

# Αρχιτεκτονική και σύνολο εντολών



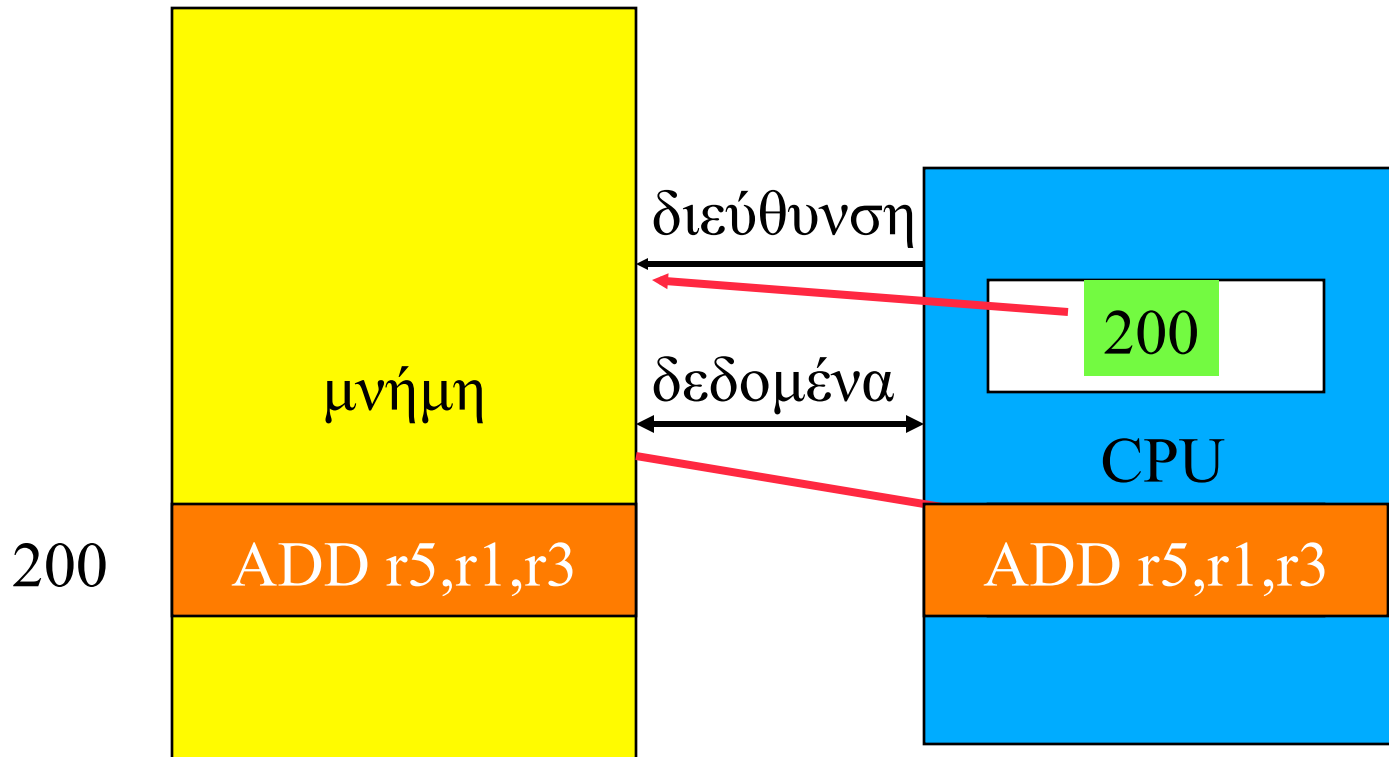
- Ταξινόμηση υπολογιστικών αρχιτεκτονικών.
- Συμβολική (assembly) γλώσσα.

