

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ



Διάρθρωση

1. Επίπεδο ISA
2. Μοντέλα μνήμης
3. Τύποι δεδομένων
4. Μορφές/τύποι εντολών
5. Επεκτεινόμενοι κωδικοί πράξης
6. Διευθυνσιοδότηση
7. Ροή ελέγχου
8. Μηχανισμοί εισόδου/εξόδου



ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΝΟΛΟΥ ΕΝΤΟΛΩΝ

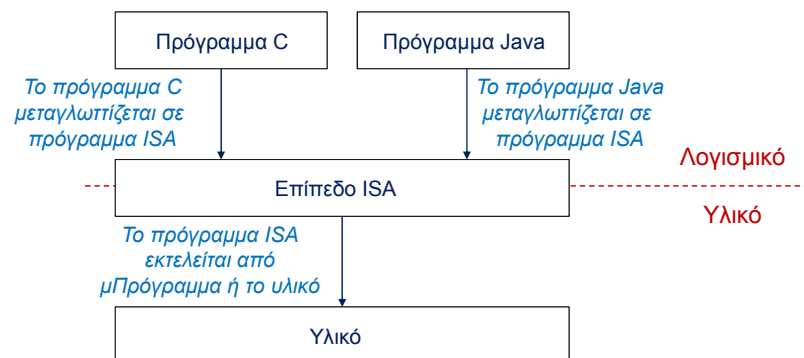


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Επίπεδο ISA

- Το επίπεδο της *αρχιτεκτονικής συνόλου εντολών* (*instruction set architecture – ISA*) αποτελεί τη διασύνδεση μεταξύ υλικού και λογισμικού (μεταγλωττιστές).



- Απαιτήσεις:
 - αποδοτικά υλοποιήσιμο από τους σχεδιαστές υλικού
 - κανονικότητα και πληρότητα εντολών για την εύκολη παραγωγή καλού κώδικα από τους σχεδιαστές λογισμικού
 - συμβατότητα προς τα πίσω



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Επισκόπηση Επιπέδου ISA

- Το επίπεδο της *αρχιτεκτονικής συνόλου εντολών* (*instruction set architecture – ISA*) είναι ο τρόπος με τον οποίο εμφανίζεται ένα υπολογιστικό σύστημα στον προγραμματιστή της γλώσσας μηχανής. Η εικόνα αυτή περιλαμβάνει:
 - το μοντέλο μνήμης
 - τους καταχωρητές
 - τις εντολές
 - τους τύπους δεδομένων
- Πληροφορίες που αφορούν ζητήματα όπως: αν η μΑρχιτεκτονική είναι μΠρογραμματιζόμενη ή όχι, αν χρησιμοποιεί δομή διοχέτευσης ή όχι, αν είναι υπερβαθμωτή ή όχι κ.τ.λ. δεν ανήκουν στο επίπεδο ISA καθώς δεν είναι ορατά στον σχεδιαστή του μεταγλωττιστή.
- Στο επίπεδο ISA συνήθως υπάρχουν τουλάχιστον δύο καταστάσεις λειτουργίας:
 - κατάσταση πυρήνα (kernel mode) που προορίζεται για την εκτέλεση του λειτουργικού συστήματος και επιτρέπει να εκτελούνται όλες οι εντολές
 - κατάσταση χρήστη (user mode) που προορίζεται για την εκτέλεση εφαρμογών και δεν επιτρέπει την εκτέλεση ορισμένων ευαίσθητων εντολών.

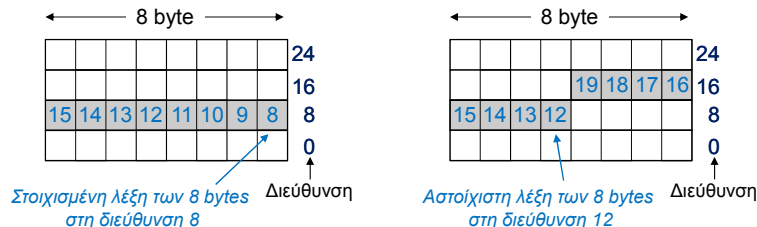


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Μοντέλα Μνήμης

- Συνηθισμένο μέγεθος μιας θέσης μνήμης είναι τα 8-bit (δηλ. 1 byte).
- Τα bytes συνήθως ομαδοποιούνται σε λέξεις των 4 bytes ή 8 bytes και όπως είδαμε υπάρχουν εντολές χειρισμού ολόκληρων λέξεων.
- Πολλές αρχιτεκτονικές απαιτούν οι λέξεις στη μνήμη να είναι *στοιχισμένες* στα “φυσικά τους όρια”. Π.χ. μια λέξη των 4-bytes να αρχίζει στη διεύθυνση 0, 4, 8, κ.τ.λ. και όχι στη διεύθυνση 1 ή 2. Παρόμοια μια λέξη των 8-bytes να αρχίζει στη διεύθυνση 0, 8, 16, κ.τ.λ., όπως στο παράδειγμα.



- Υπάρχουν μηχανές που έχουν ξεχωριστούς χώρους για δεδομένα και εντολές, χρησιμοποιώντας διευθύνσεις ίδιου αριθμού bits.
 - διπλασιασμός μνήμης
 - δεν επιτρέπεται εγγραφή στη μνήμη εντολών



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Μοντέλο Μνήμης IJVM

- Η διαχείριση της μνήμης μπορεί να γίνει με δύο τρόπους:
 - ως πίνακας 4.294.967.296 byte (4GB)
 - ως πίνακας 1.073.741.824 λέξεων των 4 byte
- Οι διευθύνσεις μνήμης δεν είναι άμεσα ορατές στο επίπεδο ISA.
- Υπάρχουν υπονοούμενες διευθύνσεις με βάση *δείκτες διευθύνσεων (pointers)*.
- Η προσπέλαση στη μνήμη γίνεται μόνο με *σχετική διευθυνσιοδότηση* με βάση τους δείκτες διευθύνσεων.



