκτονικής υπολογιστών με όσο το δυνατόν μεγαλύτερη ακρίβεια και έδωσα ένα πλήθος από εργασίες για το σπίτι προκειμένου να ενισχύσω τις έννοιες αυτές. Πολλοί διδάσκοντες επιθυμούν να εφοδιάσουν το υλικό αυτό με εργασίες. Στο παράρτημα αυτό δίνονται μερικές οδηγίες ως προς αυτό το ζήτημα και περιγράφεται το υποστηρικτικό υλικό το οποίο είναι διαθέσιμο στο εγχειρίδιο του καθηγητή. Το υποστηρικτικό υλικό καλύπτει έξι μορφές εργασιών και άλλων ασκήσεων:

- Αλληλεπιδραστικές προσομοιώσεις
- Ερευνητικές εργασίες
- Εργασίες προσομοίωσης
- Εργασίες σε γλώσσα συμβολομετάφρασης
- Αναθέσεις εργασιών μελέτης/συγγραφής αναφοράς
- Γραπτές εργασίες
- Ερωτήσεις εξέτασης

Α.1 ΑΛΛΗΛΕΠΙΔΡΑΣΤΙΚΕΣ ΠΡΟΣΟΜΟΙΩΣΕΙΣ

Η ενσωμάτωση εργασιών αλληλεπιδραστικής προσομοίωσης είναι νέα σε αυτή την έκδοση. Αυτές οι προσομοιώσεις παρέχουν ένα ισχυρό εργαλείο για την κατανόηση των πολύπλοκων χαρακτηριστικών της σχεδίασης ενός σύγχρονου υπολογιστικού συστήματος. Οι σημερινοί φοιτητές επιθυμούν να έχουν τη δυνατότητα οπτικοποίησης στην οθόνη τους, των διαφόρων πολύπλοκων μηχανισμών ενός υπολογιστικού συστήματος. Χρησιμοποιείται ένα σύνολο από 20 προσομοιώσεις, ώστε να παρουσιαστούν οι βασικές λειτουργίες και αλγόριθμοι της οργάνωσης και σχεδίασης της αρχιτεκτονικής του υπολογιστή. Ο Πίνακας Α.1 παραθέτει τις προσομοιώσεις ανά κεφάλαιο. Στο αντίστοιχο σημείο του βιβλίου, υπάρχει μία εικόνα η οποία δείχνει ότι μία αντίστοιχη αλληλεπιδραστική προσομοίωση διατίθεται για το φοιτητή, στο δικτυακό τόπο του βιβλίου.

Επειδή οι προσομοιώσεις επιτρέπουν στο χρήστη να ορίζει τις αρχικές συνθήκες, μπορούν να χρησιμοποιηθούν ως βάση για την ανάθεση εργασιών προς αυτούς. Το Κέντρο Πηγών Διδάσκοντα (Instructor Source Center-IRC) που αφορά αυτό το βιβλίο, συμπεριλαμβάνει ένα σύνολο από εργασίες, μία για κάθε αλληλεπιδραστική προσομοίωση. Κάθε εργασία συμπεριλαμβάνει διάφορα προβλήματα, στα οποία μπορούν να εργαστούν οι φοιτητές.

Οι αλληλεπιδραστικές προσομοιώσεις αναπτύχθηκαν υπό την καθοδήγηση του Καθηγητή Israel Koren, του Πανεπιστημίου της Μασαχουσέτης, Τμήμα Ηλεκτρονικής και Μηχανικής Υπολογιστών, Ο Aswin Sreedhar, από το ίδιο Πανεπιστήμιο, ανέπτυξε τις εργασίες πάνω στα θέματα της αλληλεπιδραστικής προσομοίωσης.

Πίνακας Α.1: Οργάνωση και αρχιτεκτονική υπολογιστών- Αλληλεπιδραστικές προσομοιώσεις ανά κεφάλαιο

Κεφάλαιο 4-Κρυφή Μνήμη	
Προσομοιωτής κρυφής μνήμης	Προσομοιώνει κρυφές μνήμες μικρού μεγέθους με βάση ένα μοντέλο κρυφής μνήμης το οποίο ορίζει ο χρήστης και απεικονίζει τα περιεχόμενα της κρυφής μνήμης στο τέλος κάθε κύκλου της προσομοίωσης, βάσει της ακολουθίας εισόδων, τις οποίες, είτε εισάγει ο χρήστης είτε παράγονται τυχαία, αν επιλεγεί κάτι τέτοιο.
Χρονική ανάλυση της κρυφής μνήμης	Παρουσιάζει την ανάλυση του μέσου χρόνου προσπέλασης της κρυφής μνήμης για τις παραμέτρους οι οποίες έχουν καθοριστεί.
Παρουσίαση λειτουργίες της κρυφής μνήμης σε επίπεδο πολυδιεργασιών	Μοντελοποιεί την κρυφή μνήμη σε ένα πολυδιεργασιακό σύστημα
Επιλεκτικός προσομοιωτής θύματος κρυφής μνήμης	Συγκρίνει τρεις διαφορετικές τεχνικές υλοποίησης της κρυφής μνήμης.
Κεφάλαιο 5-Εσωτερική Μνήμη	
Προσομοιωτής διαφυλλωμένης μνήμης	Δείχνει τις συνέπειες της χρήσης μίας διαφυλλωμένης μνήμης
Κεφάλαιο 6-Εξωτερική Μνήμη	
RAID	Προσδιορισμός της αποτελεσματικότητας και αξιοπιστίας της αποθήκευσης
Κεφάλαιο 7-Είσοδος/Έξοδος	
Εργαλείο σχεδίασης συστήματος Ε/Ε	Αποτιμά το συγκριτικό κόστος και απόδοση των διαφορετικών συστημάτων Ε/Ε
Κεφάλαιο 8-Υποστήριξη του Λειτουργικού Συστήματος	
Αλγόριθμοι αντικατάστασης σελίδων	Σύγκριση των αλγορίθμων LRU, LIFO, και βέλτιστου.
Περισσότεροι αλγόριθμοι αντικατάστασης σελίδων.	Σύγκριση ενός πλήθους τεχνικών.
Κεφάλαιο 12-Δομή του Επεξεργαστή και Λειτουργία	
Αναλυτής πίνακα κρατήσεων	Αποτίμηση των πινάκων κρατήσεων, οι οποίοι αποτελούν έναν τρόπο αναπαράστασης της ροής εργασιών μίας διασωλήνωσης.
Πρόβλεψη διακλάδωσης	Δείχνει τρεις διαφορετικές τεχνικές πρόβλεψης των διακλαδώσεων.
Απομονωτής προορισμού διακλάδωσης	Συνδυαστικός προσομοιωτής εργαλείου πρόβλεψης διακλαδώσεων/απομονωτή προορισμού διακλάδωσης.
Κεφάλαιο 13-Υπολογιστές Μειωμένου Συνόλου Εντολών	
Διασωλήνωση 5 σταδίων ενός επεξεργαστή MIPS	Προσομοίωση της διασωλήνωσης.
Ξετύλιγμα βρόχων	Προσομοιώνει την τεχνική λογισμικού για το ξετύλιγμα των βρόχων με την οποία εκμεταλλευόμαστε τον παραλληλισμό σε επίπεδο εντολής
Κεφάλαιο 14-Παραλληλισμός σε Επίπεδο Εντολής και Υπερβαθμωτοί Επεξεργαστές	
Σύγκριση διασωλήνωσης με στατική και δυναμική σχεδίαση	Μία πιο πολύπλοκη προσομοίωση της διασωλήνωσης ενός επεξεργαστή MIPS
Προσομοιωτής απομονωτή αναδιάταξης	Προσομοιώνει την αναδιάταξη των εντολών σε μία διασωλήνωση, για κάποιον επεξεργαστή αρχιτεκτονικής RISC
Τεχνική διατήρησης αποτελεσμάτων για δυναμική σχεδίαση	Προσομοίωση μίας τεχνικής σχεδίασης εντολών, η οποία χρησιμοποιείται από ένα πλήθος επεξεργαστών
Αλγόριθμος του Tomasulo	Προσομοίωση μίας άλλης τεχνικής σχεδίασης εντολών
Μία εναλλακτική προσομοίωση του αλγορίθμου του Tomasulo	*Μία άλλη προσομοίωση του αλγορίθμου του Tomasulo*
Κεφάλαιο 17-Παράλληλη Επεξεργασία	
Προσομοίωση επεξεργαστή διανυσμάτων	Παρουσιάζει την εκτέλεση των εντολών επεξεργασίας διανυσμάτων.

Α.2 ΕΡΕΥΝΗΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Ένας αποτελεσματικός τρόπος ενίσχυσης των βασικών εννοιών του μαθήματος, αλλά και διδασκαλίας του τρόπου ανάπτυξης των ερευνητικών δεξιοτήτων των φοιτητών, είναι η ανάθεση μίας ερευνητικής εργασίας. Μία τέτοια εργασία θα μπορούσε να συμπεριλαμβάνει την αναζήτηση στην αρθρογραφία, αλλά και την αναζήτηση προϊόντων κατασκευαστών στο Διαδίκτυο, δραστηριότητες μέσα σε ένα ερευνητικό εργαστήριο, και προσπάθειες προτυποποίησης. Οι εργασίες είναι δυνατόν να ανατεθούν σε ομάδες, ή, για πιο μικρές εργασίες, σε ένα άτομο. Σε κάθε περίπτωση, είναι προτιμότερο να απαιτείται κάποια μορφή υποβολής πρότασης στην αρχή του εξαμήνου, ώστε να υπάρχει αρκετός χρόνος για το διδάσκοντα, προκειμένου να αξιολογήσει την πρόταση όσον αφορά την καταλληλότητα του αντικειμένου, αλλά και την απαιτούμενη προσπάθεια. Για μία ερευνητική εργασία, θα πρέπει να δίνονται στο φοιτητή τα ακόλουθα:

- Μία μορφή της υποβολής πρότασης
- Μία μορφή της τελικής αναφοράς
- Ένα χρονοδιάγραμμα με τις ενδιάμεσες και τελικές προθεσμίες
- Μία λίστα με πιθανά αντικείμενα έρευνας

Οι φοιτητές μπορούν να επιλέξουν ένα από τα θέματα, ή να σχεδιάσουν τη δική τους ερευνητική εργασία. Το Κέντρο Πηγών Διδάσκοντα συμπεριλαμβάνει μία προτεινόμενη μορφή για την υποβολή της αρχικής πρότασης και της τελικής αναφοράς, αλλά και μία λίστα με πιθανά θέματα.

Α.3 ΕΡΓΑΣΙΕΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

Ένας εξαιρετικός τρόπος για να κατανοήσει ο φοιτητής την εσωτερική λειτουργία ενός επεξεργαστή, να μελετήσει και να εκτιμήσει ορισμένους από τους συμβιβασμούς και τις συνέπειες στην απόδοση, είναι μέσα από την προσομοίωση των βασικών στοιχείων του επεξεργαστή. Δύο βασικά εργαλεία τα οποία είναι χρήσιμα για το σκοπό αυτό είναι το Superscalar και το SMPCache.

Σε σύγκριση με την πραγματική υλοποίηση του υλικού, η προσομοίωση παρέχει δύο πλεονεκτήματα για ερευνητικούς και εκπαιδευτικούς σκοπούς:

- Με την προσομοίωση, είναι εύκολο να τροποποιηθούν τα διάφορα στοιχεία μίας μορφής οργάνωσης, να διαφοροποιηθούν τα χαρακτηριστικά της απόδοσης των διαφόρων στοιχείων, και τέλος να αναλυθούν οι επιδράσεις αυτών των τροποποιήσεων.
- Η προσομοίωση παρέχει μία λεπτομερή συλλογή στατιστικών στοιχείων σχετικά με την απόδοση, τα οποία είναι δυνατόν να χρησιμοποιηθούν από το φοιτητή, για να κατανοήσει τους συμβιβασμούς που γίνονται σε σχέση με την απόδοση.

Το Εργαλείο Superscalar

Το εργαλείο [BURG97, MANJ01a, MANJ01b] είναι ένα σύνολο εργαλείων, το οποίο είναι δυνατόν να χρησιμοποιηθεί για να προσομοιωθούν πραγματικά προγράμματα σε μία περιοχή σύγχρονων επεξεργαστών και συστημάτων. Συμπεριλαμβάνεται ένας μεταγλωττιστής, συμβολομεταφραστής, εργαλείο διασύνδεσης, καθώς και εργαλεία προσομοίωσης και οπτικοποίησης. Το εργαλείο Superscalar παρέχει προσομοιωτές επεξεργαστών, οι οποίοι εκτείνονται από έναν εξαιρετικά υψηλής ταχύτητας λειτουργικό επεξεργαστή, μέχρι έναν λεπτομερή προσομοιωτή υπερβαθμωτού επεξεργαστή έκδοσης εντολών εκτός σειράς, ο οποίος υποστηρίζει κρυφές μνήμες χωρίς παρεμπόδιση, αλλά και θεωρητική

εκτέλεση των εντολών. Η αρχιτεκτονική του συνόλου των εντολών και οι παράμετροι οργάνωσης, είναι δυνατόν να τροποποιηθούν, ώστε να δημιουργηθεί μία ποικιλία από πειράματα.

Το Κέντρο Πηγών Διδάσκοντα συμπεριλαμβάνει μία συμπαγή εισαγωγή στο εργαλείο Superscalar, με εντολές σχετικά με τον τρόπο φόρτωσης και εκκίνησης του προγράμματος. Επίσης, το εγχειρίδιο περιλαμβάνει και ορισμένες προτεινόμενες εργασίες.

Το εργαλείο Superscalar είνα ένα φορητό λογισμικό, το οποίο εκτελείται στις περισσότερες πλατφόρμες UNIX. Το λογισμικό Superscalar διατίθεται για χωρίς κόστος λήψη μέσα από το Δικτυακό Τόπο του, όταν πρόκειται να χρησιμοποιηθεί για μη εμπορικούς σκοπούς.

Το Εργαλείο SMPCache

Το εργαλείο SMPCache είναι ένας προσομοιωτής καθοδηγούμενος από ίχνη, ο οποίος χρησιμοποιείται για την ανάλυση και διδασκαλία των συστημάτων μνήμης των συμμετρικών πολυεπεξεργαστών [RODR01]. Η προσομοίωση βασίζεται σε ένα μοντέλο το οποίο δημιουργείται βάσει των βασικών αρχών της αρχιτεκτονικής αυτών των συστημάτων. Ο προσομοιωτής διαθέτει ένα πλήρως γραφικό και φιλικό προς το χρήστη περιβάλλον. Ορισμένες από τις παραμέτρους τις οποίες μπορεί να μελετήσει κανείς μέσα από τον προσομοιωτή, είναι οι ακόλουθες: τοπικότητα, επίδραση του πλήθους των επεξεργαστών, πρωτόκολλα συνοχής μεταξύ των κρυφών μνημών, τεχνικές διαιτησίας στο δίαυλο, απεικόνιση, πολιτικές αντικατάστασης, μέγεθος της κρυφής μνήμης (μπλοκ δεδομένων στην κρυφή μνήμη), πλήθος των συνόλων της κρυφής μνήμης (για κρυφές μνήμες συσχέτισης συνόλων), πλήθος των λέξεων ανά μπλοκ (μέγεθος μπλοκ της μνήμης).

Το Κέντρο Πηγών Διδάσκοντα συμπεριλαμβάνει μία συμπαγή εισαγωγή στο εργαλείο SMPCache, με εντολές σχετικά με τον τρόπο φόρτωσης και εκκίνησης του προγράμματος. Επίσης, το εγχειρίδιο περιλαμβάνει και ορισμένες προτεινόμενες εργασίες.

Το εργαλείο SMPCache είνα ένα φορητό λογισμικό, το οποίο εκτελείται στις περισσότερες πλατφόρμες προσωπικών υπολογιστών με λειτουργικό σύστημα Windows. Το λογισμικό SMPCache διατίθεται για χωρίς κόστος λήψη μέσα από το Δικτυακό Τόπο του, όταν πρόκειται να χρησιμοποιηθεί για μη εμπορικούς σκοπούς.

Α.4 ΕΡΓΑΣΙΕΣ ΣΕ ΓΛΩΣΣΑ ΣΥΜΒΟΛΟΜΕΤΑΦΡΑΣΗΣ

Ο προγραμματισμός σε γλώσσα συμβολομετάφρασης χρησιμοποιείται συχνά για τη διδασκαλία στοιχείων του υλικού στο χαμηλότερο επίπεδο, αλλά και βασικών εννοιών της αρχιτεκτονικής υπολογιστών. Το CodeBlue είναι ένα απλό πρόγραμμα γλώσσας συμβολομετάφρασης, το οποίο αναπτύχθηκε στην Ακαδημία της Αεροπορίας των Η.Π.Α. Ο στόχος της εργασίας ήταν να αναπτυχθούν και να διδαχτούν οι έννοιες της γλώσσας συμβολομετάφρασης με χρήση ενός ιδεατού προσομοιωτή, τον οποίο οι φοιτητές θα μάθαιναν σε ένα μόλις μάθημα. Επίσης, οι προγραμματιστές που υλοποίησαν το CodeBlue επιθυμούσαν η γλώσσα αυτή να δώσει κίνητρα στους φοιτητές, αλλά και να την θεωρήσουν ευχάριστη στη χρήση. Η γλώσσα CodeBlue είναι πολύ πιο απλή από τα περισσότερα απλοποιημένα σύνολα εντολών μίας αρχιτεκτονικής, όπως το SC123. Επίσης, επιτρέπει στους φοιτητές να αναπτύξουν ενδιαφέροντα προγράμματα σε γλώσσα συμβολομετάφρασης, τα οποία ανταγωνίζονται σε διαγωνισμούς και είναι παρόμοια με εκείνα του πολύ πιο πολύπλοκου προσομοιωτή SPIMbot. Ακόμη πιο σημαντικό είναι το γεγονός ότι, μέσα από τον προγραμματισμό με το CodeBlue οι φοιτητές μαθαίνουν σημαντικές έννοιες της αρχιτεκτονικής υπολογιστών, όπως είναι η συγκατοίκηση εντολών και δεδομένων στη μνήμη, η υλοποίηση της δομής ελέγχου, και οι τρόποι διευθυνσιοδότησης.

Για να υπάρξει μία βάση ώστε να ανατεθούν εργασίες, οι προγραμματιστές του CodeBlue δημιούργησαν ένα γραφικό περιβάλλον ανάπτυξης, το οποίο επιτρέπει στους φοιτητές να δημιουργήσουν ένα πρόγραμμα, να δουν την αναπαράστασή του στη μνήμη, να το εκτελέσουν βήμα προς βήμα, και να προσομοιώσουν μία μάχη ανάμεσα σε ανταγωνιζόμενα προγράμματα, σε ένα γραφικό περιβάλλον

μνήμης.

Οι εργασίες είναι δυνατόν να δημιουργηθούν γύρω από την έννοια ενός διαγωνισμού Core War. Το Core War είναι ένα παιχνίδι προγραμματισμού, το οποίο εισήχθη στις αρχές της δεκαετίας του '80 και ήταν πολύ δημοφιλές για περίπου 15 χρόνια. Το Core War έχει τέσσερα βασικά στοιχεία: έναν πίνακα μνήμης αποτελούμενο από 8000 θέσεις, μία απλοποιημένη γλώσσα συμβολομετάφρασης, την RedCode, έναν προσομοιωτή με την ονομασία MARS-Memory Array Redcode Simulator, και το σύνολο των αντιμαχώμενων προγραμμάτων. Δύο από τα διαμαχόμενα προγράμματα, εισάγονται στον πίνακα της μνήμης σε τυχαία επιλεγμένες θέσεις. Κανένα πρόγραμμα δεν γνωρίζει που βρίσκεται το άλλο. Ο προσομοιωτής MARS εκτελεί τα προγράμματα με έναν απλό τρόπο μοιράσματος του χρόνου. Τα δύο προγράμματα παίρνουν σειρά: εκτελείται μία απλή εντολή ενός προγράμματος, στη συνέχεια μία απλή εντολή του άλλου, κ.ο.κ. Το τι θα κάνει ένα πρόγραμμα μάχης κατά τη διάρκεια των κύκλων εκτέλεσης που αποδίδονται σε αυτό, εξαρτάται αποκλειστικά από τον προγραμματιστή. Ο στόχος είναι να καταστραφούν οι εντολές του άλλου προγράμματος. Το περιβάλλον του CodeBlue αντικαθιστά τις εντολές του με εντολές της γλώσσας RedCode και παρέχει τη δική της αλληλεπιδρασική διεπαφή εκτέλεσης.

Το Κέντρο Πηγών Διδάσκοντα περιέχει το περιβάλλον του CodeBlue, ένα εγχειρίδιο χρήστη για τους φοιτητές, άλλο υποστηρικτικό υλικό, και προτεινόμενες εργασίες.

Α.5 ΑΝΑΘΕΣΕΙΣ ΕΡΓΑΣΙΩΝ ΜΕΛΕΤΗΣ/ΣΥΓΓΡΑΦΗΣ ΑΝΑΦΟΡΑΣ

Ένας ακόμη εξαιρετικός τρόπος ενίσχυσης των εννοιών του μαθήματος αλλά και απόκτησης ερευνητικής εμπειρίας από την πλευρά των φοιτητών, είναι η ανάθεση εργασιών μελέτης και ανάλυσης άρθρων μέσα από τη διεθνή αρθρογραφία. Το Κέντρο Πηγών Διδάσκοντα περιέχει μία λίστα από άρθρα τα οποία μπορούν να ανατεθούν για εργασία, οργανωμένα βάσει των κεφαλαίων του βιβλίου. Το Κέντρο Πηγών Διδάσκοντα περιέχει ένα αντίγραφο για καθένα από αυτά τα άρθρα. Επίσης, περιέχει ένα προτεινόμενο τρόπο συγγραφής της εργασίας.

Α.6 ΓΡΑΠΤΕΣ ΕΡΓΑΣΙΕΣ

Οι γραπτές εργασίες μπορεί να έχουν μία ισχυρή πολλαπλασιαστική επίδραση στη διαδικασία εκμάθησης ενός τεχνικού αντικειμένου όπως είναι η οργάνωση και αρχιτεκτονική υπολογιστών. Οι υποστηρικτές της κίνησης WAC-Writing Across the Curriculum **(http://wac.colostate.edu)**, αναφέρουν τα σημαντικά κέρδη που προκύπτουν από τη συγγραφή γραπτών εργασιών, όσον αφορά τη διευκόλυνση της εκμάθησης. Η συγγραφή εργασιών οδηγεί σε πιο λεπτομερρή και πολύπλοκη σκέψη πάνω σε ένα αντικείμενο. Επιπλέον, βοηθά να ξεπεραστεί η τάση των φοιτητών να προσπαθούν να σπουδάσουν ένα αντικείμενο με την ελάχιστη δυνατή προσπάθεια, μαθαίνοντας απλώς γεγονότα και τεχνικές επίλυσης προβλημάτων, χωρίς να κατανοούν σε μεγάλο βαθμό το αντικείμενο.

Το Κέντρο Πηγών Διδάσκοντα περιέχει ένα πλήθος από προτεινόμενες γραπτές εργασίες, οργανωμένες ανά κεφάλαιο. Οι διδάσκοντες πιθανόν να καταλήξουν στο συμπέρασμα ότι αυτό είναι το πιο σημαντικό μέρος της προσέγγισής τους για τη διδασκαλία του υλικού. Θα εκτιμούσα πολύ κάθε μορφή ανάδρασης σε αυτή την περιοχή και προτάσεις για επιπλέον θέματα γραπτών εργασιών.

Α.7 ΕΡΩΤΗΣΕΙΣ ΕΞΕΤΑΣΗΣ

Ένα σύνολο από ερωτήσεις εξέτασης είναι διαθέσιμο στο Κέντρο Πηγών Διδάσκοντα. Για κάθε κεφάλαιο, το σύνολο των ερωτήσεων εξέτασης συμπεριλαμβάνει ερωτήσεις Α/Ψ, πολλαπλών επιλογών, και

συμπλήρωσης των κενών. Το σύνολο των ερωτήσεων εξέτασης αποτελεί έναν αποτελεσματικό τρόπο αξιολόγησης του βαθμού στον οποίο οι φοιτητές έχουν κατανοήσει το υλικό.

ΠΑΡΑΡΤΗΜΑ Β
ΓΛΩΣΣΑ ΣΥΜΒΟΛΟΜΕΤΑΦΡΑΣΗΣ ΚΑΙ ΣΧΕΤΙΚΑ ΖΗΤΗΜΑΤΑ

Β.1 Γλώσσα Συμβολομετάφρασης

Στοιχεία της Γλώσσας Συμβολομετάφρασης

Τύποι Προτάσεων της Γλώσσας Συμβολομετάφρασης

Παράδειγμα: Πρόγραμμα Υπολογισμού του Μέγιστου Κοινού Διαιρέτη

Β.2 Συμβολομεταφραστές

Συμβολομεταφραστής Δύο Περασμάτων

Συμβολομεταφραστής Ενός Περάσματος

Παράδειγμα: Πρόγραμμα Εύρεσης Πρώτων Αριθμών

Β.3 Φόρτωση και Διασύνδεση

Επανατοποθέτηση

Φόρτωση

Διασύνδεση

Β.4 Προτεινόμενη Μελέτη και Διαδικτυακοί Τόποι

Β.5 Όροι-κλειδιά, Ερωτήσεις Επανάληψης και Προβλήματα

ΒΑΣΙΚΑ ΣΗΜΕΙΑ

- Μία γλώσσα συμβολομετάφρασης, είναι μία συμβολική αναπαράσταση της γλώσσας μηχανής ενός συγκεκριμένου επεξεργαστή, στην οποία έχουν προστεθεί και άλλοι τύποι προτάσεων με σκοπό να διευκολυνθεί η συγγραφή των προγραμμάτων και να δίνονται εντολές προς το συμβολομεταφραστή.
- Ο συμβολομεταφραστής είναι ένα πρόγραμμα, το οποίο μετατρέπει τη γλώσσα συμβολομετάφρασης σε κώδικα μηχανής.
- Το πρώτο βήμα για τη δημιουργία μίας ενεργού διεργασίας, είναι να φορτωθεί ένα πρόγραμμα στην κύρια μνήμη, και να δημιουργηθεί μία εικόνα διεργασίας.
- Ένα εργαλείο διασύνδεσης χρησιμοποιείται για να αναλύει τις αναφορές μεταξύ των μονάδων οι οποίες φορτώνονται.

Το ζήτημα της γλώσσας συμβολομετάφρασης εισήχθη με συντομία στο Κεφάλαιο 11. Αυτό το παράρτημα παρέχει περισσότερες λεπτομέρειες και καλύπτει ένα σύνολο από σχετικά ζητήματα. Υπάρχει ένα πλήθος λόγων για τους οποίους αξίζει να μελετηθεί ο προγραμματισμός με γλώσσα συμβολομετάφρασης (σε σύγκριση με τον προγραμματισμό σε γλώσσα υψηλού επιπέδου), συμπεριλαμβανομένων των ακολούθων:

1. Καθιστά πιο σαφή την εκτέλεση των εντολών
2. Δείχνει τον τρόπο αναπαράστασης των δεδομένων στη μνήμη.
3. Δείχνει τον τρόπο με τον οποίο ένα πρόγραμμα αλληλεπιδρά με το λειτουργικό σύστημα, τον επεξεργαστή, και το σύστημα Ε/Ε.
4. Καθιστά πιο σαφή τον τρόπο με τον οποίο ένα πρόγραμμα προσπελαύνει τις εξωτερικές συσκευές.
5. Η κατανόηση των προγραμμάτων που είναι γραμμένα σε γλώσσα συμβολομετάφρασης βελτιώνει τους φοιτητές στον προγραμματισμό σε γλώσσες υψηλότερου επιπέδου, δίνοντάς τους καλύτερη ιδέα σχετικά με τη γλώσσα στην οποία θα πρέπει να μεταφραστεί ένα πρόγραμμα γραμμένο σε γλώσσα υψηλότερου επιπέδου.

Θα ξεκινήσουμε αυτό το κεφάλαιο με μία μελέτη των βασικών στοιχείων μίας γλώσσας συμβολομετάφρασης, χρησιμοποιώντας την αρχιτεκτονική x86 για τα παραδείγματά μας.[1] Στη συνέχεια, θα εξετάσουμε τις λειτουργίες ενός συμβολομεταφραστή. Τέλος, ακολουθεί μία συζήτηση σχετικά με τα εργαλεία διασύνδεσης και φόρτωσης.

Ο Πίνακας Β.1 παρουσιάζει ορισμένους από τους βασικούς όρους που υπάρχουν σε αυτό το παράρτημα.

Β.1 ΓΛΩΣΣΑ ΣΥΜΒΟΛΟΜΕΤΑΦΡΑΣΗΣ

Η γλώσσα συμβολομετάφρασης (Assembly Language), είναι μία γλώσσα προγραμματισμού, η οποία βρίσκεται ένα βήμα μακρυά από τη γλώσσα μηχανής. Τυπικά, κάθε εντολή γραμμένη σε γλώσσα

[1] Υπάρχει ένα πλήθος συμβολομεταφραστών για την αρχιτεκτονική x86. Τα παραδείγματά μας χρησιμοποιούν το συμβολομεταφραστή NASM-Netwide Assembler, ο οποίος είναι ανοικτό λογισμικό. Στο δικτυακό τόπο του βιβλίου υπάρχει αντίγραφο του εγχειριδίου του συμβολομεταφραστή NASM.

Πίνακας Β.1: Βασικοί όροι για αυτό το παράρτημα

Συμβολομεταφραστής

Ένα πρόγραμμα το οποίο μεταφράζει τη γλώσσα συμβολομετάφρασης σε κώδικα μηχανής.

Γλώσσα Συμβολομετάφρασης

Μία συμβολική αναπαράσταση της γλώσσας μηχανής ενός συγκεκριμένου επεξεργαστή, στην οποία έχουν προστεθεί και άλλοι τύποι προτάσεων με σκοπό να διευκολυνθεί η συγγραφή των προγραμμάτων και να δίνονται εντολές προς το συμβολομεταφραστή.

Μεταγλωττιστής

Ένα πρόγραμμα, το οποίο μετατρέπει ένα άλλο πρόγραμμα γραμμένο σε κάποια πηγαία γλώσσα (ή γλώσσα προγραμματισμού) σε γλώσσα μηχανής (αντικειμενικό κώδικα). Ορισμένοι μεταγλωττιστές δίνουν το αποτέλεσμα εκφρασμένο σε γλώσσα συμβολομετάφρασης, η οποία στη συνέχεια μετατρέπεται σε γλώσσα μηχανής από έναν ξεχωριστό συμβολομεταφραστή. Ο μεταγλωττιστής διακρίνεται από το συμβολομεταφραστή από το γεγονός ότι κάθε πρόταση της εισόδου, σε γενικές γραμμές, δεν αντιστοιχεί σε μία απλή εντολή μηχανής ή σε μία σταθερή ακολουθία εντολών. Ο μεταγλωττιστής είναι δυνατόν να υποστηρίξει χαρακτηριστικά όπως αυτόματη κατανομή μεταβλητών, αυθαίρετες αριθμητικές εκφράσεις, δομές ελέγχου όπως είναι οι βρόχοι FOR και WHILE, εμβέλεια μεταβλητών, λειτουργίες Ε/Ε, και φορητότητα του πηγαίου κώδικα.

Εκτελέσιμος Κώδικας

Ο κώδικας μηχανής ο οποίος παράγεται από έναν επεξεργαστή γλώσσας πηγαίου κώδικα, όπως είναι ο συμβολομεταφραστής ή ο μεταγλωττιστής. Πρόκειται για λογισμικό σε μορφή εκτελέσιμη από έναν υπολογιστή.

Σύνολο Εντολών

Το σύνολο όλων των εντολών ενός συγκεκριμένου υπολογιστή. Με άλλα λόγια, το σύνολο των εντολών γλώσσας μηχανής, τις οποίες καταλαβαίνει ο συγκεκριμένος υπολογιστής.

Εργαλείο Διασύνδεσης

Ένα βοηθητικό πρόγραμμα το οποίο ενώνει ένα ή περισσότερα αρχεία αντικειμενικού κώδικα, προερχόμενα από ξεχωριστά μεταγλωττισμένα κομμάτια του προγράμματος, σε ένα αρχείο το οποίο περιέχει κώδικα που είναι δυνατόν να φορτωθεί ή να εκτελεστεί.

Εργαλείο Φόρτωσης

Μία ρουτίνα, η οποία αντιγράφει ένα εκτελέσιμο πρόγραμμα στη μνήμη, προκειμένου να εκτελεστεί.

Γλώσσα Μηχανής ή Κώδικας Μηχανής

Η δυαδική αναπαράσταση ενός προγράμματος το οποίο στην πραγματικότητα διαβάζεται και ερμηνεύεται από έναν υπολογιστή. Ένα πρόγραμμα γραμμένο σε κώδικα μηχανής αποτελείται από μία ακολουθία εντολών μηχανής (πιθανόν με διεσπαρμένα δεδομένα). Οι εντολές είναι δυαδικές συμβολοσειρές, η οποίες μπορεί είτε να έχουν το ίδιο μέγεθος (π.χ. μία λέξη μήκους 32 bits, η οποία συναντάται σε πολλούς σύγχρονους μικροεπεξεργαστές RISC) είτε διαφορετικό.

Αντικειμενικός Κώδικας

Η αναπαράσταση σε γλώσσα μηχανής του πηγαίου κώδικα προγραμματισμού. Ο αντικειμενικός κώδικας δημιουργείται από ένα μεταγλωττιστή ή συμβολομεταφραστή και στη συνέχεια μετατρέπεται σε εκτελέσιμο κώδικα με τη βοήθεια ενός εργαλείου διασύνδεσης.

συμβολομετάφρασης μεταφράζεται σε εντολή μηχανής, μέσω του συμβολομεταφραστή. Η γλώσσα συμβολομετάφρασης εξαρτάται από το υλικό, και κάθε τύπος επεξεργαστή έχει τη δική του διαφορετική γλώσσα. Πιο συγκεκριμένα, οι εντολές της γλώσσας συμβολομετάφρασης αναφέρονται σε συγκεκριμένους καταχωρητές του επεξεργαστή, συμπεριλαμβάνουν όλους τους κωδικούς εντολών του επεξεργαστή, και δείχνουν το μήκος σε bits των καταχωρητών και των παραγόντων της γλώσσας μη-

χανής. Επομένως, ένας προγραμματιστής της γλώσσας συμβολομετάφρασης, θα πρέπει να κατανοεί πλήρως την αρχιτεκτονική του συστήματος.

Οι προγραμματιστές σπάνια χρησιμοποιούν τη γλώσσα συμβολομετάφρασης για εφαρμογές, ή ακόμη και για προγράμματα του συστήματος. Οι γλώσσες υψηλού επιπέδου παρέχουν μεγάλη ισχύ, αλλά και ακρίβεια, ώστε να διευκολύνουν το έργο του προγραμματιστή. Τα μειονεκτήματα της χρησιμοποίησης μίας γλώσσας συμβολομετάφρασης σε σύγκριση με τη γλώσσα υψηλού επιπέδου, είναι τα ακόλουθα [FOG08a]:

1. **Χρόνος υλοποίησης:** Η συγγραφή κώδικα σε γλώσσα συμβολομετάφρασης είναι πολύ πιο χρονοβόρα σε σχέση με τη συγγραφή σε γλώσσα υψηλού επιπέδου.

2. **Αξιοπιστία και ασφάλεια:** Είναι πολύ εύκολο να γίνουν λάθη στον κώδικα συμβολομετάφρασης. Ο συμβολομεταφραστής δεν ελέγχει αν τηρούνται οι συμβάσεις κλήσης και αποθήκευσης στους καταχωρητές. Κανείς δεν ελέγχει αν το πλήθος των εντολών ΠΙΕΣΗΣ (PUSH) και ΩΘΗΣΗΣ (POP) είναι το ίδιο σε όλες τις πιθανές διακλαδώσεις και μονοπάτια. Υπάρχουν τόσες πολλές πιθανότητες ύπαρξης κρυμμένων σφαλμάτων μέσα στον κώδικα συμβολομετάφρασης, ώστε να επηρρεάζεται η αξιοπιστία και η ασφάλεια της εργασίας, εκτός αν ο προγραμματιστής ελέγχει και επαληθεύει το πρόγραμμα με συστηματικό τρόπο.