# Αρχιτεκτονική von Neumann

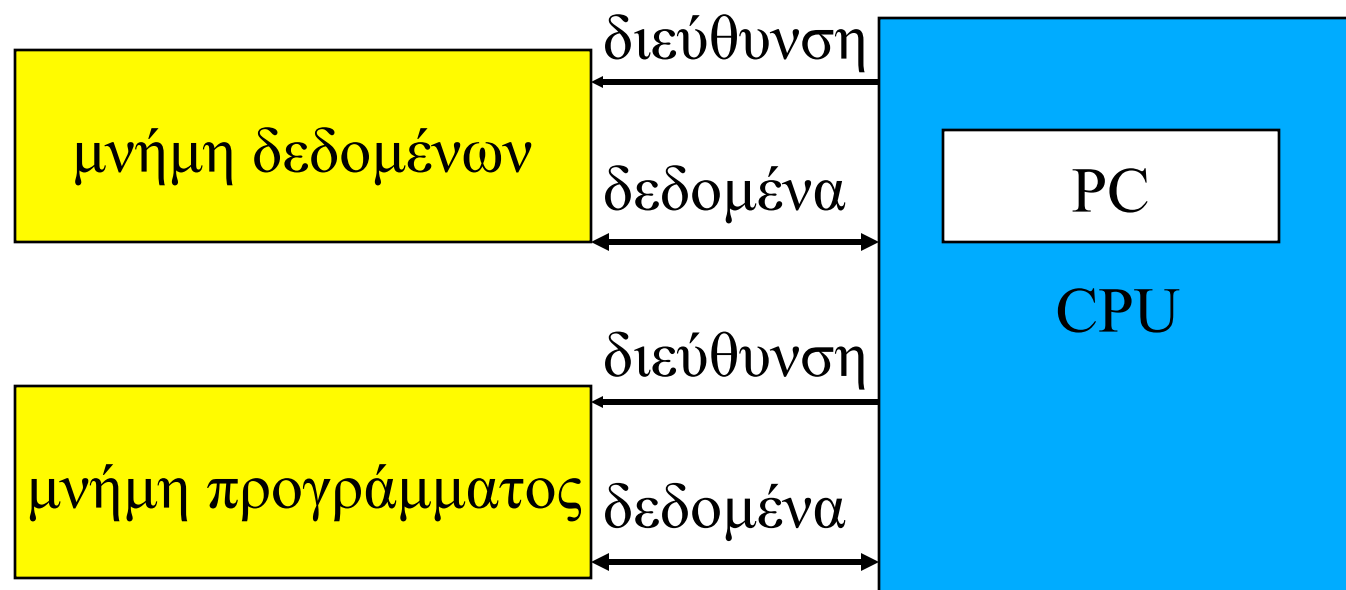


- Η μνήμη κρατά δεδομένα και εντολές.
- Η κεντρική μονάδα επεξεργασίας (CPU) εξάγει εντολές από τη μνήμη.
  - Ο διαχωρισμός της CPU και της μνήμης χαρακτηρίζει τον προγραμματιζόμενο υπολογιστή.
- CPU καταχωρητές: μετρητής προγράμματος (PC), καταχωρητής εντολών (IR), καταχωρητές γενικού σκοπού, κτλ.

# CPU + μνήμη



# Αρχιτεκτονική Harvard



# von Neumann vs. Harvard



- Η αρχιτεκτονική Harvard δεν μπορεί να χρησιμοποιήσει αυτό-μεταβαλλόμενο κώδικα.
- Η αρχιτεκτονική Harvard επιτρέπει δύο ταυτόχρονες εξαγωγές από τη μνήμη.
- Οι περισσότεροι DSP επεξεργαστές χρησιμοποιούν αρχιτεκτονική Harvard για ρέοντα δεδομένα:
  - Μεγαλύτερο bandwidth(εύρος ζώνης) μνήμης.
  - Περισσότερο προβλέψιμο bandwidth.

# RISC vs. CISC



- Υπολογιστής πολύπλοκου συνόλου εντολών (**CISC**):
  - Πολλές μορφές διευθυνσιοδότησης (addressing modes).
  - Πολλές λειτουργίες.
- Υπολογιστής μειωμένου συνόλου εντολών (**RISC**):
  - φόρτωση/αποθήκευση.
  - Διοχετεύσιμες (Pipelined) εντολές.

# Χαρακτηριστικά του συνόλου εντολών



- Σταθερό και μεταβαλλόμενο μήκος.
- Μορφές διευθυνσιοδότησης.
- Αριθμός τελεστών (operands).
- Τύποι τελεστών.