Καταχωρητές

- Δεν είναι όλοι οι καταχωρητές του επιπέδου της μΑρχιτεκτονικής “ορατοί” στο επίπεδο ISA
 - π.χ. στη Mic μηχανή οι TOS και MAR δεν είναι ορατοί ενώ οι PC και SP είναι ορατοί στο επίπεδο ISA.
- Οι καταχωρητές χωρίζονται σε δύο κατηγορίες:
 - καταχωρητές ειδικής χρήσης (σκοπού): π.χ. μετρητής προγράμματος PC ή ο δείκτης στοίβας SP
 - καταχωρητές γενικής χρήσης: με αποστολή τη φύλαξη τοπικών μεταβλητών και ενδιάμεσων αποτελεσμάτων (π.χ. ο H)
 - ο *συνήθως είναι συμμετρικοί και εναλλάξιμοι.*
- Κάποιοι καταχωρητές είναι διαθέσιμοι μόνο σε κατάσταση πυρήνα.
- *Καταχωρητής σημαιών (flags register)* που περιέχει διάφορα bit πληροφορίας:
 - *κωδικοί συνθήκης (condition codes):* δείχνουν την κατάσταση του αποτελέσματος της πιο πρόσφατης πράξης
 - ο *N (negative), Z (zero), V (overflow), C (carry), P (parity)*
 - κατάσταση λειτουργίας (πυρήνα ή χρήστη)
 - κατάσταση ενεργοποίησης διακοπών κ.α.



Τύποι Δεδομένων

- Σε έναν υπολογιστή, τα δεδομένα αναπαρίστανται με μια συγκεκριμένη μορφή.
- Υποστήριξη υλικού: οι εντολές αναμένουν δεδομένα σε συγκεκριμένη μορφή και δεν επιτρέπεται να διαλέξει ο χρήστης διαφορετική μορφή.
- Τύποι δεδομένων:
 - Αριθμητικοί τύποι
 - ο ακέραιοι (8, 16, 32, 64 bit – προσημασμένοι / απρόσημοι)
 - ο κινητής υποδιαστολής (32 bit ή 64 bit – απλής ή διπλής ακρίβειας)
 - Μη αριθμητικοί τύποι
 - ο χαρακτήρες (ascii, unicode)
 - ο αλφαριθμητικά (character strings)
 - ο λογικές τιμές (boolean)
 - ο δείκτες (pointers) δηλ. διευθύνσεις μνήμης



Τύποι Δεδομένων της JVM

- Η Java είναι μια γλώσσα με ισχυρό έλεγχο τύπων (strongly typed).
- Η JVM υποστηρίζει:
 - προσημασμένους ακέραιους στην αναπαράσταση του συμπληρώματος ως προς 2
 - πραγματικούς κινητής υποδιαστολής απλής και διπλής ακρίβειας
 - χαρακτήρες με τη μορφή unicode των 16 bit
 - δείκτες

Αριθμητικοί τύποι δεδομένων της JVM

Type	8 Bits	16 Bits	32 Bits	64 Bits	128 Bits
Signed integer	×	×	×	×	
Unsigned integer					
Binary coded decimal integer					
Floating point			×	×	



ΕΝΤΟΛΕΣ



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Μορφές Εντολών 1/2

- Μια εντολή αποτελείται από έναν *κωδικό πράξης (opcode)* και ενδεχόμενα ένα ή περισσότερα πεδία όπου προσδιορίζεται η διεύθυνση των τελεστών πάνω στους οποίους επενεργούν ή η διεύθυνση όπου καταχωρούνται τα αποτελέσματα.
- Ο προσδιορισμός της διεύθυνσης των τελεστών ονομάζεται *διευθυνσιοδότηση (addressing)*.
- Αριθμός διευθύνσεων ανά εντολή:

Εντολή μηδέν διευθύνσεων

OPCODE

Εντολή μιας διεύθυνσης

OPCODE	ΔΙΕΥΘΥΝΣΗ
--------	-----------

Εντολή δύο διευθύνσεων

OPCODE	ΔΙΕΥΘΥΝΣΗ1	ΔΙΕΥΘΥΝΣΗ2
--------	------------	------------

Εντολή τριών διευθύνσεων

OPCODE	ΔΙΕΥΘΥΝΣΗ1	ΔΙΕΥΘΥΝΣΗ2	ΔΙΕΥΘΥΝΣΗ3
--------	------------	------------	------------

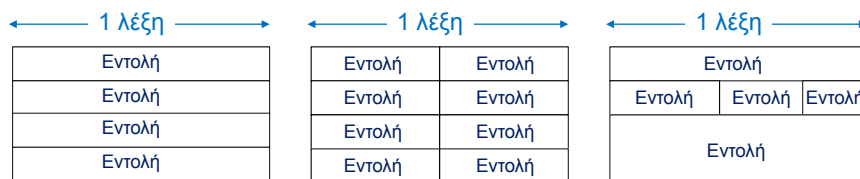


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Μορφές Εντολών 2/2

- Μήκος των εντολών (σε # bits):
 - οι εντολές έχουν το ίδιο μήκος ή διαφορετικά μήκη
 - ο απλούστερη υλοποίηση όταν όλες οι εντολές έχουν το ίδιο μήκος αλλά αυτό συνεπάγεται σπατάλη χώρου μνήμης
 - οι εντολές έχουν μήκος μικρότερο, ίσο ή μεγαλύτερο από το μήκος της λέξης