3. **Αποσφαλμάτωση και επαλήθευση:** Ο κώδικας συμβολομετάφρασης είναι δυσκολότερος στην αποσφαλμάτωση και επαλήθευση, επειδή υπάρχουν πολύ μεγαλύτερες πιθανότητες σφαλμάτων σε σύγκριση με μία συνηθισμένη γλώσσα υψηλού επιπέδου.

4. **Συντηρησιμότητα:** Ο κώδικας συμβολομετάφρασης είναι πολύ πιο δύσκολο να τροποποιηθεί και να συντηρηθεί, επειδή η γλώσσα επιτρέπει την ύπαρξη μη δομημένου κώδικα και κάθε δυνατό τέχνασμα, το οποίο είναι δύσκολο να κατανοήσει κανείς. Η λεπτομερής τεκμηρίωση και ένα σταθερό στυλ προγραμματισμού είναι εντελώς απαραίτητα.

5. **Φορητότητα:** Ο κώδικας συμβολομετάφρασης εξαρτάται από την πλατφόρμα. Είναι δύσκολη η μεταφορά του σε άλλη πλατφόρμα.

6. **Ο κώδικας του συστήματος είναι δυνατόν να χρησιμοποιήσει εγγενείς λειτουργίες αντί της συμβολομετάφρασης:** Οι καλύτεροι σύγχρονοι μεταγλωττιστές της γλώσσας C++ διαθέτουν εγγενείς λειτουργίες προσπέλασης των καταχωρητών ελέγχου και άλλων εντολών του συστήματος. Ο κώδικας συμβολομετάφρασης δεν είναι πλέον αναγκαίος στους οδηγούς των συσκευών και στον υπόλοιπο κώδικα του συστήματος, δεδομένης της διαθεσιμότητας των εγγενών λειτουργιών..

7. **Ο κώδικας της εφαρμογής είναι δυνατόν να χρησιμοποιήσει εγγενείς λειτουργίες ή κλάσεις διανυσμάτων αντί της συμβολομετάφρασης:** Οι καλύτεροι σύγχρονοι μεταγλωττιστές της γλώσσας C++ διαθέτουν εγγενείς λειτουργίες για τις διανυσματικές πράξεις, αλλά και άλλες ειδικές εντολές, οι οποίες παλαιότερα απαιτούσαν προγραμματισμό σε γλώσσα συμβολομετάφρασης.

8. **Οι μεταγλωττιστές έχουν βελτιωθεί σημαντικά τα τελευταία χρόνια:** Οι καλύτεροι μεταγλωττιστές είναι πλέον αρκετά καλοί. Απαιτεί ειδικές γνώσεις και εμπειρία για να πετύχει κανείς μεγαλύτερη βελτιστοποίηση σε σχέση με το μεταγλωττιστή της C++.

Εν τούτοις, εξακολουθούν να υπάρχουν αρκετά πλεονεκτήματα της χρήσης της γλώσσας συμβολομετάφρασης, ορισμένα από τα οποία είναι τα εξής:

1. **Αποσφαλμάτωση και επαλήθευση:** Η εξέταση του κώδικα συμβολομετάφρασης ο οποίος παράγεται από ένα μεταγλωττιστή, ή του παραθύρου αποσυμβολομετάφρασης ενός εργαλείου

αποσφαλμάτωσης, είναι χρήσιμη στην εύρεση σφαλμάτων και στον έλεγχο του πόσο καλά ο μεταγλωττιστής βελτιστοποιεί ένα συγκεκριμένο τμήμα του κώδικα.

2. **Δημιουργία μεταγλωττιστών:** Η κατανόηση των τεχνικών συγγραφής του κώδικα συμβολομετάφρασης είναι αναγκαία για τη δημιουργία μεταγλωττιστών, εργαλείων αποσφαλμάτωσης, αλλά και άλλων εργαλείων ανάπτυξης.

3. **Ενσωματωμένα συστήματα:** Τα μικρά ενσωματωμένα συστήματα έχουν λιγότερους πόρους σε σχέση με τους προσωπικούς υπολογιστές και τα μεγάλα υπολογιστικά συστήματα. Ο προγραμματισμός σε γλώσσα συμβολομετάφρασης είναι απαραίτητος για τη βελτιστοποίση του κώδικα σε ταχύτητα και μέγεθος, μέσα σε ένα ενσωματωμένο σύστημα.

4. **Οδηγοί υλικού και κώδικας συστήματος:** Η προσπέλαση στο υλικό, σε καταχωρητές ελέγχου του συστήματος, κ.ο.κ, είναι ορισμένες φορές δύσκολη ή αδύνατη, με χρήση κώδικα υψηλού επιπέδου.

5. **Προσπέλαση εντολών οι οποίες δεν είναι προσπελάσιμες από μία γλώσσα υψηλού επιπέδου:** Αρκετές εντολές συμβολομετάφρασης δεν έχουν ισοδύναμες εντολές σε γλώσσα υψηλού επιπέδου.

6. **Αυτο-τροποποιούμενος κώδικας:** Γενικά, ο αυτο-τροποποιούμενος κώδικας δεν είναι ωφέλιμος, επειδή παρεμβάλλεται στον αποτελεσματική αποθήκευση του κώδικα στην κρυφή μνήμη. Ωστόσο, μπορεί να εμφανίζει πλεονεκτήματα, για παράδειγμα, να συμπεριληφθεί ένας μικρός μεταγλωττιστής σε προγράμματα μαθηματικών, όπου μία συνάρτηση ορισμένη από το χρήστη θα πρέπει να υπολογιστεί πολλές φορές.

7. **Βελτιστοποίηση του μεγέθους του κώδικα:** Ο χώρος αποθήκευσης και η μνήμη έχουν πλέον τόσο χαμηλό κόστος, ώστε δεν αξίζει η προσπάθεια να χρησιμοποιηθεί ο κώδικας συμβολομετάφρασης για να μειωθεί το μέγεθος του κώδικα. Όμως, το μέγεθος της κρυφής μνήμης εξακολουθεί να είναι ένας σημαντικός πόρος, ώστε μερικές φορές, να είναι χρήσιμη η βελτιστοποίηση ως προς το μέγεθος ενός κρίσιμου τμήματος του κώδικα, για να χωρά στην κρυφή μνήμη κώδικα.

8. **Βελτιστοποίηση της ταχύτητας του κώδικα:** Γενικά, οι σύγχρονοι μεταγλωττιστές της γλώσσας C++ βελτιστοποιούν αρκετά καλά τον κώδικα, στις περισσότερες περιπτώσεις. Ωστόσο, εξακολουθούν να υπάρχουν περιπτώσεις στις οποίες οι μεταγλωττιστές δεν λειτουργούν με μεγάλη αποτελεσματικότητα και στις οποίες, είναι εφικτό να αυξηθεί εντυπωσιακά η ταχύτητα με προσεκτικό προγραμματισμό σε γλώσσα συμβολομετάφρασης.

9. **Βιβλιοθήκες συναρτήσεων:** Το συνολικό κέρδος της βελτιστοποίησης του κώδικα, είναι μεγαλύτερο σε βιβλιοθήκες συναρτήσεων, τις οποίες χρησιμοποιούν πολλοί προγραμματιστές.

10. **Συμβατότητα των βιβλιοθηκών συναρτήσεων με πολλούς μεταγλωττιστές και λειτουργικά συστήματα:** Είναι εφικτό να υλοποιήσουμε βιβλιοθήκες συναρτήσεων με πολλαπλές καταχωρήσεις, οι οποίες είναι συμβατές με διαφορετικούς μεταγλωττιστές και λειτουργικά συστήματα. Αυτό απαιτεί προγραμματισμό με γλώσσα συμβολομετάφρασης.

Μερικές φορές, οι όροι *γλώσσα συμβολομετάφρασης* και *γλώσσα μηχανής* χρησιμοποιούνται εσφαλμένα ως συνώνυμοι. Η γλώσσα μηχανής αποτελείται από εντολές οι οποίες είναι άμεσα εκτελέσιμες από τον επεξεργαστή. Κάθε εντολή γραμμένη σε γλώσσα μηχανής είναι μία δυαδική συμβολοσειρά, η οποία περιέχει έναν κωδικό εντολής, αναφορές παραγόντων, και πιθανόν άλλα bits τα οποία σχετίζονται με την εκτέλεση, όπως είναι οι σημαίες. Για λόγους διευκόλυνσης, αντί να γράφουμε μία εντολή σε μορφή συμβολοσειράς από bits, μπορούμε να τη γράψουμε συμβολικά, με ονομασίες οι

Σχήμα Β.1: Δομή πρότασης γραμμένης σε γλώσσα συμβολομετάφρασης

οποίες αντιστοιχούν σε κωδικούς εντολών και καταχωρητές. Μία γλώσσα συμβολομετάφρασης χρησιμοποιεί σε πολύ μεγαλύτερο βαθμό τα συμβολικά ονόματα (ΣτΜ: Στα Ελληνικά χρησιμοποιείται συχνά ο όρος συμβολική γλώσσα ως συνώνυμος του όρου γλώσσα συμβολομετάφρασης. Αν και οι δύο όροι βρίσκονται πολύ κοντά, εν τούτοις διαφέρουν όπως είδαμε στην Ενότητα 11.5), συμπεριλαμβανομένης της χρήσης ονομάτων για συγκεκριμένες θέσεις της κύριας μνήμης και συγκεκριμένες θέσεις εντολών. Η γλώσσα συμβολομετάφρασης επίσης περιλαμβάνει προτάσεις οι οποίες δεν είναι άμεσα εκτελέσιμες, αλλά χρησιμεύουν ως εντολές προς το συμβολομεταφραστή, ο οποίος παράγει κώδικα μηχανής από ένα πρόγραμμα γραμμένο σε γλώσσα συμβολομετάφρασης.

Στοιχεία της Γλώσσας Συμβολομετάφρασης

Μία πρόταση γραμμένη σε μία τυπική γλώσσα συμβολομετάφρασης, έχει τη μορφή που απεικονίζεται στο Σχήμα Β.1. Η πρόταση αποτελείται από 4 στοιχεία: ετικέτα, μνημονικό όνομα, παράγοντα, και σχόλιο.

ΕΤΙΚΕΤΑ Αν υπάρχει ετικέτα, ο συμβολομεταφραστής την ορίζει ως ισοδύναμη με τη διεύθυνση στην οποία θα φορτωθεί το πρώτο byte του αντικειμενικού κώδικα ο οποίος παράγεται για αυτή την εντολή. Στη συνέχεια, ο προγραμματιστής έχει τη δυνατότητα να χρησιμοποιήσει την ετικέτα ως διεύθυνση, ή ως δεδομένα σε ένα πεδίο διεύθυνσης μίας άλλης εντολής. Ο συμβολομεταφραστής αντικαθιστά την ετικέτα με την τιμή που έχει αποδοθεί, όταν δημιουργεί το αντικειμενικό πρόγραμμα. Οι ετικέτες χρησιμοποιούνται πιο συχνά σε εντολές διακλάδωσης.

Ως παράδειγμα, θεωρείστε το παρακάτω τμήμα ενός προγράμματος:

```
L2:   SUB   EAX,   EDX   ;  Αφαίρεση των περιεχομένων του καταχωρητή
                         ;  EDX από τα περιεχόμενα του EAX
      JG    L2           ;  άλμα στη διεύθυνση που ορίζει η ετικέτα L2
                         ;  αν το αποτέλεσμα της αφαίρεσης είναι θετικό
```

Το πρόγραμμα θα συνεχίσει να εκτελεί βρόχους με σημείο εκκίνησης τη θέση που ορίζει η ετικέτα L2, έως ότου το αποτέλεσμα μηδενιστεί ή γίνει αρνητικό. Επομένως, όταν εκτελείται η εντολή JG, αν το αποτέλεσμα είναι θετικό, ο επεξεργαστής τοποθετεί τη διεύθυνση η οποία είναι ισοδύναμη με την ετικέτα L2 στο μετρητή προγράμματος.

Οι λόγοι χρησιμοποίησης της ετικέτας είναι οι ακόλουθοι:

1. Η ετικέτα βοηθά στο να βρεθεί και να απομνημονευτεί πιο εύκολα η θέση του προγράμματος.

2. Η ετικέτα είναι εύκολο να μετακινηθεί προκειμένου να διορθωθεί ένα πρόγραμμα. Ο συμβολομεταφραστής θα αλλάξει αυτομάτως τις διευθύνσεις όλων των εντολών οι οποίες χρησιμοποιούν την ετικέτα, όταν το πρόγραμμα θα συμβολομεταφραστεί εκ νέου.

3. Ο προγραμματιστής δεν είναι αναγκαίο να υπολογίζει σχετικές ή απόλυτες διευθύνσεις της μνήμης, αλλά απλώς χρησιμοποιεί τις ετικέτες με τον τρόπο που είναι απαραίτητος.

ΜΝΗΜΟΝΙΚΟ Το μνημονικό είναι το όνομα της πράξης ή της λειτουργίας της πρότασης η οποία είναι γραμμένη σε γλώσσα συμβολομετάφρασης. Όπως θα συζητηθεί παρακάτω, μία πρόταση είναι δυνατόν να αντιστοιχεί σε μία εντολή μηχανής, μία ψευδοεντολή του συμβολομεταφραστή, ή μία μακροεντολή. Στην περίπτωση της εντολής μηχανής, το μνημονικό είναι το συμβολικό όνομα το οποίο συσχετίζεται με ένα συγκεκριμένο κωδικό εντολής.

Ο Πίνακας 10.8 παραθέτει το μνημονικό, ή όνομα εντολής, πολλών εντολών της οικογένειας x86. Το Παράρτημα Α της αναφοράς [CART06] παραθέτει τις εντολές της οικογένειας x86, μαζί με τους παράγοντες κάθε μίας και την επίδραση της εντολές στους κωδικούς συνθηκών. Το Παράρτημα Β των εγχειριδίων του συμβολομεταφραστή NASM παρέχει πιο λεπτομερή περιγραφή κάθε εντολής της οικογένειας x86. Τα δύο αυτά έγγραφα διατίθενται στο Δικτυακό τόπο του βιβλίου.

ΠΑΡΑΓΟΝΤΑΣ-ΕΣ Μία πρόταση γραμμένη σε γλώσσα συμβολομετάφρασης συμπεριλαμβάνει 0 ή περισσότερους παράγοντες. Κάθε παράγοντας προσδιορίζει μία άμεση τιμή, μία τιμή καταχωρητή, ή μία θέση μνήμης. Τυπικά, η γλώσσα συμβολομετάφρασης παρέχει συμβάσεις με τις οποίες γίνεται διάκριση ανάμεσα στους τρεις τύπους αναφορών σε παράγοντες, αλλά και συνθήκες με τις οποίες καταδεικνύεται ο τρόπος διευθυνσιοδότησης.

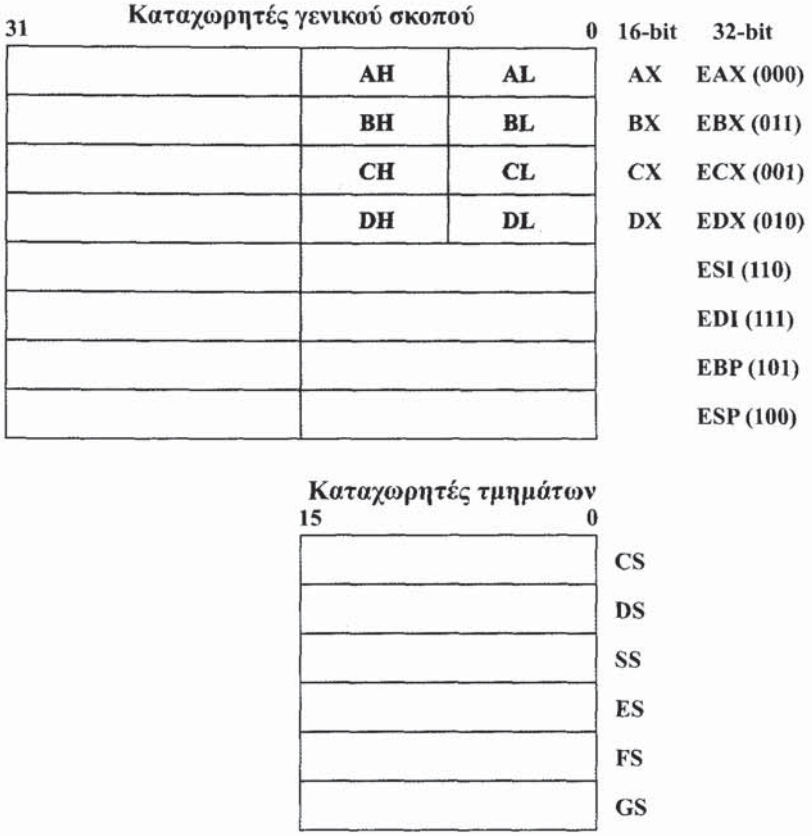
Για την αρχιτεκτονική x86, μία πρόταση γραμμένη σε γλώσσα συμβολομετάφρασης είναι πιθανό να αναφέρεται σε έναν παράγοντα καταχωρητή χρησιμοποιώντας ένα όνομα. Το Σχήμα Β.2 απεικονίζει τους καταχωρητές γενικού σκοπού του επεξεργαστή x86, μαζί με τα συμβολικά τους ονόματα και την κωδικοποίηση τους σε συμβολοσειρά bit. Ο συμβολομεταφραστής θα μεταφράσει το συμβολικό όνομα σε ένα δυαδικό αναγνωριστικό του καταχωρητή.

Όπως περιγράφηκε στην Ενότητα 11.2, η αρχιτεκτονική x86 διαθέτει ένα πλούσιο σύνολο από τύπους διευθυνσιοδότησης, καθένας από τους οποίους θα πρέπει να εκφράζεται συμβολικά στη γλώσσα συμβολομετάφρασης. Στο σημείο αυτό, παραθέτουμε μερικά από τα πιο κοινά παραδείγματα. Για τη **διευθυνσιοδότηση καταχωρητή**, χρησιμοποιείται το όνομα του καταχωρητή στην εντολή. Για παράδειγμα, η εντολή `MOV ECX, EBX` αντιγράφει τα περιεχόμενα του καταχωρητή EBX στον καταχωρητή ECX. Η άμεση διευθυνσιοδότηση δείχνει ότι η τιμή κωδικοποιείται μέσα στην εντολή. Για παράδειγμα, η εντολή `MOV EAX, 100H` αντιγράφει τη δεκαεξαδική τιμή 100 στον καταχωρητή EAX. Η άμεση τιμή είναι δυνατόν να εκφραστεί ως δυαδικός αριθμός με πρόθεμα B, ή ως δεκαδικός αριθμός χωρίς πρόθεμα. *Επομένως, οι παρακάτω προτάσεις είναι ισοδύναμες της προηγούμενης:* `MOV EAX, 100000000B` και `MOV EAX, 256`. Η **απόλυτη διευθυνσιοδότηση** αναφέρεται σε μία θέση μνήμης και εκφράζεται ως μετατόπιση από τον καταχωρητή τμήματος DS. Αυτό επεξηγείται καλύτερα με ένα παράδειγμα. Έστω ότι ο καταχωρητής τμήματος δεδομένων DS, με μήκος 16-bits περιέχει την τιμή 100H. Κατόπιν, εκτελείται η παρακάτω ακολουθία εντολών:

```
MOV AX, 1234H
MOV [3518H], AX
```

Αρχικά, ο καταχωρητής AX μήκους 16-bit αρχικοποιείται με τιμή 1234H. Στη συνέχεια, στη δεύτερη γραμμή, τα περιεχόμενά του μεταφέρονται στη λογική διεύθυνση DS3518H. Αυτή η διεύθυνση σχηματίζεται με ολίσθηση των περιεχομένων του καταχωρητή DS κατά 4 bits προς τα αριστερά και πρόσθεση της τιμής 3518H, ώστε να σχηματιστεί η λογική διεύθυνση 13518H.

ΣΧΟΛΙΟ Όλες οι γλώσσες συμβολομετάφρασης επιτρέπουν την τοποθέτηση σχολίων σε ένα πρόγραμμα. Το σχόλιο εμφανίζεται είτε στο δεξί μέρος μίας πρότασης, είτε καταλαμβάνει ολόκληρη τη γραμμή.



Σχήμα 8.2 Καταχωρητές εκτέλεσης προγράμματος του επεξεργαστή x86

Σε κάθε περίπτωση, το σχόλιο ξεκινά με ένα ειδικό χαρακτήρα, ο οποίος δηλώνει στο συμβολομεταφραστή ότι το υπόλοιπο της γραμμής (δεξιά του ειδικού χαρακτήρα) θα πρέπει να αγνοηθεί. Τυπικά, οι γλώσσες συμβολομετάφρασης της οικογένειας επεξεργαστών x86 χρησιμοποιούν το ελληνικό ερωτηματικό ως ειδικό χαρακτήρα.

Τύποι Προτάσεων της Γλώσσας Συμβολομετάφρασης

Οι προτάσεις της γλώσσας συμβολομετάφρασης χωρίζονται σε τέσσερις τύπους: εντολές, ψευδοεντολές, μακρο-ορισμούς, και σχόλια. Μία πρόταση σχολίου, είναι απλώς μία πρόταση η οποία αποτελείται πλήρως από ένα σχόλιο. Οι υπόλοιποι τύποι περιγράφονται σύντομα σε αυτή την ενότητα.

ΕΝΤΟΛΕΣ Σε ένα πρόγραμμα γραμμένο σε γλώσσα συμβολομετάφρασης, το μεγαλύτερο μέρος των προτάσεων, οι οποίες δεν είναι σχόλια, είναι συμβολικές αναπαραστάσεις των εντολών γλώσσας μηχανής. Σχεδόν χωρίς καμία παραλλαγή, υπάρχει μία σχέση μίας προς μία, ανάμεσα σε μία εντολή γλώσσας συμβολομετάφρασης και σε μία εντολή μηχανής. Ο συμβολομεταφραστής αναλύει κάθε συμβολική αναφορά και μεταφράζει την εντολή που είναι γραμμένη σε γλώσσα συμβολομετάφρασης σε μία δυαδική συμβολοσειρά, η οποία συνιστά την εντολή μηχανής.

ΨΕΥΔΟΕΝΤΟΛΕΣ Οι **ψευδοεντολές**, είναι προτάσεις της γλώσσας συμβολομετάφρασης, οι οποίες δεν μεταφράζονται άμεσα σε εντολές γλώσσας μηχανής, αλλά αποτελούν εντολές προς το συμβολομεταφραστή για να εκτελέσει συγκεκριμένες ενέργειες κατά τη διαδικασία συμβολομετάφρασης. Ορισμένα παραδείγματα είναι τα εξής:

- Ορισμός σταθερών
- Προσδιορισμός περιοχών της μνήμης για αποθήκευση δεδομένων
- Αρχικοποίηση περιοχών της μνήμης
- Τοποθέτηση πινάκων ή άλλων σταθερών δεδομένων στη μνήμη
- Επίτρεψη αναφορών σε άλλα προγράμματα

Ο Πίνακας Β.2 παραθέτει ορισμένες από τις ψευδοεντολές του συμβολομεταφραστή NASM. Για παράδειγμα, θεωρείστε την παρακάτω ακολουθία προτάσεων:

```
L2:   DB    "A"          ;  Αρχικοποίηση του byte στην τιμή του A (65),
                            σύμφωνα με τον κώδικα ASCII
      MOV   AL    [L1]   ;  αντιγραφή του byte που βρίσκεται στη
                            θέση L1, στον καταχωρητή AL
      MOV   EAX   L1     ;  αποθήκευση της διεύθυνσης του byte που
                            βρίσκεται στη θέση L1, στον EAX
      MOV   [L1]  AH     ;  αντιγραφή των περιεχομένων του καταχωρη-
                            τή AH στο byte που βρίσκεται στη θέση L1
```

Πίνακας Β.2: Ορισμένες ψευδοεντολές της γλώσσας συμβολομετάφρασης NASM

(α) Γράμματα που αντιστοιχούν στις ψευδοεντολές Res_x και D_x

Μονάδα	Γράμμα
byte	B
λέξη (2 bytes)	W
διπλή λέξη (4 bytes)	D
τετραπλή λέξη (8 bytes)	Q
10 bytes)	T

(β) Ψευδοεντολές

Όνομα	Περιγραφή	Παράδειγμα
DB,DW DD,DQ DT	Αρχικοποίηση θέσεων	`L6 DD 1A92H` ; η διπλή λέξη της θέσης L6 αρχικοποιείται με τιμή 1A92H
RESB, RESW RESD RESQ REST	Δέσμευση μη αρχικοποιημένων θέσεων	`BUFFER RESB 64` ; δέσμευση 64 bytes ξεκινώντας από τη θέση BUFFER
INCBIN	Συμπερίληψη δυαδικού αρχείου στην έξοδο	`INCBIN "file.dat"` ; Συμπερίληψη αυτού του αρχείου
EQU	Ορισμός ενός συμβόλου σε μία δεδομένη σταθερή τιμή	`MSGLEN EQU 25` ; η σταθερά MSGLEN ισούται με τη δεκαδική τιμή 25.
TIME	Επανάληψη μίας εντολής πολλές φορές.	`ZEROBUF TIMES 64 DB 0` ; αρχικοποίηση του απομονωτή 64-byte με μηδενικές τιμές σε όλες τις θέσεις

Αν χρησιμοποιείται μία απλή ετικέτα, αυτή ερμηνεύεται ως η διεύθυνση (ή μετατόπιση) των δεδομένων. Αν η ετικέτα είναι τοποθετημένη σε άγκιστρα, ερμηνεύεται ως τα δεδομένα της διεύθυνσης.

ΜΑΚΡΟ-ΟΡΙΣΜΟΙ Ένας μακρο-ορισμός μοιάζει με υπορουτίνα κατά πολλούς τρόπους. Η υπορουτίνα, είναι ένα τμήμα του προγράμματος, το οποίο γράφεται μία φορά και είναι δυνατόν να χρησιμοποιηθεί πολλές φορές μέσω της κλήσης της από οποιοδήποτε σημείο του προγράμματος. Όταν ένα πρόγραμμα έχει μεταγλωττιστεί ή έχει ολοκηρωθεί η συμβολομετάφρασή του, η υπορουτίνα φορτώνεται μία μόλις φορά. Η κλήση σε μία υπορουτίνα μεταφέρει τον έλεγχο σε αυτή και μία εντολή επιστροφής εντός της υπορουτίνας επιστρέφει τον έλεγχο στο σημείο κλήσης. Ομοίως, ένας μακρο-ορισμός είναι ένα τμήμα κώδικα, το οποίο ο προγραμματιστής γράφει μία φορά και μπορεί να το χρησιμοποιήσει πολλές φορές. Η βασική διαφορά βρίσκεται στο γεγονός ότι όταν ο συμβολομεταφραστής συναντήσει μία κλήση σε μακρο-ορισμό, αντικαθιστά την κλήση με τον ίδιο το μακρο-ορισμό. Αυτή η διαδικασία λέγεται **ανάπτυξη του μακρο-ορισμού**. Επομένως, αν ένας μακρο-ορισμός οριστεί σε ένα πρόγραμμα συμβολομετάφρασης και ενεργοποιηθεί 10 φορές, τότε ο κώδικας συμβολομετάφρασης θα περιέχει 10 στιγμιότυπά του. Ουσιαστικά, ο χειρισμός των υπορουτινών γίνεται από το υλικό κατά τη διάρκεια της εκτέλεσης του προγράμματος, ενώ ο χειρισμός των μακρο-λειτουργιών γίνεται από το συμβολομεταφραστή κατά τη διάρκεια της συμβολομετάφρασης. Οι μακρο-συμβολισμοί παρέχουν τα ίδια πλεονεκτήματα με τις υπορουτίνες, με όρους προγραμματισμού με αυτόνομες μονάδες κώδικα, αλλά χωρίς την επιβάρυνση την οποία εμφανίζει η κλήση μίας υπορουτίνας κατά τη διάρκεια του χρόνου εκτέλεσης. Ο συμβιβασμός είναι ότι η προσέγγιση των μακρο-ορισμών χρησιμοποιεί περισσότερο χώρο στον αντικειμενικό κώδικα.

Στο συμβολομεταφραστή NASM, αλλά και σε άλλους συμβολομεταφραστές, γίνεται διάκριση ανάμεσα σε έναν μακρο-ορισμό αποτελούμενο από μία γραμμή και σε ένα μακρο-ορισμό αποτελούμενο από πολλές γραμμές. Στο NASM, οι μακρο-ορισμοί μίας γραμμής ορίζονται χρησιμοποιώντας την ψευδοεντολή %DEFINE. Παρακάτω, δίνεται ένα παράδειγμα στο οποίο αναπτύσσοονται πολλοί μακρο-ορισμοί μίας γραμμής. Αρχικά, ορίζουμε δύο μακρο-ορισμούς:

```
% DEFINE B(X)=2*X
% DEFINE A(X)=1+B(X)
```

Σε κάποιο σημείο του προγράμματος, εμφανίζεται η παρακάτω πρόταση:

```
MOV AX, A(8)
```

Ο συμβολομεταφραστής αναπτύσσει αυτή την πρόταση ως ακολούθως:

```
MOV AX,1+2*8
```

η οποία μεταφράζεται συμβολικά σε μία εντολή μηχανής η οποία μετακινεί την άμεση τιμή 17 στον καταχωρητή AX.

Οι μακρο-ορισμοί πολλαπλών γραμμών ορίζονται χρησιμοποιώντας το μνημονικό όνομα &MACRO. Ένα παράδειγμα, είναι το ακόλουθο:

```
%MACRO   PROLOGUE 1
         PUSH EBP          ;   πίεση των περιεχομένων του EBP στη στοίβα
                           ;   στην οποία δείχνει ο ESP
                           ;   μείωση των περιέχομένων του ESP κατά 4
         MOVE EBP,   ESP   ;   αντιγραφή των περιεχομένων του ESP στον EBP
```

```
SUB ESP,  %1   ;  αφαίρεση της τιμής της πρώτης παραμέτρου
                  από τα περιεχόμενα του ESP
```

Ο αριθμός 1 που ακολουθεί την ονομασία του μακρο-ορισμού στη γραμμή %MACRO ορίζει το πλήθος των παραμέτρων τις οποίες αναμένει να λάβει ο μακρο-ορισμός. Η χρήση της μονάδας %1 μέσα στο μακρο-ορισμό αναφέρεται στην πρώτη παράμετρο, στην κλήση του.

Η κλήση του μακρο-ορισμού

```
MYFUNC: PROLOGUE 12
```

αναπτύσσεται στις ακόλουθες γραμμές κώδικα:

```
MYFUNC:   PUSH  EBP
          MOV   EBP,  ESP
          SUB   ESP,  12
```

Παράδειγμα: Πρόγραμμα Υπολογισμού του Μέγιστου Κοινού Διαιρέτη

Ως παράδειγμα της χρήσης της γλώσσας συμβολομετάφρασης, εξετάζουμε ένα πρόγραμμα υπολογισμού του μέγιστου κοινού διαιρέτη δύο ακεραίων. Ορίζουμε το μέγιστο κοινό διαιρέτη των ακεραίων a και b, ως ακολούθως:

$$\gcd(a,b) = \max[k, \text{τέτοιο ώστε το } k \text{ να διαιρεί το } a \text{ και το } k \text{ να διαιρεί το } b]$$

όπου λέμε ότι το k διαιρεί το a αν δεν υπάρχει υπόλοιπο. Ο αλγόριθμος του Ευκλείδη για τον υπολογισμό του μέγιστου κοινού διαιρέτη βασίζεται στο ακόλουθο θεώρημα. Για κάθε μη αρνητικό ακέραιο a και κάθε ακέραιο b,

$$\gcd(a,b) = \gcd(b, a \bmod b)$$

Ένα πρόγραμμα γραμμένο σε γλώσσα C το οποίο υλοποιεί τον αλγόριθμο του Ευκλείδη, είναι το ακόλουθο:

```
unsigned int gcd (unsigned int a, unsigned int b)
{
    if (a == 0 && b == 0)
        b = 1;
    else if (b == 0)
        b = a;
    else if (a != 0)
        while (a != b)
            if (a <b)
                b-=a;
            else
                a-=b;
    return b;
}
```

```
gcd:    mov     ebx,eax
        mov     eax,edx
        test    ebx,ebx
        jne     L1
        test    edx,edx
        jne     L1
        mov     eax,1
        ret
L1:     test    eax,eax
        jne     L2
        mov     eax,ebx
        ret
L2:     test    ebx,ebx
        je      L5
L3;     cmp     ebx,eax
        je      L5
        jae     L4
        sub     eax,ebx
        jmp     L3
L4:     sub     ebx,eax
        jmp     L3
L5:     ret
```

(α) Μεταγλωττισμένο πρόγραμμα

```
gcd:    neg     eax
        je      L3
L1:     neg     eax
        xchg    eax,edx
L2:     sub     eax,edx
        jg      L2
        jne     L1
L3:     add     eax,edx
        jne     L4
        inc     eax
L4:     ret
```

(β) Γράφεται άμεσα σε γλώσσα συμβολομετάφρασης

Σχήμα Β.3: Προγράμματα γραμμένα σε γλώσσα συμβολομετάφρασης, για τον υπολογισμό του Μέγιστου Κοινού Διαιρέτη

Το Σχήμα Β.3 παρουσιάζει δύο προγράμματα γραμμένα σε γλώσσα συμβολομετάφρασης, τα οποίο αντιστοιχούν στο παραπάνω πρόγραμμα. Το πρόγραμμα στο αριστερό μέρος παρήχθη από ένα μεταγλωττιστή της γλώσσας C. Το πρόγραμμα στο δεξί μέρος γράφτηκε με το χέρι. Το τελευταίο, χρησιμοποιεί ένα πλήθος από προγραμματιστικά τεχνάσματα ώστε να παράγει μία πιο συμπαγή και αποτελεσματική υλοποίηση.