# Μοντέλο προγραμματισμού



- **Μοντέλο προγραμματισμού**: καταχωρητές ορατοί στον προγραμματιστή.
- Κάποιοι καταχωρητές δεν είναι ορατοί (IR).



# Πολλαπλές υλοποιήσεις



- Οι επιτυχημένες αρχιτεκτονικές έχουν πολλαπλές υλοποιήσεις:
  - Μεταβαλλόμενες ταχύτητες ρολογιού.
  - Διαφορετικά εύρη διαύλου.
  - Διαφορετικά μεγέθη κρυφής μνήμης.
  - κτλ.

# Γλώσσα assembly



- Εντολές μία προς μία (λιγότερο ή περισσότερο).
- Βασικά χαρακτηριστικά:
  - Μια εντολή ανά γραμμή.
  - Οι ετικέτες παρέχουν ονόματα σε διευθύνσεις (συνήθως στην πρώτη στήλη).
  - Οι εντολές ξεκινούν συνήθως σε επόμενες στήλες.
  - Οι στήλες εκτείνονται ως το τέλος της γραμμής.

# Παράδειγμα γλώσσας assembly για ARM

επικέτα1 ADR r4,c  
LDR r0,[r4] ; σχόλιο  
ADR r4,d  
LDR r1,[r4]  
SUB r0,r0,r1 ; σχόλιο

↑  
προορισμός

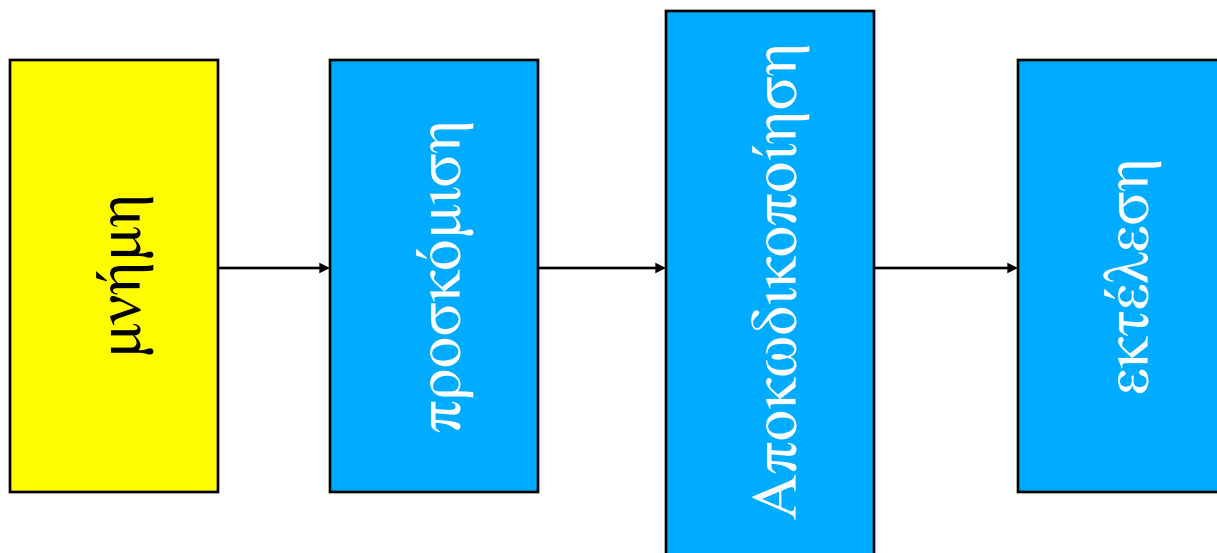
# Ψευδολειτουργίες (pseudops)



- Κάποιες εντολές στον συμβολομεταφραστή δεν αντιστοιχούν κατ' ευθείαν σε πραγματικές εντολές:
  - Καθορίζουν την παρούσα διέυθυνση.
  - Κατοχυρώνουν τη θέση αποθήκευσης.
  - Σταθερές.

