Υπερβαθμωτή Οργάνωση Υπολογιστών

Από τις βαθμωτές στις υπερβαθμωτές αρχιτεκτονικές αγωγού…

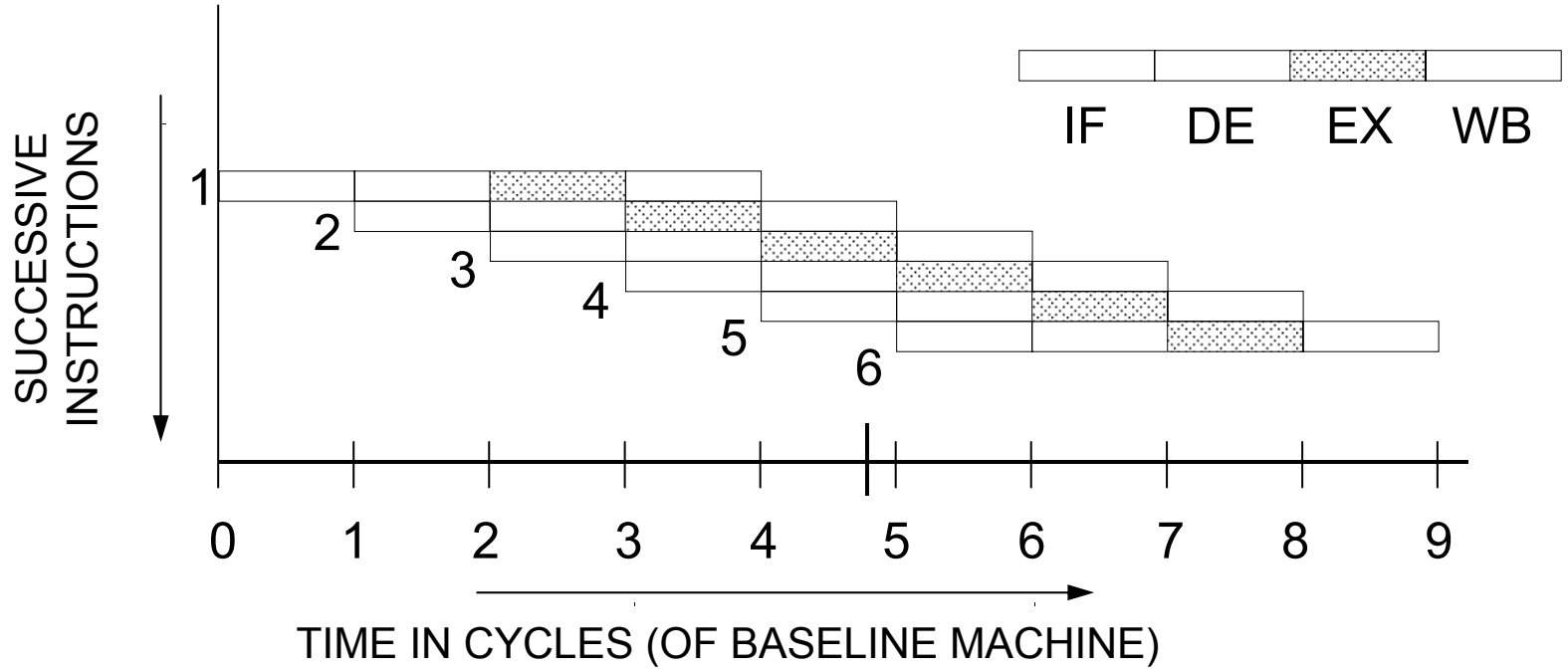
Τα όρια του Παραλληλισμού σε επίπεδο εντολών (Instruction Level Parallelism - ILP)

Weiss and Smith [1984]	1.58
Sohi and Vajapeyam [1987]	1.81
Tjaden and Flynn [1970]	1.86 (Flynn's bottleneck)
Tjaden and Flynn [1973]	1.96
Uht [1986]	2.00
Smith et al. [1989]	2.00
Jouppi and Wall [1988]	2.40
Johnson [1991]	2.50
Acosta et al. [1986]	2.79
Wedig [1982]	3.00
Butler et al. [1991]	5.8
Melvin and Patt [1991]	6
Wall [1991]	7 (Jouppi disagreed)
Kuck et al. [1972]	8
Riseman and Foster [1972]	51 (no control dependences)
Nicolau and Fisher [1984]	90 (Fisher's optimism)