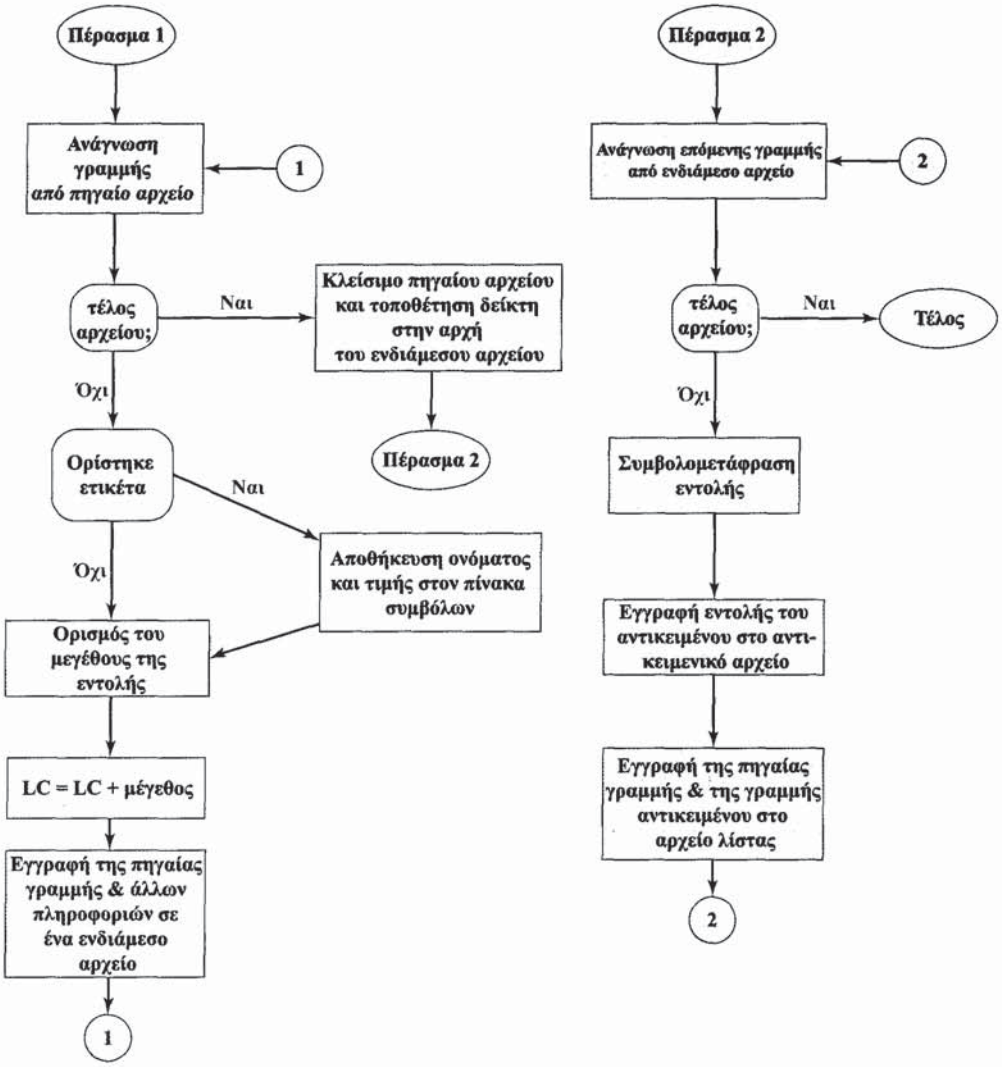
Β.2 ΣΥΜΒΟΛΟΜΕΤΑΦΡΑΣΤΕΣ

Ο συμβολομεταφραστής είναι ένα βοηθητικό λογισμικό, το οποίο λαμβάνει ως είσοδο ένα πρόγραμμα γραμμένο σε γλώσσα συμβολομετάφρασης και παράγει ως έξοδο τον αντίστοιχο αντικειμενικό κώδικα. Ο αντικειμενικός κώδικας είναι ένα δυαδικό αρχείο. Ο συμβολομεταφραστής βλέπει αυτό το αρχείο ως ένα μπλοκ μνήμης το οποίο ξεκινά από τη σχετική θέση 0.

Υπάρχουν δύο γενικότερες προσεγγίσεις όσον αφορά τους συμβολομεταφραστές: η προσέγγιση δύο περασμάτων και η προσέγγιση ενός περάσματος.

Συμβολομεταφραστής Δύο Περασμάτων

Αρχικά, θα εξετάσουμε τον συμβολομεταφραστή δύο περασμάτων, ο οποίος είναι πιο συνηθισμένος και ευκολότερος στην κατανόηση. Ο συμβολομεταφραστής κάνει δύο περάσματα στον πηγαίο κώδικα (Σχήμα Β.4):

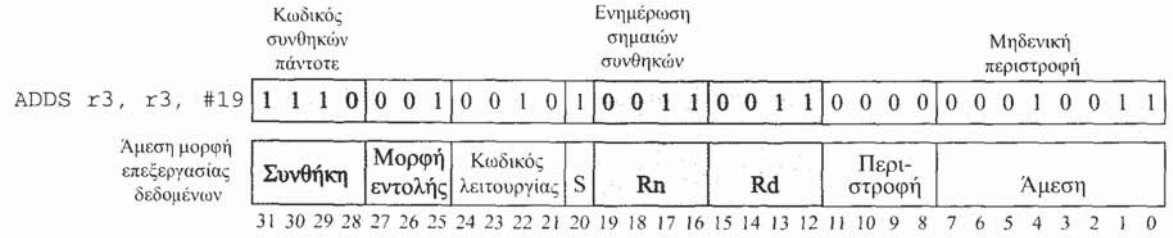


Σχήμα Β.1 Διάγραμμα ροής συμβολομεταφραστή δύο περασμάτων

ΠΡΩΤΟ ΠΕΡΑΣΜΑ Στο πρώτο πέρασμα, ο συμβολομεταφραστής ενδιαφέρεται μόνον για τους ορισμούς ετικετών. Το πρώτο πέρασμα χρησιμοποιείται για να κατασκευαστεί ένας **πίνακας συμβόλων**, ο οποίος περιέχει μία λίστα με όλες τις ετικέτες και τις αντίστοιχες τιμές του **μετρητή θέσης** αυτών. Στο πρώτο byte του αντικειμενικού κώδικα η τιμή του μετρητή θέσης θα είναι ίση με 0. Το πρώτο πέρασμα εξετάζει κάθε πρόταση του κώδικα συμβολομετάφρασης. Παρά το γεγονός ότι ο συμβολομεταφραστής δεν είναι ακόμη έτοιμος να μεταφράσει εντολές, θα πρέπει να εξετάσει κάθε εντολή επαρκώς, ώστε να προσδιορίσει το μήκος της αντίστοιχης εντολής μηχανής και επομένως κατά πόσο θα αυξήσει την τιμή του μετρητή θέσης. Αυτό είναι πιθανό να απαιτεί όχι μόνον την εξέταση του κωδικού εντολής, αλλά επίσης και των παραγόντων, αλλά και των τρόπων διευθυνσιοδότησης.

Οι ψευδοεντολές όπως οι DQ και REST (δείτε τον Πίνακα Β.2) αναγκάζουν το μετρητή θέσης να ρυθμιστεί αναλόγως με το πόση μνήμη προσδιορίζεται.

Όταν ο συμβολομεταφραστής συναντήσει μία πρόταση με ετικέτα, θα τοποθετήσει την ετικέτα στον πίνακα συμβόλων, μαζί με την τρέχουσα τιμή του μετρητή θέσης. Ο συμβολομεταφραστής συνεχίζει έως ότου διαβάσει όλες τις προτάσεις της γλώσσας συμβολομετάφρασης.



Σχήμα Β.5: Μετάφραση μίας εντολής συμβολομετάφρασης του επεξεργαστή ARM σε δυαδική εντολή μηχανής

ΔΕΥΤΕΡΟ ΠΕΡΑΣΜΑ Το δεύτερο πέρασμα διαβάζει το πρόγραμμα εκ νέου από την αρχή. Κάθε εντολή μεταφράζεται στον κατάλληλο δυαδικό κώδικα μηχανής. Η μετάφραση συμπεριλαμβάνει τις εξής λειτουργίες:

1. Μετάφραση του μνημονικού ονόματος σε δυαδικό κωδικό εντολής.
2. Χρήση του κωδικού εντολής για τον προσδιορισμό της μορφής της εντολής, της θέσης και του μήκους των πεδίων της εντολής.
3. Μετάφραση του ονόματος κάθε παράγοντα στον κατάλληλο καταχωρητή ή κωδικό μνήμης.
4. Μετάφραση κάθε άμεσης τιμής σε μία δυαδική συμβολοσειρά.
5. Μετάφραση κάθε αναφοράς σε ετικέτες στην κατάλληλη τιμή του μετρητή θέσης με χρήση του πίνακα συμβόλων.
6. Ορισμός κάθε άλλου bit της εντολής το οποίο είναι αναγκαίο, συμπεριλαμβανομένων των δεικτών του τρόπου διευθυνσιοδότησης, των bits που αντιστοιχούν σε κωδικούς συνθηκών, κ.ο.κ.

Ένα απλό παράδειγμα, στο οποίο χρησιμοποιείται η γλώσσα συμβολομετάφρασης του επεξεργαστή ARM, παρουσιάζεται στο Σχήμα Β.5. Η εντολή ADDS r3, r3, #19 της γλώσσας, μεταφράζεται στη δυαδική εντολή μηχανής 1110 0010 0101 0011 0011 0000 0001 0011.

ΜΗΔΕΝΙΚΟ ΠΕΡΑΣΜΑ Οι περισσότερες γλώσσες συμβολομετάφρασης συμπεριλαμβάνουν τη δυνατότητα ορισμού μακρο-ορισμών. Όταν αυτοί οι μακρο-ορισμοί είναι δυαθέσιμοι, υπάρχει ένα επιπλέον πέρασμα, το οποίο θα πρέπει να κάνει ο συμβολομεταφραστής, πριν από το πρώτο. Τυπικά, η γλώσσα συμβολομετάφρασης απαιτεί ότι όλοι οι μακρο-ορισμοί θα πρέπει να εμφανίζονται στην αρχή του προγράμματος.

Ο συμβολομεταφραστής ξεκινά αυτό το "μηδενικό πέρασμα", διαβάζοντας όλους τους μακρο-ορισμούς. Από τη στιγμή που αυτοί οι μακρο-ορισμοί θα αναγνωριστούν, ο συμβολομεταφραστής εξετάζει τον πηγαίο κώδικα και εξαπλώνει τους μακρο-ορισμούς με τις αντίστοιχες παραμέτρους τους, όταν συναντήσει κλήση σε αυτούς. Το πέρασμα αυτό παράγει μία νέα έκδοση του πηγαίου κώδικα, όπου όλες οι μορφές εξάπλωσης βρίσκονται στη θέση τους, ενώ οι μακρο-ορισμοί έχουν αφαιρεθεί.

Συμβολομεταφραστής Ενός Περάσματος

Είναι δυνατόν να υλοποιήσουμε ένα συμβολομεταφραστή, ο οποίος κάνει μόνον ένα πέρασμα στον πηγαίο κώδικα (χωρίς να μετράει το μηδενικό πέρασμα επεξεργασίας των μακρο-ορισμών). Η βασική δυσκολία που συναντάμε κατά την προσπάθεια συμβολομετάφρασης ενός προγράμματος με ένα μόνον πέρασμα, σχετίζεται με την προώθηση των αναφορών σε ετικέτες. Οι παράγοντες των εντολών πιθανόν να είναι κάποια σύμβολα τα οποία δεν έχουν οριστεί ακόμη στο πηγαίο πρόγραμμα. Επομένως, ο

συμβολομεταφραστής δεν γνωρίζει ποιά σχετική διεύθυνση θα πρέπει να εισάγει στη μεταφρασμένη εντολή.

Ουσιαστικά, η διαδικασία ανάλυσης της προώθησης αναφορών λειτουργεί ως ακολούθως: Όταν ο συμβολομεταφραστής συναντήσει έναν παράγοντα εντολής, ο οποίος είναι ένα σύμβολο που δεν έχει οριστεί ακόμη, θα κάνει τις εξής ενέργειες:

1. Αφήνει το πεδίο του παράγοντα κενό (μόνον μηδενικά), στη δυαδική εντολή η οποία προκύπτει μετά από τη συμβολομετάφραση.

2. Το σύμβολο το οποίο χρησιμοποιείται ως παράγοντας εισάγεται στον πίνακα συμβόλων. Η καταχώρηση του πίνακα λαμβάνει μία τιμή σημαίας, ώστε να δειχτεί ότι πρόκειται για μη ορισμένο σύμβολο.

3. Η διεύθυνση του πεδίου του παράγοντα που αναφέρεται στο μη ορισμένο σύμβολο, προστίθεται σε μία λίστα προώθησης αναφορών, η οποία σχετίζεται με την καταχώρηση του πίνακα συμβόλων.

Όταν συναντάται ο ορισμός του συμβόλου, έτσι ώστε να είναι δυνατή η συσχέτιση μίας τιμής του μετρητή θέσης με αυτό, ο συμβολομεταφραστής εισάγει την τιμή του μετρητή θέσης στην κατάλληλη καταχώρηση του πίνακα συμβόλων. Αν υπάρχει μία λίστα αναφορών προώθησης η οποία σχετίζεται με το σύμβολο, τότε ο συμβολομεταφραστής εισάγει την κατάλληλη διεύθυνση σε κάθε εντολή η οποία έχει παραχθεί προηγουμένως και βρίσκεται στη λίστα προώθησης αναφορών.

Παράδειγμα: Πρόγραμμα Εύρεσης Πρώτων Αριθμών

Στο σημείο αυτό, παρουσιάζουμε ένα παράδειγμα με ψευδοεντολές. Το παράδειγμα εξετάζει ένα πρόγραμμα εύρεσης πρώτων αριθμών. Θυμηθείτε ότι οι πρώτοι αριθμοί διαιρούνται με το 1 και τον εαυτό τους. Δεν υπάρχει κάποια σχέση βάσει της οποίας θα υλοποιηθεί αυτό το πρόγραμμα. Η μέθοδος που χρησιμοποιείται βασίζεται στην εύρεση όλων των παραγόντων των περιττών αριθμών, οι οποίοι βρίσκονται κάτω από ένα δοθέν όριο. Αν δεν είναι δυνατή η εύρεση ενός παράγοντα για έναν περιττό αριθμό, τότε είναι πρώτος. Το Σχήμα Β.6 δείχνει το βασικό αλγόριθμο, που γράφτηκε σε γλώσσα C. Το Σχήμα Β.7 δείχνει τον ίδιο αλγόριθμο, γραμμένο στη γλώσσα συμβολομετάφρασης NASM.

```
unsigned guess;                    /* τρέχουσα εικασία για πρώτο αριθμό */
unsigned factor;                   /* πιθανός παράγοντας εικασίας */
unsigned limit;                    /* εύρεση πρώτων αριθμών ως αυτή την τιμή */

printf ("Find primes up to : ");
scanf("%u", &limit);
printf ("2\n");                    /* χειρισμός των 2 αρχικών πρώτων αριθμών ως */
printf ("3\n");                    /* ειδική περίπτωση */
guess = 5;                         /* αρχική εικασία */
while (guess < = limit) {          /* αναζήτηση για παράγοντα εικασίας */
    factor = 3;
    while (factor * factor < guess && guess% factor != 0)
    factor + = 2;
    if (guess % factor != 0)
        printf ("%d\n", guess);
    guess += 2;                    /* εξέταση μόνο των περιττών αριθμών */
}
```

Πρόγραμμα γραμμένο σε γλώσσα C, το οποίο ελέγχει αν δύο αριθμοί είναι πρώτοι

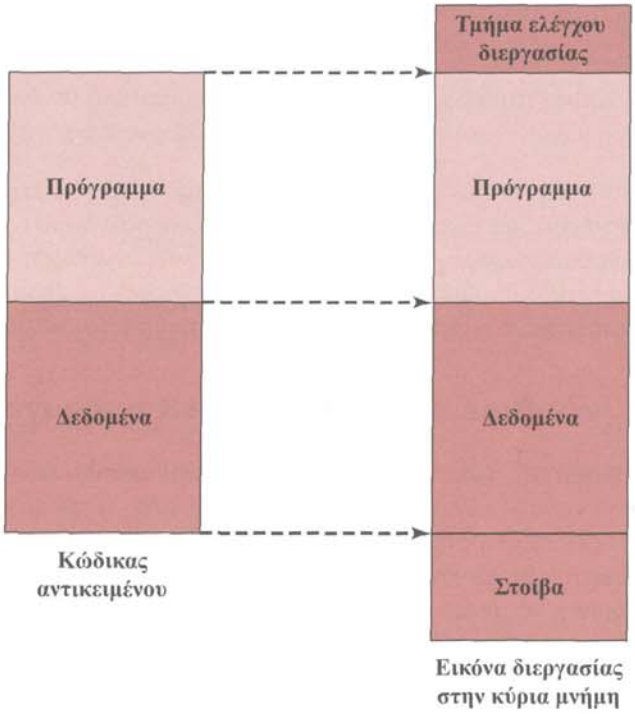
```
%include "asm_io.inc"
segment .data
Message db "Find primes up to: ", 0

segment .bss
Limit resd 1                        ; εύρεση πρώτων αριθμών μέχρι αυτό το όριο
Guess resd 1                        ; τρέχουσα εικασία για πρώτο αριθμό

segment .text
     global _asm_main
_asm_main:
     enter 0,0                      ; ρουτίνα εγκατάστασης
     pusha

     mov eax, Message
     call print_string
     call read_int                  ; scanf("%u", & limit);
     mov [Limit], eax
     mov eax, 2                     ; printf("2\n");
     call print_int
     call print_nl
     mov eax, 3                     ; printf("3\n");
     call print_int
     call print_nl

     mov dword [Guess], 5           ; Guess = 5;
while_limit:                        ; while (Guess <= Limit)
     mov eax, [Guess]
     cmp eax, [Limit]
     jnbe end_while_limit           ; χρήση της jnbe επειδή οι αριθμοί είναι μη προσημασμένοι

     mov ebx, 3                     ; η τιμή ebx είναι παράγοντας = 3;
while_factor:
     mov eax,ebx
     mul eax                        ; edx:eax = eax*eax
     jo end_while_factor            ; αν η απαίτηση δεν χωράει στον eax
     cmp eax, [Guess]
     jnb end_while_factor           ; if !(factor*factor < guess)
     mov eax,[Guess]
     mov edx,0
     div ebx                        ; edx = edx:eax% ebx
     cmp edx, 0
     je end_while_factor            ; if !(guess% factor != 0)

     add ebx,2; factor += 2;
     jmp while_factor
end_while_factor:
     je end_if                      ; if !(guess% factor != 0)
     mov eax,[Guess]                ; printf("%u\n")
     call print_int
     call print_nl
end_if:
     add dword [Guess], 2           ; guess += 2
     jmp while_limit
end_while_limit:

     popa
     mov eax, 0                     ; επιστροφή στο c
     leave
     ret
```

Πρόγραμμα γραμμένο σε γλώσσα συμβολομετάφρασης, το οποίο ελέγχει αν δύο αριθμοί είναι πρώτοι

Β.3

Το πρώτο βήμα για τη δημιουργία μίας ενεργού διεργασίας, είναι να φορτώσουμε το πρόγραμμα στην κύρια μνήμη και να δημιουργήσουμε μία εικόνα της διεργασίας (Σχήμα Β.8). Το Σχήμα Β.9 απεικονίζει ένα σενάριο, το οποίο είναι τυπικό στα περισσότερα συστήματα. Η εφαρμογή αποτελείται από ένα πλήθος μονάδων κώδικα, οι οποίες έχουν περάσει από τη διαδικασία συμβολομετάφρασης ή μεταγλώττισης και έχουν μετατραπεί σε αντικειμενικό κώδικα. Τα τμήματα έχουν διασυνδεθεί για

να αναλυθεί κάθε αναφορά μεταξύ τους. Την ίδια στιγμή, έχουν αναλυθεί οι αναφορές σε ρουτίνες βιβλιοθηκών. Οι ρουτίνες αυτές, είναι πιθανό να ενσωματωθούν στο πρόγραμμα ή να αναφερθούν ως κοινόχρηστος κώδικας, τον οποίο θα πρέπει να παρέχει το λειτουργικό σύστημα κατά το χρόνο εκτέλεσης. Σε αυτή την ενότητα, συνοψίζουμε τα βασικά γνωρίσματα των εργαλείων φόρτωσης και διασύνδεσης. Αρχικά, θα συζητήσουμε την έννοια της επανατοποθέτησης. Στη συνέχεια, για λόγους αποσαφήνισης, θα περιγράψουμε την εργασία της φόρτωσης, όταν χρησιμοποιείται ένα αυτόνομο πρόγραμμα. Στην περίπτωση αυτή, δεν απαιτείται διασύνδεση. Στη συνέχεια, μπορούμε να εξετάσουμε συνολικά τις λειτουργίες διασύνδεσης και φόρτωσης.

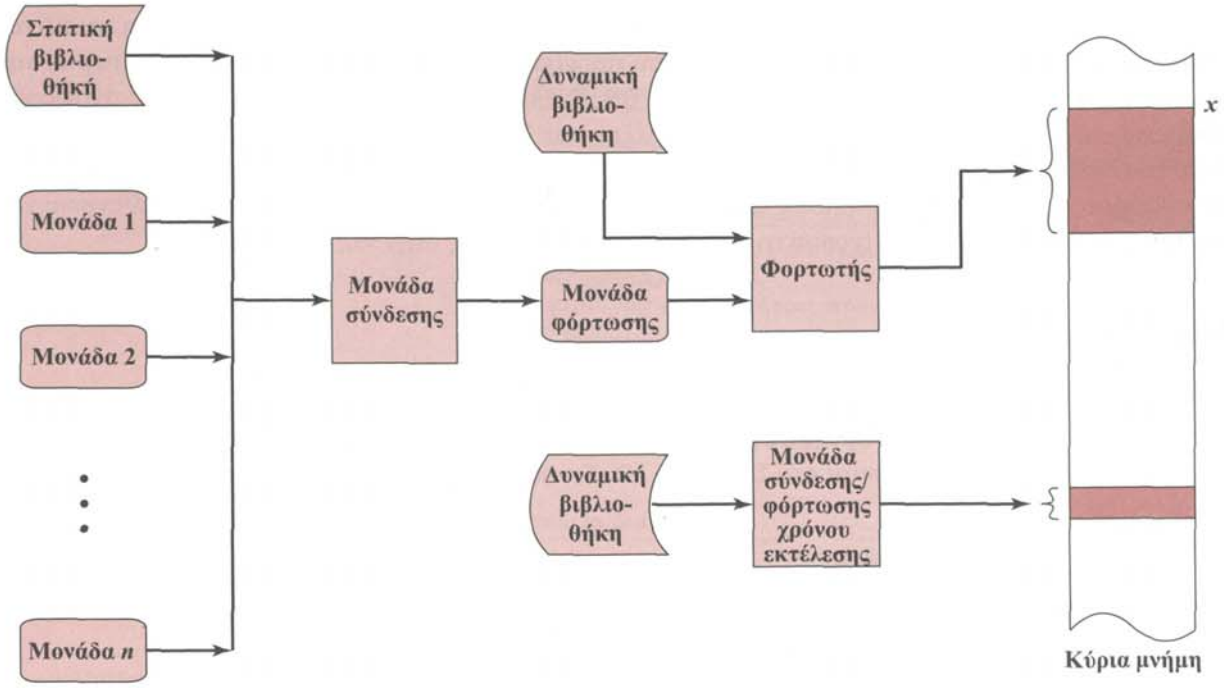


Σχήμα Β.8: Η λειτουργία φόρτωσης

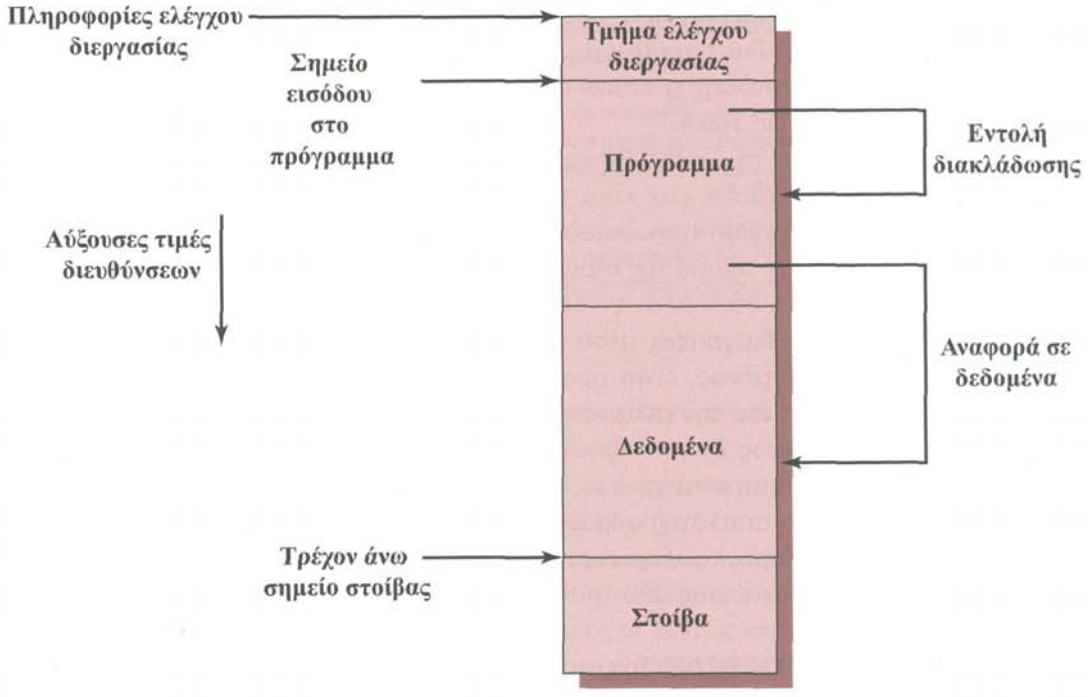
Επανατοποθέτηση

Σε ένα σύστημα πολυπρογραμματισμού, η διαθέσιμη μνήμη είναι γενικά μοιρασμένη ανάμεσα σε ένα πλήθος διεργασιών. Τυπικά, δεν είναι εφικτό για τον προγραμματιστή να γνωρίζει από την αρχή ποια άλλα προγράμματα θα βρίσκονται στη μνήμη όταν εκτελεί το πρόγραμμά του. Επιπλέον, επιθυμούμε να έχουμε τη δυνατότητα να εναλλάσσουμε τις ενεργές διεργασίες αφαιρώντας και επαναφέροντάς τες στη μνήμη, ώστε να μεγιστοποιήσουμε το βαθμό χρήσης του επεξεργαστή μέσα από τη δημιουργία μίας μεγάλης ομάδας έτοιμων προς εκτέλεση διεργασιών. Από τη στιγμή που ένα πρόγραμμα έχει βγει μέσω εναλλαγής από τη μνήμη και έχει τοποθετηθεί στο δίσκο, θα είναι αρκετά περιοριστικό να δηλώσουμε ότι όταν ξαναεισέλθει στη μνήμη θα πρέπει να τοποθετηθεί στις ίδιες θέσεις. Αντίθετα, είναι πιθανό να πρέπει να **επανατοποθετήσουμε** τη διεργασία, σε άλλη περιοχή της μνήμης.

Επομένως, δεν είμαστε σε θέση να γνωρίζουμε από πριν που θα τοποθετηθεί ένα πρόγραμμα και θα πρέπει να επιτρέψουμε τη μετακίνηση ενός προγράμματος η οποία οφείλεται στην εναλλαγή μέσα στην κύρια μνήμη. Αυτά τα γεγονότα εγείρουν ορισμένα τεχνικά ζητήματα όσον αφορά τη διευθυνσιοδότηση, όπως δείχνει το Σχήμα Β.10, όπου απεικονίζεται η εικόνα μίας διεργασίας. Για



Σχήμα Β.9: Ένα σενάριο διασύνδεσης και φόρτωσης



Σχήμα Β.10: Απαιτήσεις διευθυνσιοδότησης μίας διεργασίας

λόγους απλότητας, ας υποθέσουμε ότι η εικόνα της διεργασίας καταλαμβάνει μία συνεχή περιοχή της μνήμης. Προφανώς, το λειτουργικό σύστημα θα πρέπει να γνωρίζει τη θέση των πληροφοριών ελέγχου της διεργασίας και της στοίβας εκτέλεσης, καθώς και το σημείο εισαγωγής από το οποίο θα ξεκινήσει η εκτέλεση του προγράμματος για αυτή τη διεργασία. Επειδή το λειτουργικό σύστημα διαχειρίζεται τη μνήμη και είναι υπεύθυνο για την προσκόμιση της διεργασίας στην κύρια μνήμη,

αυτές οι διευθύνσεις είναι εύκολο να βρεθούν. Ωστόσο, ο επεξεργαστής θα πρέπει επίσης να χειρίζεται αναφορές στη μνήμη, οι οποίες γίνονται εντός του προγράμματος. Οι εντολές διακλάδωσης περιέχουν μία διεύθυνση αναφοράς της εντολής η οποία θα εκτελεστεί στη συνέχεια. Οι εντολές αναφοράς δεδομένων περιέχουν τη διεύθυνση του byte ή λέξης των δεδομένων αναφοράς. Με κάποιο τρόπο, το υλικό του επεξεργαστή και το λειτουργικό σύστημα θα πρέπει να έχουν τη δυνατότητα να μεταφράζουν της αναφορές της μνήμης της οποίες βρίσκουν μέσα στον κώδικα, σε πραγματικές διευθύνσεις της φυσικής μνήμης, οι οποίες δείχνουν τη θέση του προγράμματος στην κύρια μνήμη.

Φόρτωση

Στο Σχήμα Β.9, ο φορτωτής τοποθετεί τη μονάδα φόρτωσης στην κύρια μνήμη, ξεκινώντας από τη θέση x. Κατά τη φόρτωση του προγράμματος, θα πρέπει να ικανοποιηθούν οι απαιτήσεις διευθυνσιοδότησης, οι οποίες απεικονίζονται στο Σχήμα Β.10. Γενικά, υπάρχουν τρεις προσεγγίσεις:

- Απόλυτη φόρτωση
- Επανατοποθετούμενη φόρτωση
- Δυναμική φόρτωση, κατά τη διάρκεια εκτέλεσης του προγράμματος

ΑΠΟΛΥΤΗ ΦΟΡΤΩΣΗ Ένα εργαλείο απόλυτης φόρτωσης απαιτεί από μία μονάδα φόρτωσης να φορτώνεται πάντοτε στην ίδια θέση μνήμης. Επομένως, στη μονάδα φόρτωσης η οποία παρουσιάζεται στο φορτωτή, όλες οι αναφορές διευθύνσεων θα πρέπει να γίνονται σε συγκεκριμένες ή απόλυτες διευθύνσεις της κύριας μνήμης. Για παράδειγμα, στο Σχήμα Β.9, αν η θέση x ισούται με 1024, τότε η πρώτη λέξη μίας μονάδας φόρτωσης η οποία προορίζεται να φορτωθεί σε εκείνη την περιοχή της μνήμης, θα έχει διεύθυνση ίση με 1024.

Η εκχώρηση συγκεκριμένων τιμών διεύθυνσης σε αναφορές μνήμης εντός ενός προγράμματος, γίνονται είτε από τον προγραμματιστή, είτε κατά τη διαδικασία μεταγλώττισης ή συμβολομετάφρασης (Πίνακας Β.3α). Η πρώτη προσέγγιση παρουσιάζει διάφορα μειονεκτήματα. Πρώτον, κάθε προγραμματιστής θα πρέπει να γνωρίζει το σκοπό της στρατηγικής εκχώρησης βάσει της οποίας τοποθετούνται οι μονάδες στην κύρια μνήμη. Δεύτερον, αν γίνουν οποιεσδήποτε αλλαγές στο πρόγραμμα, οι οποίες συμπεριλαμβάνουν εισαγωγές ή διαγραφές μέσα στο σώμα της μονάδας, τότε θα πρέπει να τροποποιηθούν όλες οι διευθύνσεις. Επομένως, είναι προτιμότερο να επιτρέψουμε τις συμβολικές αναφορές μνήμης μέσα στα προγράμματα και την ανάλυσή τους κατά τη διαδικασία της μεταγλώττισης ή συμβολομετάφρασης. Αυτό το γεγονός απεικονίζεται στο Σχήμα Β.11. Κάθε αναφορά σε μία εντολή ή αντικείμενο δεδομένων αναπαρίσταται αρχικά με ένα σύμβολο. Για να ετοιμαστεί η μονάδα προκειμένου να εισαχθεί σε ένα εργαλείο απόλυτης φόρτωσης, ο συμβολομεταφραστής ή ο μεταγλωττιστής θα μετατρέψει όλες αυτές τις αναφορές σε συγκεκριμένες διευθύνσεις (σε αυτό το παράδειγμα, για μία μονάδα η οποία θα φορτωθεί ξεκινώντας από τη θέση 1024), όπως φαίνεται στο Σχήμα Β.11β.

ΕΠΑΝΑΤΟΠΟΘΕΤΟΥΜΕΝΗ ΦΟΡΤΩΣΗ Το μειονέκτημα της δέσμευσης συγκεκριμένων διευθύνσεων για συγκεκριμένες αναφορές μνήμης πριν από τη φόρτωση, είναι ότι η μονάδα φόρτωσης η οποία προκύπτει, μπορεί να φορτωθεί σε μία μόνον περιοχή της μνήμης. Ωστόσο, όταν η μνήμη μοιράζεται σε πολλά προγράμματα, πιθανόν να μην είναι επιθυμητό να αποφασίζουμε από πριν σε ποια περιοχή θα πρέπει να τοποθετηθεί μία συγκεκριμένη μονάδα. Είναι προτιμότερο να λάβουμε αυτή την απόφαση κατά τη διάρκεια της φόρτωσης. Επομένως, χρειαζόμαστε μία μονάδα φόρτωσης, η οποία είναι δυνατόν να φορτωθεί οπουδήποτε στη μνήμη.

Για να ικανοποιηθεί αυτή η νέα απαίτηση, ο μεταγλωττιστής ή ο συμβολομεταφραστής δεν παράγει πραγματικές διευθύνσεις κύριας μνήμης (απόλυτες διευθύνσεις), αλλά διευθύνσεις σχετικές με κάποιο γνωστό σημείο, όπως είναι το σημείο έναρξης ενός προγράμματος. Αυτή η τεχνική απεικονίζεται στο

Πίνακας Β.3: Δέσμευση διευθύνσεων

(α) Φορτωτής

Χρόνος Δέσμευσης	Λειτουργία
Χρόνος προγραμματισμού	Όλες οι πραγματικές φυσικές διευθύνσεις προσδιορίζονται άμεσα από τον προγραμματιστή μέσα στο ίδιο το πρόγραμμα
Χρόνος μεταγλώττισης ή συμβολομετάφρασης	Το πρόγραμμα περιέχει αναφορές σε συμβολικές διευθύνσεις οι οποίες μετατρέπονται σε πραγματικές φυσικές διευθύνσεις από το μεταγλωττιστή ή το συμβολομεταφραστή.
Χρόνος φόρτωσης	Ο μεταγλωττιστής ή ο συμβολομεταφραστής παράγει τις σχετικές διευθύνσεις. Ο φορτωτής μεταφράζει αυτές τις διευθύνσεις σε απόλυτες, κατά τη διάρκεια φόρτωσης του προγράμματος.
Χρόνος εκτέλεσης	Το πρόγραμμα το οποίο φορτώνεται διατηρεί τις σχετικές διευθύνσεις. Αυτές μετατρέπονται δυναμικά σε απόλυτες, από το υλικό του επεξεργαστή.