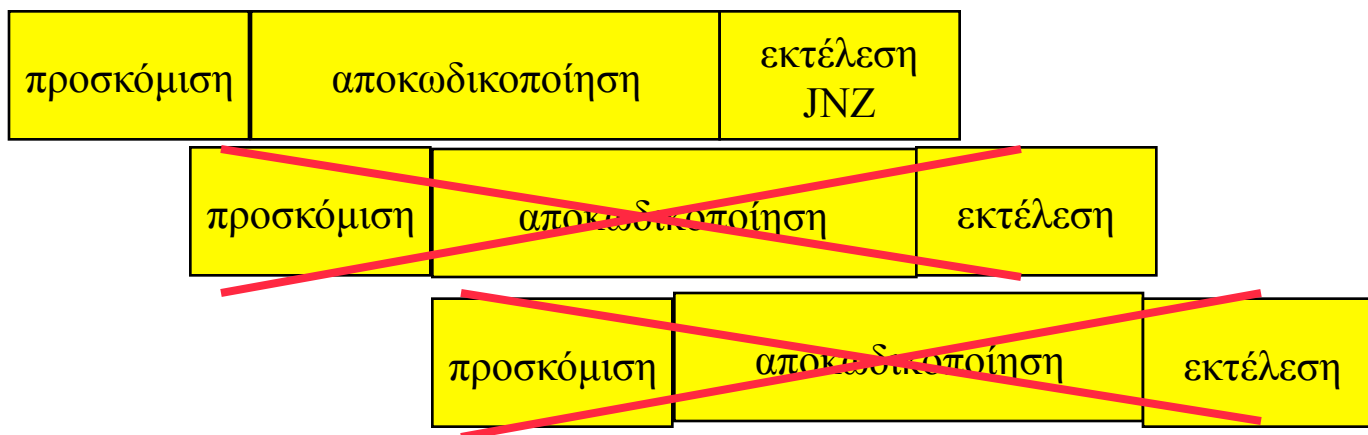
# Διασωλήνωση (Pipelining)

- Ταυτόχρονη εκτέλεση πολλαπλών εντολών αλλά σε διαφορετικές βαθμίδες.
- Απλή διοχέτευση τριών βαθμίδων:



# Επιπλοκές διοχέτευσης

- Δεν είναι πάντα ικανή να προβλέψει την επόμενη εντολή:
  - Διακλάδωση υπό συνθήκη.
- Προκαλεί bubble στη διοχέτευση:



# Υπερβάθμωση



- Η RISC διοχέτευση εκτελεί (συνήθως) μια εντολή ανά κύκλο ρολογιού.
- Οι μηχανές υπερβάθμωσης εκτελούν πολλές εντολές ανά κύκλο ρολογιού.
  - Γρηγορότερη εκτέλεση.
  - Μεγαλύτερη διακύμανση στους χρόνους εκτέλεσης.
  - Πιο ακριβές CPU.

# Απλή υπερβάθμωση



- Εκτελεί ταυτόχρονα μια εντολή κινητής υποδιαστολής και μια ακέραια εντολή.
  - Χρήση διαφορετικών καταχωρητών.
  - Οι λειτουργίες κινητής υποδιαστολής χρησιμοποιούν δικιά τους μονάδα υλικού.
- Αναμένει όταν οι μονάδα κινητής υποδιαστολής και η ακέραια μονάδα επικοινωνούν μεταξύ τους.



# Κόστος



- Τα καλά νέα: μπορεί να βρεθεί παραλληλία κατά το χρόνο εκτέλεσης.
  - Τα κακά νέα: προκαλείται διακύμανση στους χρόνους εκτέλεσης.
- Απαιτεί πολύ υλικό.
  - $n^2$  υλικό για τη μονάδα εντολών για παραλληλοποίηση  $n$  εντολών.