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Κριτήρια Σχεδιασμού Εντολών

- Τεχνολογία: π.χ. αν έχουμε γρήγορη μνήμη τότε η χρήση εντολών διαχείρισης στοίβας (JVM) είναι καλή επιλογή, αν όχι η σωστή επιλογή είναι η χρήση πολλών καταχωρητών (UltraSPARC II).
- Μήκος των εντολών
 - μικρότερες εντολές $\Rightarrow$ μικρότερο μέγεθος προγράμματος
 - το εύρος ζώνης της μνήμης (# bits/sec που μεταφέρονται) αξιοποιείται καλύτερα με μικρότερες εντολές
 - ο αν το εύρος ζώνης μιας μνήμης είναι t bps (bits per second) και οι εντολές έχουν μέσο μήκος r bits, τότε η μνήμη μπορεί να παρέχει t/r εντολές ανά sec. Συνεπώς, μικρότερες εντολές $\Rightarrow$ μικρότερο $r \Rightarrow$ περισσότερες εντολές ανά sec.
- Δυνατότητα επέκτασης συνόλου εντολών: αν ζητάμε 2^n πράξεις τότε θέλουμε εντολές με κωδικό πράξης τουλάχιστον n bits. Στην πράξη, για να έχουμε δυνατότητα επέκτασης, ο κωδικός πράξης θα πρέπει να είναι $> n$.
- Αριθμός των bits ενός πεδίου διεύθυνσης: π.χ. μνήμη με 2^{32} bytes
 - προσπέλαση κατά 1 byte $\Rightarrow$ διεύθυνση των 32 bits
 - προσπέλαση κατά 2 byte $\Rightarrow$ διεύθυνση των 31 bits
 - προσπέλαση κατά 4 byte $\Rightarrow$ διεύθυνση των 30 bits
 - ο απλούστερη σχεδίαση αλλά για χρήση 1 byte θα μεταφερθούν 4 bytes



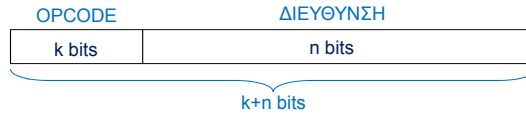
Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Μέγεθος Κωδικού Πράξης

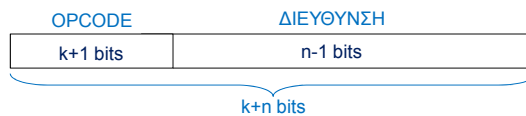
- Έστω εντολή των $k+n$ bits, όπου k -bits είναι ο κωδικός πράξης και n -bits είναι το πεδίο της διεύθυνσης.
- Η εντολή επιτρέπει 2^k διαφορετικές πράξεις και 2^n διευθύνσεις μνήμης.

Εντολή των $k+n$ bits
 2^k πράξεις
 2^n διευθύνσεις



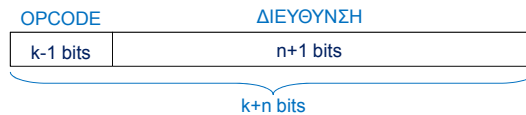
- Εντολή με $k+1$ bits κωδικό πράξης και $n-1$ bits πεδίο διεύθυνσης, επιτρέπει τις διπλασίες πράξεις (2^{k+1}) και τις μισές διευθύνσεις (2^{n-1}) μνήμης.

Εντολή των $k+n$ bits
 2^{k+1} πράξεις
 2^{n-1} διευθύνσεις



- Εντολή με $k-1$ bits κωδικό πράξης και $n+1$ bits πεδίο διεύθυνσης, επιτρέπει τις μισές πράξεις (2^{k-1}) και τις διπλασίες διευθύνσεις (2^{n+1}) μνήμης.

Εντολή των $k+n$ bits
 2^{k-1} πράξεις
 2^{n+1} διευθύνσεις



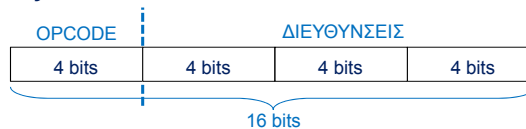
Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

15

Επεκτεινόμενοι Κωδικοί Πράξης 1/3

- Έστω μηχανή με 16 καταχωρητές, με εντολές των 16 bits, όπου λογικά αναμένεται οι διευθύνσεις να είναι των 4 bits.
- Ένας πιθανός σχεδιασμός είναι να έχουμε κωδικό πράξης των 4 bits και 3 διευθύνσεις, δηλ. 16 εντολές των 3 διευθύνσεων.

Εντολή των 16 bits
 $2^4 = 16$ πράξεις
 3 διευθύνσεις των 4 bits



- Έστω ότι χρειαζόμαστε:
 - 15 εντολές με 3 διευθύνσεις
 - 14 εντολές με 2 διευθύνσεις
 - 31 εντολές με 1 διεύθυνση και
 - 16 εντολές χωρίς διευθύνσεις
- Απαιτούνται 76 εντολές και συνεπώς ο κωδικός πράξης πρέπει να έχει 7 bits.
- Συνεπώς, σε μια απλοϊκή προσέγγιση, τα 16 bit δεν αρκούν για εντολές 3 διευθύνσεων.