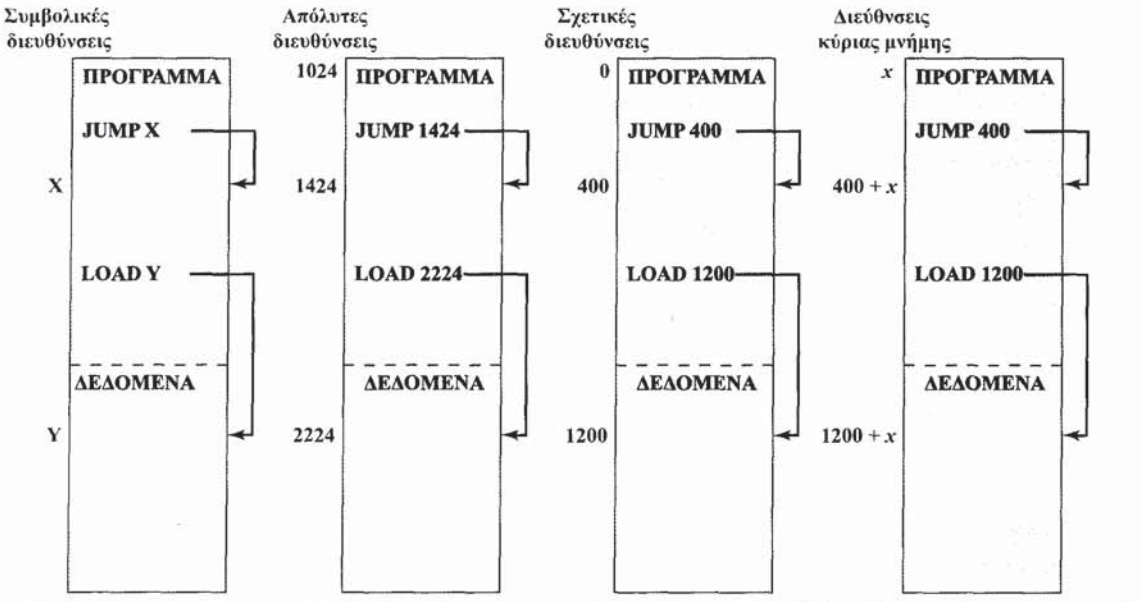
(β) Εργαλείο Διασύνδεσης

Χρόνος Διασύνδεσης	Λειτουργία
Χρόνος προγραμματισμού	Δεν επιτρέπονται εξωτερικές αναφορές προγραμμάτων ή δεδομένων. Ο προγραμματιστής θα πρέπει να τοποθετήσει στο πρόγραμμα τον πηγαίο κώδικα για κάθε υποπρόγραμμα το οποίο αναφέρεται.
Χρόνος μεταγλώττισης ή συμβολομετάφρασης	Ο συμβολομεταφραστής θα πρέπει να προσκομίσει τον πηγαίο κώδικα κάθε υπορουτίνας η οποία αναφέρεται και να συνθέσει όλες αυτές τις ρουτίνες σε μία μονάδα.
Δημιουργία μονάδων φόρτωσης	Όλες οι μονάδες έχουν περάσει από τη διαδικασία συμβολομετάφρασης, με χρήση σχετικών διευθύνσεων. Αυτές οι μονάδες διασυνδέονται μεταξύ τους και όλες οι αναφορές αναδιατυπώνονται σε σχέση με το σημείο έναρξης της τελικής μονάδας φόρτωσης.
Χρόνος φόρτωσης	Οι εξωτερικές αναφορές δεν αναλύονται έως ότου η μονάδα φόρτωσης φτάσει στο σημείο να φορτωθεί στη μνήμη. Εκείνη τη χρονική στιγμή, οι αναφερόμενες δυναμικές μονάδες διασύνδεσης επικολλούνται στη μονάδα φόρτωσης και ολόκληρο το πακέτο φορτώνεται στην κύρια ή στην ιδεατή μνήμη.
Χρόνος εκτέλεσης	Οι εξωτεικές αναφορές δεν αναλύονται έως ότου εκτελεστεί η εξωτερική κλήση από τον επεξεργαστή. Εκείνη τη στιγμή, η διεργασία διακόπτεται και η επιθυμητή μονάδα διασυνδέεται με το πρόγραμμα κλήσης.

Σχήμα Β.11γ. Στο σημείο έναρξης της μονάδας φόρτωσης εκχωρείται η σχετική διεύθυνση 0 και όλες οι άλλες αναφορές μνήμης εντός της μονάδας εκφράζονται σε σχέση με το σημείο έναρξής της.

Με όλες τις αναφορές στη μνήμη εκφρασμένες με μία σχετική μορφή, η τοποθέτηση της μονάδας στην επιθυμητή θέση γίνεται σχετικά εύκολη για το φορτωτή. Αν η μονάδα πρόκειται να φορτωθεί με σημείο εκκίνησης τη θέση x, τότε ο φορτωτής θα πρέπει απλώς να προσθέσει το x σε κάθε αναφορά μνήμης καθώς φορτώνει τη μονάδα στη μνήμη. Για να διευκολύνει αυτή την εργασία, η μονάδα φόρτωσης θα πρέπει να συμπεριλαμβάνει πληροφορίες οι οποίες αναφέρουν στο φορτωτή που βρίσκονται οι αναφορές διευθύνσεων και πως θα πρέπει να ερμηνευτούν (συνήθως σε σχέση με την αρχή του προγράμματος, αλλά επίσης πιθανόν σε σχέση και με κάποιο άλλο σημείο του προγράμματος, όπως είναι η τρέχουσα θέση). Αυτό το σύνολο πληροφοριών δημιουργείται από το μεταγλωττιστή ή το συμβολομεταφραστή και αναφέρεται συνήθως ως λεξικό επανατοποθέτησης.

ΔΥΝΑΜΙΚΗ ΦΟΡΤΩΣΗ, ΚΑΤΑ ΤΗ ΔΙΑΡΚΕΙΑ ΕΚΤΕΛΕΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ Οι φορτωτές επανατοποθέτησης είναι συνηθισμένοι και παρέχουν προφανή ωφέλη σε σχέση με τους απόλυτους φορτωτές. Ωστόσο, σε ένα περιβάλλον πολυπρογραμματισμού, ακόμη και αν αυτό δεν εξαρτάται από την ιδεατή μνήμη, η τεχνική φόρτωσης με επανατοποθέτηση δεν είναι επαρκής. Έχουμε ήδη αναφέρει την ανάγκη εναλλαγής των εικόνων των διεργασιών, οι οποίες διαδοχικά εισέρχονται στη



(α) Μονάδα αντικειμένου (β) Μονάδα απόλυτης φόρτωσης (γ) Μονάδα σχετικής φόρτωσης (δ) Μονάδα σχετικής φόρτωσης η οποία φορτώνεται στην κύρια μνήμη ξεκινώντας από τη θέση X

Σχήμα Β.11: Απόλυτη και επανατοποθετούμενη φόρτωση μονάδων

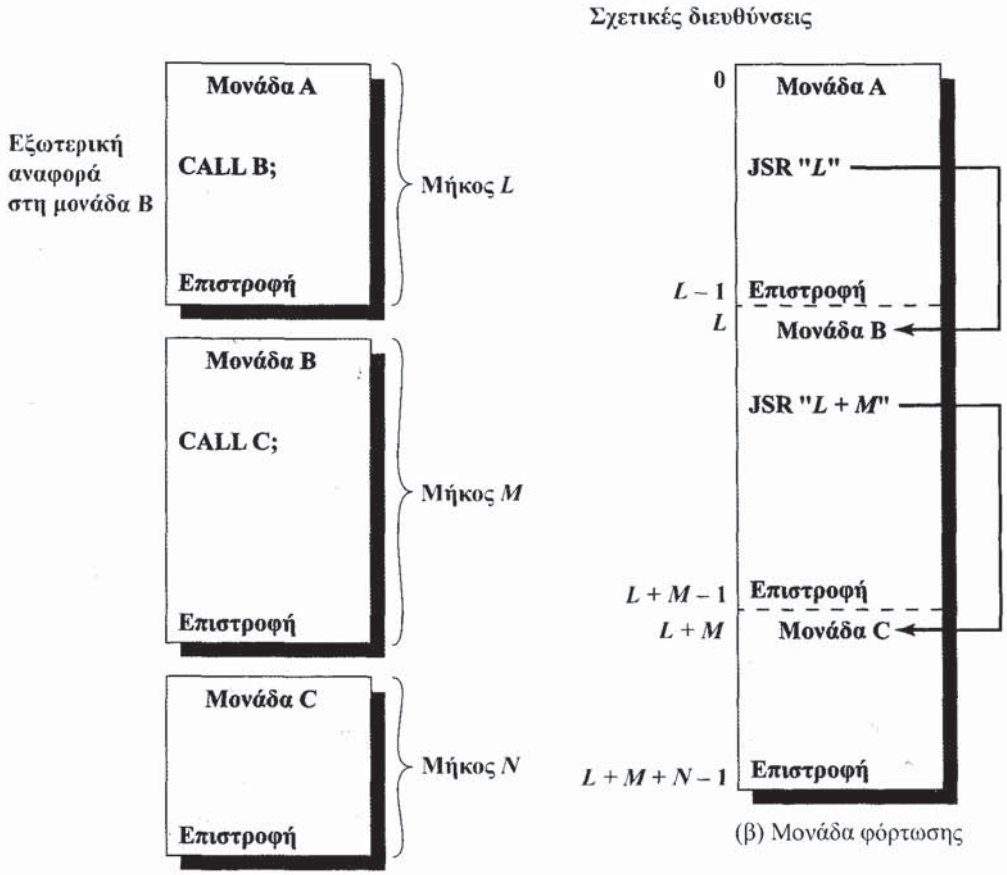
μνήμη και εξέρχονται από αυτή, ώστε να μεγιστοποιηθεί ο βαθμός χρήσης του επεξεργαστή. Για να βελτιστοποιηθεί ο βαθμός χρήσης της μνήμης, επιθυμούμε να έχουμε τη δυνατότητα να τοποθετούμε την εικόνα μίας διεργασίας πίσω στη μνήμη, σε διαφορετικές θέσεις και σε διαφορετικές χρονικές στιγμές. Επομένως, από τη στιγμή που θα φορτωθεί ένα πρόγραμμα είναι πιθανόν να εξέλθει από τη μνήμη και να τοποθετηθεί στο δίσκο και στη συνέχεια να εισέλθει εκ νέου σε άλλη περιοχή της μνήμης. Αυτό δεν θα ήταν εφικτό αν οι αναφορές στη μνήμη δεσμευόταν με απόλυτες διευθύνσεις κατά το χρόνο έναρξης της φόρτωσης.

Η εναλλακτική λύση είναι να αναβάλλουμε τον υπολογισμό της απόλυτης διεύθυνσης έως ότου αυτή χρειαστεί κατά τη διάρκεια εκτέλεσης του προγράμματος. Για το σκοπό αυτό, η μονάδα φόρτωσης φορτώνεται στην κύρια μνήμη με όλες τις αναφορές μνήμης σε σχετική μορφή (Σχήμα Β.11γ). Η απόλυτη διεύθυνση υπολογίζεται όταν εκτελείται πραγματικά η εντολή. Για να εξασφαλίσουμε ότι αυτή η λειτουργία δεν μειώνει την απόδοση, θα πρέπει να υλοποιηθεί σε ειδικό υλικό του επεξεργαστή και όχι με λογισμικό. Αυτό το υλικό περιγράφηκε στο Κεφάλαιο 8.

Ο δυναμικός υπολογισμός της διεύθυνσης παρέχει πλήρη ευελιξία. Κάθε πρόγραμμα είναι δυνατόν να φορτωθεί σε κάθε περιοχή της κύριας μνήμης. Επομένως, η εκτέλεση ενός προγράμματος είναι δυνατόν να διακοπεί, το πρόγραμμα να εξέλθει από την κύρια μνήμη και στη συνέχεια να επανατοποθετηθεί σε διαφορετική περιοχή.

Διασύνδεση

Η λειτουργία ενός εργαλείου διασύνδεσης είναι να λαμβάνει ως είσοδο ένα σύνολο από μονάδες αντικειμένων και να παράγει μία μονάδα φόρτωσης, αποτελούμενη από ένα ενιαίο σύνολο μονάδων του προγράμματος και μονάδων δεδομένων, ώστε στη συνέχεια, αυτό το σύνολο να μεταβιβαστεί στο φορτωτή. Σε κάθε μονάδα αντικειμένων, είναι πιθανόν να υπάρχουν αναφορές διευθύνσεων σε θέσεις άλλων μονάδων. Σε οποιαδήποτε μη συνδεδεμένη μονάδα αντικειμένων, κάθε αναφορά αυτής της μορφής είναι δυνατόν να εκφραστεί μόνον συμβολικά. Το εργαλείο διασύνδεσης δημιουργεί μία απλή



Σχήμα Β.12: Η λειτουργία της διασύνδεσης

μονάδα φόρτωσης η οποία αποτελεί τη λογική συνένωση όλων των άλλων μονάδων αντικειμένων. Κάθε αναφορά μεταξύ μονάδων θα πρέπει να μεταβάλλεται από μία συμβολική διεύθυνση σε μία αναφορά θέσης εντός της συνολικής μονάδας φόρτωσης. Για παράδειγμα, η μονάδα Α του Σχήματος Β.12α περιλαμβάνει την ενεργοποίηση μίας διαδικασίας η οποία ανήκει στη μονάδα Β. Όταν αυτές οι μονάδες συνενωθούν στη μονάδα φόρτωσης, αυτή η συμβολική αναφορά στη μονάδα Β μετατρέπεται σε μία συγκεκριμένη αναφορά στη θέση του σημείου εισόδου της μονάδας Β εντός της μονάδας φόρτωσης.

ΕΚΔΟΤΗΣ ΔΙΑΣΥΝΔΕΣΗΣ Η φύση αυτής της διασύνδεσης διευθύνσεων εξαρτάται από τον τύπο της μονάδας φόρτωσης η οποία θα δημιουργηθεί, αλλά και από το χρονικό σημείο που θα γίνει η διασύνδεση (Πίνακας Β.3β). Αν, ως συνήθως, επιθυμείται μία επανατοποθετούμενη μονάδα φόρτωσης, τότε η διασύνδεση γίνεται συνήθως με τον ακόλουθο τρόπο. Μέσα από μία διαδικασία μεταγλώττισης ή συμβολομετάφρασης, κάθε μονάδα αντικειμένων δημιουργείται με αναφορές σχετικές με το σημείο έναρξης της μονάδας αντικειμένων. Όλες αυτές οι μονάδες τοποθετούνται μαζί σε μία επανατοποθετούμενη μονάδα φόρτωσης, με όλες τις αναφορές να είναι σχετικές με το σημείο έναρξης της μονάδας φόρτωσης. Αυτή η μονάδα είναι δυνατόν να χρησιμοποιηθεί ως είσοδος για την επανατοποθετούμενη φόρτωση ή για τη δυναμική φόρτωση, κατά τη διάρκεια εκτέλεσης του προγράμματος.

Ένα πρόγραμμα διασύνδεσης το οποίο παράγει μία επανατοποθετούμενη μονάδα φόρτωσης, αναφέρεται συχνά και ως εκδότης διασύνδεσης. Το Σχήμα Β.12 απεικονίζει τη λειτουργία του εκδότη διασύνδεσης.

ΔΥΝΑΜΙΚΟ ΕΡΓΑΛΕΙΟ ΔΙΑΣΥΝΔΕΣΗΣ Όπως συμβαίνει και με τη φόρτωση, υπάρχει δυνατότητα να αναβάλλουμε προσωρινά ορισμένες λειτουργίες της διασύνδεσης. Ο όρος *δυναμική δισύνδεση* χρησιμοποιείται για να αναφερθεί στην πρακτική αναβολής της διασύνδεσης ορισμένων εξωτερικών μονάδων, μέχρι την ολοκλήρωση της δημιουργίας της μονάδας φόρτωσης. Επομένως, η μονάδα φόρτωσης περιέχει αναφορές, οι οποίες δεν έχουν εξεταστεί σε άλλα προγράμματα. Αυτές οι αναφορές εξετάζονται είτε κατά τη διάρκεια της φόρτωσης, είτε κατά τη διάρκεια της εκτέλεσης.

Για τη **δυναμική διασύνδεση κατά τη διάρκεια της φόρτωσης** (η οποία συμπεριλαμβάνει τη δυναμική βιβλιοθήκη στο άνω μέρος του Σχήματος Β.9), λαμβάνουν χώρα τα εξής γεγονότα. Η μονάδα φόρτωσης (μονάδα εφαρμογών) που πρόκειται να φορτωθεί τοποθετείται στη μνήμη. Κάθε αναφορά σε μία εξωτερική μονάδα (μονάδα προορισμού) αναγκάζει το φορτωτή να βρει τη μονάδα προορισμού, να τη φορτώσει, και να μετατρέψει την αναφορά σε σχετική διεύθυνση της μνήμης, από το σημείο έναρξης της μονάδας εφαρμογών. Αυτή η προσέγγιση εμφανίζει διάφορα πλεονεκτήματα, σε σχέση με τη στατική διασύνδεση:

- Καθίσταται ευκολότερη η ενσωμάτωση τροποποιημένων ή ενημερωμένων εκδόσεων της μονάδας προορισμού, η οποία μπορεί να είναι ένα βοηθητικό πρόγραμμα του λειτουργικού συστήματος, ή κάποια άλλη ρουτίνα γενικού σκοπού. Με τη στατική διασύνδεση, η αλλαγή σε μία τέτοια μονάδα υποστήριξης θα απαιτούσε την επανασύνδεση ολόκληρης της μονάδας εφαρμογών. Αυτό όχι μόνο είναι αναποτελεσματικό, αλλά σε μερικές περιπτώσεις είναι και ανέφικτο. Για παράδειγμα, στο πεδίο ενός προσωπικού υπολογιστή, το μεγαλύτερο μέρος του εμπορικού λογισμικού εκδίδεται σε μορφή μονάδων φόρτωσης. Δεν εκδίδονται οι πηγαίες μονάδες και οι αντικειμενικές μονάδες.

- Έχοντας τον κώδικα προορισμού σε ένα αρχείο δυναμικής διασύνδεσης, ανοίγει ο δρόμος για αυτόματη κοινή χρήση του κώδικα. Το λειτουργικό σύστημα έχει τη δυνατότητα αναγνώρισης του γεγονότος ότι περισσότερες από μία εφαρμογές χρησιμοποιούν τον ίδιο κώδικα προορισμού, επειδή φόρτωσε και διασύνδεσε αυτόν τον κώδικα. Το λειτουργικό σύστημα μπορεί να χρησιμοποιήσει αυτές τις πληροφορίες για να φορτώσει ένα απλό αντίγραφο του κώδικα προορισμού και να το διασυνδέσει με περισσότερες εφαρμογές, αντί να φορτώνει ένα αντίγραφο σε κάθε εφαρμογή.

- Καθίσταται ευκολότερο για τους ανεξάρτητους προγραμματιστές να επεκτείνουν τη λειτουργικότητα ενός ευρέως χρησιμοποιούμενου λειτουργικού συστήματος, όπως είναι το Linux. Ένας προγραμματιστής μπορεί να δημιουργήσει μία νέα συνάρτηση, η οποία να είναι χρήσιμη σε μία ποικιλία εφαρμογών και να τη συσκευάσει ως δυναμική μονάδα διασύνδεσης.

Με τη τη **δυναμική διασύνδεση κατά τη διάρκεια της εκτέλεσης** (η οποία συμπεριλαμβάνει τη δυναμική βιβλιοθήκη στο κάτω μέρος του Σχήματος Β.9), ένα μέρος της διασύνδεσης αναβάλλεται μέχρι τον χρόνο της εκτέλεσης. Οι εξωτερικές αναφορές σε μονάδες προορισμού παραμένουν στο πρόγραμμα που έχει φορτωθεί. Όταν γίνει μία κλήση στη μονάδα η οποία απουσιάζει, το λειτουργικό σύστημα θα εντοπίσει αυτή τη μονάδα, θα τη φορτώσει, και θα τη διασυνδέσει με τη μονάδα κλήσης. Αυτές οι μονάδες συνήθως είναι κοινόχρηστες. Στο περιβάλλον των Windows, αυτές οι μονάδες ονομάζονται βιβλιοθήκες δυναμικής διασύνδεσης (Dynamic Link Libraries-DLLs). Επομένως, αν μία διεργασία ήδη χρησιμοποιεί μία δυναμικά διασυνδεδεμένη κοινόχρηστη μονάδα, τότε αυτή η μονάδα βρίσκεται στην κύρια μνήμη και μία νέα διεργασία μπορεί απλώς να διασυνδεθεί στη μονάδα που έχει ήδη φορτωθεί.

Η χρήση των DLLs είναι δυνατόν να οδηγήσει σε ένα πρόβλημα γνωστό και ως η **κόλαση των DLLs**. Μία κόλαση DLL εμφανίζεται όταν δύο ή περισσότερες διεργασίες μοιράζονται μία μονάδα DLL, αλλά αναμένουν διαφορετικές εκδόσεις της μονάδας. Για παράδειγμα, μία εφαρμογή ή συνάρτηση συστήματος είναι πιθανό να επανεγκατασταθεί και να φέρει μαζί της μία παλαιότερη έκδοση ενός αρχείου DLL.

Έχουμε δει ότι η δυναμική φόρτωση επιτρέπει τη μετακίνηση ολόκληρης της μονάδας φόρτωσης. Ωστόσο, η δομή της μονάδας είναι στατική και δεν αλλάζει κατά την εκτέλεση μίας διεργασίας, αλλά και μεταξύ διαφορετικών εκτελέσεων. Ωστόσο, σε μερικές περιπτώσεις, δεν είναι δυνατόν να προσδιορίσουμε πριν από την εκτέλεση ποιες μονάδες αντικειμένων θα χρειαστούν. Αυτή η κατάσταση εμφανίζεται τυπικά σε εφαρμογές επεξεργασίας συναλλαγών, όπως το σύστημα κράτησης αεροπορικών εισητηρίων ή οι τραπεζικές εφαρμογές. Η φύση της συναλλαγής υπαγορεύει ποιες μονάδες του προγράμματος είναι απαραίτητες. Οι μονάδες αυτές φορτώνονται ως κατάλληλες και διασυνδέονται με το κύριο πρόγραμμα. Το πλεονέκτημα της χρήσης ενός δυναμικού εργαλείου διασύνδεσης, είναι ότι δεν είναι απαραίτητη η κατανομή μνήμης σε μονάδες του προγράμματος, εκτός αν υπάρχουν αναφορές σε αυτές. Αυτή η δυνατότητα χρησιμοποιείται για την υποστήριξη συστημάτων τμηματοποίησης.

Επίσης, είναι εφικτή μία ακόμη βελτίωση: Μία εφαρμογή δεν είναι δυνατόν να γνωρίζει τα ονόματα όλων των μονάδων ή σημείων έναρξης που είναι πιθανό να κληθούν. Για παράδειγμα, ένα σχεδιαστικό πρόγραμμα είναι πιθανό να έχει γραφτεί ώστε να λειτουργεί με μία ποικιλία συσκευών εκτύπωσης σχεδίων, κάθε μία από τις οποίες οδηγείται από ένα διαφορετικό πακέτο. Η εφαρμογή μπορεί να μάθει το όνομα της συσκευής η οποία είναι εγκατεστημένη στο σύστημα μέσω μίας άλλης διεργασίας, ή εξετάζοντας ένα αρχείο διαμόρφωσης. Αυτό το γεγονός επιτρέπει στο χρήστη της εφαρμογής να εγκαταστήσει μία νέα συσκευή η οποία δεν υπήρχε το χρόνο που γράφτηκε η εφαρμογή.

Β.4 ΠΡΟΤΕΙΝΟΜΕΝΗ ΜΕΛΕΤΗ

Η αναφορά [SALO93] καλύπτει τα ζητήματα σχεδίασης συμβολομεταφραστών και φορτωτών.

Τα ζητήματα της διασύνδεσης και φόρτωσης καλύπτονται σε πολλά βιβλία προγραμματισμού, αρχιτεκτονικής υπολογιστών, και λειτουργικών συστημάτων. Λεπτομερής παρουσίαση υπάρχει στο βιβλίο [BECK97]. Στο βιβλίο [CLAR98] επίσης υπάρχει καλή περιγραφή. Μία λεπτομερής συζήτηση πάνω στο αντικείμενο, με πολλά παραδείγματα λειτουργικών συστημάτων, υπάρχει στο βιβλίο [LEVI00].

Το βιβλίο [BART03] περιέχει εξαιρετική παρουσίαση για την εκμάθηση της γλώσσας συμβολομετάφρασης που χρησιμοποιεί η οικογένεια επεξεργαστών x86 και είναι χρήσιμο για ατομική μελέτη. Το βιβλίο [CART06] καλύπτει τη γλώσσας συμβολομετάφρασης των μηχανών x86. Για ένα σοβαρό προγραμματιστή που χρησιμοποιεί αυτή τη γλώσσα, το βιβλίο [FOG08a] είναι εξαιρετικά χρήσιμο. Τέλος, η γλώσσας συμβολομετάφρασης που χρησιμοποιεί η οικογένεια επεξεργαστών ARM περιγράφεται λεπτομερώς στην αναφορά [KNAG04].

BART03 Bartlett, J. *Programming from the Ground Up.* 2003. Διαθέσιμο στο Δικτυακό τόπο αυτού του βιβλίου.

BECK97 Beck, L. *System Software.* Reading, MA:Addison-Wesley, 1997.

CART06 Carter,P *PC Assembly Language.* July 23, 2006. Διαθέσιμο στο Δικτυακό τόπο αυτού του βιβλίου.

CLAR98 Clarke, D., and Merusi, D. *System Software Programming: The Way Things Work.* Upper Saddle River, NJ: Prentice Hall, 1998.

FOG08a Fog,A. *Optimizing Subroutines in Assembly Language:An Optimization Guide for x86 Platforms.* Copenhagen University College of Engineering, 2008. http:// www.agner.org/optimize/

KNAG04 Knaggs,P.,and Welsh,S *ARM:Assembly Language Programming.* Bournemouth University School of Design, Engineering & Computing.August 31, 2004. Διαθέσιμο στο Δικτυακό τόπο αυτού του βιβλίου.

LEVI00 Levine, J. *Linkers and Loaders.* San Francisco: Morgan Kaufmann, 2000.

SALO93 Salomon, D. *Assemblers and Loaders.* Ellis Horwood Ltd, 1993. Διαθέσιμο στο Δικτυακό τόπο αυτού του βιβλίου.

Προτεινόμενοι Διαδικτυακοί Τόποι:

- **Gavin's Guide to 80x86 Assembly:** Μία καλή, λεπτομερής εξέταση της γλώσσας συμβολομετάφρασης που χρησιμοποιεί η οικογένεια επεξεργαστών x86.
- **The Art of Assembly Language Programming:** Ένα πολύ μεγάλο online βιβλίο, περίπου 1500 σελίδες, με αντικείμενο τον προγραμματισμό σε γλώσσα συμβολομετάφρασης. Θα πρέπει να αρκεί σε ένα φοιτητή, ο οποίος μελετά το αντικείμενο.

Β.5 ΒΑΣΙΚΟΙ ΟΡΟΙ, ΕΡΩΤΗΣΕΙΣ ΕΠΑΝΑΛΗΨΗΣ ΚΑΙ ΠΡΟΒΛΗΜΑΤΑ

Βασικοί Όροι

γλώσσα συμβολομετάφρασης	επανατοποθέτηση	συμβολομεταφραστής
διασύνδεση	εργαλείο δυναμικής διασύνδεσης	συμβολομεταφραστής δύο περασμάτων
δυναμική διασύνδεση κατά τη διάρκεια της εκτέλεσης	ετικέτα	συμβολομεταφραστής ενός περάσματος
δυναμική διασύνδεση κατά τη διάρκεια φόρτωσης	μακρο-ορισμός	σχόλια
εκδότης διασύνδεσης	μνημονικό όνομα	φόρτωση
εντολή	παράγοντας	ψευδοεντολή

Ερωτήσεις Επανάληψης

Β.1. Να παραθέσετε ορισμένους λόγους για τους οποίους αξίζει να μελετηθεί ο προγραμματισμός σε γλώσσα συμβολομετάφρασης.

Β.2. Τι είναι η γλώσσα συμβολομετάφρασης;

Β.3. Να παραθέσετε ορισμένα μειονεκτήματα της γλώσσα συμβολομετάφρασης σε σχέση με τις γλώσσες υψηλού επιπέδου.

Β.4. Να παραθέσετε ορισμένα πλεονεκτήματα της γλώσσα συμβολομετάφρασης σε σχέση με τις γλώσσες υψηλού επιπέδου.

Β.5. Ποιά είναι τα τυπικά στοιχεία μίας πρότασης γραμμένης σε γλώσσα συμβολομετάφρασης;

Β.6. Να παραθέσετε και να ορίσετε σύντομα τέσσερα διαφορετικά είδη προτάσεων που είναι γραμμένες σε γλώσσα συμβολομετάφρασης.

Β.7. Ποιά είναι η διαφορά ανάμεσα στο συμβολομεταφραστή ενός και δύο περασμάτων;

Προβλήματα

Β.1 Το Core War είναι ένα παιχνίδι προγραμματισμού, το οποίο εισήχθη στις αρχές της δεκαετίας του '80 και ήταν πολύ δημοφιλές για περίπου 15 χρόνια. Το Core War έχει τέσσερα βασικά στοιχεία: έναν πίνακα μνήμης αποτελούμενο από 8000 θέσεις, μία απλοποιημένη γλώσσα συμβολομετάφρασης, την RedCode, έναν προσομοιωτή με την ονομασία MARS-Memory Array Redcode Simulator, και το σύνολο των αντιμαχώμενων προγραμμάτων. Δύο από τα διαμαχόμενα

προγράμματα, εισάγονται στον πίνακα της μνήμης σε τυχαία επιλεγμένες θέσεις. Κανένα πρόγραμμα δεν γνωρίζει που βρίσκεται το άλλο. Ο προσομοιωτής MARS εκτελεί τα προγράμματα με έναν απλό τρόπο μοιράσματος του χρόνου. Τα δύο προγράμματα παίρνουν σειρά: εκτελείται μία απλή εντολή ενός προγράμματος, στη συνέχεια μία απλή εντολή του άλλου, κ.ο.κ. Το τι θα κάνει ένα πρόγραμμα μάχης κατά τη διάρκεια των κύκλων εκτέλεσης που κατανέμονται σε αυτό, εξαρτάται αποκλειστικά από τον προγραμματιστή. Ο στόχος είναι να καταστραφούν οι εντολές του άλλου προγράμματος. Σε αυτό το πρόβλημα και σε διάφορα άλλα, χρησιμοποιούμε μία ακόμη πιο απλή γλώσσα, η οποία ονομάζεται CodeBlue, ώστε να μελετήσουμε ορισμένες έννοιες του CoreWar.

Η γλώσσα CodeBlue περιέχει μόνον 5 προτάσεις γραμμένες σε γλώσσα συμβολομετάφρασης και χρησιμοποιεί τρεις τρόπους διεθυνσιοδότησης (Πίνακας Β.4). Οι διευθύνσεις περιτυλίγονται, έτσι ώστε για την τελευταία θέση της μνήμης, η σχετική διεύθυνση του +1 αναφέρεται στην πρώτη θέση της μνήμης. Για παράδειγμα, η εντολή `ADD #4, 6` προσθέτει 4 στα περιεχόμενα της σχετικής θέσης 6 και αποθηκεύει τα αποτελέσματα στη θέση 6. Η εντολή `JUMP @5` μεταφέρει την εκτέλεση στη διεύθυνση μνήμης η οποία περιέχεται στη θέση η οποία βρίσκεται 5 θέσεις μετά από τη θέση της τρέχουσας εντολής `JUMP`.

Πίνακας Β.4: Η γλώσσα συμβολομετάφρασης CodeBlue

(α) Σύνολο Εντολών

Μορφή	**Σημασία**
DATA <value>	η τιμή του πεδίου <value> ορίζεται ως τιμή της τρέχουσας θέσης
COPY A, B	αντιγραφή της αφετηρίας A στον προορισμό B
ADD A, B	πρόσθεση των περιεχομένων της θέσης A στα περιεχόμενα της θέσης B, αποθήκευση του αποτελέσματος στη θέση B
JUMP A	μεταφορά της εκτέλεσης στη θέση A
JUMPZ A, B	Αν B=0, μεταφορά στη θέση A

(β) Τρόποι Διευθυνσιοδότησης

Τρόπος	**Μορφή**	**Σημασία**
Κυριολεκτικός	# ακολουθούμενο από τιμή	Άμεσος τρόπος, η τιμή του παράγοντα υπάρχει στην εντολή
Σχετικός	Τιμή	Η τιμή αναπαριστά μία μετατόπιση από την τρέχουσα θέση, η οποία περιέχει τον παράγοντα.
Έμμεσος	@ ακολουθούμενο από τιμή	Η τιμή αναπαριστά μία μετατόπιση από την τρέχουσα θέση. Η θέση της μετατόπισης περιέχει τη σχετική διεύθυνση της θέσης η οποία περιέχει τον παράγοντα.

α. Το πρόγραμμα Ζιζάνιο (Imp) αποτελείται από την απλή εντολή `COPY 0, 1`. Τι κάνει αυτό το πρόγραμμα;

β. Το πρόγραμμα Νάνος (Dwarf) αποτελείται από την παρακάτω ακολουθία εντολών:

```
ADD #4, 3
COPY 2,
JUMP -2
DATA 0
```

Τι κάνει αυτό το πρόγραμμα;

γ. Να ξαναγράψετε το πρόγραμμα Νάνος, χρησιμοποιώντας σύμβολα, ώστε να μοιάζει περισσότερο με ένα τυπικό πρόγραμμα γλώσσας συμβολομετάφρασης.

Β.2 Τι θα συμβεί αν μονομαχήσει ο Νάνος με το Ζιζάνιο;

Β.3 Να γράψετε ένα πρόγραμμα "βομβαρδισμού", το οποίο μηδενίζει ολόκληρη τη μνήμη (με πιθανή εξαίρεση τις θέσεις του προγράμματος).

Β.4 Τι θα επιτύχει το παρακάτω πρόγραμμα εναντίον του Ζιζανίου;

```
LOOP    COPY  #0, 1
        JUMP  -1
```

Υπόδειξη: Θυμηθείτε ότι η εκτέλεση των εντολών εναλλάσσεται ανάμεσα στα δύο αντιμαχόμενα προγράμματα.

Β.5 **α.** Ποιά είναι η τιμή της σημαίας κατάστασης C μετά την εκτέλεση της παρακάτω ακολουθίας;

```
mov al, 3
add al, 4
```

β. Ποιά είναι η τιμή της σημαίας κατάστασης C μετά την εκτέλεση της παρακάτω ακολουθίας;

```
mov al, 3
sub al, 4
```

Β.6 Θεωρείστε την παρακάτω εντολή της γλώσσας NAMS:

```
cmp vleft, vright
```

Για προσημασμένους ακέραιους, υπάρχουν τρεις σημαίες κατάστασης, οι οποίες είναι σχετικές. Αν vleft = vright, τότε η σημαία ZF λαμβάνει τιμή 1. Αν vleft > vright, τότε η σημαία ZF μηδενίζεται και SF=OF. Αν vleft < vright, η σημαία ZF μηδενίζεται και SF≠OF. Γιατί ισχύει ότι SF=OF, αν vleft > vright;

Β.7 Θεωρείστε τον παρακάτω κώδικα της γλώσσας NAMS:

```
mov al, 0
cmp al, al
je next
```

Να γράψετε ένα ισοδύναμο πρόγραμμα, αποτελούμενο από μία μόνον εντολή.

Β.8 Θεωρείστε το παρακάτω πρόγραμμα, το οποίο είναι γραμμένο σε γλώσσα C:

```
/ Απλό πρόγραμμα υπολογισμού της μέσης τιμής τριών ακεραίων /
main ()
{ int avg;
   int i1 = 20;
   int i2 = 13;
   int i3 = 82;
   avg = (i1 + i2 + i3)/3;
}
```

Να γράψετε μία έκδοση του προγράμματος σε γλώσσα NASM.

Β.9 Θεωρείστε το παρακάτω τμήμα κώδικα, το οποίο είναι γραμμένο σε γλώσσα C:

```
if (EAX == 0) EBX = 1;
 else EBX = 2;
```

Να γράψετε μία ισοδύναμη έκδοση του προγράμματος σε γλώσσα NASM.

Β.10 Οι ψευδοεντολές αρχικοποίησης δεδομένων είναι δυνατόν να χρησιμοποιηθούν για να αρχικοποιηθούν πολλαπλές θέσεις. Για παράδειγμα, η εντολή

```
db 0x55,0x56,0x57
```

δεσμεύει 3 bytes και αρχικοποιεί τις τιμές τους.

Η γλώσσα NASM υποστηρίζει το ειδικό σύμβολο $ για να επιτρέπει στους υπολογισμούς να χρησιμοποιούν την τρέχουσα θέση συμβολομετάφρασης. Με άλλα λόγια, το σύμβολο $ αποτιμά τη θέση στην αρχή της γραμμής η οποία περιέχει την έκφραση. Έχοντας τα παραπάνω γεγονότα υπόψη, *θεωρείστε την παρακάτω ακολουθία ψευδοεντολών:*

```
message db "hello, world"
msglen equ $-message
```

Ποιά τιμή εκχωρείται στο σύμβολο msglen;

Β.11 Έστω ότι τρεις συμβολικές μεταβλητές V1, V2, V3 περιέχουν ακέραιες τιμές. Να γράψετε ένα τμήμα κώδικα σε γλώσσα NASM, ώστε η μικρότερη τιμή να μετακινείται στον ακέραιο ax. Να χρησιμοποιήσετε μόνον τις εντολές `moc`, `cmp` και `jbe`.

Β.12 Να περιγράψετε το αποτέλεσμα της εντολής `cmp eax, 1`.