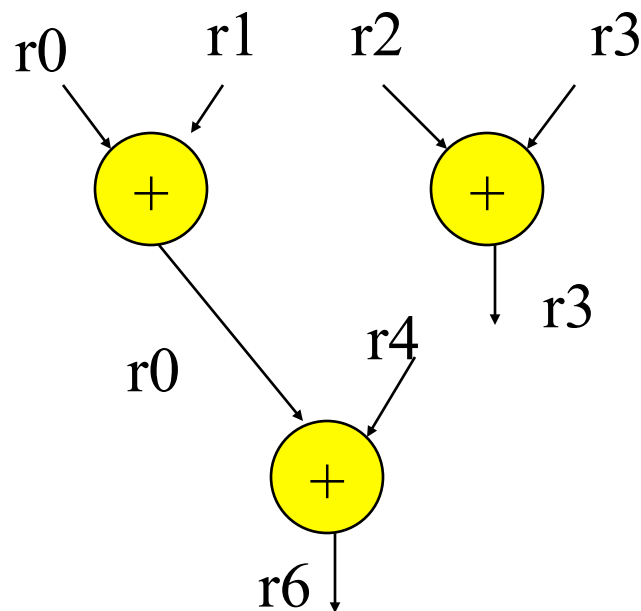
# Βρίσκοντας παραλληλισμό

- Ανεξάρτητες εντολές μπορούν να εκτελεστούν παράλληλα:

ADD r0, r0, r1

ADD r3, r2, r3

ADD r6, r4, r0



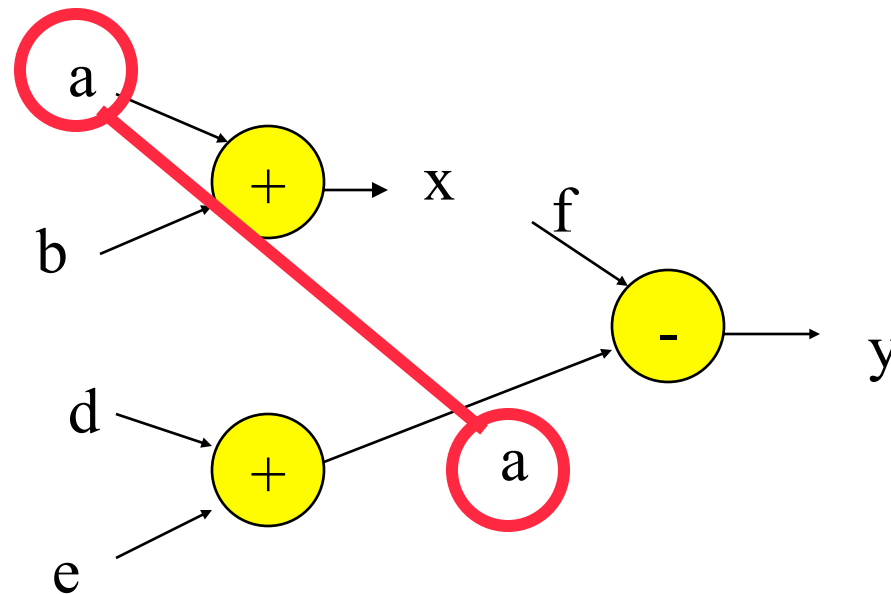
# Επικινδυνότητα της διοχέτευσης

- Δύο λειτουργίες που απαιτούν τον ίδιο πόρο δεν μπορούν να εκτελεστούν παράλληλα:

$$x = a + b.$$

$$a = d + e.$$

$$y = a - f.$$



# Πινακοποίηση (Scoreboarding)

- Σε έναν πίνακα καταγράφονται οι πόροι που χρησιμοποιεί κάθε εντολή:

	Reg file	ALU	FP
instr1	X	X	
instr2			X

# Σειρά εκτέλεσης



- Σε σειρά (in-order):

- Η μηχανή σταματά να εκδίδει εντολές όταν η επόμενη εντολή δε μπορεί να εκτελεστεί.

- Εκτός σειράς (out-of-order):

- Η μηχανή θα αλλάξει τη σειρά των εντολών για να διατηρήσει την εκτέλεσή τους.
- Σημαντικά γρηγορότερη μέθοδος αλλά και πιο πολύπλοκη.

# Αρχιτεκτονικές VLIW



- Η επεξεργασία με πολύ μεγάλες λέξεις εντολών (VLIW) παρέχει αξιοσημείωτο παραλληλισμό.
- Βασίζεται στους μεταγλωττιστές για τον προσδιορισμό της παραλληλίας.