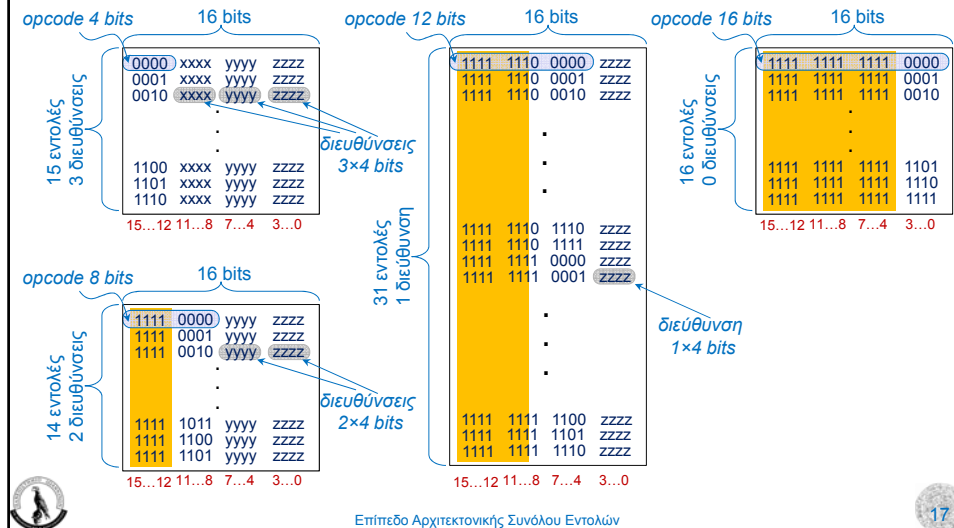


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

16

Επεκτεινόμενοι Κωδικοί Πράξης 2/3

- Η λύση στο προηγούμενο πρόβλημα είναι η χρήση *επεκτεινόμενων κωδικών πράξης* (expanding opcodes).

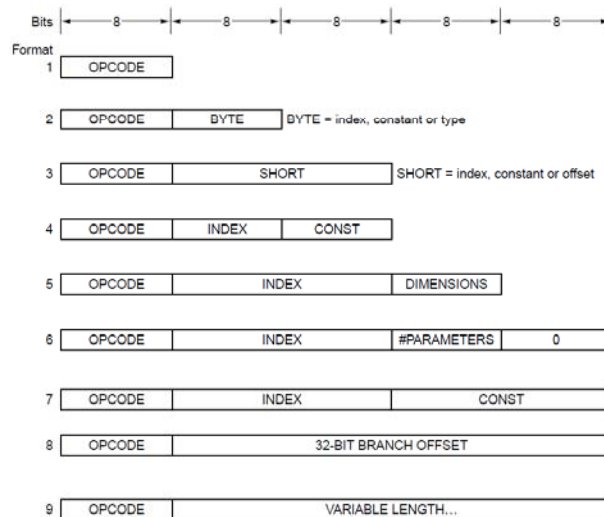


Επεκτεινόμενοι Κωδικοί Πράξης 3/3

- Οι επεκτεινόμενοι κωδικοί πράξης συνήθως χρησιμοποιούνται με τους ακόλουθους δύο τρόπους:
 - Οι εντολές διατηρούνται όλες στο ίδιο μέγεθος και εκχωρούνται μικρότεροι κωδικοί πράξης στις εντολές που απαιτούν τα περισσότερα bits για τον προσδιορισμό άλλων πληροφοριών. Ευνοεί την στοίχιση των εντολών στη μνήμη.
 - Ελαχιστοποιείται το μέσο μέγεθος των εντολών με την επιλογή μικρότερων κωδικών πράξης για τις πιο συνηθισμένες εντολές και μεγαλύτερων κωδικών πράξης για τις πιο σπάνιες εντολές.



Μορφές Εντολών της JVM



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

19

ΔΙΕΥΘΥΝΣΙΟΔΟΤΗΣΗ



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

20

Διευθυνσιοδότηση

- Σημαντικό αντικείμενο στο επίπεδο της αρχιτεκτονικής ISA είναι ο προσεκτικός σχεδιασμός των κωδικών πράξης.
- Όμως το μεγαλύτερο μέρος των bits μιας εντολής (και συνεπώς των bits ενός προγράμματος) χρησιμοποιείται για τον καθορισμό της προέλευσης και του προορισμού των τελεστών (*διευθυνσιοδότηση – addressing*).
 - Έστω εντολή ADD με τρεις τελεστέους, 2 πηγές και 1 προορισμός.
 - Αν οι διευθύνσεις μνήμης είναι των 32-bits, η πιο απλή μορφή αυτής της εντολής θα ήταν με τη χρήση 3 πεδίων διευθύνσεων των 32-bits, δηλ. συνολικά 96-bits!
- Δύο μέθοδοι για να μειωθεί ο # bits για τον προσδιορισμό των τελεστών:
 - Χρήση καταχωρητή για τελεστέους που χρησιμοποιούνται συχνά
 - ο π.χ. αν $\exists$ 32 καταχωρητές απαιτούνται 5 bits για τον προσδιορισμό τους, άρα η ADD θα απαιτούσε 15 bits για τους 3 τελεστέους. Όμως απαιτούνται 2 εντολές LOAD και μια STORE με δ/σεις των 32 bit για τη μεταφορά των τελεστών.
 - Καθορισμός των τελεστών με υπονοούμενο τρόπο
 - ο αρχική ADD 3 δ/σεων: $\text{ΠΡΟΟΡΙΣΜΟΣ} = \text{ΠΗΓΗ1} + \text{ΠΗΓΗ2}$
 - ο νέα ADD 2 δ/σεων: $\text{ΚΑΤΑΧΩΡΗΤΗΣ2} = \text{ΚΑΤΑΧΩΡΗΤΗΣ2} + \text{ΠΗΓΗ1}$
 - ο χρήση συσσωρευτή οπότε η ADD απαιτεί μόνο μία δ/ση: $\text{ACC} = \text{ACC} + \text{ΠΗΓΗ}$
 - ο χρήση στοίβας οπότε δεν απαιτείται καμία δ/ση.