Κατηγοριοποίηση των μηχανημάτων βάσει του ILP

[Jouppi, DECWRL 1991]

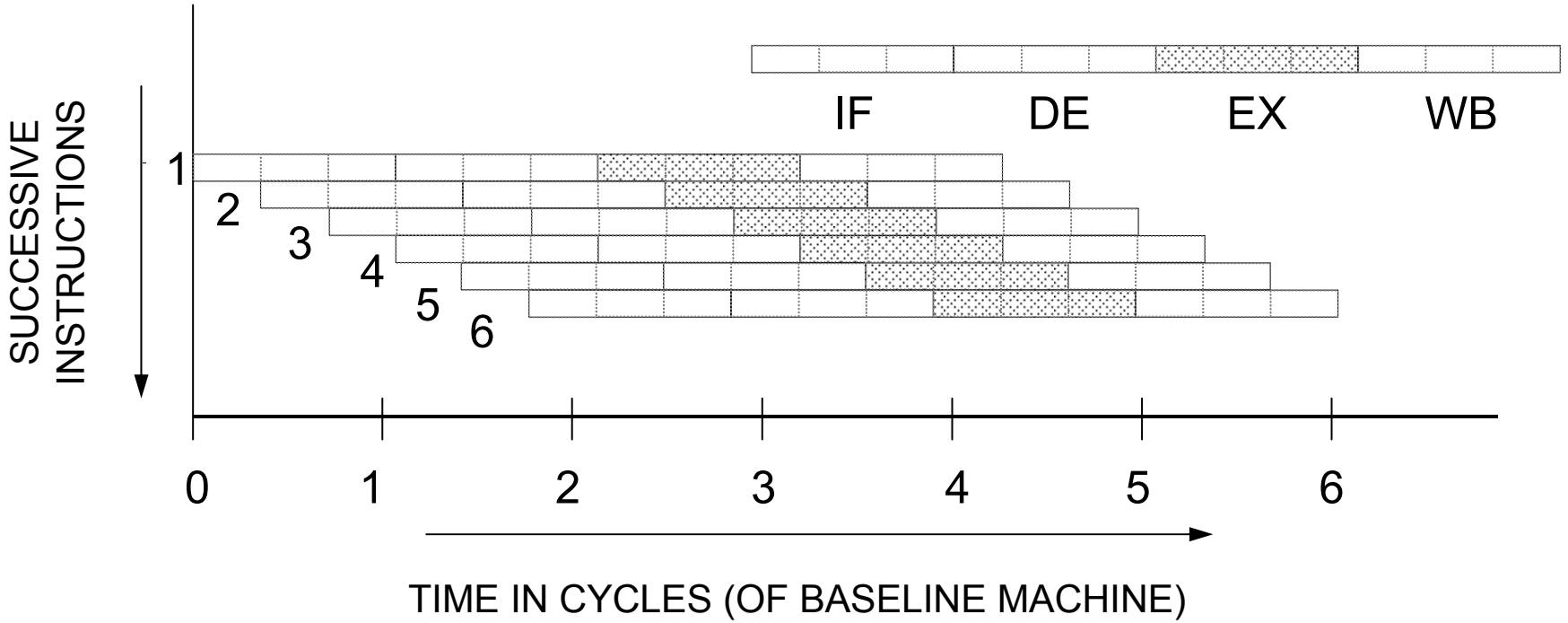
- Baseline **scalar** RISC
  - Παραλληλισμός διανομής (Issue parallelism) = IP = 1
  - Καθυστέρηση λειτουργίας (Operation latency) = OP = 1
  - Μέγιστο IPC = 1



Κατηγοριοποίηση των μηχανημάτων βάσει του ILP

[Jouppi, DECWRL 1991]