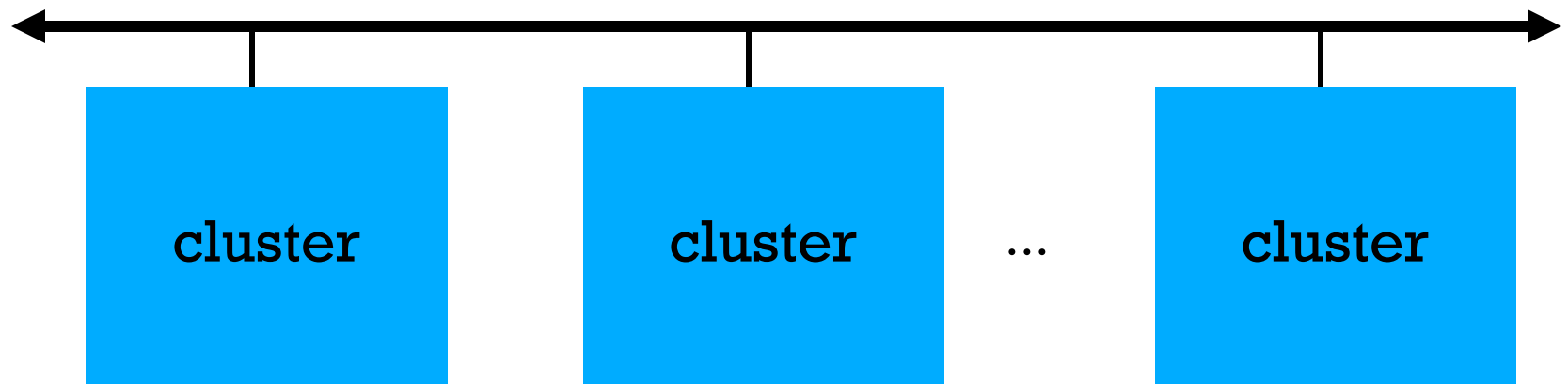
# Τι είναι η VLIW?

- Μονάδες παράλληλης λειτουργίας με κοινό (shared) αρχείο καταχωρητών:



# Συστοιχία (cluster) VLIW

- Οργανώνονται σε συστοιχίες για καλύτερη χρησιμοποίηση του εύρους των καταχωρητών (register bandwidth):



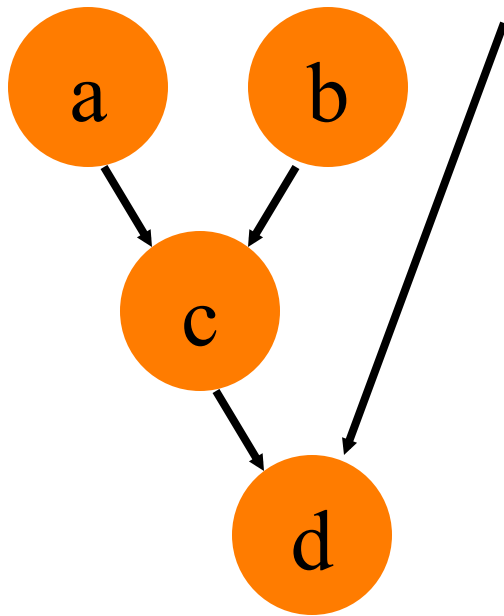


# VLIW και μεταγλωττιστές

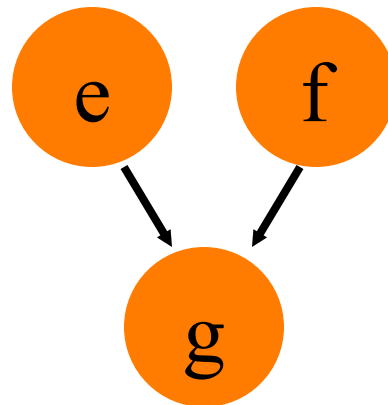


- Η VLIW απαιτεί σημαντικά πιο σύνθετη τεχνολογία μεταγλωττιστή σε σχέση με τις παραδοσιακές αρχιτεκτονικές: πρέπει να μπορεί να εξάγει τον παραλληλισμό για να διατηρήσει τις εντολές πλήρεις.
- Πολλές VLIWs έχουν καλή υποστήριξη ως προς το μεταγλωττιστή.