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

21

Τρόποι Διευθυνσιοδότησης

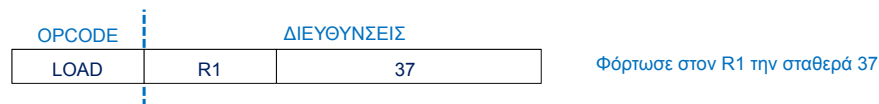
- Υπάρχει μια πλειάδα τρόπων για τον προσδιορισμό των τελεστών:
 - Διευθυνσιοδότηση με άμεσο καθορισμό τιμής (immediate addressing)
 - Άμεση διευθυνσιοδότηση (direct addressing)
 - Διευθυνσιοδότηση μέσω καταχωρητή (register addressing/mode)
 - Έμμεση διευθυνσιοδότηση μέσω καταχωρητή (register indirect mode)
 - Διευθυνσιοδότηση με δείκτη θέσης (indexed addressing)
 - Διευθυνσιοδότηση με βάση και δείκτη θέσης (based-indexed addressing)
 - Διευθυνσιοδότηση μέσω στοίβας (stack-based addressing)
- Ορθογωνιότητα των κωδικών πράξης και των τρόπων διευθ/τησης:
 - Όλοι οι κωδικοί πράξης πρέπει να επιτρέπουν όλους τους τρόπους διευθ/τησης (όπου αυτό έχει νόημα) και
 - όλοι οι καταχωρητές θα πρέπει να είναι διαθέσιμοι σε όλους τους τρόπους διευθ/τησης μέσω καταχωρητή.



22

Διευθ/τηση με Άμεσο Καθορισμό Τιμής

- Η *διευθυνσιοδότηση με άμεσο καθορισμό τιμής (immediate addressing)* είναι ο απλούστερος τρόπος διευθ/τησης για να καθοριστεί ένας τελεστέος. Το τμήμα δ/σης της εντολής περιέχει τον ίδιο τον τελεστέο και όχι κάποια δ/ση ή άλλη πληροφορία για τον προσδιορισμό του.
 - Πλεονέκτημα: δεν απαιτεί επιπλέον αναφορά στη μνήμη για την προσκόμιση του τελεστέου.
 - Μειονέκτημα: με αυτόν τον τρόπο παρέχονται μόνο σταθερές.



- Παράδειγμα JVM:

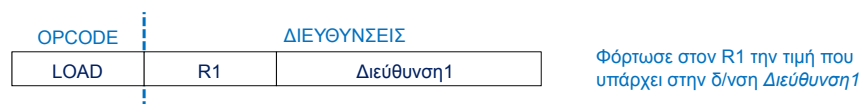


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

23

Άμεση Διευθυνσιοδότηση

- Με την *άμεση διευθυνσιοδότηση (direct addressing)* για τον καθορισμό του τελεστέου χρησιμοποιείται η πλήρης διεύθυνσή του.
 - Περιορισμένη χρησιμότητα.
 - Προσπέλαση πάντα της ίδιας θέσης μνήμης. Προσπέλαση καθολικών μεταβλητών.



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

24

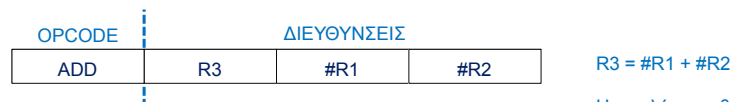
Διευθ/τηση μέσω Καταχωρητή

- Η *διευθυνσιοδότηση μέσω καταχωρητή (register addressing / mode)* είναι παρόμοια με την άμεση διευθ/τηση μόνο που καθορίζεται ένας καταχωρητής αντί για μια θέση μνήμης.
 - Είναι ο πιο συνηθισμένος τρόπος διευθ/τησης.



Έμμεση Διευθ/τηση μέσω Καταχωρητή

- Η *έμμεση διευθυνσιοδότηση μέσω καταχωρητή (register indirect addressing)* προσδιορίζει έναν καταχωρητή ο οποίος περιέχει τη διεύθυνση του τελούμενου στη μνήμη.
- Όταν μια δ/ση χρησιμοποιείται με αυτόν τον τρόπο καλείται *δείκτης (pointer)*.
 - Πλεονέκτημα η προσπέλαση της μνήμης χωρίς να υπάρχει η ανάγκη παρουσίας μιας πλήρους δ/σης μνήμης στην εντολή.



Η εντολή προσθέτει τα τελούμενα τα οποία προσκομίζονται από τις θέσεις μνήμης που περιέχονται στους καταχωρητές R1 και R2 και αποθηκεύει το αποτέλεσμα στον καταχωρητή R3.



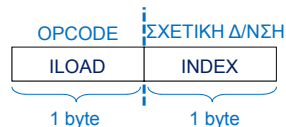
Διευθ/τηση με Δείκτη Θέσης

- Η *διευθυνσιοδότηση με δείκτη θέσης (indexed addressing)* πραγματοποιεί προσπελάσεις στην μνήμη με τον καθορισμό ενός καταχωρητή (με ρητό ή υπονοούμενο τρόπο) ο οποίος δείχνει σε μια θέση μνήμης αναφοράς και μιας *σχετικής διεύθυνσης (index)* ως προς τη δ/ση σε αυτόν τον καταχωρητή.
 - Κλασσικό παράδειγμα στην JVM η χρήση του καταχωρητή της βάσης της στοίβας (LV) και η σχετική διευθ/τηση ως προς αυτόν τον καταχωρητή.



Φόρτωσε στον R1 τη μεταβλητή που βρίσκεται στη διεύθυνση μνήμης $\#LV1 + INDEX$ (ο καταχωρητής αναφοράς LV1 ορίζεται ρητά).

- Παράδειγμα JVM:



Φόρτωσε στη κορυφή της στοίβας τη μεταβλητή που βρίσκεται στη διεύθυνση μνήμης $\#LV + INDEX$ (ο καταχωρητής LV υπονοείται όπως και η κορυφή της στοίβας που βρίσκεται στον SP).



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

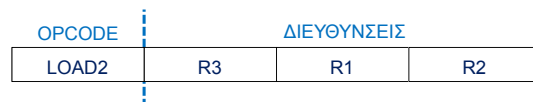
27

Διευθ/τηση με Βάση & Δείκτη Θέσης

- Η *διευθυνσιοδότηση με βάση και δείκτη θέσης (based-indexed addressing)* πραγματοποιεί προσπελάσεις στην μνήμη με πρόσθεση των περιεχομένων 2 καταχωρητών και μιας (προαιρετικής) *σχετικής διεύθυνσης (index)*.
- Ο ένας καταχωρητής είναι η βάση και ο άλλος ο δείκτης θέσης.