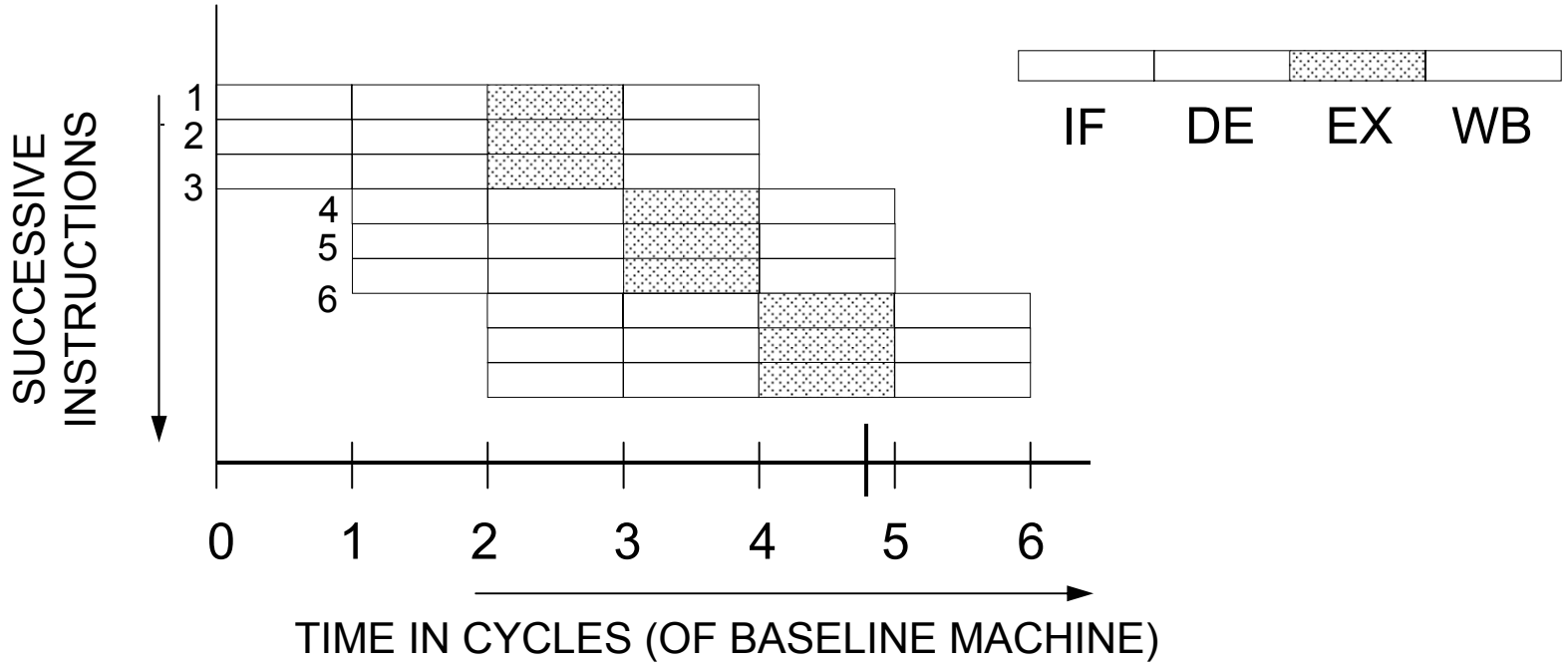
- **Superpipelined**: κύκλος ρολογιού = 1/m του baseline
  - Παραλληλισμός διανομής = IP = 1 εντολή / μικρο-κύκλο
  - Καθυστέρηση λειτουργίας = OP = m μικρο-κύκλοι
  - Μέγιστο IPC = m εντολές / βασικό κύκλο (m x speedup?)