# Στατική χρονοδρομολόγηση



εκφράσεις



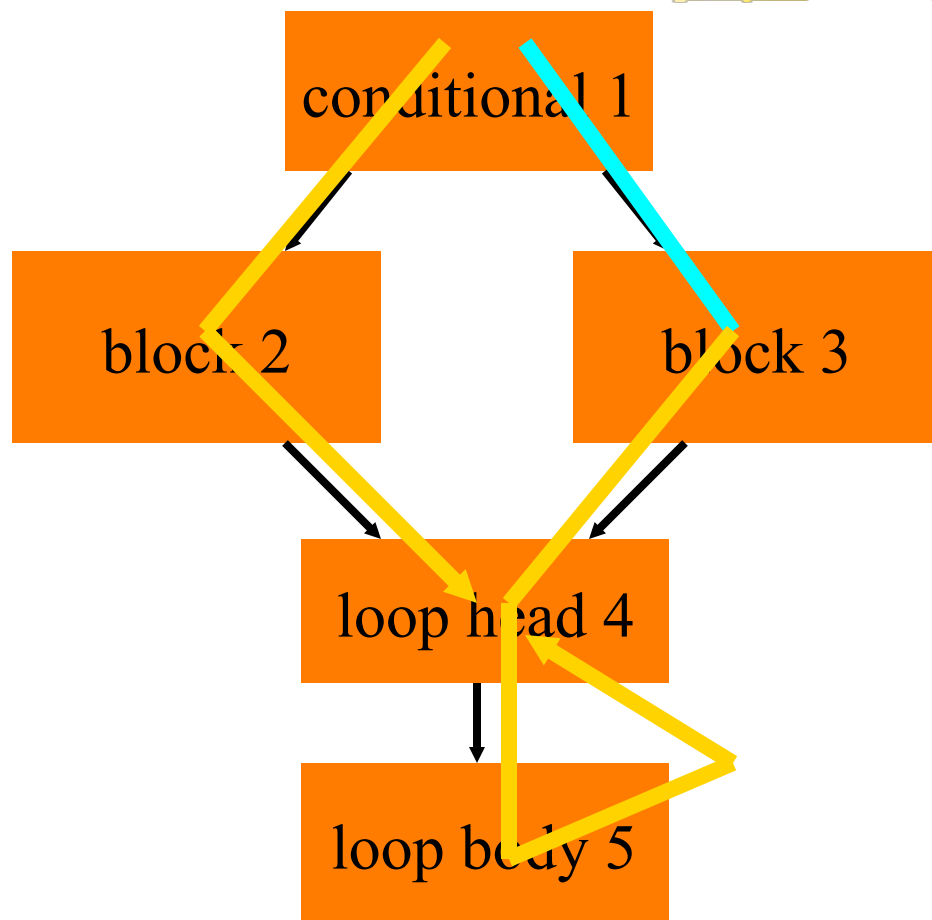
εντολές

a b e

f c nop

d g nop

# Χρονοδρομολόγηση ίχνους



Κατάταξη ίχνών  
ως προς τη συχνότητα.

Χρονοδρομολόγηση  
ίχνών ως προς τη  
συχνότητα.