Φόρτωσε σε έναν υπονοούμενο καταχωρητή τη μεταβλητή που βρίσκεται στη διεύθυνση μνήμης $(R1 + R2 + INDEX)$, όπου ο R1 είναι η βάση και ο R2 είναι ο δείκτης.



Φόρτωσε στον R3 τη μεταβλητή που βρίσκεται στη διεύθυνση μνήμης $(R1 + R2)$, όπου ο R1 είναι η βάση και ο R2 είναι ο δείκτης.

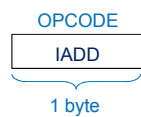


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

28

Διευθ/τηση μέσω Στοίβας

- Η *διευθυνσιοδότηση μέσω στοίβας (stack-based addressing)* επιτρέπει την πραγματοποίηση προσπελάσεων στην μνήμη χωρίς τη χρήση διευθύνσεων στις εντολές.
- Σε αυτήν την περίπτωση τα τελούμενα είναι στην κορυφή της στοίβας και η προσπέλασή τους γίνεται με τη χρήση του υπονοούμενου καταχωρητή δείκτη στοίβας (stack pointer) SP. Φυσικά απαιτείται πρότερη μεταφορά των τελούμενων στην κορυφή της στοίβας.
- Παράδειγμα IJVM:



Απόσυρε (pop) τις δύο μεταβλητές από την κορυφή της στοίβας που δείχνει ο SP. Ακολούθως, πρόσθεσε τις δύο μεταβλητές και απέθεσε το αποτέλεσμα στην κορυφή της στοίβας. Θυμηθείτε ότι ο καταχωρητής TOS περιέχει πάντα (προς διευκόλυνση) τα περιεχόμενα της κορυφής της στοίβας.



Τύποι Εντολών

- Υπάρχουν κάποιοι γενικοί τύποι εντολών στις υπάρχουσες μηχανές:
 - Εντολές μετακίνησης δεδομένων [ILOAD]
 - Εντολές διμελών πράξεων (συνδυάζουν 2 τελεστέους για να δώσουν ένα αποτέλεσμα) [IADD, IOR, ...]
 - Εντολές μονομελών πράξεων (επιδρούν πάνω σε έναν τελεστέο για να δώσουν ένα αποτέλεσμα) [IINC, DUP, εντολές ολίσθησης ή περιστροφής]
 - Εντολές σύγκρισης και διακλάδωσης υπό συνθήκη [IFEQ, IFLT]
 - Εντολές κλίσης διαδικασιών [INVOKEVIRTUAL, IRETURN]
 - Εντολές ελέγχου βρόχων [GOTO]
 - Εντολές εισόδου / εξόδου



ΡΟΗ ΕΛΕΓΧΟΥ

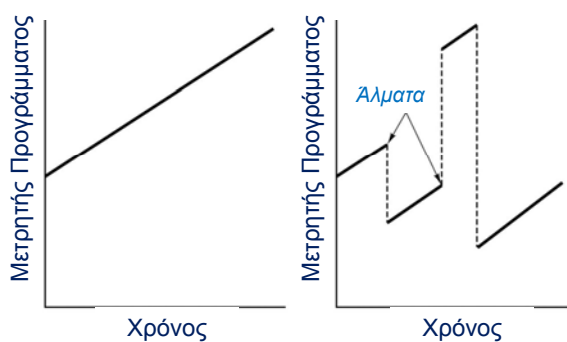


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Σειριακή Ροή & Διακλαδώσεις

- Ως ροή ελέγχου (flow control) ορίζουμε τη σειρά με την οποία εκτελούνται οι εντολές σε ένα πρόγραμμα.
- Η ροή ελέγχου είναι συνήθως σειριακή (οι διαδοχικά εκτελούμενες εντολές προσκομίζονται από συνεχόμενες θέσεις μνήμης).
- Οι διακλαδώσεις, οι κλήσεις διαδικασιών, οι διακοπές και οι παγίδες προκαλούν αλλοίωση στη σειριακή ροή του ελέγχου.