Β.13 Η εντολή `xchg` χρησιμοποιείται για την εναλλαγή των περιεχομένων δύο καταχωρητών. Έστω ότι το σύνολο εντολών της μηχανής x86 δεν υποστήριζε αυτή την εντολή.

α. Να υλοποιήσετε την εντολή `xchg ax, bx` χρησιμοποιώντας μόνον τις εντολές `push` και `pop`.

β. Να υλοποιήσετε την εντολή `xchg ax, bx` χρησιμοποιώντας μόνον την εντολή `xor` (μην χρησιμοποιήσετε άλλους καταχωρητές).

Β.14 Στο παρακάτω πρόγραμμα, υποθέστε ότι τα a, b, x, y είναι σύμβολα θέσεων της κύριας μνήμης. Τι κάνει το πρόγραμμα αυτό; Μπορείτε να απαντήσετε γράφοντας το αντίστοιχο πρόγραμμα στη γλώσσα C.

```
        mov     eax,a
        mov     ebx,b
        xor     eax,x
        xor     ebx,y
        or      eax,ebx
        jnz     L2
L1:                         ;   ακολουθία εντολών....
        jmp     L3
L2:                         ;   μία άλλη ακολουθία εντολών....
L3:
```

Β.15 Η Ενότητα Β.1 συμπεριλαμβάνει ένα πρόγραμμα γραμμένο σε γλώσσα C το οποίο υπολογίζει το μέγιστο κοινό διαιρέτη δύο ακεραίων.

α. Να περιγράψετε λεκτικά τον αλγόριθμο και να δείξετε τον τρόπο με τον οποίο το πρόγραμμα υλοποιεί τον αλγόριθμο του Ευκλείδη για τον υπολογισμό του μέγιστου κοινού διαιρέτη.

β. *Να προσθέσετε σχόλια στο πρόγραμμα συμβολομετάφρασης* του Σχήματος Β.3α, ώστε να αποσαφηνίσετε ότι υλοποιεί τη λογική του προγράμματος που είναι γραμμένο σε γλώσσα C.

γ. Να επαναλάβετε το ερώτημα (β) για το πρόγραμμα του Σχήματος Β.3β.

Β.16 **α.** Ένας συμβολομεταφραστής δύο περασμάτων έχει τη δυνατότητα να διαχειριστεί τα μελλοντικά σύμβολα, επομένως, μία εντολή μπορεί να χρησιμοποιήσει ένα μελλοντικό σύμβολο ως παράγοντα. Αυτό δεν ισχύει πάντοτε για τις ψευδοεντολές. Για παράδειγμα, η ψευδοεντολή EQU δεν έχει τη δυνατότητα να χρησιμοποιήσει ένα μελλοντικό σύμβολο. Η ψευδοεντολή 'A EQU B+1' είναι εύκολο να εκτελεστεί αν έχει οριστεί το σύμβολο B από πριν, αλλά αδύνατο να εκτελεστεί αν το B είναι ένα μελλοντικό σύμβολο. Για ποιό λόγο συμβαίνει αυτό;

β. Να προτείνετε ένα τρόπο ώστε ο συμβολομεταφραστής να εξαλείψει τον περιορισμό, έτσι ώστε κάθε πηγαία γραμμή να έχει τη δυνατότητα να χρησιμοποιεί τα μελλοντικά σύμβολα.

Β.17 Θεωρείστε τη συμβολική ψευδοεντολή MAX, η οποία έχει την ακόλουθη μορφή:

σύμβολο MAX λίστα εκφράσεων

Η ετικέτα είναι υποχρεωτική και εκχωρείται σε αυτή η τιμή της μεγαλύτερης έκφρασης που υπάρχει στο πεδίο του παράγοντα. Για παράδειγμα,

```
MSGLEN MAX A, B, C ;  όπου A, B, C είναι σύμβολα
                      που έχουν οριστεί
```

Με ποιό τρόπο εκτελείται η ψευδοεντολή MAX από το συμβολομεταφραστή και σε ποιό πέρασμα;

ΛΕΞΙΚΟ ΟΡΩΝ

Αγγλικοί Όροι

ASCII American Standard Code for Information Exchange. Ο κώδικας ASCII είναι ένας κώδικας 7-bit, ο οποίος χρησιμοποιείται για να αναπαραστήσει αριθμητικούς, αλφαβητικούς, και ειδικούς εκτυπώσιμους χαρακτήρες. Επίσης, περιλαμβάνει κωδικούς για *χαρακτήρες ελέγχου*, οι οποίοι δεν εκτυπώνονται και δεν εμφανίζονται, αλλά προσδιορίζουν μία λειτουργία ελέγχου.

bit Σε ένα αμιγώς δυαδικό σύστημα, ένα από τα ψηφία 0 ή 1.

byte Μία ακολουθία 8 bit. Επίσης, αναφέρεται ως *οκτάδα*.

CD-ROM Compact Disk Read Only Memory Ένας μη διαγράψιμος δίσκος που χρησιμοποιείται για την αποθήκευση δεδομένων. Τυπικά, χρησιμοποιούνται δίσκοι 12-cm και είναι δυνατή η αποθήκευση περισσότερων από 550 Mbytes.

flip-flop Ένα κύκλωμα ή μονάδα, η οποία περιέχει ενεργά στοιχεία, τα οποία έχουν τη δυνατότητα να λάβουν μία από τις δύο σταθερές καταστάσεις σε κάθε χρονική στιγμή.

G Πρόθεμα το οποίο έχει την τιμή 2^{30}.

k Πρόθεμα το οποίο έχει την τιμή 2^{10}.

M Πρόθεμα το οποίο έχει την τιμή $2^{20} = 1,048,576$. Επομένως, 2Mb=2,097,152 bits.

Ελληνικοί Όροι

ακολουθιακό κύκλωμα Ένα λογικό κύκλωμα του οποίου η έξοδος εξαρτάται από την τρέχουσα είσοδο και την κατάσταση του κυκλώματος. Επομένως διαθέτει το χαρακτηριστικό της μνήμης.

άλμα χωρίς συνθήκη Ένα άλμα το οποίο λαμβάνει χώρα οποιεδήποτε εκτελείται η εντολή η οποία προσδιορίζεται.

άλμα υπό συνθήκη Ένα άλμα το οποίο λαμβάνει χώρα μόνον όταν εκτελείται η εντολή η οποία το προσδιορίζει και ικανοποιούνται οι καθορισμένες συνθήκες. Να συγκρίνετε με το *άλμα χωρίς συνθήκη*.

αλυσίδωση Μία μέθοδος διασύνδεσης συσκευών για τον προσδιορισμό των προτεραιοτήτων σχετικά με τις διακοπές. Η μέθοδος βασίζεται στη σειριακή σύνδεση των πηγών που προκαλούν τις διακοπές.

άμεση διεύθυνση Τα περιεχόμενα ενός τμήματος διεύθυνσης, τα οποία περιέχουν την τιμή ενός παράγοντα και όχι τη διεύθυνση. Συνώνυμο με τη *διεύθυνση μηδενικού επιπέδου*.

άμεση προσπέλαση Η δυνατότητα λήψης δεδομένων από μία μονάδα αποθήκευσης ή της εισαγωγής δεδομένων σε μία μονάδα, μέσω διευθύνσεων οι οποίες δείχνουν τη φυσική θέση των δεδομένων.

αναπαράσταση προσημασμένου μέτρου Χρησιμοποιείται για να αναπαριστά δυαδικούς ακεραίους. Σε μία λέξη N-bit, το αριστερότερο bit είναι το πρόσημο (0=θετικό, 1=αρνητικό) και τα υπόλοιπα $N-1$ bits συνιστούν

την ποσότητα του αριθμού.

αναπαράσταση συμπληρώματος ως προς 1 Χρησιμοποιείται για να αναπαραστήσει δυαδικούς ακέραιους. Ένας θετικός ακέραιος αναπαρίσταται ως προσημασμένο μέτρο. Ένας αρνητικός ακέραιος αναπαρίσταται αντιστρέφοντας κάθε bit της αναπαράστασης του αντίστοιχου θετικού ακεραίου.

αναπαράσταση συμπληρώματος ως προς 2 Χρησιμοποιείται για να αναπαραστήσει δυαδικούς ακέραιους. Ένας θετικός ακέραιος αναπαρίσταται ως προσημασμένο μέτρο. Ένας αρνητικός ακέραιος αναπαρίσταται αντιστρέφοντας κάθε bit της αναπαράστασης του αντίστοιχου θετικού ακεραίου και προσθέτοντας μία μονάδα στο υπόδειγμα bit το οποίο προκύπτει, θεωρώντας το μη προσημασμένο ακέραιο αριθμό.

απενεργοποιημένη διακοπή Μία συνθήκη, η οποία συνήθως δημιουργείται από τον επεξεργαστή, κατά τη διάρκεια της οποίας, ο επεξεργαστής αγνοεί κάθε σήμα αίτησης διακοπής μίας καθορισμένης κατηγορίας.

αποκωδικοποιητής Μία συσκευή, η οποία διαθέτει ένα πλήθος γραμμών εισόδου, από τις οποίες, οποιεσδήποτε είναι δυνατόν να μεταφέρουν σήματα, και ένα πλήθος γραμμών εξόδου από τις οποίες, μία μόνον είναι δυνατόν να μεταφέρει σήμα. Επομένως, υπάρχει μία αντιστοιχία μίας προς έναν ανάμεσα στις εξόδους και στους συνδυασμούς των σημάτων εισόδου.

απόλυτη διεύθυνση Μία διεύθυνση, η οποία καθορίζει τη θέση μνήμης ενός αντικειμένου δεδομένων, το οποίο θα χρησιμοποιηθεί ως παράγοντας. Συνώνυμο με την *διεύθυνση ενός επιπέδου*.

απομονωμένη είσοδος-έξοδος Μία μέθοδος διευθυνσιοδότησης των μονάδων Ε/Ε και των εξωτερικών συσκευών. Ο χώρος διευθύνσεων Ε/Ε αντιμετωπίζεται ξεχωριστά από το χώρο διευθύνσεων της κύριας μνήμης. Πρέπει να χρησιμοποιηθούν ειδικές εντολές μηχανής για την Ε/Ε. Συγκρίνετε με την *Ε/Ε απεικόνισης στη μνήμη*.

απόλυτη διεύθυνση Μία διεύθυνση σε μία γλώσσα υπολογιστή, η οποία προσδιορίζει μία θέση μνήμης ή μία συσκευή, χωρίς να χρησιμοποιείται έμμεση αναφορά

απομονωτής Μία μονάδα αποθήκευσης, η οποία χρησιμοποιείται για να αντισταθμιστεί η διαφορά που υπάρχει στο ρυθμό της ροής δεδομένων, ή στο χρόνο εμφάνισης των γεγονότων, όταν μεταφέρονται δεδομένα από μία συσκευή σε μία άλλη.

απομονωτής κρυφής μνήμης Ένας ειδικός απομονωτής, μικρότερος και ταχύτερος από την κύρια μνήμη, ο οποίος χρησιμοποιείται για να αποθηκεύει ένα αντίγραφο των εντολών και δεδομένων της κύριας μνήμης, τα οποία είναι πιθανό να χρειαστούν στον επεξεργαστή και τα οποία έχουν ληφθεί αυτόματα από την κύρια μνήμη.

αριθμητική και λογική μονάδα (ALU) Ένα τμήμα του υπολογιστή, το οποίο εκτελεί τις αριθμητικές, λογικές, και άλλες σχετικές πράξεις

αρχιτεκτονική υπολογιστών Τα χαρακτηριστικά ενός συστήματος, τα οποία είναι ορατά σε έναν προγραμματιστή, ή διαφορετικά, τα χαρακτηριστικά τα οποία έχουν άμεση επίδραση στη λογική εκτέλεση ενός προγράμματος. Παραδείγματα χαρακτηριστικών της αρχιτεκτονικής αποτελούν το σύνολο εντολών, το πλήθος των bits τα οποία χρησιμοποιούνται για να αναπαρασταθούν διάφοροι τύποι δεδομένων (π.χ. αριθμοί, χαρακτήρες), οι μηχανισμοί Ε/Ε, και οι τεχνικές διευθυνσιοδότησης της μνήμης.

ασύγχρονος χρονισμός Μία τεχνική στην οποία η εμφάνιση ενός γεγονότος σε ένα δίαυλο ακολουθεί και εξαρτάται από την εμφάνιση ενός προηγούμενου γεγονότος.

αυτοδεικτοδότηση Μία μορφή διευθυνσιοδότησης με δείκτες, όπου ο καταχωρητής δείκτη αυξάνεται ή μειώνεται αυτόματα με κάθε αναφορά στη μνήμη

αφέντης διαύλου Μία συσκευή η οποία συνδέεται σε ένα δίαυλο και είναι σε θέση να εκκινήσει και να ελέγχει την επικοινωνία πάνω σε αυτόν.

βαθμωτή ποσότητα Μία ποσότητα η οποία χαρακτηρίζεται από μία τιμή.

βάση Σε ένα σύστημα απαρίθμησης το οποίο είναι συνηθισμένο σε επιστημονικά άρθρα, ο αριθμός ο οποίος υψώνεται στη δύναμη την οποία δηλώνει ο εκθέτης, και έπειτα πολλαπλασιάζεται με το σημαντικό μέρος, για να προσδιορίσει τον πραγματικό αριθμό ο οποίος αναπαρίσταται (π.χ. ο αριθμός 10 στην έκφραση $2.7 \times 10^2 = 270$

γλώσσα μικροπρογραμματισμού Ένα σύνολο εντολών, το οποίο χρησιμοποιείται για να προσδιορίσει μικροπρογράμματα.

γλώσσα συμβολομετάφρασης Μία γλώσσα υπολογιστή, της οποίας οι εντολές αντιστοιχούν μία προς μία με τις εντολές του υπολογιστή και παρέχει διευκολύνσεις, όπως είναι η χρήση μακροεντολών. Είναι συνώνυμη με την έννοια της *εξαρτημένης από του υπολογιστή γλώσσας*

γραμμή κρυφής μνήμης Ένα μπλοκ δεδομένων, το οποίο σχετίζεται με μία ετικέτα κρυφής μνήμης και τη μονάδα η οποία μεταφέρεται ανάμεσα στην κρυφή και την κύρια μνήμη.

δεικτοδότηση Μία τεχνική αλλαγής της διεύθυνσης με χρήση καταχωρητών δείκτη.

δεικτοδοτούμενη διεύθυνση Μία διεύθυνση η οποία τροποποιείται από τα περιεχόμενα ενός καταχωρητή δείκτη, πριν ή κατά τη διάρκεια της εκτέλεσης μίας εντολής.

δευτερεύουσα μνήμη Μνήμη η οποία βρίσκεται έξω από το ίδιο το σύστημα του υπολογιστή, με άλλα λόγια δεν είναι δυνατή η άμεση επεξεργασία των δεδομένων της από τον επεξεργαστή. Αρχικά, θα πρέπει να αντιγράφεται στην κύρια μνήμη. Παραδείγματα είναι ο δίσκος και η ταινία.

διαγράψιμος οπτικός δίσκος Ένας δίσκος οπτικής τεχνολογίας, ο οποίοα είναι εύκολο να διαγραφεί και να επαναγραφεί. Χρησιμοποιούνται δίσκοι 3.25 και 5.25 ιντσών. Η τυπική χωρητικότητά τους είναι 650 MB.

διαιτησία διαύλου Η διαδικασία μέσω της οποίας αποφασίζεται σε ποιον από τους ανταγωνιζόμενους αφέντες του διαύλου θα επιτραπεί η προσπέλαση σε αυτόν.

διακοπή Η αναστολή μίας διεργασίας, όπως η εκτέλεση ενός προγράμματος, η οποία προκαλείται από ένα γεγονός εξωτερικό προς αυτή, με τέτοι τρόπο, ώστε να είναι δυνατή η επαναφορά της.

διάνυσμα Μία ποσότητα η οποία συνήθως χαρακτηρίζεται από ένα διατεταγμένο σύνολο βαθμωτών τιμών.

διασωλήνωση Μία μορφή οργάνωσης του επεξεργαστή, στην οποία αυτός αποτελείται από ένα πλήθος σταδίων, επιτρέποντας την ταυτόχρονη εκτέλεση πολλών εντολών.

δίαυλος Ένα κοινόχρηστο μονοπάτι επικοινωνιών, το οποίο αποτελείται από μία γραμμή ή ένα σύνολο γραμμών. Σε ορισμένα συστήματα υπολογιστών, ο επεξεργαστής, η μνήμη, και τα στοιχεία Ε/Ε συνδέονται με ένα κοινό δίαυλο. Από τη στιγμή που οι γραμμές μοιράζονται από όλα τα στοιχεία, ένα στοιχείο είναι δυνατόν να μεταδώσει δεδομένα σε κάθε χρονική στιγμή.

δίαυλος δεδομένων Το τμήμα του διαύλου συστήματος, το οποίο χρησιμοποιείται για τη μεταφορά δεδομένων.

δίαυλος διευθύνσεων Είναι το τμήμα του διαύλου συστήματος, το οποίο χρησιμοποιείται για τη μεταφορά μίας διεύθυνσης. Τυπικά, η διεύθυνση προσδιορίζει μία θέση μνήμης, ή μία μονάδα Ε/Ε.

δίαυλος ελέγχου Το τμήμα ενός διαύλου συστήματος, το οποίο χρησιμοποιείται για τη μεταφορά σημάτων ελέγχου.

δίαυλος συστήματος Ένας δίαυλος ο οποίος χρησιμοποιείται για να διασυνδέει τα βασικά στοιχεία του υπολογιστή (επεξεργαστής, μνήμη, Ε/Ε).

διεργασία Ένα πρόγραμμα σε εκτέλεση. Μία διεργασία χρονοδρομολογείται και ελέγχεται από το λειτουργικό σύστημα.

διεύθυνση βάσης Μία αριθμητική τιμή, η οποία χρησιμοποιείται ως αναφορά κατά τον υπολογισμό των διευθύνσεων, όταν εκτελείται ένα πρόγραμμα.

δισκέτα Ένας εύκαμπτος μαγνητικός δίσκος ο οποίος βρίσκεται μέσα σε προστατευτικό υλικό. Συνώνυμο του εύκαμπτου δίσκου.

δυαδικό ψηφίο (bit) ισοτιμίας Ένα δυαδικό ψηφίο το οποίο ενσωματώνεται σε μία ομάδα δυαδικών ψηφίων, ώστε το άθροισμα των όλων των ψηφίων να είναι πάντοτε άρτιο (άρτια ισοτιμία) ή πάντοτε περιττό (περιττή ισοτιμία)

δυαδικός τελεστής Ένας τελεστής ο οποίος αναπαριστά μία πράξη πάνω σε δύο και μόνον δύο παράγοντες.

δυναμική RAM Μία μνήμη RAM, της οποίας τα κελιά υλοποιούνται από πυκνωτές. Μία δυναμική RAM βαθμιαία θα χάνει τα περιεχόμενά της, εκτός αν ανανεώνεται περιοδικά.

είσοδος-έξοδος (Ε/Ε) Αναφέρεται είτε σε είσοδο είτε σε έξοδο, είτε και στις δύο λειτουργίες μαζί. Αφορά τη διακίνηση δεδομένων ανάμεσα στον υπολογιστή και ένα άμεσα συνδεδεμένο περιφερειακό.

είσοδος-έξοδος καθοδηγούμενη από διακοπή Μία μορφή Ε/Ε. Ο επεξεργαστής εκδίδει μία εντολή Ε/Ε, συνεχίζει να εκτελεί τις επόμενες εντολές και διακόπτεται από τη μονάδα Ε/Ε, όταν αυτή ολοκληρώσει την εργασία της.

είσοδος-έξοδος με απεικόνιση στη μνήμη Μία μέθοδος διευθυνσιοδότησης μονάδων Ε/Ε και εξωτερικών συσκευών. Χεησιμοποιείται ένας απλός χώρος διευθύνσεων τόσο για την κύρια μνήμη, όσο και για τις διευθύνσεις Ε/Ε, ενώ οι ίδιες εντολές μηχανής χρησιμοποιούνται τόσο για τις λειτουργίες ανάγνωσης/εγγραφής, όσο και για τις λειτουργίες Ε/Ε.

έκδοση εντολής Η διαδικασία αρχικοποίησης της εκτέλεσης μίας εντολής στις λειτουργικές μονάδες του επεξεργαστή. Η έκδοση λαμβάνει χώρα όταν μία εντολή μετακινείται από το στάδιο αποκωδικοποίησης της διασωλήνωσης προς το πρώτο στάδιο εκτέλεσης.

ελεγκτής Ε/Ε Μία σχετικά απλή μονάδα Ε/Ε, η οποία απαιτεί λεπτομερή έλεγχο από τον επεξεργαστή ή από ένα κανάλι Ε/Ε. Συνώνυμο της έννοιας του *ελεγκτή συσκευής*.

έμμεση διεύθυνση Μία διεύθυνση μίας μονάδας μνήμης, η οποία περιέχει μία άλλη διεύθυνση.

έμμεσος κύκλος Το τμήμα του κύκλου εντολής, κατά τη διάρκεια του οποίου ο επεξεργαστής προσπελαύνει τη μνήμη για να μετατρέψει μία έμμεση διεύθυνση σε άμεση.

ενδιάμεσος καταχωρητής μνήμης (MBR) Ένας καταχωρητής, ο οποίος περιέχει δεδομένα τα οποία διαβάζονται από τη μνήμη ή που πρόκειται να γραφτούν σε αυτή.

εντολή υπολογιστή Μία εντολή η οποία αναγνωρίζεται από τη μονάδα επεξεργασίας του υπολογιστή για τον οποίο έχει σχεδιαστεί. Συνώνυμο με την *εντολή μηχανής*.

εξομοίωση η διαδικασία μίμησης της συμπεριφοράς ενός συστήματος ή μέρους αυτού από ένα άλλο, κυρίως μέσω υλικού, ώστε το σύστημα μίμησης να αποδέχεται τα ίδια δεδομένα, να εκτελεί τα ίδια προγράμματα, και να επιτυγχάνει τα ίδια αποτελέσματα με το μιμούμενο σύστημα.

επεξεργαστής Σε έναν υπολογιστή, μία λειτουργική μονάδα η οποία ερμηνεύει και εκτελεί εντολές. Ένας επεξεργαστής αποτελείται από τουλάχιστον μία μονάδα ελέγχου εντολών και μία αριθμητική μονάδα.

επεξεργαστής εισόδου-εξόδου Μία μονάδα Ε/Ε, η οποία διαθέτει το δικό της επεξεργαστή και έχει τη δυνατότητα να εκτελέσει τις δικές της ειδικές εντολές Ε/Ε ή, σε μερικές περιπτώσεις, εντολές μηχανής γενικού σκοπού.

επεξεργαστής υπερδιασωλήνωσης Μία σχεδίαση επεξεργαστών, όπου η διασωλήνωση αποτελείται από πολλά μικρά στάδια, έτσι ώστε να είναι δυνατή η εκτέλεση περισσότερων από ενός σταδίων κατά τη διάρκεια ενός κύκλου ρολογιού, με αποτέλεσμα να βρίσκεται ταυτόχρονα στη διασωλήνωση ένα μεγάλο πλήθος εντολών.

επικοινωνία δεδομένων Η μεταφορά δεδομένων μεταξύ συσκευών. Ο όρος σε γενικές γραμμές αποκλείει την Ε/Ε.

επιτρεπόμενη διακοπή Μία συνθήκη, η οποία συνήθως παράγεται από τον επεξεργαστή κατά τη διάρκεια της οποίας ο επεξεργαστής θα αποκρίνεται σε σήματα αίτησης διακοπής μίας συγκεκριμένης κατηγορίας.

εύκαμπτος δίσκος (CD) Ένας μη διαγράψιμος δίσκος αποθήκευσης ψηφιοποιημένων πληροφοριών ήχου.

ημιαγωγός Μία στερεή κρυσταλλική ουσία όπως πυρίτιο ή γερμάνιο της οποία η ηλεκτρική αγωγιμότητα βρίσκεται ανάμεσα στους μονωτές και τους καλούς αγωγούς. Χρησιμοποιείται για την κατασκευή τρανζιστορς και στοιχείων στερεάς κατάστασης.

ιδεατή μνήμη Ο χώρος αποθήκευσης ο οποίος μπορεί να θεωρηθεί από το χρήστη ως διευθυνσιοδοτούμενη κύρια μνήμη, όπου οι ιδεατές διευθύνσεις απεικονίζονται σε πραγματικές. Το μέγεθος της ιδεατής μνήμης περιορίζεται από την τεχνική διευθυνσιοδότησης του συστήματος και από την ποσότητα της βοηθητικής μνήμης η οποία είναι διαθέσιμη, και όχι από το πραγματικό πλήθος των θέσεων της κύριας μνήμης.

καθολική μεταβλητή Μία μεταβλητή η οποία ορίζεται σε ένα τμήμα του προγράμματος και χρησιμοποιείται τουλάχιστον σε ένα άλλο τμήμα.

κανάλι Ε/Ε Μία σχετικά πολύπλοκη μονάδα Ε/Ε, η οποία ανακουφίζει τον επεξεργαστή από τις λεπτομέρειες των πράξεων Ε/Ε. Ένα κανάλι Ε/Ε θα εκτελέσει μία ακολουθία εντολών Ε/Ε από την κύρια μνήμη, χωρίς να υπάρχει ανάγκη συμμετοχής του επεξεργαστή.

κανάλι επιλογέα Ένα κανάλι Ε/Ε, το οποίο είναι σχεδιασμένο για να λειτουργεί με μία μόνον μονάδα Ε/Ε κάθε φορά. Από τη στιγμή που επιλέγεται η συσκευή, μεταφέρεται μία πλήρης εγγραφή, ένα byte τη φορά. Να συγκρίνετε με τους όρους *κανάλι πολυπλέκτη μπλοκ* και *κανάλι πολυπλέκτη*.

κανάλι πολυπλέκτη Ένα κανάλι το οποίο είναι σχεδιασμένο για να λειτουργεί ταυτόχρονα με ένα πλήθος συσκευών Ε/Ε. Διάφορες συσκευές είναι δυνατόν να μεταφέρουν εγγραφές ταυτόχρονα, παρεμβάλλοντας αντικείμενα δεδομένων. Δείτε επίσης τις έννοιες *κανάλι πολυπλέκτη μπλοκ* και *κανάλι πολυπλέκτη byte*

κανάλι πολυπλέκτη byte Ένα κανάλι πολυπλέκτη, το οποίο παρεμβάλλει bytes δεδομένων. Δείτε επίσης τον όρο *κανάλι πολυπλέκτη μπλοκ*. Να συγκρίνετε με τον όρο *κανάλι επιλογέα*.

κανάλι πολυπλέκτη μπλοκ Ένα κανάλι πολυπλέκτη, το οποίο παρεμβάλλει μπλοκ δεδομένων. Δείτε επίσης τον όρο *κανάλι πολυπλέκτη byte*. Να συγκρίνετε με τον όρο *κανάλι επιλογέα*.

καταχωρητές Μνήμη υψηλής ταχύτητας, η οποία βρίσκεται στο εσωτερικό του επεξεργαστή. Ορισμένοι καταχωρητές είναι ορατοί από το χρήστη, δηλαδή διαθέσιμοι στον προγραμματισμό μέσα από το σύνολο εντολών του επεξεργαστή. Άλλοι καταχωρητές χρησιμοποιούνται μόνον από τον επεξεργαστή, για να εξυπηρετήσουν τους σκοπούς του ελέγχου.

καταχωρητές ελέγχου Καταχωρητές του επεξεργαστή, οι οποίοι χρησιμοποιούνται για να ελέγξουν τη λειτουργία του. Οι περισσότεροι από αυτούς δεν είναι ορατοί στο χρήστη.

καταχωρητές ορατοί από το χρήστη Καταχωρητές του επεξεργαστή οι οποίοι είναι δυνατόν να αναφερθούν από τον προγραμματιστή. Η μορφή του συνόλου εντολών επιτρέπει σε έναν ή περισσότερους καταχωρητές να προσδιοριστούν ως παράγοντες ή ως διευθύνσεις παραγόντων.

καταχωρητής γενικού σκοπού Ένας καταχωρητής, γενικά άμεσα διευθυνσιοδοτούμενος, ο οποίος βρίσκεται σε ένα σύνολο καταχωρητών και χρησιμοποιείται για διαφορετικούς σκοπούς, για παράδειγμα ως συσσωρευτής, ως καταχωρητής δείκτη, ή ως ειδική μονάδα διαχείρισης των δεδομένων.

καταχωρητής δείκτη Ένας καταχωρητής του οποίου τα περιεχόμενα είναι δυνατόν να χρησιμοποιηθούν για να αλλάξουν τη διεύθυνση ενός παράγοντα κατά τη διάρκεια εκτέλεσης των εντολών. Επίσης, χρησιμοποιείται ως μετρητής. Ένας καταχωρητής δείκτη χρησιμοποιείται για να ελέγξει την εκτέλεση ενός βρόχου, τη χρήση ενός πίνακα, ως διακόπτης, για εξέταση πινάκων, ή απλά ως δείκτης.

καταχωρητής διεύθυνσης εντολής Ένας καταχωρητής ειδικού σκοπού, ο οποίος χρησιμοποιείται για να αποθηκεύει τη διεύθυνση της επόμενης εντολής η οποία πρόκειται να εκτελεστεί.

καταχωρητής διεύθυνσης μνήμης (MAR) Ένας καταχωρητής που βρίσκεται μέσα στον επεξεργαστή, ο οποίος περιέχει τη διεύθυνση της θέσης μνήμης η οποία προσπελαύνεται.

καταχωρητής εντολής Ο καταχωρητής ο οποίος χρησιμοποιείται για να αποθηκεύσει μία εντολή προς ερμηνεία.

κεντρική μονάδα επεξεργασίας (CPU) Το τμήμα του υπολογιστή, το οποίο προσκομίζει και εκτελεί εντολές. Αποτελείται από την αριθμητική και λογική μονάδα, τη μονάδα ελέγχου, και τους καταχωρητές. Συχνά, αναφέρεται απλώς ως *επεξεργαστής*.

κρυφή μνήμη Μία σχετικά μικρή μνήμη υψηλής ταχύτητας, η οποία παρεμβάλλεται ανάμεσα στη μεγαλύτερη και πιο αργή μνήμη, και το μονάδα η οποία προσπελαύνει την πιο αργή μνήμη. Η κρυφή μνήμη αποθηκεύει τα δεδομένα τα οποία προσπελάστηκαν πρόσφατα και είναι σχεδιασμένη ώστε να αυξάνει την ταχύτητα διαδοχικών προσπελάσεων στα ίδια δεδομένα.

κύκλος ανάκλησης Το τμήμα του κύκλου εντολής, κατά τη διάρκεια του οποίου ο επεξεργαστής προσκομίζει από τη μνήμη, την εντολή που πρόκειται να εκτελεστεί.

κύκλος διακοπής Το τμήμα του κύκλου εντολής, κατά τη διάρκεια του οποίου γίνεται έλεγχος για την ύπαρξη διακοπών. Αν υπάρχει εκκρεμής διακοπή, ο επεξεργαστής αποθηκεύει την κατάσταση του τρέχοντος προγράμματος και συνεχίζει την επεξεργασία από μία ρουτίνα εξυπηρέτησης της διακοπής.

κύκλος εκτέλεσης Το τμήμα του κύκλου εντολής κατά τη διάρκεια του οποίου ο επεξεργαστής εκτελεί τη λειτουργία την οποία καθορίζει ο κωδικός εντολής.

κύκλος εντολής Η επεξεργασία που κάνει η κεντρική μονάδα επεξεργασίας, ώστε να εκτελέσει μία εντολή.

κύρια μνήμη Διευθυνσιοδοτούμενη από πρόγραμμα μονάδα αποθήκευσης, από την οποία οι εντολές και άλλα δεδομένα είναι δυνατόν να φορτωθούν άμεσα σε καταχωρητές για τη μετέπειτα εκτέλεση ή επεξεργασία.

κώδικας διόρθωσης σφαλμάτων Ένας κώδικας στον οποίο κάθε χαρακτήρας ή σήμα συμμορφώνεται σε συγκεκριμένους κανόνες κατασκευής, έτσι ώστε οι αποκλίσεις από αυτούς τους κανόνες να δείχνουν την ύπαρξη σφάλματος.

κωδικός εντολής Ένας κωδικός ο οποίος χρησιμοποιείται για να αναπαραστήσει τις λειτουργίες ενός υπολογιστή. Η αγγλική συντομογραφία είναι opcode.

κωδικός συνθηκών Ένας κωδικός ο οποίος δείχνει το αποτέλεσμα μίας προηγούμενης πράξης (π.χ. αριθμητικής). Ένας επεξεργαστής πιθανόν να συμπεριλαμβάνει έναν ή περισσότερους κωδικούς συνθηκών, οι οποίοι είναι δυνατόν να αποθηκευτούν ξεχωριστά σε αυτόν, ή ως μέρος ενός μεγαλύτερου καταχωρητή ελέγχου. Επίσης, είναι γνωστός και ως *σημαία*.

λειτουργικό σύστημα Λογισμικό το οποίο ελέγχει την εκτέλεση προγραμμάτων και παρέχει υπηρεσίες όπως κατανομή πόρων, χρονοδρομολόγηση, έλεγχος εισόδου/εξόδου, και διαχείρισης δεδομένων

λέξη Ένα διατεταγμένο σύνολο από bytes ή bits, το οποίο αποτελεί την τυπική μονάδα αποθήκευσης, αποστολής, ή επεξεργασίας των πληροφοριών ενός υπολογιστή. Τυπικά, αν ένας επεξεργαστής διαθέτει σταθερού μήκους σύνολο εντολών, τότε το μήκος της εντολής είναι ίσο με το μήκος της λέξης.

λέξη κατάστασης προγράμματος (PSW) Μία περιοχή μνήμης, η οποία χρησιμοποιείται γιαν να δείξει τη σειρά με την οποία εκτελούνται οι εντολές και για να αποθηκεύει και να δείχνει την κατάσταση του συστήματος. Συνώνυμη με την έννοια της *λέξη κατάστασης επεξεργαστή*.

λωρίδες δίσκων Μία μορφή απεικόνισης παράταξης δίσκων, όπου λογικά συνεχή μπλοκ δεδομένων ή λωρίδες απεικονίζονται κυκλικά σε διαδοχικά μέλη της παράταξης. Ένα σύνολο από διαδοχικές λωρίδες το οποίο απεικονίζει ακριβώς μία λωρίδα σε κάθε μέλος της παράταξης, ονομάζεται ράβδωση.

μαγνητικός δίσκος Ένας επίπεδος κυκλικός δίσκος με ένα στρώμα μαγνητικής επιφάνειας, όπου είναι δυνατή η αποθήκευση δεδομένων σε μία ή και στις δύο πλευρές.

μεγάλο υπολογιστικό σύστημα Μετά από την εμφάνιση των μικρότερων σε όγκο μίνι-υπολογιστών στην αρχή της δεκαετίας του '70, οι παραδοσιακές μεγάλες μηχανές περιγράφονταν με τον όρο μεγάλα υπολογιστικά συστήματα. Τυπικά χαρακτηριστικά του μεγάλου υπολογιστικού συστήματος είναι ότι υποστηρίζει μεγάλες βάσεις δεδομένων, έχει λεπτομερές υλικό Ε/Ε, και χρησιμοποιείται σε λειτουργίες κεντρικής επεξεργασίας δεδομένων.

μετρητής προγράμματος Καταχωρητής διεύθυνσης εντολής.

μη πτητική μνήμη Μνήμη της οποίας τα περιεχόμενα είναι σταθερά και δεν απαιτούν συνεχή παροχή ρεύματος.

μικροεντολή Μία εντολή η οποία ελέγχει τη ροή των δεδομένων και την ακολουθία τους, σε πιο βασικό επίπεδο σε σχέση με τις εντολές μηχανής. Οι ξεχωριστές εντολές μηχανής και πιθανών άλλες λειτουργίες είναι πιθανό να υλοποιούνται από μικροπρογράμματα.

μικροεπεξεργαστής Ένας επεξεργαστής του οποίου τα στοιχεία έχουν συσκευαστεί σε ένα ή λίγα μικρού μεγέθους ολοκληρωμένα κυκλώματα.