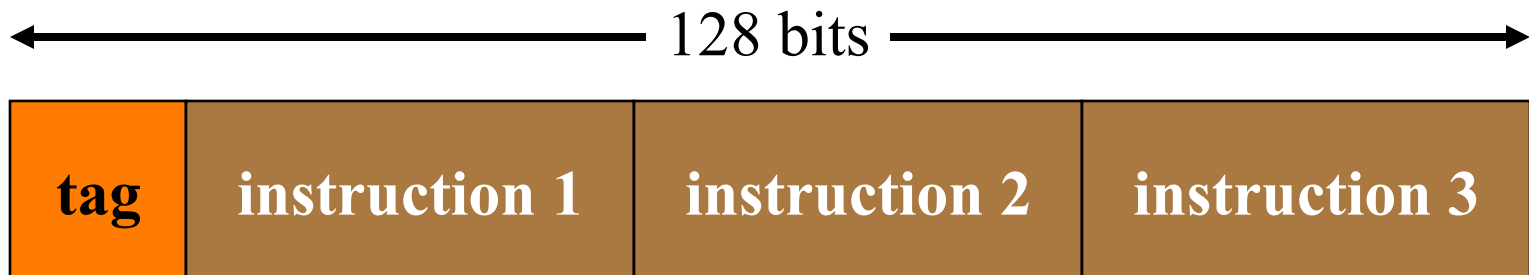
# EPIC



- Αποκλειστικά παράλληλος υπολογισμός εντολών (EPIC = Explicitly parallel instruction computing).
- Χρησιμοποιείται στην Intel/HP Merced (IA-64) μηχανή.
- Ενσωματώνει πολλά χαρακτηριστικά που επιτρέπουν στη μηχανή να βρίσκει και να εκμεταλλεύεται υψηλό παραλληλισμό.

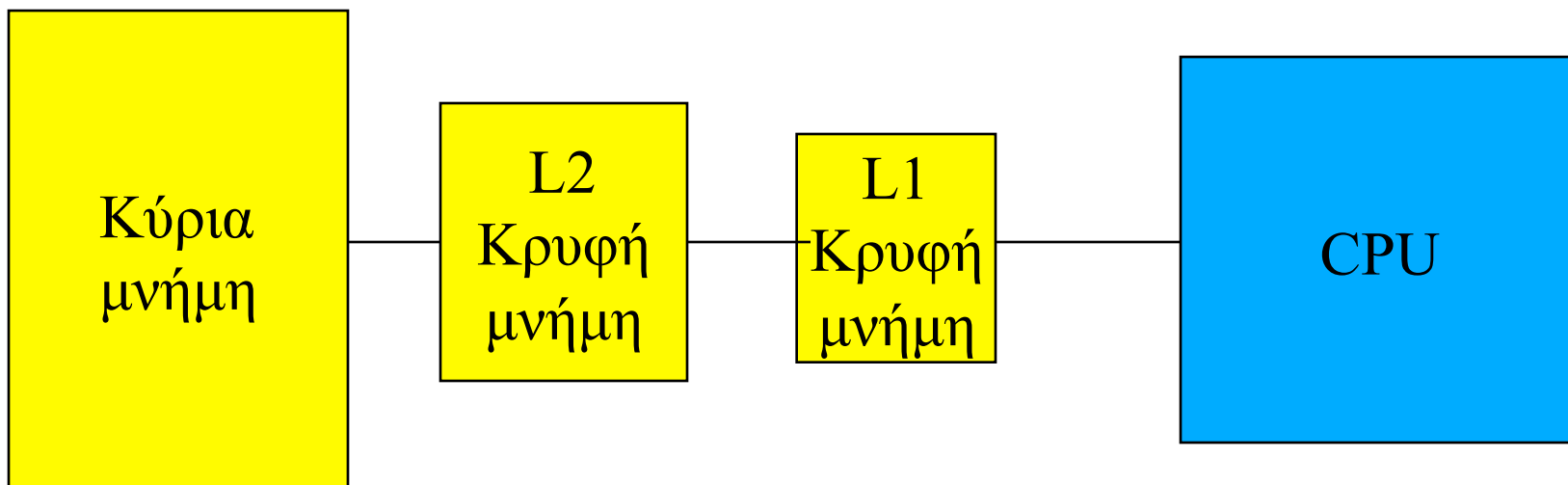
# Ο τύπος της εντολής για IA-64

- Στην εντολή προστίθεται ετικέτα (tag) που δείχνει τις εντολές που μπορούν να εκτελεστούν παράλληλα:



# Σύστημα μνήμης

- Η CPU εξάγει δεδομένα και εντολές από μια ιεραρχημένη μνήμη:



# Επιπλοκές της ιεραρχημένης μνήμης

- Η συμπεριφορά του προγράμματος είναι πολύ περισσότερο εξαρτημένη από τις καταστάσεις.
  - Εξαρτάται από το πόσο νωρίτερα η εκτέλεση εγκατέλειψε την κρυφή μνήμη.
- Ο χρόνος εκτέλεσης είναι λιγότερο προβλέψιμος.
  - Οι χρόνοι προσπέλασης στη μνήμη μπορεί να ποικίλουν ακόμα και σε τάξη του 100X.