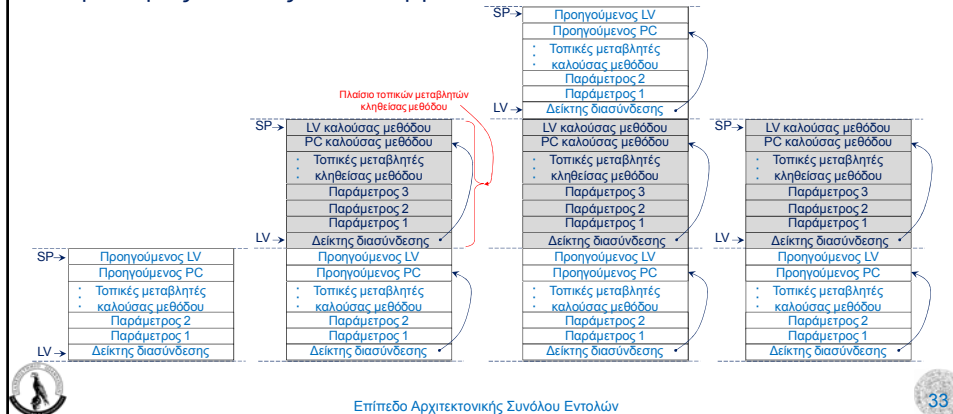


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Διαδικασίες 1/2

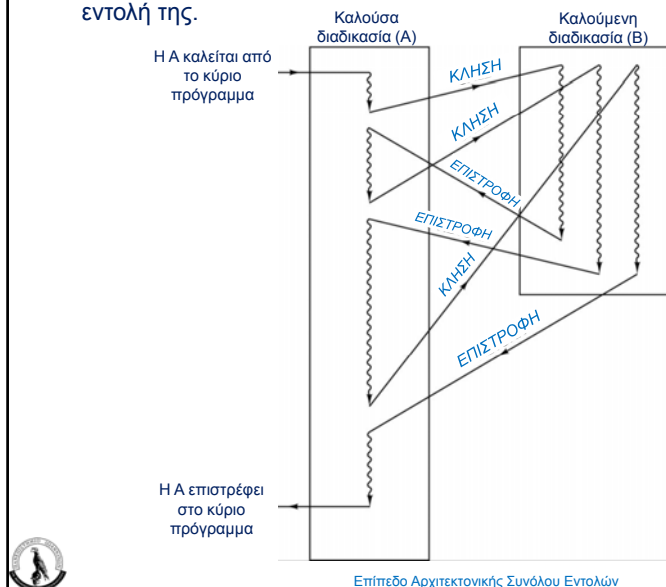
- Τα προγράμματα δομούνται με τη χρήση *διαδικασιών (procedures)*.
- Οι διαδικασίες αλλάζουν τη ροή του ελέγχου όπως ακριβώς και οι διακλαδώσεις αλλά αντίθετα με αυτές όταν μια διαδικασία ολοκληρωθεί επιστρέφει τον έλεγχο πάντα στην επόμενη εντολή που ακολουθεί μετά την κλήση τους.
- Ιδιαίτερο είδος διαδικασιών είναι οι *αναδρομικές διαδικασίες (recursive procedures)*, όπου μια διαδικασία καλεί τον εαυτό της είτε άμεσα, είτε έμμεσα μέσω μιας αλυσίδας άλλων διεργασιών.



33

Διαδικασίες 2/2

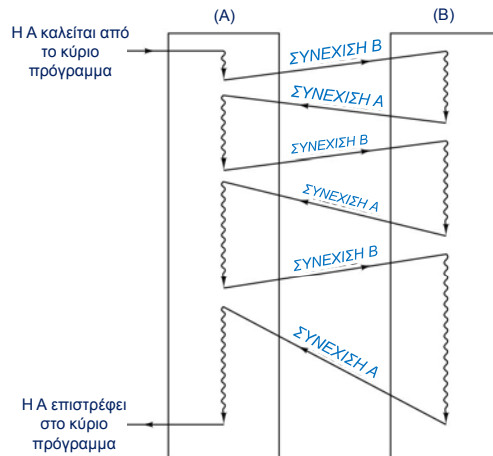
- Όταν καλείται μια διαδικασία, η εκτέλεσή της αρχίζει πάντα από την πρώτη εντολή της.



34

Συρρουτίνες

- Δύο διαδικασίες που καλούν η μία την άλλη με συνέχιση της εκτέλεσης από το σημείο που διακόπηκε καλούνται *συρρουτίνες (co-routines)*.



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Μηχανισμοί Εισόδου/Εξόδου

- Στους περισσότερους υπολογιστές χρησιμοποιούνται συνήθως τρεις διαφορετικοί μηχανισμοί εισόδου/εξόδου:
 - Προγραμματισμένη είσοδος/έξοδος με “αναμονή με απασχόληση” (busy waiting)
 - Η CPU μπαίνει σε ένα βρόχο αναμένοντας την I/O συσκευή να ετοιμαστεί.
 - Είσοδος/έξοδος οδηγούμενη από διακοπές (interrupt driven)
 - Η CPU ξεκινάει την επικοινωνία με μια συσκευή I/O και αναμένει μια διακοπή όταν αυτή είναι έτοιμη.
 - Είσοδος/έξοδος με άμεση προσπέλαση μνήμης (direct memory access - DMA)
 - Αναθέτουμε την προγραμματισμένη είσοδο/έξοδο σε μια αφιερωμένη για αυτό το σκοπό συσκευή όπως ο ελεγκτής άμεσης προσπέλασης μνήμης.

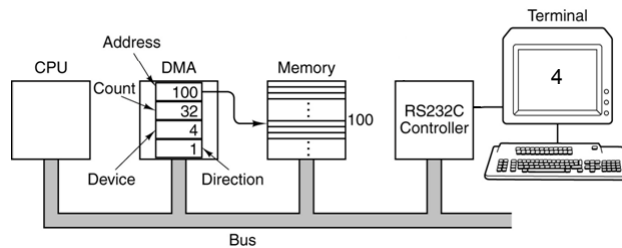


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών



Άμεση Προσπέλαση Μνήμης

- Ο *ελεγκτής άμεσης προσπέλασης μνήμης* (direct access memory – DMA controller), είναι υλικό (ολοκληρωμένο κύκλωμα ή ενσωματωμένο κύκλωμα στη CPU) που αναλαμβάνει την προγραμματισμένη είσοδο/έξοδο και το οποίο έχει άμεση πρόσβαση στο δίαυλο.
- Ο DMA έχει υψηλότερη προτεραιότητα διαύλου από τη CPU, με αποτέλεσμα να αφαιρεί κύκλους διαύλου από τη CPU (*υπεξαίρεση κύκλων – cycle stealing*).
- Ο DMA περιέχει τουλάχιστον 4 καταχωρητές, οι οποίοι γράφονται από τη CPU.
- Οι DMA ελεγκτές περιέχουν περισσότερες από μία τετράδες (σύνολα) καταχωρητών για να ελέγχουν πολλές ταυτόχρονες μεταφορές.