μικρολειτουργία Μία βασική λειτουργία του επεξεργαστή, η οποία εκτελείται κατά τη διάρκεια ενός χρονικού παλμού.

μικροπρόγραμμα Μία ακολουθία μικροεντολών οι οποίες βρίσκονται σε μία ειδική μονάδα μνήμης από όπου είναι δυνατή η προσπέλασή τους, προκειμένου να εκτελεστούν διάφορες λειτουργίες.

μικροπρογραμματιζόμενος επεξεργαστής Ένας επεξεργαστής του οποίου η αριθμητική και λογική μονάδα έχει υλοποιηθεί με μικροπρογραμματισμό.

μικροϋπολογιστής *Ένα σύστημα υπολογιστή, του οποίου η μονάδα επεξεργασίας* είναι ένας μικροεπεξεργαστής. Ένας βασικός μικροϋπολογιστής συμπεριλαμβάνει έναν μικροεπεξεργαστή, μνήμη, και δυνατότητα Ε/Ε, η οποία πιθανόν να περιλαβάνεται ή να μην περιλαμβάνεται εντός ενός chip.

μνήμη άμεσης προσπέλασης (DMA) Μία μορφή Ε/Ε, όπου μία ειδική μονάδα με την ονομασία *μονάδα DMA*, ελέγχει την ανταλλαγή δεδομένων ανάμεσα στην κύρια μνήμη και μία μονάδα Ε/Ε. Ο επεξεργαστής στέλνει μία αίτηση για τη μεταφορά ενός μπλοκ δεδομένων προς τη μονάδα DMA και διακόπτεται αφότου έχει ολοκληρωθεί η μεταφορά ολόκληρου του μπλοκ.

μνήμη ανάγνωσης μόνο (ROM) Μνήμη ημιαγωγών της οποίας τα περιεχόμενα δεν είναι δυνατόν να αλλάξουν, εκτός αν καταστραφεί η μονάδα αποθήκευσης. Μη διαγραφόμενη μνήμη.

μνήμη ελέγχου Ένα τμήμα της μνήμης, το οποίο περιέχει μικροκώδικα.

μνήμη συσχέτισης Μία μνήμη της οποίας οι θέσεις καθορίζονται από τα περιεχόμενά τους, ή από ένα τμήμα των περιεχομένων τους και όχι από τα ονόματα ή τις λογικές θέσεις τους.

μνήμη τυχαίας προσπέλασης (RAM) Μνήμη στην οποία κάθε διευθυνσιοδοτούμενη θέση έχει μοναδικό μηχανισμό διευθυνσιοδότησης. Ο χρόνος προσπέλασης κάθε θέσης είναι ανεξάρτητος από την ακολουθία προηγούμενων προσπελάσεων.

μονάδα εισόδου-εξόδου Ένα από τα βασικά μέρη ενός υπολογιστή. Είναι υπεύθυνη για τον έλεγχο μίας ή περισσότερων εξωτερικών συσκευών (περιφερειακά) και για την ανταλλαγή δεδομένων ανάμεσα σε αυτές τις συσκευές και τη μνήμη και/ή τους καταχωρητές.

μονάδα ελέγχου Το τμήμα του επεξεργαστή το οποίο ελέγχει τις λειτουργίες του, συμπεριλαμβανομένων των πράξεων της αριθμητικής και λογικής μονάδας, της διακίνησης δεδομένων μέσα στον επεξεργαστή, και της ανταλλαγής δεδομένων και σημάτων ελέγχου μεταξύ εξωτερικών διεπαφών (π.χ. ο δίαυλος συστήματος).

μοναδιαίος τελεστής Ένας τελεστής ο οποίος αναπαριστά μία πράξη πάνω σε έναν και μόνον έναν παράγοντα.

μορφή εντολής Η μορφή μίας εντολής ως ακολουθία από bits. Η μορφή διαιρεί την εντολή σε πεδία, τα οποία αντιστοιχούν στις συνιστώσες της εντολής (π.χ. κωδικός εντολής, παράγοντες).

μπλοκ ελέγχου διεργασιών Η ενεργοποίηση μίας διεργασίας σε ένα λειτουργικό σύστημα. Πρόκειται για τη δομή δεδομένων η οποία περιέχει πληροφορίες σχετικά με τα χαρακτηριστικά και την κατάσταση μίας διεργασίας.

ολοκληρωμένο κύκλωμα (IC) Ένα μικρό κομμάτι από συμπαγές υλικό, όπως πυρίτιο, πάνω στο οποίο τυπώνεται μία ομάδα από ηλεκτρονικά στοιχεία, αλλά και η διασύνδεσή τους.

οργάνωση υπολογιστή Αναφέρεται στις λειτουργικές μονάδες και τις διασυνδέσεις τους, με τις οποίες υλοποιούνται οι προδιαγραφές της αρχιτεκτονικής. Τα χαρακτηριστικά της οργάνωσης συμπεριλαμβάνουν και τις λεπτομέρειες του υλικού, οι οποίες δεν είναι ορατές στον προγραμματιστή, όπως είναι τα σήματα ελέγχου, οι διεπαφές ανάμεσα στον υπολογιστή και τις περιφερειακές μονάδες, αλλά και η τεχνολογία μνήμης η οποία χρησιμοποιείται.

ορθογωνικότητα Μία αρχή με την οποία δύο μεταβλητές ή διαστάσεις είναι ανεξάρτητες μεταξύ τους. Στο περιβάλλον ενός συνόλου εντολών, ο όρος χρησιμοποιείται γενικά για να δείξει ότι τα άλλα στοιχεία μίας εντολής (τρόπος διευθυνσιοδότησης, πλήθος παραγόντων, μήκος παράγοντα), είναι ανεξάρτητα (δεν καθορίζονται) από τον κωδικό εντολής.

παράγοντας Μία οντότητα στην οποία εκτελείται μία λειτουργία.

πεδίο υπολογιστών υψηλής απόδοσης Μία περιοχή έρευνας η οποία επικεντρώνεται στους υπερ-υπολογιστές και στο λογισμικό το οποίο εκτελείται σε αυτούς. Έμφαση δίνεται σε επιστημονικές εφαρμογές, οι οποίες πιθανόν να συμπεριλαμβάνουν ευρεία χρήση υπολογισμών με διανύσματα και πίνακες, αλλά και παράλληλους αλγορίθμους.

πίνακας αληθείας Ένας πίνακας ο οποίος περιγράφει μία λογική συνάρτηση, παραθέτοντας όλους τους πιθανούς συνδυασμούς των τιμών εισόδου και δείχνοντας, για κάθε συνδυασμό, την τιμή της εξόδου.

πίνακας προγραμματιζόμενης λογικής (PLA) Μία παράταξη πυλών των οποίων οι διασυνδέσεις είναι δυνατόν να προγραμματιστούν, ώστε να εκτελείται μία συγκεκριμένη λογική πράξη.

πλαίσιο σελίδας Μία περιοχή της κύριας μνήμης, η οποία χρησιμοποιείται για να αποθηκευτεί μία σελίδα.

πλεονάζουσα παράταξη ανεξάρτητων δίσκων (RAID) Μία παράταξη δίσκων, όπου ένα τμήμα της φυσικής χωρητικότητας χρησιμοποιείται για να αποθηκεύονται περιττές πληροφορίες σχετικά με τα δεδομένα τα οποία αποθηκεύονται στον υπόλοιπο χώρο. Οι πλεονάζουσες πληροφορίες επιτρέπουν την αναπαραγωγή των δεδομένων του χρήστη σε περίπτωση που ένας από τους δίσκους της παράταξης αντιμετωπίσει βλάβη, ή αποτύχει η προσπέλαση σε αυτόν.

πολύ μεγάλου μήκους λέξη εντολής (VLIW) Αναφέρεται στη χρήση εντολών οι οποίες περιέχουν πολλαπλές λειτουργίες. Στην ουσία, σε μία λέξη συμπεριλαμβάνονται πολλές εντολές. Τυπικά, μία VLIW κατασκευάζεται από ένα μεταγλωττιστή, ο οποίος τοποθετεί στη λέξη, λειτουργίες οι οποίες είναι δυνατόν να εκτελεστούν παράλληλα.

πολυεπεξεργαστής Ένας υπολογιστής με δύο ή περισσότερους επεξεργαστές, οι οποίοι έχουν κοινή πρόσβαση σε μία κύρια μνήμη.

πολυεπεξεργαστής μη ομοιόμορφης προσπέλασης μνήμης (NUMA) Ένας πολυεπεξεργαστής με κοινόχρηστη μνήμη, όπου ο χρόνος προσπέλασης από έναν επεξεργαστή προς μία λέξη μνήμης ποικίλει ανάλογα με τη θέση της λέξης μνήμης.

πολυπλέκτης Ένα συνδυαστικό κύκλωμα το οποίο συνδέει πολλαπλές εισόδους σε μία μόνον έξοδο. Κάθε χρονική στιγμή, μία μόνον από τις εισόδους επιλέγεται για να εξαχθεί στην έξοδο.

πολυπρογραμματισμός Ένας τρόπος λειτουργίας, ο οποίος παρέχει τη δυνατότητα εκτέλεσης δύο ή περισσότερων προγραμμάτων από έναν επεξεργαστή.

πρόβλεψη διακλάδωσης Ένας μηχανισμός ο οποίος χρησιμοποιείται από τον επεξεργαστή για να προβλεφτεί το αποτέλεσμα μίας διακλάδωσης, πριν από την εκτέλεσή της.

προβλεφθείσα εκτέλεση Ένας μηχανισμός ο οποίος υποστηρίζει την υπό συνθήκη εκτέλεση ξεχωριστών εντολών. Με τον τρόπο αυτό, καθίσταται δυνατόν να εκτελεστούν εικοτολογικά και οι δύο διακλαδώσεις μίας εντολής διακλάδωσης, και να διατηρηθούν τα αποτελέσματα εκείνης η οποία τελικά ακολουθείται.

προγραμματιζόμενη είσοδος-έξοδος Μία μορφή Ε/Ε, στην οποία η κεντρική μονάδα επεξεργασίας εκδίδει μία εντολή Ε/Ε προς μία μονάδα Ε/Ε και στη συνέχεια αναμένει την ολοκλήρωση της λειτουργίας πριν προχωρήσει.

προγραμματιζόμενη μνήμη ανάγνωσης μόνο (PROM) Μνήμη ημιαγωγών, της οποίας τα περιεχόμενα είναι δυνατόν να οριστούν μία μόνον φορά. Η διαδικασία εγγραφής γίνεται ηλεκτρικά και υπάρχει δυνατότητα να υλοποιηθεί από το χρήστη σε μία χρονική στιγμή μετά από την κατασκευή του chip.

πρωτόκολλο συνοχής της κρυφής μνήμης Ένας μηχανισμός διατήρισης της εγκυρότητας των δεδομένων που αποθηκεύονται ανάμεσα σε πολλαπλές κρυφές μνήμες, ώστε κάθε προσπέλαση να λαμβάνει την πιο πρόσφατη έκδοση των περιεχομένων μίας λέξης της κύριας μνήμης.

πτητική μνήμη Μία μνήμη στην οποία απαιτείται σταθερή πηγή ηλεκτρικής ισχύος, ώστε να διατηρούνται τα περιεχόμενα της. Αν αποκοπεί η ισχύς, οι αποθηκευμένες πληροφορίες χάνονται.

πύλη Ένα ηλεκτρονικό κύκλωμα το οποίο παράγει ένα σήμα εξόδου. Το σήμα αυτό είναι μία απλή Boolean πράξη πάνω στα σήματα εισόδου της πύλης.

πυρήνας Το τμήμα ενός λειτουργικού συστήματος, το οποίο περιέχει τις πιο βασικές και συχνά χρησιμοποιούμενες λειτουργίες. Συχνά, ο πυρήνας παραμένει εντός της κύριας μνήμης.

σελίδα Σε ένα ιδεατό σύστημα αποθήκευσης, ένα μπλοκ σταθερού μήκους το οποίο έχει μία ιδεατή διεύθυνση και μεταφέρεται ως μονάδα ανάμεσα στην πραγματική και τη βοηθητική μνήμη.

σελιδοποίηση κατ' απαίτηση Η μεταφορά μίας σελίδας από μία βοηθητική μονάδα αποθήκευσης στην πραγ-

ματική, όταν αυτό χρειαστεί.

σταθερό τμήμα λογισμικού Μικροκώδικας αποθηκευμένος στη μνήμη ανάγνωσης μόνο.

στατική RAM Μία μνήμη RAM, της οποίας τα κελιά υλοποιούνται με flip-flops. Η μνήμη αυτή διατηρεί τα περιεχόμενά της όσο παρέχεται ρεύμα σε αυτή και δεν απαιτείται περιοδική ανανέωση.

στοίβα Μία διατεταγμένη λίστα της οποίας τα αντικείμενα εισάγονται και διαγράφονται από το ίδιο άκρο, γνωστό και ως κορυφή. Επομένως, το επόμενο αντικείμενο που εισάγεται στη λίστα τοποθετείται στην κορυφή, και το επόμενο προς διαγραφή στοιχείο, είναι εκείνο το οποίο έχει παραμείνει στη λίστα για το μικρότερο χρονικό διάστημα. Αυτή η μέθοδος ακολουθεί την πειθαρχία lifo, last-in-first-out

σύγχρονος χρονισμός Μία τεχνική στην οποία η εμφάνιση των γεγονότων στο δίαυλο προσδιορίζεται από ένα ρολόι. Το ρολόι ορίζει χρονικές σχισμές ίσου πλάτους, και τα γεγονότα ξεκινούν μόνον στην αρχή κάθε σχισμής.

συμμετρική πολυεπεξεργασία (SMP) Μία μορφή πολυεπεξεργασίας η οποία επιτρέπει στο λειτουργικό σύστημα να εκτελείται σε κάθε διαθέσιμο επεξεργαστή ή σε διάφορους διαθέσιμους επεξεργαστές ταυτόχρονα.

συνδυαστικό κύκλωμα Μία λογική μονάδα, της οποίας οι έξοδοι, σε κάθε χρονική στιγμή, εξαρτώνται μόνον από τις τιμές των εισόδων. Ένα συνδυαστικό κύκλωμα είναι μία ειδική περίπτωση ενός ακολουθιακού κυκλώματος, το οποίο δεν έχει δυνατότητα αποθήκευσης.

σύνολο εντολών υπολογιστή Ένα πλήρες σύνολο των τελεστών των εντολών ενός υπολογιστή, μαζί με περιγραφή της σημασίας που αποδίδεται στους παράγοντες αυτών των τελεστών. Συνώνυμο με το *σύνολο εντολών μηχανής*.

συσσωρευτής Το όνομα του καταχωρητή του επεξεργαστή σε μορφή εντολής μίας διεύθυνσης. Ο συσσωρευτής ή AC, είναι εμμέσως ένας από τους δύο παράγοντες της εντολής.

συστάδα Μία ομάδα διασυνδεδεμένων αυτόνομων υπολογιστών οι οποίοι εργάζονται ως μονάδα και δημιουργούν την ψευδαίσθηση ότι αποτελούν μία μηχανή. Ο όρος *αυτόνομος υπολογιστής*, δηλώνει έναν υπολογιστή ο οποίος μπορεί να λειτουργήσει από μόνος του, ανεξαρτήτως της συστάδας.

σύστημα αναπαράστασης κινητής υποδιαστολής Ένα αριθμητικό σύστημα στο οποίο ένας πραγματικός αριθμός αναπαρίσταται από ένα ζεύγος διαφορετικών ποσοτήτων, όπου ο πραγματικός αριθμός είναι γινόμενο του σταθερού τμήματος, το οποίο αποτελεί τη μία ποσότητα, και μίας τιμής η οποία λαμβάνεται αν υψώσουμε τη βάση της κινητής υποδιαστολής σε μία δύναμη η οποία δηλώνεται από τον εκθέτη της αναπαράστασης, ο οποίος είναι η δεύτερη ποσότητα.

σύστημα αναπαράστασης σταθερής υποδιαστολής Ένα σύστημα αρίθμησης, στο οποίο το σημείο υποδιαστολής υποννοείται ως σταθερό μέσα στην ακολουθία ψηφίων, με τη βοήθεια ενός συμβιβασμού μέσω του οποίου έχει γίνει η συμφωνία σχετικά με τη θέση του σημείου.

σφάλμα σελίδας Εμφανίζεται όταν η σελίδα, η οποία περιέχει μία αναφερόμενη λέξη, δεν βρίσκεται στην κύρια μνήμη. Αυτό το γεγονός προκαλεί διακοπή και απαιτείται από το λειτουργικό σύστημα να προσκομίσει τη σελίδα.

τοπική μεταβλητή Μία μεταβλητή η οποία ορίζεται και χρησιμοποιείται μόνον μέσα σε ένα τμήμα του προγράμματος.

τοπικότητα αναφοράς Η τάση ενός επεξεργαστή να προσπελαύνει επαναληπτικά το ίδιο σύνολο θέσεων μνήμης μέσα σε μία μικρή χρονική περίοδο.

υπερβαθμωτός επεξεργαστής Μία σχεδίαση επεξεργαστών η οποία περιλαμβάνει διασωληνώσεις πολλαπλών εντολών, έτσι ώστε να είναι δυνατή η εκτέλεση περισσότερων από μίας εντολών, ταυτόχρονα, στο ίδιο στάδιο της διασωλήνωσης.

χρόνος κύκλου επεξεργαστή Ο χρόνος που απαιτείται για την εκτέλεση της συντομότερης καλά ορισμένης μικρολειτουργίας του επεξεργαστή. Πρόκειται για τη βασική χρονική μονάδα, με την οποία μετρώνται όλες οι λειτουργίες του επεξεργαστή. Είναι συνώνυμη με την έννοια του *χρόνου κύκλου μηχανής*.

χρόνος κύκλου μνήμης Το αντίστροφο του ρυθμού με τον οποίο προσπελαύνεται η μνήμη. Πρόκειται για τον ελάχιστο χρόνο ανάμεσα στην απόκριση μίας αίτησης προσπέλασης (ανάγνωσης ή εγγραφής) και στην απόκριση της επόμενης αίτησης προσπέλασης.

χώρος διευθύνσεων Το εύρος των διευθύνσεων (μνήμη, Ε/Ε) στις οποίες είναι δυνατόν να γίνει αναφορά.

ΑΝΑΦΟΡΕΣ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ACM Association for Computing Machinery

IBM International Business Machines Corporation

IEEE Institute of Electrical and Electronics Engineers

ABBO04 Abbot,D. *PCI Bus Demystified.* New York: Elsevier, 2004.

ACOS86 Acosta, R.; Kjelstrup, J.; and Torng,H. "An Instruction Issuing Approach to Enhancing Performance in Multiple Functional Unit Processors." *IEEE Transactions on Computers*, September 1986.

ADAM91 Adamek, J. *Foundations of Coding.* New York:Wiley, 1991.

AGAR89 Agarwal, A. *Analysis of Cache Performance for Operating Systems and Multiprogramming* Boston: Kluwer Academic Publishers, 1989.

AGER87 Agerwala, T., and Cocke, J. *High Performance Reduced Instruction Set Processors.* Technical Report RC12434 (55845).Yorktown, NY: IBM Thomas J.Watson Research Center, January 1987.

AMDA67 Amdahl, G. "Validity of the Single-Processor Approach to Achieving Large-Scale Computing Capability." *Proceedings, of the AFIPS Conference*, 1967.

ANDE67a Anderson, D.; Sparacio, F.; and Tomasulo, F. "The IBM System/360 Model 91: Machine Philosophy and Instruction Handling." *IBM Journal of Research and Development*, January 1967.

ANDE67b Anderson, S., et al."The IBM System/360 Model 91: Floating-Point Execution Unit." *IBM Journal of Research and Development*, January 1967. Reprinted in [SWAR90,Volume 1].

ANDE03 Anderson,D."You Don"t Know Jack About Disks." *ACM Queue*, June 2003.

ANDE98 Anderson,D. *FireWire System Architecture.* Reading, MA:Addison-Wesley, 1998.

ANTH08 Anthes,G. "What"s Next for the x86?" *ComputerWorld*, June 16, 2008.

ARM08a ARM Limited. *Cortex-A8 Technical Reference Manual.* ARM DDI 0344E, 2008. www. arm.com

ARM08b ARM Limited. *ARM11 MPCore Processor Technical Reference Manual.* ARM DDI 0360E, 2008. www.arm.com

ASH90 Ash, R. *Information Theory.* New York: Dover, 1990.

ATKI96 Atkins, M."PC Software Performance Tuning." *IEEE Computer*, August 1996.

AZIM92 Azimi, M.; Prasad, B.; and Bhat, K. "Two Level Cache Architectures." *Proceedings COMPCON "92*, February 1992.

BACO94 Bacon, F.; Graham, S.; and Sharp,O." Compiler Transformations for High-Performance Computing." *ACM Computing Surveys*, December 1994.

BAIL93 Bailey, D. "RISC Microprocessors and Scientific Computing." *Proceedings, Supercomputing "93*, 1993.

BART03 Bartlett, J. *Programming from the Ground Up.* 2003.Available at this book"s Web site.

BASH81 Bashe,C.; Bucholtz,W.; Hawkins,G.; Ingram, J.; and Rochester,N."The Architecture of IBM"s Early Computers." *IBM Journal of Research and Development*, September 1981.

BASH91 Bashteen, A.; Lui, I.; and Mullan, J. "A Superpipeline Approach to the MIPS Architecture." *Proceedings, COMPCON Spring "91*, February 1991.

BECK97 Beck, L. *System Software.* Reading, MA:Addison-Wesley, 1997.

BELL70 Bell, C.; Cady, R.; McFarland, H.; Delagi, B.; O"Loughlin, J.; and Noonan, R. "A New Architecture for Minicomputers - The DEC PDP-11." *Proceedings, Spring Joint Computer Conference*, 1970.

BELL71 Bell, C., and Newell, A. *Computer Structures: Readings and Examples.* New York: McGraw-Hill, 1971.

BELL74 Bell, J.; Casasent, D.; and Bell, C. "An Investigation into Alternative Cache Organizations." *IEEE Transactions on Computers*, April 1974.
http://research.microsoft.com/ users/GBell/gbvita.htm

BELL78a Bell,C.; Mudge, J.; and McNamara, J. *Computer Engineering:A DEC View of Hardware Systems Design*. Bedford, MA: Digital Press, 1978.

BELL78b Bell, C.; Newell, A.; and Siewiorek,D. "Structural Levels of the PDP-8." In [BELL78a].

BELL78c Bell, C.; Kotok, A.; Hastings, T.; and Hill, R. "The Evolution of the DEC System-10." *Communications of the ACM*, January 1978.

BENH92 Benham, J. "A Geometric Approach to Presenting Computer Representations of Integers." *SIGCSE Bulletin*, December 1992.

BETK97 Betker, M.; Fernando, J.; and Whalen, S."The History of the Microprocessor." *Bell Labs Technical Journal*, Autumn 1997.

BEZ03 Bez, R.; et al. Introduction to Flash Memory. *Proceedings of the IEEE*, April 2003.

BHAR00 Bharandwaj, J., et al. "The Intel IA-64 Compiler Code Generator." *IEEE Micro*, September/October 2000.

BLAA97 Blaauw, G., and Brooks, F. *Computer Architecture: Concepts and Evolution*. Reading, MA:Addison-Wesley, 1997.

BLAH83 Blahut, R. *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1983.

BOHR98 Bohr, M. "Silicon Trends and Limits for Advanced Microprocessors." *Communications of the ACM*, March 1998.

BOHR03 Bohr, M." High Performance Logic Technology and Reliability Challenges." *International Reliability Physics Symposium*, March 2003.
http://www.irps.org/03-41st

BORK03 Borkar, S. "Getting Gigascale Chips: Challenges and Opportunities in Continuing Moore"s Law." *ACM Queue*, October 2003.

BORK07 Borkar, S."Thousand Core Chips-A Technology Perspective." Proceedings, *ACM/IEEE Design Automation Conference*, 2007.

BRAD91a Bradlee,D.; Eggers, S.; and Henry,R."The Effect on RISC Performance of Register Set Size and Structure Versus Code Generation Strategy." *Proceedings, 18th Annual International Symposium on Computer Architecture*, May 1991.

BRAD91b Bradlee, D.; Eggers, S.; and Henry, R. "Integrating Register Allocation and Instruction Scheduling for RISCs." *Proceedings, Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, April 1991.

BREW97 Brewer, E. "Clustering: Multiply and Conquer." *Data Communications*, July 1997.

BREY09 Brey, B. *The Intel Microprocessors: 8086/8066, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4 and Core2 with 64-bit Extensions*. Upper Saddle River, NJ: Prentice Hall, 2009.

BROW96 Brown, S., and Rose, S. "Architecture of FPGAs and CPLDs:A Tutorial." *IEEE Design and Test of Computers*, Vol. 13, No. 2, 1996.

BURG97 Burger,D., and Austin,T."The SimpleScalar Tool Set,Version 2.0." *Computer Architecture News*, June 1997.

BURK46 Burks, A.; Goldstine, H.; and von Neumann, J. *Preliminary Discussion of the Logical Design of an Electronic Computer Instrument*. Report prepared for U.S. Army Ordnance Dept., 1946, reprinted in [BELL71].

BUYY99a Buyya, R. *High Performance Cluster Computing: Architectures and Systems*. Upper Saddle River, NJ: Prentice Hall, 1999.

BUYY99b Buyya, R. *High Performance Cluster Computing: Programming and Applications*. Upper Saddle River, NJ: Prentice Hall, 1999.

CANT01 Cantin, J., and Hill, H. "Cache Performance for Selected SPEC CPU2000 Benchmarks." *Computer Architecture News*, September 2001.

CART96 Carter, J. *Carter, J. Microprocesser Architecture and Microprogramming*. Upper Saddle River, NJ: Prentice Hall, 1996.

CART06 Carter, P. *PC Assembly Language*. July 23, 2006.Available at this book"s Web site.

CATA94 Catanzaro, B. *Multiprocessor System Architectures*. Mountain View, CA: Sunsoft Press, 1994.

CEKL97 Cekleov, M., and Dubois, M. "Virtual-Address Caches, Part 1: Problems and Solutions in Uniprocessors." *IEEE Micro*, September/October 1997.

CHAI82 Chaitin, G. "Register Allocation and Spilling via Graph Coloring." *Proceedings, SIGPLAN Symposium on Compiler Construction*, June 1982.

CHAS00 Chasin,A."Predication, Speculation, and Modern CPUs." *Dr.Dobb"s Journal*, May 2000.

CHEN94 Chen, P.; Lee, E.; Gibson, G.; Katz, R.; and Patterson, D. "RAID: High-Performance, Reliable Secondary Storage. " *ACM Computing Surveys*, June 1994.

CHEN96 Chen, S., and Towsley, D. "A Performance Evaluation of RAID Architectures." *IEEE Transactions on Computers*, October 1996.

CHOW86 Chow, F.; Himmelstein, M.; Killian, E.; and Weber, L. "Engineering a RISC Compiler System." *Proceedings, COMPCON Spring "86*, March 1986.

CHOW87 Chow, F.; Correll, S.; Himmelstein, M.; Killian, E.; and Weber, L. "How Many Addressing Modes Are Enough?" *Proceedings, Second International Conference on Architectural Support for Programming Languages and Operating Systems*, October 1987.

CHOW90 Chow, F., and Hennessy, J. "The Priority-Based Coloring Approach to Register Allocation." *ACM Transactions on Programming Languages*, October 1990.

CLAR85 Clark,D., and Emer, J."Performance of the VAX-11/780 Translation Buffer: Simulation and Measurement." *ACM Transactions on Computer Systems*, February 1985.

CLAR98 Clarke, D., and Merusi, D. *System Software Programming: The Way Things Work*. Upper Saddle River, NJ: Prentice Hall, 1998.

CLEM00 Clements, A. "The Undergraduate Curriculum in Computer Architecture." *IEEE Micro*, May/June 2000.

COHE81 Cohen,D. "On Holy Wars and a Plea for Peace." *Computer*, October 1981.

COLW85a Colwell, R. Hitchcock, C. Jensen, E. Brinkley-Sprunt, H. and Kollar, C. "Computers, Complexity, and Controversy." *Computer*, September 1985.

COLW85b Colwell, R.; Hitchcock, C.; Jensen, E.; and Sprunt, H. "More Controversy About "Computers, Complexity, and Controversy."" *Computer*, December 1985.

COME00 Comerford, R. "Magnetic Storage: The Medium that Wouldn"t Die." *IEEE Spectrum*, December 2000.

COOK82 Cook, R., and Dande,N."An Experiment to Improve Operand Addressing." *Proceedings, Symposium on Architecture Support for Programming Languages and Operating Systems*, March 1982.

COON81 Coonen J. "Underflow and Denormalized Numbers." *IEEE Computer*, March 1981.

COUT86 Coutant, D.; Hammond, C.; and Kelley, J. "Compilers for the New Generation of Hewlett-Packard Computers." *Proceedings, COMPCON Spring "86*, March 1986.

CRAG79 Cragon, H. "An Evaluation of Code Space Requirements and Performance of Various Architectures." *Computer Architecture News*, February 1979.

CRAG92 Cragon, H. *Branch Strategy Taxonomy and Performance Models*. Los Alamitos, CA: IEEE Computer Society Press, 1992.

CRAW90 Crawford, J."The i486 CPU: Executing Instructions in One Clock Cycle." *IEEE Micro*, February 1990.

CRIS97 Crisp, R. "Direct RAMBUS Technology: The New Main Memory Standard." *IEEE Micro*, November/December 1997.

CUPP01 Cuppu, V., et al. "High Performance DRAMS in Workstation Environments." *IEEE Transactions on Computers*, November 2001.

DATT93 Dattatreya, G. "A Systematic Approach to Teaching Binary Arithmetic in a First Course." *IEEE Transactions on Education*, February 1993.

DAVI87 Davidson, J., and Vaughan, R. "The Effect of Instruction Set Complexity on Program Size and Memory Performance." *Proceedings, Second International Conference on Architectural Support for Programming Languages and Operating Systems*, October 1987.

DENN68 Denning, P. "The Working Set Model for Program Behavior." *Communications of the ACM*, May 1968.

DERO87 DeRosa, J., and Levy, H. "An Evaluation of Branch Architectures." *Proceedings, Fourteenth Annual International Symposium on Computer Architecture*, 1987.

DESA05 Desai,D., et al. "BladeCenter System Overview." IBM Journal of Research and Development." November 2005.

DEWA90 Dewar, R., and Smosna, M. *Microprocessors: A Programmer"s View*. New York: McGraw-Hill, 1990.

DEWD84 Dewdney, A. "In the Game Called Core War Hostile Programs Engage in a Battle of Bits." *Scientific American*, May 1984.

DIJK63 Dijkstra, E. "Making an ALGOL Translator for the X1." In *Annual Review of Automatic Programming,Volume 4*. Pergamon, 1963.

DOWD98 Dowd, K., and Severance, C. *High Performance Computing*. Sebastopol, CA: O'Reilly, 1998.

DUBE91 Dubey, P., and Flynn, M."Branch Strategies: Modeling and Optimization." *IEEE Transactions on Computers*, October 1991.

DULO98 Dulong, C. "The IA-64 Architecture at Work." *Computer*, July 1998.

ECKE90 Eckert, R. "Communication Between Computers and Peripheral Devices-An Analogy." *ACM SIGCSE Bulletin*, September 1990.

ELAY85 El-Ayat, K., and Agarwal, R. "The Intel 80386-Architecture and Implementation." *IEEE Micro*, December 1985.

ERCE04 Ercegovac, M., and Lang,T. *Digital Arithmetic*. San Francisco: Morgan Kaufmann, 2004.

EISC07 Eischen, C. "RAID 6 Covers More Bases." *Network World*, April 9, 2007.

EVAN03 Evans, J., and Trimper, G. *Itanium Architecture for Programmers*. Upper Saddle River, NJ: Prentice Hall, 2003.

EVEN00a Even,G., and Paul,W."On the Design of IEEE Compliant Floating-Point Units." *IEEE Transactions on Computers*, May 2000.

EVEN00b Even,G., and Seidel, P."A Comparison of Three Rounding Algorithms for IEEE Floating- Point Multiplication." *IEEE Transactions on Computers*, July 2000.

EVER98 Evers, M., et al."An Analysis of Correlation and Predictability:What Makes Two-Level Branch Predictors Work." *Proceedings, 25th Annual International Symposium on Microarchitecture*, July 1998.

EVER01 Evers, M., and Yeh, T. "Understanding Branches and Designing Branch Predictors for High-Performance Microprocessors." *Proceedings of the IEEE*, November 2001.

FARH04 Farhat, H. *Digital Design and Computer Organization*. Boca Raton, FL: CRC Press, 2004.

FARM92 Farmwald, M., and Mooring, D. "A Fast Path to One Memory." *IEEE Spectrum*, October 1992.

FLEM86 Fleming, P., and Wallace, J. "How Not to Lie with Statistics: The Correct Way to Summarize Benchmark Results." *Communications of the ACM*, March 1986.

FLYN72 Flynn, M. "Some Computer Organizations and Their Effectiveness." *IEEE Transactions on Computers*, September 1972.

FLYN85 Flynn, M.; Johnson, J.; and Wakefield, S."On Instruction Sets and Their Formats." *IEEE Transactions on Computers*, March 1985.

FLYN87 Flynn, M.; Mitchell, C.; and Mulder, J. "And Now a Case for More Complex Instruction Sets." *Computer*, September 1987.

FLYN01 Flynn, M., and Oberman, S. *Advanced Computer Arithmetic Design*. New York:Wiley, 2001.

FOG08a Fog,A. *Optimizing Subroutines in Assembly Language: An Optimization Guide for x86 Platforms*. Copenhagen University College of Engineering, 2008. http://www.agner.org/ optimize/

FOG08b Fog, A. *The Microarchitecture of Intel and AMD CPUs*. Copenhagen University College of Engineering, 2008. http://www.agner.org/optimize/

FRAI83 Frailey, D. "Word Length of a Computer Architecture: Definitions and Applications." *Computer Architecture News*, June 1983.

FRIE96 Friedman, M."RAID Keeps Going and Going and . . ." *IEEE Spectrum*, April 1996.

FURB00 Furber, S. *ARM System-On-Chip Architecture*. Reading, MA:Addison-Wesley, 2000.

FURH87 Furht, B., and Milutinovic,V. "A Survey of Microprocessor Architectures for Memory Management." *Computer*, March 1987.

FUTR01 Futral,W. *InfiniBand Architecture: Development and Deployment*. Hillsboro, OR: Intel Press, 2001.

GENU04 Genu, P. A Cache Primer. Application Note AN2663. Freescale Semiconductor, Inc., 2004. www.freescale.com/files/32bit/doc/app_note/AN2663.pdf

GHAI98 Ghai, S.; Joyner, J.; and John, L. *Investigating the Effectiveness of a Third Level Cache*. Technical Report TR-980501-01, Laboratory for Computer Architecture, University of Texas at Austin. http://lca.ece.utexas.edu/pubs-by-type.html

GIBB04 Gibbs,W. "A Split at the Core." *Scientific American*, November 2004.

GIFF87 Gifford, D., and Spector, A. "Case Study: IBM"s System/360-370 Architecture." *Communications of the ACM*, April 1987.

GOCH06 Gochman, S., et al."Introduction to Intel Core Duo Processor Architecture." *Intel Technology Journal*, May 2006.

GOLD91 Goldberg, D. "What Every Computer Scientist Should Know About Floating-Point Arithmetic." *ACM Computing Surveys*, March 1991.

GOOD83 Goodman, J. "Using Cache Memory to Reduce Processor-Memory Bandwidth." *Proceedings, 10th*

Annual International Symposium on Computer Architecture, 1983. Reprinted in [HILL00].

GOOD05 Goodacre, J., and Sloss, A. "Parallelism and the ARM Instruction Set Architecture." *Computer*, July 2005.

GREG98 Gregg, J. *Ones and Zeros: Understanding Boolean Algebra, Digital Circuits, and the Logic of Sets*. New York:Wiley, 1998.