Κατηγοριοποίηση των μηχανημάτων βάσει του ILP

[Jouppi, DECWRL 1991]

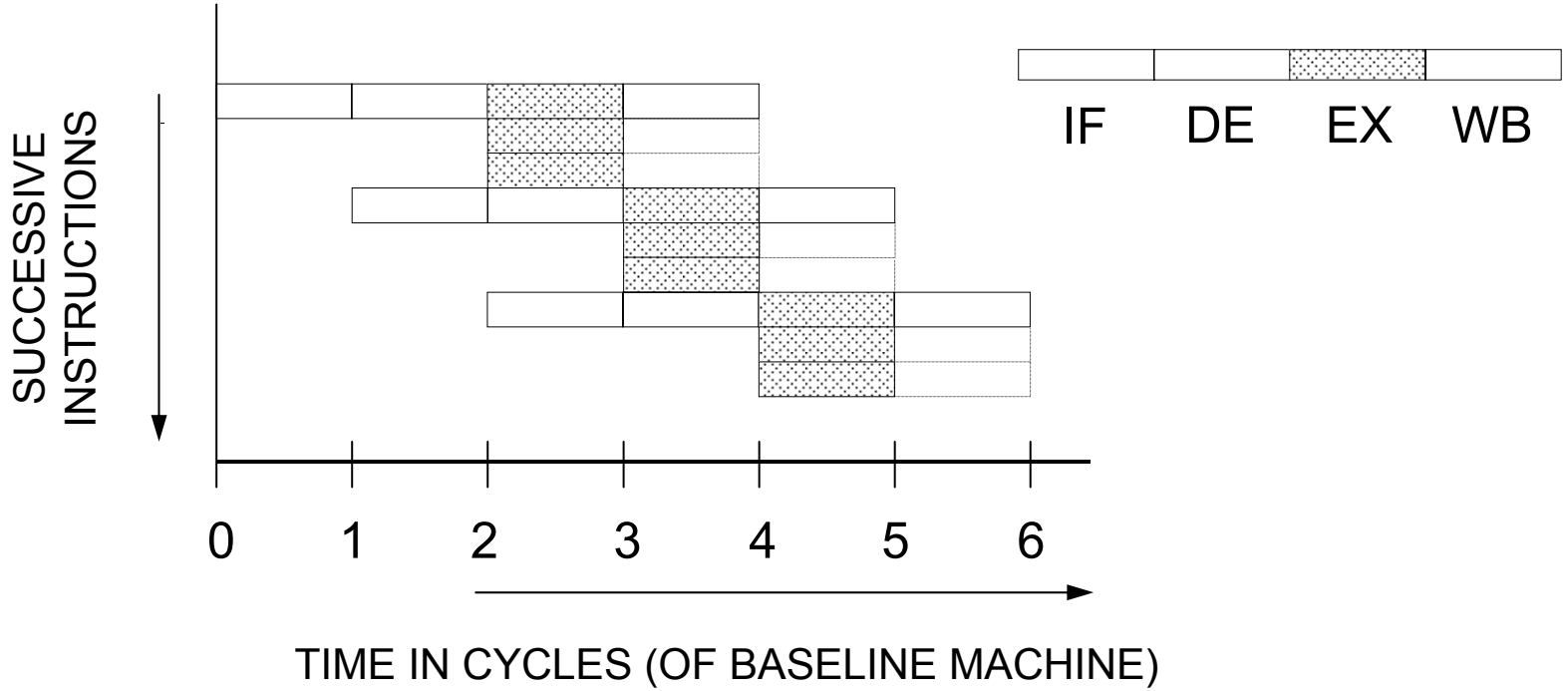
- **Superscalar**:
  - Παραλληλισμός διανομής = IP = n εντολές / κύκλο
  - Καθυστέρηση λειτουργίας = OP = 1 κύκλος
  - Μέγιστο IPC = n εντολές / κύκλο (n x speedup?)



Κατηγοριοποίηση των μηχανημάτων βάσει του ILP

[Jouppi, DECWRL 1991]