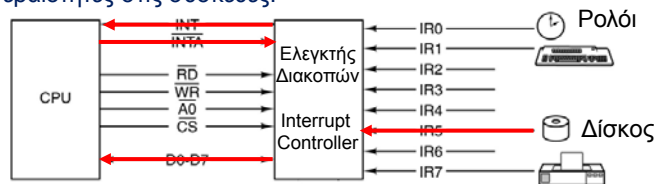


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

37

Διακοπές 1/3

- Η CPU αναθέτει διεργασίες σε συσκευές και αναμένει μια διακοπή όταν ολοκληρωθεί η διεργασία.
- Για τη διαχείριση πολλαπλών διακοπών υπάρχει ένας ελεγκτής διακοπών που δίνει προτεραιότητες στις συσκευές.



- Η συσκευή αιτείται διακοπή.
- Ο ελεγκτής ενεργοποιεί το σήμα *INT* για να ειδοποιήσει τη CPU.
- Όταν η CPU είναι έτοιμη επιβεβαιώνει τη λήψη της διακοπής ενεργοποιώντας το σήμα *INTA*.
- Ο ελεγκτής τοποθετεί στο δίαυλο δεδομένων τον κωδικό αναγνώρισης της συσκευής (*διάνυσμα διακοπής – interrupt vector*).
- Η CPU αποθέτει στη στοίβα το μετρητή προγράμματος και με το διάνυσμα διακοπής ως δείκτη εντοπίζει ένα νέο μετρητή προγράμματος που δείχνει την ρουτίνα εξυπηρέτησης της διακοπής που αντιστοιχεί στη συγκεκριμένη συσκευή.



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

38

Διακοπές 2/3

- Η ρουτίνα εξυπηρέτησης της διακοπής εκτελεί τις ακόλουθες ενέργειες:
 - Αποθήκευση των καταχωρητών (στη στοίβα ή σε πίνακα) ώστε να μπορούν να αποκατασταθούν αργότερα.
 - Επειδή το δίδυμο διακοπής αντιστοιχεί σε όλες τις συσκευές ίδιου τύπου, απαιτείται η ανάγνωση κάποιου καταχωρητή συσκευής (ενδεχόμενα στον σχετικό ελεγκτή) για τον απόλυτο προσδιορισμό της συσκευής.
 - Ολοκληρώνεται η ρουτίνα εξυπηρέτησης.
 - Στέλνεται από τη CPU, προς τη συσκευή ή τον ελεγκτή, ειδικός κωδικός που ενημερώνει για τη διεκπεραίωση της διακοπής.
 - Αποκαθίστανται όλοι οι αποθηκευμένοι καταχωρητές και ο μετρητής προγράμματος.
 - Εκτελείται μια εντολή RETURN FROM INTERRUPT που επαναφέρει τη CPU στον τρόπο λειτουργίας και την κατάσταση που ήταν πριν τη διακοπή. Ο υπολογιστής συνεχίζει τη λειτουργία από το σημείο που είχε μείνει.
- Διακοπές και *διαφάνεια* (transparency): Μετά την εξυπηρέτηση μιας διακοπής ο υπολογιστής επανέρχεται στην ίδια ακριβώς κατάσταση που ήταν πριν από τη διακοπή.

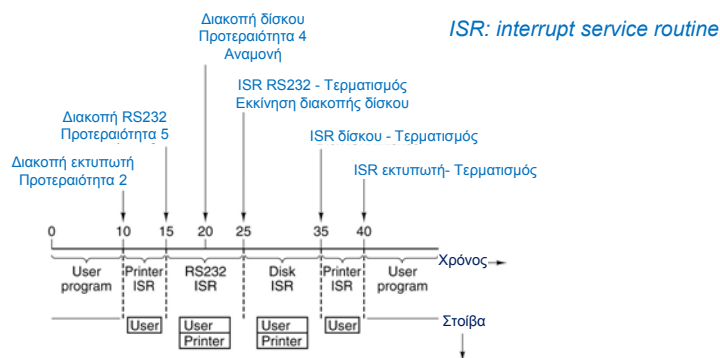


Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

39

Διακοπές 3/3

- Οι διακοπές είναι *ασύγχρονες*. Συνεπώς οι ρουτίνες εξυπηρέτησης διακοπών μπορούν και αυτές να διακόπτονται από άλλες διακοπές.
- Ο καλύτερος τρόπος για να διατηρηθεί απλή η διαχείριση των διακοπών, είναι να εξασφαλιστεί ότι όλες οι διακοπές θα είναι διαφανείς.
- Συνήθως, σε κάθε συσκευή ή τύπο συσκευής I/O εκχωρείται μια *προτεραιότητα*, υψηλή για κρίσιμες συσκευές και χαμηλή για συσκευές μικρότερης κρίσιμότητας.



Επίπεδο Αρχιτεκτονικής Συνόλου Εντολών

40

Παγίδες

- Κατά την εκτέλεση ενός προγράμματος εμφανίζονται *παγίδες (traps)* δηλ. αυτόματες κλήσεις διαδικασιών οι οποίες προκαλούνται από κάποια συνθήκη (συνήθως μια σημαντική αλλά σπάνια συνθήκη, π.χ. υπερχείλιση).
- Οι παγίδες είναι *σύγχρονες*. Όσες φορές και αν εκτελεστεί το πρόγραμμα, με τα ίδια δεδομένα εισόδου, οι παγίδες θα εμφανίζονται πάντα στο ίδιο σημείο του προγράμματος.
- Στην περίπτωση παγίδας, η ροή ελέγχου μεταφέρεται σε καθορισμένη θέση μνήμης όπου υπάρχει μια διακλάδωση προς μια διαδικασία που καλείται *χειριστής παγίδων (trap handler)*.
- Συνθήκες εμφάνισης παγίδας:
 - υπερχείλιση ή ανεπάρκεια αριθμών κινητής υποδιαστολής
 - υπερχείλιση ακέραιων
 - υπερχείλιση στοίβας
 - ο μη ορισμένος κωδικός πράξης
 - η απόπειρα χρήσης ανύπαρκτης συσκευής
 - η διαίρεση με το μηδέν ...