GRIM05 Grimheden, M., and Torngren, M."What is Embedded Systems and How Should It Be Taught?-Results from a Didactic Analysis." *ACM Transactions on Embedded Computing Systems*, August 2005.

GUST88 Gustafson, J. "Reevaluating Amdahl"s Law." *Communications of the ACM*, May 1988.

HALF97 Halfhill,T. "Beyond Pentium II." *Byte*, December 1997.

HAMM97 Hammond, L. Nayfay,B. and Olukotun, K."A Single-Chip Multiprocessor." *Computer*, September 1997.

HAND98 Handy, J. *The Cache Memory Book*. San Diego: Academic Press, 1993.

HARR06 Harris,W. "Multi-core in the Source Engine." bit-tech.net technical paper, November 2, 2006. bit-tech.net/gaming/2006/11/02/Multi_core_in_the_ Source_Engin/1

HAUE07 Haeusser, B., et al. *IBM System Storage Tape Library Guide for Open Systems*. IBM Redbook SG24-5946-05, October 2007. ibm.com/redbooks

HAYE98 Hayes, J. *Computer Architecture and Organization*. New York: McGraw-Hill, 1998.

HEAT84 Heath, J. "Re-Evaluation of RISC 1." *Heath, J. "Re-Evaluation of RISC 1."* March 1984.

HENN82 Hennessy,J.,et al."Hardware/SoftwareTradeoffs for Increased Performance." *Proceedings, Symposium on Architectural Support for Programming Languages and Operating Systems*, March 1982.

HENN84 Hennessy, J. "VLSI Processor Architecture." *IEEE Transactions on Computers*, December 1984.

HENN91 Hennessy, J., and Jouppi, N. "Computer Technology and Architecture: An Evolving Interaction." *Computer*, September 1991.

HENN06 Henning, J. "SPEC CPU2006 Benchmark Descriptions." *Computer Architecture News*, September 2006.

HENN07 Henning, J."SPEC CPU Suite Growth:An Historical Perspective." *Computer Architecture News*, March 2007.

HIDA90 Hidaka, H.; Matsuda,Y.; Asakura, M.; and Kazuyasu, F. "The Cache DRAM Architecture: A DRAM with an On-Chip Cache Memory." *IEEE Micro*, April 1990.

HIGB90 Higbie, L. "Quick and Easy Cache Performance Analysis." *Computer Architecture News*, June 1990.

HILL64 Hill, R. "Stored Logic Programming and Applications." *Datamation*, February 1964.

HILL89 Hill, M. "Evaluating Associativity in CPU Caches." *IEEE Transactions on Computers*, December 1989.

HILL00 Hill, M.; Jouppi, N.; and Sohi, G. *Readings in Computer Architecture*. San Francisco: Morgan Kaufmann, 2000.

HINT01 Hinton,G., et al."The Microarchitecture of the Pentium 4 Processor." *Intel Technology Journal*, Q1 2001. http://developer.intel.com/technology/itj/

HIRA07 Hirata, K., and Goodacre, J. "ARM MPCore: The Streamlined and Scalable ARM11 processor core." *Proceedings, 2007 Conference on Asia South Pacific Design Automation*, 2007.

HUCK83 Huck,T. *Comparative Analysis of Computer Architectures*. Stanford University Technical Report No. 83-243, May 1983.

HUCK00 Huck, J., et al. "Introducing the IA-64 Architecture." *IEEE Micro*, September/October 2000.

HUGU91 Huguet,M.,and Lang,T."Architectural Support for Reduced Register Saving/Restoring in Single-Window Register Files." *ACMTransactions on Computer Systems*, February 1991.

HUTC96 Hutcheson, G., and Hutcheson, J. "Technology and Economics in the Semiconductor Industry." *Scientific American*, January 1996.

HWAN93 Hwang, K. *Advanced Computer Architecture*. New York: McGraw-Hill, 1993.

HWAN99 Hwang, K. et al. "Designing SSI Clusters with Hierarchical Checkpointing and Single I/O Space." *IEEE Concurrency*, January-March 1999.

HWU98 Hwu,W. "Introduction to Predicated Execution." *Computer*, January 1998.

HWU01 Hwu,W.:August,D.; and Sias, J. "Program Decision Logic Optimization Using Predication and Control Speculation." *Proceedings of the IEEE*, November 2001.

IBM01 International Business Machines, Inc. *64 Mb Synchronous DRAM*. IBM Data Sheet 364164,

January 2001.

INTE98 Intel Corp. *Pentium Pro and Pentium II Processors and Related Products.* Aurora, CO, 1998.

INTE00a Intel Corp *Intel IA-64 Architecture Software Developer"s Manual (4 volumes).* Document 245317 through 245320.Aurora, CO, 2000.

INTE00b Intel Corp *Itanium Processor Microarchitecture Reference for Software Optimization.* Aurora, CO, Document 245473.August 2000.

INTE01a Intel Corp. *Intel Pentium 4 Processor Optimization Reference Manual.* Document 248966-04 2001. http://developer.intel.com/design/Pentium4/ documentation.htm

INTE01b Intel Corp. *Desktop Performance and Optimization for Intel Pentium 4 Processor.* Document 248966-04 2001 http://developer.intel.com/design/ Pentium4/documentation.htm

INTE04a Intel Corp. *IA-32 Intel Architecture Software Developer"s Manual (4 volumes).* Document 253665 through 253668. 2004. http://developer.intel. com/design/Pentium4/ documentation.htm

INTE04b Intel Research and Development. *Architecting the Era of Tera.* Intel White Paper, February 2004. http://www.intel.com/labs/teraera/index.htm

INTE04b Intel Corp. *Endianness White Paper.* November 15, 2004.

INTE08 Intel Corp. Intel ® 64 and *IA-32 Intel Architectures Software Developer"s Manual (3 volumes).* Denver, CO, 2008. intel.com/products/processor/ manuals

JACO08 Jacob, B.; Ng, S.; and Wang, D. *Memory Systems: Cache, DRAM, Disk.* Boston: Morgan Kaufmann, 2008.

JAME90 James,D."Multiplexed Buses:The Endian Wars Continue." *IEEE Micro,* September 1983.

JARP01 Jarp, S. "Optimizing IA-64 Performance." *Dr. Dobb"s Journal,* July 2001.

JERR05 Jerraya, A., and Wolf,W., eds. *Multiprocessor Systems-on-Chips.* San Francisco: Morgan Kaufmann, 2005.

JOHN91 Johnson, M. *Superscalar Microprocessor Design.* Englewood Cliffs, NJ: Prentice Hall, 1991.

JOHN08 John, E., and Rubio, J. *Unique Chips and Systems.* Boca Raton, FL: CRC Press, 2008.

JOUP88 Jouppi, N. "Superscalar versus Superpipelined Machines." *Computer Architecture News,* June 1988.

JOUP89a Jouppi, N., and Wall, D. "Available Instruction-Level Parallelism for Superscalar and Superpipelined Machines." *Proceedings, Third International Conference on Architectural Support for Programming Languages and Operating Systems,* April 1989.

JOUP89b Jouppi,N."The Nonuniform Distribution of Instruction-Level and Machine Parallelism and Its Effect on Performance." *IEEE Transactions on Computers,* December 1989.

KAEL91 Kaeli, D., and Emma, P. "Branch History Table Prediction of Moving Target Branches Due to Subroutine Returns." *Proceedings, 18th Annual International Symposium on Computer Architecture,* May 1991.

KAGA01 Kagan, M. "InfiniBand: Thinking Outside the Box Design." *Communications System Design,* September 2001. www.csdmag.com

KALL04 Kalla, R.; Sinharoy, B.; and Tendler, J. "IBM Power5 Chip: A Dual-Core Multithreaded Processor." *IEEE Micro,* March-April 2004.

KANE92 Kane, G., and Heinrich, J. *MIPS RISC Architecture.* Englewood Cliffs, NJ: Prentice Hall, 1992.

KAPP00 Kapp, C. "Managing Cluster Computers." *Dr. Dobb"s Journal,* July 2000.

KATE83 Katevenis, M. Reduced Instruction Set Computer Architectures for VLSI. PhD dissertation, Computer Science Department, University of California at Berkeley, October 1983. Reprinted by MIT Press, Cambridge, MA, 1985.

KATH01 Kathail, B.; Schlansker, M.; and Rau, B. "Compiling for EPIC Architectures." *Proceedings of the IEEE,* November 2001.

KATZ89 Katz, R.; Gibson, G.; and Patterson, D. "Disk System Architecture for High Performance Computing." *Proceedings of the IEEE,* December 1989.

KEET01 Keeth,B., and Baker,R. *DRAM Circuit Design:A Tutorial.* Piscataway,NJ: IEEE Press, 2001.

KHUR01 Khurshudov, A. *The Essential Guide to Computer Data Storage.* Upper Saddle River, NJ: Prentice Hall, 2001.

KNAG04 Knaggs, P., and Welsh, S. *ARM:Assembly Language Programming.* Bournemouth University,

School of Design, Engineering, and Computing,August 31, 2004. www.freetechbooks .com/arm-assembly-language-programming-t729.html

KNUT71 Knuth,D. "An Empirical Study of FORTRAN Programs." *Software Practice and Experience*, vol. 1, 1971.

KNUT98 Knuth, D. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms.* Reading, MA:Addison-Wesley, 1998.

KOOP96 Koopman, P. "Embedded System Design Issues (the Rest of the Story). *Proceedings, 1996 International Conference on Computer Design*, 1996.

KUCK77 Kuck, D.; Parker, D.; and Sameh, A. "An Analysis of Rounding Methods in Floating- Point Arithmetic." *IEEE Transactions on Computers*. July 1977.

KUGA91 Kuga, M. Murakami, K. and Tomita, S."DSNS (Dynamically-hazard resolved, Statically code-scheduled, Nonuniform Superscalar): Yet Another Superscalar Processor Architecture." *Computer Architecture News*, June 1991.

LEE91 Lee, R.; Kwok, A.; and Briggs, F. "The Floating Point Performance of a Superscalar SPARC Processor." *Proceedings, Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, April 1991.

LEON07 Leonard, T. "Dragged Kicking and Screaming: Source Multicore." *Proceedings, Game Developers Conference 2007*, March 2007.

LEON08 Leong, p. "Recent Trends in FPGA Architectures and Applications." *Proceedings, 4th IEEE International symposium on Electronic Design,Test, and Applications*, 2008.

LEVI00 Levine, J. Linkers and Loaders. San Francisco: Morgan Kaufmann, 2000.

LILJ88 Lilja, D. "Reducing the Branch Penalty in Pipelined Processors." *Computer*, July 1988.

LILJ93 Lilja, D. "Cache Coherence in Large-Scale Shared-Memory Multiprocessors: Issues and Comparisons." *ACM Computing Surveys*, September 1993.

LOVE96 Lovett, T., and Clapp, R. "Implementation and Performance of a CC-NUMA System." *Proceedings, 23rd Annual International Symposium on Computer Architecture*, May 1996.

LUND77 Lunde, A. "Empirical Evaluation of Some Features of Instruction Set Processor Architectures." *Communications of the ACM*, March 1977.

LYNC93 Lynch, M. *Microprogrammed State Machine Design.* Boca Raton, FL: CRC Press, 1993.

MACD84 MacDougall, M. "Instruction-level Program and Process Modeling." *IEEE Computer*, July 1984.

MAHL94 Mahlke, S., et al. "Characterizing the Impact of Predicated Execution on Branch Prediction." *Proceedings, 27th International Symposium on Microarchitecture*, December 1994.

MAHL95 Mahlke, S., et al. "A Comparison of Full and Partial Predicated Execution Support for ILP Processors." *Proceedings, 22nd International Symposium on Computer Architecture*, June 1995.

MAK04 Mak, P., et al. "Processor Subsystem Interconnect for a Large Symmetric Multiprocessing System." *IBM Journal of Research and Development*, May/July 2004.

MANJ01a Manjikian,N. "More Enhancements of the SimpleScalar Tool Set." *Computer Architecture News*, September 2001.

MANJ01b Manjikian,N. "Multiprocessor Enhancements of the SimpleScalar Tool Set." *Computer Architecture News*, March 2001.

MANO04 Mano, M. *Logic and Computer Design Fundamentals.* Upper Saddle River,NJ: Prentice Hall, 2004.

MANS97 Mansuripur, M., and Sincerbox, G. "Principles and Techniques of Optical Data Storage." *Proceedings of the IEEE*, November 1997.

MARC90 Marchant, A. *Optical Recording.* Reading, MA:Addison-Wesley, 1990.

MARK00 Markstein, P. *IA-64 and Elementary Functions.* Upper Saddle River, NJ: Prentice Hall PTR, 2000.

MARR02 Marr, D. et al. "Hyper-Threading Technology Architecture and Microarchitecture." *Intel Technology Journal*, First Quarter, 2002.

MASH95 Mashey, J. "CISC vs. RISC (or what is RISC really)." *USENET comp.arch newsgroup, article 46782*, February 1995.

MAYB84 Mayberry,W., and Efland, G. "Cache Boosts Multiprocessor Performance." *Computer Design*, November 1984.

MAZI03 Mazidi, M., and Mazidi, J. *The 80x86 IBM PC and Compatible Computers: Assembly Language, Design and Interfacing.* Upper Saddle River, NJ: Prentice Hall, 2003.

MCDO05 MCDO05 McDougall, R. "Extreme Software Scaling." *ACM Queue*, September 2005.

MCDO06 McDougall, R., and Laudon, J. "Multi-Core Microprocessors are Here." ; *login*, October 2006.

MCEL85 McEliece,R."The Reliability of Computer Memories." *Scientific American*, January 1985.

MCNA03 McNairy, C., and Soltis, D. "Itanium 2 Processor Microarchitecture." *IEEE Micro*, March-April 2003.

MEE96a Mee, C., and Daniel, E. eds. *Magnetic Recording Technology*. New York: McGraw-Hill, 1996.

MEE96b Mee,C., and Daniel, E. eds. *Magnetic Storage Handbook*. New York: McGraw-Hill, 1996.

MEND06 Mendelson, A., et al. "CMP Implementation in Systems Based on the Intel Core Duo Processor." *Intel Technology Journal*, May 2006.

MILE00 Milenkovic, A. "Achieving High Performance in Bus-Based Shared-Memory Multiprocessors." *IEEE Concurrency*, July-September 2000.

MIRA92 Mirapuri, S.; Woodacre, M.; and Vasseghi, N. "The MIPS R4000 Processor." *IEEE Micro*, April 1992.

MOOR65 Moore, G. "Cramming More Components Onto Integrated Circuits." *Electronics Magazine*, April 19, 1965.

MORS78 Morse, S.; Pohlman,W.; and Ravenel, B. "The Intel 8086 Microprocessor: A 16-bit Evolution of the 8080." *Computer*, June 1978.

MOSH01 Moshovos, A., and Sohi, G. "Microarchitectural Innovations: Boosting Microprocessor Performance Beyond Semiconductor Technology Scaling." *Proceedings of the IEEE*, November 2001.

MYER78 Myers, G. "The Evaluation of Expressions in a Storage-to-Storage Architecture."*Computer Architecture News*, June 1978.

NAFF02 Naffziger, S., et al."The Implementation of the Itanium 2 Microprocessor." *IEEE Journal of Solid-State Circuits*, November 2002.

NOER05 Noergarrd, T. *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*. New York: Elsevier, 2005.

NOVI93 Novitsky, J.; Azimi, M.; and Ghaznavi, R. "Optimizing Systems Performance Based on Pentium Processors." *Proceedings COMPCON "92*, February 1993.

NOWE07 Nowell, M.;Vusirikala,V.; and Hays, R. "Overview of Requirements and Applications for 40 Gigabit and 100 Gigabit Ethernet." *Ethernet Alliance White Paper*, August 2007.

OBER97a Oberman, S., and Flynn, M."Design Issues in Division and Other Floating-Point Operations." *IEEE Transactions on Computers*, February 1997.

OBER97b Oberman, S., and Flynn, M. "Division Algorithms and Implementations." *IEEE Transactions on Computers*, August 1997.

OLUK96 Olukotun, K., et al. "The Case for a Single-Chip Multiprocessor." *Proceedings, Seventh International Conference on Architectural Support for Programming Languages and Operating Systems*, 1996.

OLUK05 Olukotun, K., and Hammond, L. "The Future of Microprocessors." *ACM Queue*, September 2005.

OLUK07 Olukotun, K.; Hammond, L.; and Laudon, J. *Chip Multiprocessor Architecture: Techniques to Improve Throughput and Latency*. San Rafael,CA: Morgan Claypool, 2007.

OMON99 Omondi, A. *The Microarchitecture of Pipelined and Superscalar Computers*. Boston: Kluwer, 1999.

OVER01 Overton, M. *Numerical Computing with IEEE Floating Point Arithmetic*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2001.

PADE81 Padegs, A. "System/360 and Beyond." *IBM Journal of Research and Development*, September 1981.

PADE88 Padegs, A.; Moore, B.; Smith, R.; and Buchholz,W."The IBM System/370 Vector Architecture: Design Considerations." *IEEE Transactions on Communications*, May 1988.

PARH00 Parhami, B. *Computer Arithmetic: Algorithms and Hardware Design*. Oxford: Oxford University Press, 2000.

PARK89 Parker, A., and Hamblen, J. *An Introduction to Microprogramming with Exercises Designed for the Texas Instruments SN74ACT8800 Software Development Board*. Dallas, TX:Texas Instruments, 1989.

PATT82a Patterson,D., and Sequin, C. "A VLSI RISC." *Computer*, September 1982.

PATT82b Patterson, D., and Piepho, R. "Assessing RISCs in High-Level Language Support." *IEEE Micro*, November 1982.

PATT84 Patterson,D."RISC Watch." *Computer Architecture News*, March 1984.

PATT85a Patterson, D. "Reduced Instruction Set Computers." *Communications of the ACM*, January 1985.

PATT85b Patterson, D., and Hennessy, J. "Response to "Computers, Complexity, and Controversy."" *Computer*, November 1985.

PATT88 Patterson, D.; Gibson, G.; and Katz, R. "A Case for Redundant Arrays of Inexpensive Disks

(RAID)." *Proceedings, ACM SIGMOD Conference of Management of Data*, June 1988.

PATT01 Patt, Y. "Requirements, Bottlenecks, and Good Fortune: Agents for Microprocessor Evolution." *Proceedings of the IEEE*, November 2001.

PEIR99 Peir, J.; Hsu,W.; and Smith,A."Functional Implementation Techniques for CPU Cache Memories." *IEEE Transactions on Computers*, February 1999.

PELE97 Peleg, A.;Wilkie, S.; and Weiser,U. "Intel MMX for Multimedia PCs." *Communications of the ACM*, January 1997.

PFIS98 Pfister,G. *In Search of Clusters*. Upper Saddle River, NJ: Prentice Hall, 1998.

POLL99 Pollack, F. "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies (keynote address)." *Proceedings of the 32nd annual ACM/IEEE International Symposium on Microarchitecture*, 1999.

POPE91 Popescu,V., et al."The Metaflow Architecture." *IEEE Micro*, June 1991.

PRES01 Pressel, D. "Fundamental Limitations on the Use of Prefetching and Stream Buffers for Scientific Applications." *Proceedings, ACM Symposium on Applied Computing*, March 2001.

PRIN97 Prince, B. *Semiconductor Memories*. New York:Wiley, 1997.

PRIN02 Prince, B. *Emerging Memories: Technologies and Trends*. Norwell, MA: Kluwer, 2002.

PRZY88 Przybylski, S.; Horowitz, M.; and Hennessy, J. "Performance Trade-offs in Cache Design." *Proceedings, Fifteenth Annual International Symposium on Computer Architecture*, June 1988.

PRZY90 Przybylski, S."The Performance Impact of Block Size and Fetch Strategies." *Proceedings, 17th Annual International Symposium on Computer Architecture*, May 1990.

RADD08 Radding, A. "Small Disks, Big Specs." *Storage Magazine*, September 2008.

RADI83 Radin, G. "The 801 Minicomputer." *IBM Journal of Research and Development*, May 1983.

RAGA83 Ragan-Kelley, R., and Clark, R. "Applying RISC Theory to a Large Computer." *Computer Design*, November 1983.

RAMA77 Ramamoorthy, C. "Pipeline Architecture." *Computing Surveys*, March 1977.

RECH98 Reches, S., and Weiss, S. "Implementation and Analysis of Path History in Dynamic Branch Prediction Schemes." *IEEE Transactions on Computers*, August 1998.

REDD76 Reddi, S., and Feustel, E. "A Conceptual Framework for Computer Architecture." *Computing Surveys*, June 1976.

REIM06 Reimer, J. "Valve Goes Multicore." ars technica, November 5, 2006. arstechnica.com/ articles/ paedia/cpu/valve-multicore.ars

RICH07 Riches, S., et al. "A Fully Automated High Performance Implementation of ARM Cortex- A8." IQ Online, Vol. 6, No. 3, 2007. www.arm.com/iqonline

RODR01 Rodriguez, M.; Perez, J.; and Pulido, J. "An Educational Tool for Testing Caches on Symmetric Multiprocessors." *Microprocessors and Microsystems*, June 2001.

ROSC03 Rosch,W.Winn L. *Rosch Hardware Bible*. Indianapolis, IN: Que Publishing, 2003.

SAKA02 Sakai, S. "CMP on SoC: architect"s view." *Proceedings. 15th International Symposium on System Synthesis*, 2002.

SALO93 Salomon,D *Assemblers and Loaders*. Ellis Horwood Ltd, 1993.Available at this book"s Web site.

SATY81 Satyanarayanan, M., and Bhandarkar, D. "Design Trade-Offs in VAX-11 Translation Buffer Organization." *Computer*, December 1981.

SCHA97 Schaller, R. "Moore"s Law: Past, Present, and Future." *IEEE Spectrum*, June 1997.

SCHL00a Schlansker, M.; and Rau, B. "EPIC: Explicitly Parallel Instruction Computting." *Computer*, February 2000.

SCHL00b Schlansker, M.; and Rau, B. *EPIC: An Architecture for Instruction-Level Parallel Processors*. HPL Technical Report HPL-1999-111, Hewlett-Packard Laboratories (www.hpl.hp.com), February 2000.

SCHW99 Schwarz, E., and Krygowski, C. "The S/390 G5 Floating-Point Unit." *IBM Journal of Research and Development*, September/November 1999.

SEAL00 Seal, D., ed. *ARM Architecture Reference Manual*. Reading, MA: Addison-Wesley, 2000.

SEBE76 Sebern, M. "A Minicomputer-compatible Microcomputer System: The DEC LSI-11." *Proceedings of the IEEE*, June 1976.

SEGA95 Segars, S.; Clarke, K.; and Goudge, L. "Embedded Control Problems, Thumb, and the ARM7TDMI." *IEEE Micro*, October 1995.

SEGE91 Segee, B., and Field, J. *Microprogramming and Computer Architecture*. New York: Wiley, 1991.

SERL86 Serlin,O."MIPS, Dhrystones, and Other Tales." *Datamation*, June 1, 1986.

SHAN38 Shannon, C. "Symbolic Analysis of Relay and Switching Circuits." *AIEE Transactions*, vol. 57, 1938.

SHAN99 Shanley, T., and Anderson, D. *PCI Systems Architecture*. Richardson, TX: Mindshare Press, 1999.

SHAN03 Shanley,T. *InfinBand Network Architecture*. Reading, MA:Addison-Wesley, 2003.

SHAN05 Shanley, T. *Unabridged Pentium 4, The: IA32 Processor Genealogy*. Reading, MA: Addison-Wesley, 2005.

SHAR97 Sharma, A. *Semiconductor Memories: Technology, Testing, and Reliability*. New York: IEEE Press, 1997.

SHAR00 Sharangpani, H., and Arona, K. "Itanium Processor Microarchitecture." *IEEE Micro*, September/October 2000.

SHAR03 Sharma, A. *Advanced Semiconductor Memories: Architectures, Designs, and Applications*. New York: IEEE Press, 2003.

SHEN05 Shen, J., and Lipasti, M. *Modern Processor Design: Fundamentals of Superscalar Processors*. New York: McGraw-Hill, 2005.

SIEG04 Siegel, T.; Pfeffer, E.; and Magee, A. "The IBM z990 Microprocessor." *IBM Journal of Research and Development*, May/July 2004.

SIEW82 Siewiorek, D.; Bell, C.; and Newell, A. *Computer Structures: Principles and Examples*. New York: McGraw-Hill, 1982.

SIMA97 Sima,D. "Superscalar Instruction Issue." *IEEE Micro*, September/October 1997.

SIMA04 Sima, D. "Decisive Aspects in the Evolution of Microprocessors." *Proceedings of the IEEE*, December 2004.

SIMO96 Simon, H. *The Sciences of the Artificial*. Cambridge, MA: MIT Press, 1996.

SLOS04 Sloss, A.; Symes, D.; and Wright, C. *ARM System Developer"s Guide*. San Francisco: Morgan Kaufmann, 2004.

SMIT82 Smith, A. "Cache Memories." *ACM Computing Surveys*, September 1992.

SMIT95 Smith, J., and Sohi, G. "The Microarchitecture of Superscalar Processors." *Proceedings of the IEEE*, December 1995.

SMIT87 Smith,A."Line (Block) Size Choice for CPU Cache Memories." *IEEE Transactions on Communications*, September 1987.

SMIT88 Smith, J. "Characterizing Computer Performance with a Single Number." *Communications of the ACM*, October 1988.

SMIT89 Smith, M.; Johnson, M.; and Horowitz, M. "Limits on Multiple Instruction Issue." *Proceedings,Third International Conference on Architectural Support for Programming Languages and Operating Systems*, April 1989.

SMIT08 Smith,B."ARM and Intel Battle over the Mobile Chip"s Future." Computer, May 2008. SODE96 Soderquist, P., and Leeser, M. "Area and Performance Tradeoffs in Floating-Point Divide and Square-Root Implementations." *ACM Computing Surveys*, September 1996.

SOHI90 Sohi, G. "Instruction Issue Logic for High-Performance Interruptable, Multiple Functional Unit, Pipelined Computers." *IEEE Transactions on Computers*, March 1990.

STAL88 Stallings, W. "Reduced Instruction Set Computer Architecture." *Proceedings of the IEEE*, January 1988.

STAL07 Stallings,W. *Data and Computer Communications, Eighth Edition*. Upper Saddle River, NJ: Prentice Hall, 2007.

STAL09 Stallings,W. *Operating Systems, Internals and Design Principles, Sixth Edition*. Upper Saddle River, NJ: Prentice Hall, 2009.

STEN90 Stenstrom, P. "A Survey of Cache Coherence Schemes of Multiprocessors." *Computer*, June 1990.

STEV64 Stevens,W. "The Structure of System/360, Part II: System Implementation." *IBM Systems Journal*, Vol. 3, No. 2, 1964. Reprinted in [SIEW82].

STON93 Stone, H. *High-Performance Computer Architecture*. Reading, MA: Addison-Wesley, 1993.

STON96 Stonham,T. *Digital Logic Techniques*. London: Chapman & Hall, 1996.

STRE78 Strecker,W. "VAX-11/780: A Virtual Address Extension to the DEC PDP-11 Family." *Proceedings, National Computer Conference*, 1978.

STRE83 Strecker,W."Transient Behavior of Cache Memories." *ACM Transactions on Computer Systems*, November 1983.

STRI79 Stritter, E., and Gunter, T. "A Microprocessor Architecture for a Changing World: The Motorola

68000." *Computer*, February 1979.

SWAR90 Swartzlander, E., editor. *Computer Arithmetic, Volumes I and II*. Los Alamitos, CA: IEEE Computer Society Press, 1990.

TAMI83 Tamir, Y., and Sequin, C. "Strategies for Managing the Register File in RISC." *IEEE Transactions on Computers*, November 1983.

TANE78 Tanenbaum, A. "Implications of Structured Programming for Machine Architecture." *Communications of the ACM*, March 1978.

TANE97 Tanenbaum, A., and Woodhull, A. *Operating Systems: Design and Implementation*. Upper Saddle River, NJ: Prentice Hall, 1997.

THOM00 Thompson, D. "IEEE 1394: Changing the Way We Do Multimedia Communications." *IEEE Multimedia*, April-June 2000.

TI90 Texas Instruments Inc. *SN74ACT880 Family Data Manual*. SCSS006C, 1990.

TJAD70 Tjaden, G., and Flynn, M. "Detection and Parallel Execution of Independent Instructions." *IEEE Transactions on Computers*, October 1970.

TOMA93 Tomasevic, M., and Milutinovic, V. *The Cache Coherence Problem in Shared-Memory Multiprocessors: Hardware Solutions*. Los Alamitos, CA: IEEE Computer Society Press, 1993.

TOON81 Toong, H., and Gupta, A. "An Architectural Comparison of Contemporary 16-Bit Microprocessors." *IEEE Micro*, May 1981.

TRIE01 Triebel,W *Itanium Architecture for Software Developers*. Intel Press, 2001.

TUCK67 Tucker, S. "Microprogram Control for System/360." *IBM Systems Journal*, No. 4, 1967.

TUCK87 Tucker, S. "The IBM 3090 System Design with Emphasis on the Vector Facility." *Proceedings, COMPCON Spring "87*, February 1987.

UNGE02 Ungerer, T.; Rubic, B.; and Silc, J. "Multithreaded Processors." *The Computer Journal*, No. 3, 2002.

UNGE03 Ungerer,T.; Rubic, B.; and Silc, J. "A Survey of Processors with Explicit Multithreading." *ACM Computing Surveys*, March 2003.

VASS03 Vassiliadis, S.;Wong, S.; and Cotofana, S."Microcode Processing: Positioning and Directions." *IEEE Micro*, July-August 2003.

VOEL88 Voelker, J. "The PDP-8." *IEEE Spectrum*, November 1988.

VOGL94 Vogley, B. "800 Megabyte Per Second Systems Via Use of Synchronous DRAM." *Proceedings, COMPCON "94*, March 1994.

VONN45 Von Neumann, J. *First Draft of a Report on the EDVAC*. Moore School, University of Pennsylvania, 1945. Reprinted in *IEEE Annals on the History of Computing*, No. 4, 1993.

VRAN80 Vranesic, Z., and Thurber, K. "Teaching Computer Structures." *Computer*, June 1980.

WALL85 Wallich, P."Toward Simpler, Faster Computers." *IEEE Spectrum*, August 1985.

WALL91 Wall, D. "Limits of Instruction-Level Parallelism." *Proceedings, Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, April 1991.

WANG99 Wang, G., and Tafti, D. "Performance Enhancement on Microprocessors with Hierarchical Memory Systems for Solving Large Sparse Linear Systems." *International Journal of Supercomputing Applications*, vol. 13, 1999.

WEIC90 Weicker, R. "An Overview of Common Benchmarks." *Computer*, December 1990.

WEIN75 Weinberg,G. *An Introduction to General Systems Thinking*. New York:Wiley, 1975.

WEIS84 Weiss, S., and Smith, J. "Instruction Issue Logic in Pipelined Supercomputers." *IEEE Transactions on Computers*, November 1984.

WEYG01 Weygant, P. *Clusters for High Availability*. Upper Saddle River, NJ: Prentice Hall, 2001.

WHIT97 Whitney, S., et al. "The SGI Origin Software Environment and Application Performance." *Proceedings, COMPCON Spring "97*, February 1997.

WICK97 Wickelgren, I."The Facts About FireWire." *IEEE Spectrum*, April 1997.

WILK51 Wilkes, M. "The Best Way to Design an Automatic Calculating Machine." *Proceedings, Manchester University Computer Inaugural Conference*, July 1951.

WILK53 Wilkes, M., and Stringer, J. "Microprogramming and the Design of the Control Circuits in an Electronic Digital Computer." *Proceedings of the Cambridge Philosophical Society*, April 1953. Reprinted in [SIEW82].

WILK65 Wilkes, M. "Slave memories and dynamic storage allocation," *IEEE Transactions on Electronic Computers*, April 1965. Reprinted in [HILL00].

WILL90 Williams, F., and Steven, G. "Address and Data Register Separation on the M68000 Family."

Computer Architecture News, June 1990.

YEH91 Yeh, T., and Patt, N. "Two-Level Adapting Training Branch Prediction." *Proceedings, 24th Annual International Symposium on Microarchitecture*, 1991.

ZHAN01 Zhang, Z.; Zhu, Z.; and Zhang, X. "Cached DRAM for ILP Processor Memory Access Latency Reduction." *IEEE Micro*, July-August 2001.