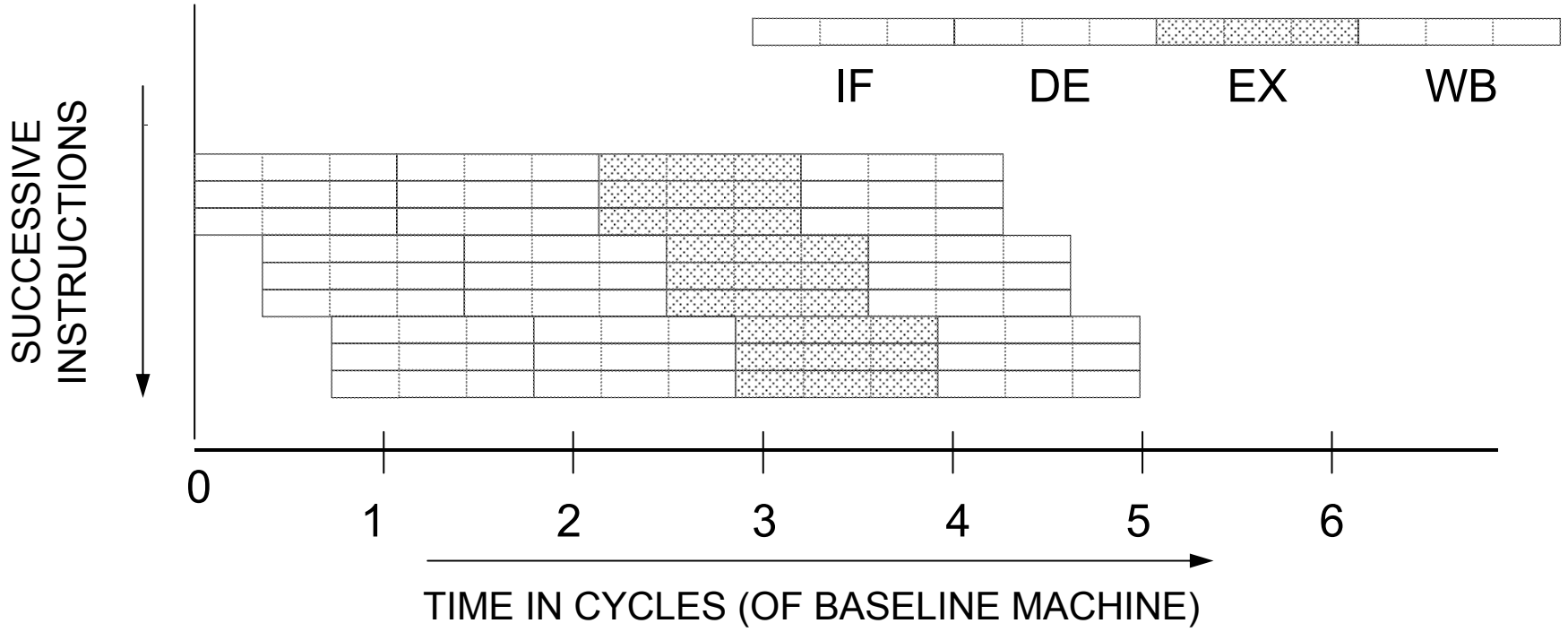
- **VLIW**: Very Long Instruction Word
  - Παραλληλισμός διανομής = IP = n εντολές / κύκλο
  - Καθυστέρηση λειτουργίας = OP = 1 κύκλος
  - Μέγιστο IPC = n εντολές / κύκλο = 1 VLIW / κύκλο