ΕΥΡΕΤΗΡΙΟ

AGU, 587
Alan Turing, 21
ALU, 16, 21, 326, 465
Arithmetic and Logic Unit, 16
ARM11 MPCore, 751
    Διαχείριση Διακοπών, 752
    Συνοχή της Κρυφής Μνήμης, 755
ARM, 47, 48, 51, 502
    CORTEX-A8, 584
    Επεξεργασία των Διακοπών, 507
    Μορφές Εντολών, 453
    Οργάνωση του Επεξεργαστή, 502
    Οργάνωση των Καταχωρητών, 505
    Τρόποι Λειτουργίας του Επεξεργαστή, 504
    Τύποι Δεδομένων, 386
    Τύποι Εξαιρέσεων, 507
ASCII, 384
Assembly Language, 767
Assembly, 457
Bell, 28
Bits Φρουροί, 361
Blu-ray DVD, 221
Boolean Πράξεις, 32
Booth, 343
Bus, 89
Byte, 119
DVD, 220
DDR SDRAM, 187
DEC, 36
Disk Cache, 125
DMA, 68, 87, 250
DRAM, 168
EEPROM, 172
Embedded Systems, 49
ENIAC, 20
Execute Cycle, 26
EPROM, 172
Hardwired Program, 71
HD DVD, 221
Java Virtual Machine, 57
I/O, 13, 15
IAS, 21, 62, 377
IBM
    80286, 47
    80386, 47
    8046, 48
    8080, 47
    8086, 47
IBM 3033
    Εκτέλεση Μικροεντολών, 654
IBM 3090
    Καταχωρητές, 720
    Οργάνωση, 719
    Σύνθετες Εντολές, 723
    Σύνολο Εντολών, 723
IBM
    Power5, 699
    System/370, 11
    7094, 28
IEEE
    Πρότυπο 754, 354, 364
Instruction Set, 376
Intel
    4004, 39
    8008, 39
    80386, 39
    8080, 39
    8086, 39, 470
Intel x86, 47, 493
    Διακοπές και Εξαιρέσεις, 499
    Διαχείριση Διακοπών, 500
    Επεξεργασία Διακοπών, 499
    Καταχωρητές Ελέγχου, 496
    Καταχωρητές MMX, 498
    Καταχωρητής EFLAGS, 495
    Μορφές Εντολών, 451
    Οργάνωση Καταχωρητών, 493
    Πίνακας Διανυσμάτων Διακοπών, 500
    Τύποι Δεδομένων, 385
    Τύποι Πράξεων, 402
Interrupts, 78
Infiniband, 264
    Αρχιτεκτονική, 264
    Ιδεατοί Διάδρομοι, 266
    Καταχώρηση Ουράς Εργασίας, 266
IRA, 384
Locality of Reference, 160
LSI-11

Ακολουθία Μικροεντολών, 644
Εκτέλεση Μικροεντολών, 650
Μορφή Μικροεντολής, 654
Οργάνωση Μονάδας Ελέγχου, 651
LSI, 38
Main Memory, 118
MIMD, 676
MIPS R4000, 541
Διασωλήνωση Εντολών, 545
Σύνολο Εντολών, 542
MIPS, 54
MISD, 676
Micro-operations, 577
Microcomputers, 11
MMX, 385
MMU, 129
Moore, 33, 41
Motorola MC68000, 470
Multiplexor, 29
Multiprogramming, 285
Multitasking, 47, 285
MFLOPS, 55
NCR, 28
NUMA, 677
OEM, 36
Omnibus, 36
Operating System, 276
Opcode, 24, 74
Overflow, 336
PCI, 98
Parallel Processing, 714
PDP-10, 446
PDP-11, 448
PDP-8, 445
Pentium 4, 48, 577
Διασωλήνωση, 578
Παραγωγή Μικρολειτουργιών, 579
Pentium III, 48
Pentium II, 48
Pentium Pro, 48
Pentium, 48
Pins, 33
Pipeline, 476
Reservation Table, 484
Speedup, 481
Pipelining, 46
Port, 87
Printed Circuit Board, 91
PROM, 172
Procedure, 398
Processor, 15
Process, 288
RC Delay, 46
RCA, 28
RAID, 196, 206, 701
Rambus DRAM, 168, 186
RAM, 169
Δυναμική, 118
Registers, 16, 24
AC and MQ, 25
AC, 74
I/OAR, 72
I/OBAR, 72
ICC, 610
IBR, 25
IR, 25, 74, 606
MAR, 24, 72, 606
MBR, 24, 72, 469, 606
PC, 25, 74, 469, 606
PSW, 469
Reduced Instruction Set Computer, 11, 382
RISC, 11, 47, 440, 518, 530
σε Αντιδιαστολή με CISC, 535
Σε Αντιδιαστολή με CISC, 554
ROM, 121, 171
SCSI, 93
SIMD, 385, 676
SISD, 676
SMPs, 676, 678
SMP, 677
SPARC, 525, 549
Μορφή Εντολών, 552
Σύνολο Εντολών, 551
Σύνολο Καταχωρητών, 549
Ψευδοεντολές, 558
SPEC, 56
Speedup, 59
SRAM, 168
SSE, 385
SSI, 33
Sun Blade 6250, 57
Superscalar, 46
System Bus, 89
System Resource Manager, 321
Thrashing, 300
Thumb, 455
TI-8800, 655
Αριθμητική και Λογική Μονάδα με Καταχωρητές, 663
Μονάδα Ακολουθίας Μικροεντολών, 660
TLB, 303
ULSI, 38
UMA, 708
UNIVAC, 27
Utilities, 277
Wilkes, 630, 636

Αδιάκριτα Πρωτόκολλα, 688
Ακύρωσης Εγγραφής, 688
Ενημέρωσης Εγγραφής, 689
Αθροιστής, 338
Αίτηση Διακοπής, 80
Ακέραιοι Αριθμοί, 326, 327
Αντίστροφος, 333
Ακολουθία Μικροεντολών, 639
Παραγωγή Διευθύνσεων, 643
Τεχνικές, 640
Άκρο
Διπλό, 424, 428
Επιλογή, 428
Μεγάλο, 424
Μικρό, 424
Ακροδέκτες, 33

Άλγεβρα Boole, 326
Αλγόριθμοι Αντικατάστασης
LRU, 142
LFU, 143
FIFO, 143
Αλγόριθμος
Yeh, 582
Αλγόριθμος Dijksta, 417
Αλγόριθμος Dijkstra, 423
Αλγόριθμος Tomasulo, 572
Αλγόριθμος του Booth, 343, 369
Αλλαγή Φάσης, 219
Αλυσίδωση, 717
Αλώνισμα, 300
Άμεση Εγγραφή, 228
Άμεση Προσπέλαση Μνήμης, 87, 250
Ανάγνωση, 13
Ανάκληση Εντολής, 77
Ανάκληση Παράγοντα, 77
Ανάλυση Ροής Δεδομένων, 42
Αναπαράσταση Κινητής Υποδιαστολής, 350
Απλή Ακρίβεια, 353
Βάση, 350
Διπλή Ακρίβεια, 353
Εκθέτης, 351
Κανονικοποίηση, 351
Σημαντικό Μέρος, 351
Σταθμισμένη, 351
Αναπαράσταση Προσημασμένου Μέτρου, 328
Αναπαράσταση Σταθερής Υποδιαστολής, 333
Αναπαράσταση Συμπληρώματος ως Προς 2, 328
Αναφορά Αφετηρίας Εντολής, 376
Αναφορά Επόμενης Εντολής, 375, 376
Αναφορά Παράγοντα, 430
Αναφορά Παράγοντα Αποτελέσματος, 376
Αναφορά Παράγοντα Αφετηρίας, 375
Αναφορά Παράγοντα Προορισμού, 375
Αντικατάσταση Σελίδας, 300
Αντίστροφη Πολωνική Αναπαράσταση, 422
Αξιολόγηση Απόδοσης, 52
Απεικόνιση Κύριας-Κρυφής Μνήμης, 130
Άμεση, 131
Απεικόνιση Συσχέτισης Συνόλων, 137
Συσχετιστική, 135
Άπειρο, 364
Απόδοση Μνήμης, 120
Παράμετροι, 120
Αποθηκευμένο Πρόγραμμα, 21
Αποθήκευση, 32
Αποθήκευση Δεδομένων, 13
Αποθήκευση Παράγοντα, 77
Αποθήκευσης Δεδομένων, 379
Αποκωδικοποίηση Λειτουργίας Εντολής, 77
Αποκωδικοποιητής, 623
Απομονωτής Αναδιάταξης, 583
Απομονωτής Επαναδιάταξης, 572
Αποστολέας, 289
Αποτελεσματικότητα Προσπέλασης, 164
Αριθμητικές Εντολές, 379
Αριθμητική Διαστημάτων, 363
Αριθμητική και Λογική Μονάδα, 16, 21, 326, 465
Αριθμητική και Λογική Μονάδα με Διασωλήνωση, 715
Αριθμητική Κινητής Υποδιαστολής, 355
Πολλαπλασιασμός και Διαίρεση, 360
Πρόσθεση και Αφαίρεση, 357
Αριθμητική Κορεσμού, 407
Αριθμητική Ολίσθηση, 344, 394
Αριθμητική Υπολογιστών, 326
Αριθμοί Κινητής Υποδιαστολής, 326
Αρχείο Καταχωρητών, 523
Αρχιτεκτονική
IA-64, 525
Συνόλου Εντολών, 54
Αρχιτεκτονική von Neumann, 71
Αρχιτεκτονική ARM, 51
Αρχιτεκτονική Κεντρικής Μεταγωγής, 36
Αρχιτεκτονική Υπολογιστών, 10
Αστοχία, 123
Αφαιρούμενες Μονάδες, 124
Αφέντης Κύκλου, 263

Βαθμιαία Υποχείλιση, 365, 371
Βοηθητικά Προγράμματα, 277
Βοηθητική Μνήμη, 124

Γλώσσα Ελέγχου Εργασίας, 282
Γλώσσα Μικροπρογραμματισμού, 631
Γλώσσα Συμβολομετάφρασης, 767
Μνημονικό, 456
Στοιχεία, 771
Συμβολική Διεύθυνση, 457
Συμβολικό Πρόγραμμα, 456
Τύποι Προτάσεων, 773
Ψευδοεντολή, 457
Γλώσσα Υψηλού Επιπέδου, 378
Γλώσσες Προγραμματισμού Υψηλού Επιπέδου, 518
Γραμμική Ανοικτή Ταινία, 223

Δευτερεύουσα Μνήμη, 124
Διάγραμμα Καταστάσεων, 76
Διάγραμμα Πεπερασμένων Καταστάσεων, 489
Διαγράμματα Χρονισμού, 114
Διαδικασία, 398
Εντολές Επιστροφής, 402
Εντολές Κλήσης, 402
Επανεισόδου, 400
Διαίρεση, 347
Διαιτησία, 94
Διακίνηση, 32
Διακίνηση Δεδομένων, 13
Διακοπές, 78
Αλυσιδωτή Εξέταση, 245
Απενεργοποιημένες, 83
Διαιτησία Διαύλου, 245
Επιλογή Μέσω Λογισμικού, 245
Κύκλος Εντολής, 80
Με Ειδική Μάσκα, 246
Με Περιστροφή, 246
Πλήρως Φωλιασμένες, 246
Πολλαπλές Γραμμές, 245
Πρόγραμμα Ε/Ε, 78
Χειραψία, 248

Διαμάχες Πόρων, 567
Διαμερισμός, 295
Διαμοιρασμός Χρόνου, 287
Διανύσματα Εξαίρεσης, 507
Διανυσματική Διακοπή, 245
Διασύνδεση, 786
  Δυναμική, 788
Διασύνδεση Επεξεργαστή, 465
Διασύνδεση Συστήματος, 15
Διασύνδεση της CPU, 16
Διασωλήνωση, 46, 476, 562
  RISC, 537
  Απόδοση, 481
  Αύξηση Ταχύτητας, 481
  Διαχείριση Διακλαδώσεων, 485
  Κίνδυνοι, 483
  Με Εντολές Διακλάδωσης Υπό Συνθήκη, 478
  Πίνακας Ιστορικού Διακλαδώσεων, 489
  Πίνακας Κρατήσεων, 484
Διασωλήνωση Εντολών, 523
Διάταξη
  Bits, 428
  Bytes, 425
Διάταξη Μετατροπής, 233
Δίαυλος, 89
  PCI, 98
  Αλγόριθμοι Διαιτησίας, 106
  Ασύγχρονος, 96
  Γραμμές Δεδομένων, 89
  Γραμμές Διευθύνσεων, 89
  Γραμμές Ελέγχου, 90
  Γραμμές Μετάδοσης, 89
  Δεδομένων, 89
  Διαιτησία, 106
  Διαιτητής, 94
  Δομή, 89
  Ελεγκτής, 94
  Εύρος, 97
  Πάνω στο Chip, 91
  Σύγχρονος, 95
  Συναλλαγή Αφέντη-Σκλάβου, 95
  Συστήματος, 89, 465
  Σχεδίαση, 93
  Τοπικός, 92
  Τύπος Μεταφοράς Δεδομένων, 97
  Χρονισμός, 95
Δίαυλος Μικροεντολής, 651
Δίαυλος Συστήματος, 16
Διαχείριση Διακλαδώσεων
  Απομονωτής Βρόχου, 486
  Καθυστερημένη Διακλάδωση, 491
  Πολλαπλές Ροές, 486
  Προανάκληση Εντολής Προορισμού Διακλάδωσης, 486
  Πρόβλεψη Διακλάδωσης, 487
Διαχείριση Μνήμης, 286, 294
  Αρχιτεκτονική ARM, 311
  Επεξεργαστής Pentium, 305
Διαχειριστής Διακοπών, 80
Διαχειριστής Πόρων Συστήματος, 321
Διεθνές Αλφάβητο Αναφοράς, 384
Διεργασία, 288, 693
Διεργασίες, 679
Διερμηνέας Εντολών, 71
Διεύθυνση Βάσης, 298
Διεύθυνση Επιστροφής, 400
Διευθυνσιοδότηση, 430
  Άμεση, 432
  Απόλυτη, 433
  Έμμεση, 433
  Έμμεση Καταχωρητή, 434
  Καταχωρητή, 434
  Με Απεικόνιση στη Μνήμη, 240
  Μετατόπισης, 435
  Πεδίο Τρόπου Διευθυνσιοδότησης, 432
  Στοίβας, 437
Διευθυνσιοδότηση Μετατόπισης
  Δεικτοδότησης, 435
  Καταχωρητή Βάσης, 435
  Σχετική, 435
Δισκίο Πυριτίου, 32
Δίσκος
  Σταθερής Κεφαλής, 200
  Μετακινούμενης Κεφαλής, 200
Δομές Διασύνδεσης, 87
Δομή Διαύλου, 36

Ε/Ε
  Επικοινωνία του Επεξεργαστή, 235
  Άμεση Προσπέλαση Μνήμης, 238
  Απομονωμένη, 240
  Διεπαφή από Σημείο σε Σημείο, 259
  Διεπαφή Πολλαπλών Σημείων, 259
  Ελεγκτές, 119
  Ελεγκτής, 237
  Έλεγχος και Χρονισμός, 235
  Εντολές, 238
  Επεξεργαστής, 237
  Επικοινωνία των Συσκευών, 236
  Καθηδηγούμενη από Διακοπές, 241
  Κανάλι, 237
  Μονάδα, 237
  Οδηγούμενη από Διακοπές, 238
  Προγραμματιζόμενη, 237
  Σειριακή και Παράλληλη Διεπαφή, 258
Εγγραφή, 13
Εγγράψιμο CD, 219
Είσοδος-Έξοδος, 13, 15, 22, 43
Εκατομμύρια Εντολές ανά Δευτερόλεπτο, 54
Έκδοση Εντολών, 568
Εκτέλεση Μικροεντολών, 645
Ελεγκτές Ε/Ε, 119
Ελεγκτής DMA Intel 8237A, 252
Ελεγκτής Διακοπών Intel 82C58A, 246
Έλεγχος, 13, 32
Έλεγχος Wilkes, 636
Έμμεσος Κύκλος, 472, 608
Εναλλαγή, 294
Ενεργός Διεύθυνση, 432
Ενθεματική Αναπαράσταση, 422
Ενσωματωμένα Συστήματα, 48, 49
Εντολές
  Αλλαγής Διεύθυνσης, 27

Αριθμητικές, 27
Διακλάδωσης με Συνθήκη, 27
Διακλάδωσης χωρίς Συνθήκη, 27
Διαχείρισης Μνήμης, 405
Δύο Διευθύνσεων, 380
Μεταφοράς Δεδομένων, 27
Μηδέν Διευθύνσεων, 380
Τριών Διευθύνσεων, 380
Εντολές SIMD, 406
Εντολές Διακλάδωσης, 379, 397
Εντολές Διαύλου PCI, 101
Εντολές Ε/Ε, 379
Εντολές Ελέγχου, 379
Εντολές Κλήσης Διαδικασιών, 398
Εντολές Μηχανής, 376, 379
Εντολές Μνήμης, 379
Εντολές Υπερπήδησης, 397
Εντολή, 24
Κωδικός Εντολής, 24
Εντολή Μηχανής
Στοιχεία, 376
Εξαρτήσεις Διαδικασιών, 567
Εξελιγμένος Προγραμματιζόμενος Ελεγκτής Διακοπών, 748
Εξωτερικές Συσκευές, 232
Εξωτερική Μνήμη, 196
Μαγνητικός Δίσκος, 196
Επανατοποθέτηση, 782
Επανεγγράψιμο CD, 219
Επέκταση Προσήμου, 332
Επεξεργασία, 32
Επεξεργασία Δεδομένων, 13
Επεξεργασία Διανυσμάτων, 713
Επεξεργαστές
Πολλαπλών Πυρήνων, 736
Υπερβαθμωτοί, 46
Επεξεργαστές Υπερδιασωλήνωσης, 545
Επεξεργαστής, 15
Υπερβαθμωτός, 562
Επεξεργαστής Ε/Ε, 256
Επεξεργαστής Πινάκων, 713
Επικάλυψη Ανάκλησης, 476
Επικοινωνίες Δεδομένων, 13
Εργασία, 281
Εσωτερικός Δίαυλος Επεξεργαστή, 465

Θύρα, 87

Ιδεατή Διεύθυνση, 129
Ιδεατή Κρυφή Μνήμη, 129
Ιδεατή Μνήμη, 160, 276, 300
Ιεραρχία Μνήμης, 122
Ισορροπία Απόδοσης, 42
Ισόχρονο Κενό, 263
Ισόχρονος Κύκλος, 263
Ίχνη Δίσκου, 198

Καθολικές Μεταβλητές, 526
Καθυστερημένη Διακλάδωση, 538, 576
Καθυστερημένη Φόρτωση, 540
Καθυστέρηση Αντίστασης-Πυκνότητας, 46
Κανάλια Δεδομένων, 29
Κανάλια Ε/Ε, 256
Ελεγκτής, 257
Επιλογέας, 257
Κανάλι Πολυπλέκτης, 257
Πολυπλέκτης Μπλοκ, 257
Κανόνας του Pollack, 740
Κατακόρυφες Μικροεντολές, 636
Καταστάσεις Διεργασιών, 289
Κατάσταση Λειτουργίας Πυρήνα, 504
Καταχωρητές, 16, 24, 72, 118
Ελέγχου Συστήματος, 388
Εντολής, 377, 606
Κωδικού Κύκλου Εντολής, 610
Μετρητής Προγράμματος, 606
Καταχωρητής, 465
Γενικού Σκοπού, 467
Δεδομένων, 467
Δείκτη, 436, 467
Διευθύνσεων Ε/Ε, 72
Διεύθυνσης, 467
Διεύθυνσης Μνήμης, 24, 72, 469, 606
Ελέγχου, 396
Ελέγχου και Κατάστασης, 469
Ενδιάμεσης Αποθήκευσης Ε/Ε, 72
Ενδιάμεσος Καταχωρητής Εντολής, 25
Ενδιάμεσος Καταχωρητής Μνήμης, 24, 72, 469, 606
Εντολής, 25, 74, 469
Λέξη Κατάστασης Προγράμματος, 469
Μετρητής Προγράμματος, 25, 74, 469
Ορατός στο Χρήστη, 466
Συσσωρευτής, 74, 445
Συσσωρευτής και Πηλίκο Πολλαπλασιαστή, 25
Τμημάτων, 467
Καταχωρητής Διανυσμάτων, 717
Καταχωρητής Εφεδρικής Αποθήκευσης Εντολών, 29
Κελί Μνήμης, 32, 168
Κεντρική Μονάδα Επεξεργασίας, 15
Κεφαλή Δίσκου, 197
Κίνδυνοι Διασωλήνωσης
Δεδομένων, 484
Ελέγχου, 485
Πόρων, 483
Κίνδυνος Δεδομένων
Αντιεξάρτηση, 485
Εξάρτηση Εξόδου, 485
Πραγματική Εξάρτηση, 485
Κλάσεις Προτεραιοτήτων Διακοπών και Εξαιρέσεων, 500
Κλήσεις Διαδικασιών, 522
Κλήση Εγγραφής, 78
Κλοιτή κύκλου, 250
Κρατούμενο, 335
Κρυμμένη Διαιτησία, 108
Κρυφή CDRAM, 188
Κρυφή Μνήμη, 118
Αλώνισμα, 134
Αρχές, 125
Αρχές Σχεδίασης, 128
Γραμμές, 126
Διευθύνσεις, 128
Επίπεδα, 118
Ετικέτα, 126

Θύμα, 134
Ιχνών, 580
Μέγεθος, 129
Μέγεθος Γραμμής, 144
Συνοχή, 144
Κρυφή Μνήμη Δίσκου, 160
Κυκλικός Απομονωτής, 525
Κύκλος Ανάκλησης, 26, 73, 606
Κύκλος Διακοπής, 80, 473, 608
Κύκλος Διαύλου, 95
Κύκλος Εκτέλεσης, 26, 73, 609
Κύκλος Εντολής, 70, 73, 472, 605, 610
Κύκλος Μηχανής, 532
Κύκλος Μικροεντολής, 645
Κύκλος Ρολογιού, 95
Κύλινδρος, 201
Κύρια Μνήμη, 15, 21, 72, 118
Κώδικας ASCII, 384
Κώδικας EBCDIC, 384
Κώδικας Hamming, 179
Κώδικες Διόρθωσης Σφαλμάτων, 179
Κωδικοί Συνθηκών, 405
Κωδικοποίηση
Άμεση, 650
Έμμεση, 650
Κωδικός Εντολής, 24, 74, 375
Κωδικός Λειτουργίας, 376

Λανθάνων Δίσκος, 125
Λανθάνων Χρόνος, 120
Λανθάνων Χρόνος Μνήμης, 46
Λανθάνων Χρόνος Πράξης, 568
Λειτουργία Ε/Ε, 78, 86
Λειτουργικό Σύστημα, 276
Αλληλεπίδρασης, 280
Δέσμης, 280
Διαχείριση Μνήμης, 276
Ομαδοποίησης, 281
Παρακολούθησης, 281
Πόροι, 279
Πυρήνας, 279
Χρονοδρομολόγηση, 276
Λέξεις Μνήμης, 23
Λέξη Ελέγχου, 631
Λέξη Κατάστασης Προγράμματος, 243
Λέξη Μνήμης, 74, 119
Λογικές Εντολές, 379
Λογικές Πράξεις, 32
Λογική Διαχείρισης Ισχύος, 749
Λογική Εγγραφή, 227
Λογική Ελέγχου, 233
Λογική Κρυφή Μνήμη, 129
Λογισμικό, 71
Λογισμικό Συστήματος, 28
Λόγος Ευστοχίας, 66, 123
Λυχνίες Κενού, 20

Μαγνητική Ταινία, 221
Ελικοειδής Εγγραφή, 222
Παράλληλη Εγγραφή, 222
Σειριακή Εγγραφή, 222
Μαγνητικός Δίσκος, 196
Μειωμένο Σύνολο Εντολών, 518, 530
Μερικό Υπόλοιπο, 347
Μέσο Πλήθος Κύκλων ανά Εντολή, 54
Μεταθεματική Αναπαράσταση, 422
Μετακίνησης Δεδομένων, 379
Μετονομασία Καταχωρητών, 573
Μετρητής Προγράμματος, 469
Μέτρο Ρυθμού, 58
Μέτρο Ταχύτητας, 57
Μετροπρογράμματα, 56
Μη Κανονικοποιημένοι Αριθμοί, 364
Μη Ομοιόμορφη Προσπέλαση Μνήμης, 677, 708
θετικά και Αρνητικά Σημεία, 711
Κίνητρα, 708
Οργάνωση, 709
Μήμη
Επίπεδα, 124
Μηχανή Von Neumman, 21
Μικροεντολές, 16, 630
Εκτέλεση, 645
Ζητήματα Σχεδίασης, 639
Κωδικοποίηση, 649
Ταξινόμηση, 645
Μικροεντολή, 445, 631
Μικροεπεξεργαστές, 39
Μικρολειτουργίες, 149, 577, 605
Μικροπρόγραμμα, 630, 631
Μικροπρογραμματιζόμενη Μονάδα Ελέγχου, 633
Μικροπρογραμματισμός, 16
Μικρουπολογιστές, 11
Μνήμη, 72, 118
DDR SDRAM, 187
DRAM, 168, 170
EEPROM, 172
EPROM, 172
Rambus DRAM, 186
SDRAM, 184
SRAM, 168, 170
Flash, 172
μη Πτητική, 121
Ανάγνωσης Μόνο, 121, 171
Απόδοση, 120
Διαφυλλωμένη, 177
Εξωτερική, 118, 196
Εσωτερική, 118
Ημιαγωγών, 38
Ιδεατή, 124
Ιεραρχία, 122
Κελί, 168
Κελιά, 32
Κρυφή, 118
Κρυφή DRAM, 188
Κύρια, 118
Κυρίως Ανάγνωσης, 172
Λανθάνων Χρόνος, 46
Μαγνητικών Πυρήνων, 38
Μονάδα Μεταφοράς, 119
Μονάδες, 119
Προγραμματιζόμενη ROM, 172
Προσπέλαση, 118, 119

Πτητική, 121
Τμήματα, 119
Τυχαίας Προσπέλασης, 118, 169
Φυσική Διεύθυνση, 129
Φυσικοί Τύποι, 121
Χαρακτηριστικά, 118
Χρόνος Κύκλου, 29
Χωρητικότητα, 119, 120
Μνήμη Ελέγχου, 632
Μονάδα DMA, 250
Μονάδα Ανάκλησης Εντολών, 586
Μονάδα Αποκωδικοποίησης Εντολών, 587
Μονάδα Διαχείρισης Μνήμης, 129
Μονάδα Ε/Ε, 87
Μονάδα Εκτέλεσης Πράξεων με Ακέραιους, 588
Μονάδα Ελέγχου, 13, 16, 21, 24
Λειτουργικές Απαιτήσεις, 611
Χρονισμός και Έλεγχος, 619
Μονάδα Θερμικού Ελέγχου, 747
Μονάδα Παραγωγής Διευθύνσεων, 587
Μονορογραμματισμός, 280
Μορφές Εντολών, 442
Κατανομή των Bits, 444
Μήκος Εντολής, 443
Μορφές Λειτουργικών Συστημάτων, 279
Μπλοκ Ελέγχου Διεργασίας, 290

Νήματα, 679
Νόμος του Amdahl, 53, 58, 741
Νόμος του Amdhal, 66

Ξετύλιγμα Βρόχων, 540

Οικογένειες Υπολογιστών, 35
Ολοκληρωμένα Κυκλώματα, 31
Μεγάλης Κλίμακας, 38
Μικρής Κλίμακας, 33
Πολύ Μεγάλη Κλίμακα, 38
Υπερβολικά Μεγάλης Κλίμακας, 38
Οπτικοί Δίσκοι, 118
Οπτικοί Δίσκοι Υψηλής Ευκρίνειας, 221
Οργάνωση, 121
Οργάνωση Υπολογιστών, 10
Οριζόντια Μικροεντολή, 632

Παράθυρα Καταχωρητών, 524
Καταχωρητές Παραμέτρων, 524
Προσωρινοί Καταχωρητές, 524
Τοπικοί Καταχωρητές, 524
Παράθυρο Εντολών, 571
Παράλληλες Αριθμητικές και Λογικές Μονάδες, 715
Παράλληλη Επεξεργασία, 714
Παραλληλισμός
Μηχανής, 568, 574
Σε Επίπεδο Εντολής, 562, 565, 568
Παράλληλοι Επεξεργαστές, 715
Παράπλευρος Απομονωτής Μετάφρασης, 303
Περιστροφή, 395
Περιφερειακά, 13
Περιφερειακή Συσκευή, 232
Πίνακας Σελίδων, 298
Αντεστραμμένος, 302
Πλαίσια Σελίδων, 298
Πλαίσιο Στοίβας, 401
Πλακέτα Τυπωμένων Κυκλωμάτων, 91
Πλήθος Διυθύνσεων, 379
Πολιτικές Εγγραφής
Δια μέσου Εγγραφή, 143
Πίσω-εγγραφή, 143
Πολιτική Έκδοσης Εντολών, 568
Έκδοση Εκτός Σειράς και Ολοκλήρωση Εκτός Σειράς, 571
Έκδοση με Σειρά και Ολοκλήρωση Εκτός Σειράς, 570
Έκδοση με Σειρά και Ολοκλήρωση με Σειρά, 569
Πολλαπλασιασμός, 339
Πολλαπλασιασμός Συμπληρώματος ως Προς 2, 341
Πολλαπλές Διακοπές, 82
Πολλαπλοί Δίαυλοι
Ιεραρχία, 91
Πολλαπλοί Επεξεργαστές
Μορφές Οργάνωσης, 676
Πολλαπλοί Πυρήνες, 46, 695, 736
Intel, 747
Οργάνωση, 745
Πολυδιεργασία, 47
Πολυεπεξεργασία, 285
Πολυνημάτωση, 693
Άμεση, 695
Έμμεση, 693
Παρεμποδισμένη, 695
Ταυτόχρονη, 695
Φυλλωμένη, 695
Πολυπλέκτης, 29
Πολυπλεξία Χρόνου, 94
Πολυπρογραμματισμός, 280, 285
Πολυπυρήνες, 46
Πραγματική Εξάρτηση Δεδομένων, 565
Πράξεις από Καταχωρητή σε Καταχωρητή, 533
Πράξη Πάνω στα Δδεδομένα, 77
Προανάκληση Εντολής, 476
Πρόβλεψη Διακλαδώσεων, 575
Πρόβλεψη Διακλάδωσης, 41
Πρόβλημα των Μήλων, 272
Πρόγραμμα Ε/Ε, 78
Προγραμματιζόμενη Περιφερειακή Διεπαφή Intel 82C55A, 246
Προδιαγραφή EIA-232, 259
Πρόσθεση και Αφαίρεση, 336
Προσπέλαση Μνήμης
Άμεση, 120
Σειριακή, 120
Συσχετιστική, 120
Τυχαία, 120
Προτεραιότητα Διακοπών, 83
Πρωτόκολλα Καταλόγου, 688
Πρωτόκολλο Mesi, 687, 689
Πύλες, 32
ΚΑΙ, 32

Ρολόι, 53
Ρυθμός, 53
Στιγμή, 53

Ταχύτητα, 53
Ρυθμαπόδοση, 58
Ρυθμός MIPS, 54
Ρυθμός MFLOPS, 55
Ρυθμός Εκτέλεσης Εντολών, 54
Ρυθμός Μετάδοσης, 120
Ρυθμός Ρολογιού, 53

Σειριακός Δίαυλος FireWire, 259
Στοίβα Πρωτοκόλλων, 260
Σελίδες, 298
Σελιδοποίηση, 298
Pentium II, 307
Κατ' απαίτηση, 300
Σήμα Διακοπής, 241
Σήμα Ελέγχου, 32
Ανάγνωσης, 32
Εγγραφής, 32
Σημαίες, 468
Σημαίες Κατάστασης, 405
Σήματα
Αρνητική Ακμή, 114
Θετική Ακμή, 114
Σήματα Ελέγχου, 612
Σκληρός Δίσκος, 124
Σταθερή Γραμμική Ταχύτητα, 218
Σταθερή Γωνιακή Ταχύτητα, 218
Σταθερολό Τμήμα Λογισμικού, 631
Στιγμή Ρολογιού, 53
Στοίβα, 381
Αύξουσα, 421
Κενή , 422
Κορυφή, 420
Μήκος, 420
Πίεση, 420
Πλήρης, 422
Υλοποίηση, 420
Φθίνουσα, 421
Ώθηση, 420
Στοίβα Συστήματος, 85
Στοίβες, 420
Στοιχεία Ε/Ε, 71
Στρογγυλοποίηση, 362
Σύγχρονη SDRAM, 184
Συγχρονισμένη DRAM, 168
Συμβολική Αναπαράσταση, 377
Συμβολικός Παράγοντας, 378
Συμβολομεταφραστές, 777
Δύο Περασμάτων, 777
Ενός Περάσματος, 779
Συμβολομεταφραστής, 457, 767
Συμμετρικοί Πολυεπεξεργαστές, 676, 678
Μεγάλο Υπολογιστικό Σύστημα SMP, 683
Συμμετρικός Πολυεπεξεργαστής, 677
Συμπαγής Δίσκος, 215
Συμπλήρωμα ως προς 1, 368
Σύνθετο Σύνολο Εντολών, 530
Σύνολα Καταχωρητών, 444
Σύνολο Εντολών, 376
Thumb, 455
Σχεδίαση, 381

Συνοχή Κρυφής Μνήμης, 686
Λύσεις Λογισμικού, 687
Λύσεις Υλικού, 687
Πρωτόκολλα, 687
Συσκευασμένοι Δεκαδικοί, 383
Συσκευή Ε/Ε, 87
Συσσωρευτής, 380
Συστάδα, 677
Κόμβος, 699
Συστάδες, 699
Αρχιτεκτονική Υπολογιστών, 705
Διατάξεις, 701
Εικόνα Μονού Συστήματος, 705
Λειτουργικά Συστήματα, 703
Σύγκριση με Συμμετρικούς Πολυεπεξεργαστές, 707
Φέτες Εξυπηρετών, 707
Συσταδοποίηση, 699
Σύστημα RAID
Διαχωρισμός σε Λωρίδες, 209
Σύστημα Παρακολούθησης, 281
Σφάλμα Σελίδας, 300

Ταυτόχρονη Πολυνημάτωση, 747
Ταχύτητα Επεξεργαστή, 44
Ταχύτητα Μικροεπεξεργαστή, 41
Ταχύτητα Ρολογιού, 53
Τεχνικές Χρονοδρομολόγησης, 290
Τεχνολογία MMX, 406
Τιμή NaN, 364
Σηματοδότησης, 364
Σιωπηρή, 364
Τμηματοποίηση, 305, 435
Pentium II, 306
Τομείς Δίσκου, 198
Τοπικότητα, 160
Τοπικότητα Αναφοράς, 123, 160, 435
Τρανζίστορ, 28
Τράπεζα Μνήμης, 177
Τρόποι Λειτουργίας σε Κατάσταση Εξαίρεσης, 504
Τρόπος Αναπαράστασης Αριθμών, 326
Τύποι Δεδομένων
Αριθμοί, 383
Λογικά Δεδομένα, 384
Χαρακτήρες, 384
Τύποι Διαύλων, 93
Δεσμευμένοι, 93
Πολυπλεγμένοι, 93
Τύποι Εντολών, 378
Επεξεργασίας Δεδομένων, 379
Τύποι Παραγόντων, 382
Τύποι Πράξεων, 388
Αριθμητικές, 391
Εισόδου/Εξόδου, 396
Ελέγχου Συστήματος, 396
Λογικές, 392
Μετατροπής, 395
Μεταφοράς Δεδομένων, 389
Μεταφοράς Ελέγχου, 396

Υπερβαθμωτή Σχεδίαση, 562
Υπερβαθμωτοί Επεξεργαστές, 46, 545, 562

Εκτέλεση Προγραμμάτων, 576
Σε Αντιδιαστολή με Επεξεργαστές Υπερδιασωλήνωσης, 564
Υλοποίηση, 577
Υπερνημάτωση, 698
Υπερχείλιση, 329, 336
Υποδιαστολή, 327
Υποθετική Εκτέλεση Εντολών, 42
Υπολογισμοί Διανυσμάτων, 712
IBM 3090, 718
Προσεγγίσεις, 713
Υπολογισμός Διεύθυνσης Εντολής, 76
Υπολογισμός Διεύθυνσης Παράγοντα, 77
Υπολογιστές
Οικογένειες, 35
Υπολογιστής
IAS, 21
Αποθηκευμένων Προγραμμάτων, 21
Αποθήκευσης Προγραμμάτων, 28
Μειωμένου Συνόλου Εντολών, 382, 518
Σύνθετου Συνόλου Εντολών, 55
Υπολογιστής Μειωμένου Συνόλου Εντολών, 11

Φόρτωση, 784
Φυσική Εγγραφή, 227

Χαρακτήρας, 234
Χαρακτήρες Ελέγχου, 384
Χρονική Ακολουθία Εκτέλεσης Εντολών, 605
Χρονική Τοπικότητα, 162
Χρονοδρομολόγηση, 286, 288
Βραχυπρόθεσμη, 289
Μακροπρόθεσμη, 288
Μεσοπρόθεσμη, 289
Χρόνος Αναζήτησης, 203
Χρόνος Κύκλου, 53
Χρόνος Κύκλου Μνήμης, 29, 120
Χρόνος Κύκλου ρολογιού, 613
Χρόνος Μεταφοράς, 203
Χρόνος Ολοκλήρωσης, 318
Χρόνος Περιστροφής, 203
Χρόνος Προσπέλασης, 203
Χρόνος Προσπέλασης Μνήμης, 120
Χρωματισμός Γραφημάτων, 529
Χωρική Τοπικότητα, 162
Χώρος Διευθύνσεων
Pentium II, 306

Ψηφιακός Ευπροσάρμοστος Δίσκος, 220

Acronyms

ACM	Association for Computing Machinery
ALU	Arithmetic Logic Unit
ASCII	American Standards Code for Information Interchange
ANSI	American National Standards Institute
BCD	Binary Coded Decimal
CD	Compact Disk
CD-ROM	Compact Disk-Read Only Memory
CPU	Central Processing Unit
CISC	Complex Instruction Set Computer
DRAM	Dynamic Random-Access Memory
DMA	Direct Memory Access
DVD	Digital Versatile Disk
EPIC	Explicitly Parallel Instruction Computing
EPROM	Erasable Programmable Read-Only Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
HLL	High-Level Language
I/O	Input/Output
IAR	Instruction Address Register
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
ILP	Instruction-Level Parallelism
IR	Instruction Register
LRU	Least Recently Used
LSI	Large-Scale Integration
MAR	Memory Address Register
MBR	Memory Buffer Register
MESI	Modify-Exclusive-Shared-Invalid
MMU	Memory Management Unit
MSI	Medium-Scale Integration
NUMA	Nonuniform Memory Access
OS	Operating System
PC	Program Counter
PCI	Peripheral Component Interconnect
PROM	Programmable Read-Only Memory
PSW	Processor Status Word
PCB	Process Control Block
RAID	Redundant Array of Independent Disks
RALU	Register/Arithmetic-Logic Unit
RAM	Random-Access Memory
RISC	Reduced Instruction Set Computer
ROM	Read-Only Memory
SCSI	Small Computer System Interface
SMP	Symmetric Multiprocessors
SRAM	Static Random-Access Memory
SSI	Small-Scale Integration
ULSI	Ultra Large-Scale Integration
VLSI	Very Large-Scale Integration
VLIW	Very Long Instruction Word