Κατηγοριοποίηση των μηχανημάτων βάσει του ILP

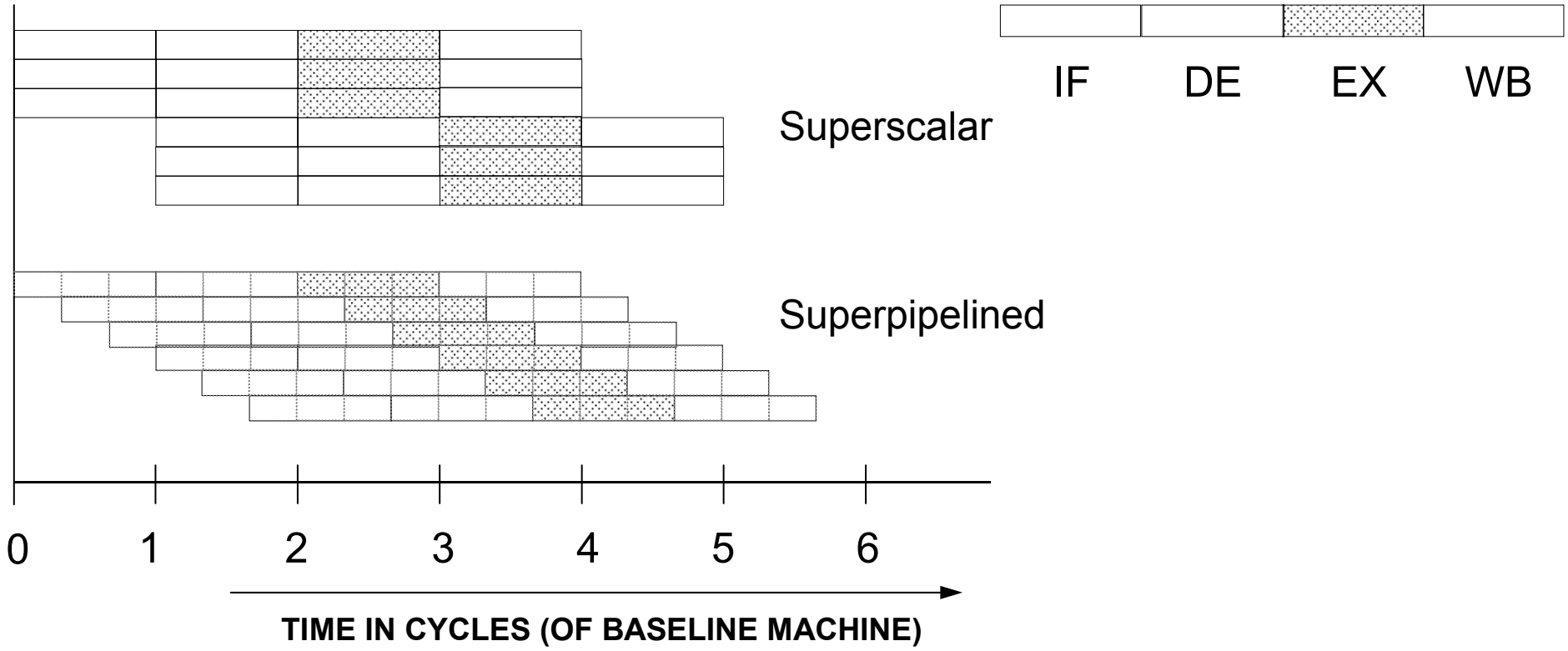
[Jouppi, DECWRL 1991]

- **Superpipelined-Superscalar**
  - Παραλληλισμός διανομής = IP = n εντολές / μικρο-κύκλο
  - Καθυστέρηση λειτουργίας = OP = m μικρο-κύκλοι
  - Μέγιστο IPC = n x m εντολές / βασικό κύκλο



Superscalar vs. Superpipelined

- Περίπου ισοδύναμη επίδοση
  - Αν n = m τότε και τα δύο έχουν το ίδιο IPC
  - Parallelism exposed in space vs. time

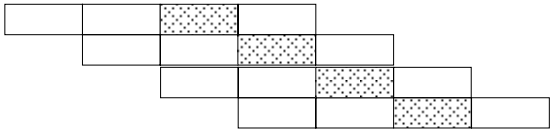


Superpipelining

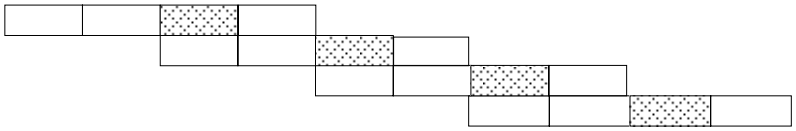
"Superpipelining is a new and special term meaning pipelining. The prefix is attached to increase the probability of funding for research proposals. There is no theoretical basis distinguishing superpipelining from pipelining. Etymology of the term is probably similar to the derivation of the now-common terms, methodology and functionality as pompous substitutes for method and function. The novelty of the term superpipelining lies in its reliance on a prefix rather than a suffix for the pompous extension of the root word."

- Nick Tredennick, 1991

Superpipelining: Hype vs. Reality

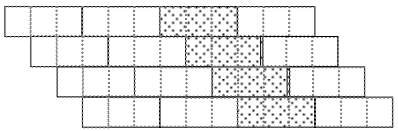


Base machine



Underpipelined machine

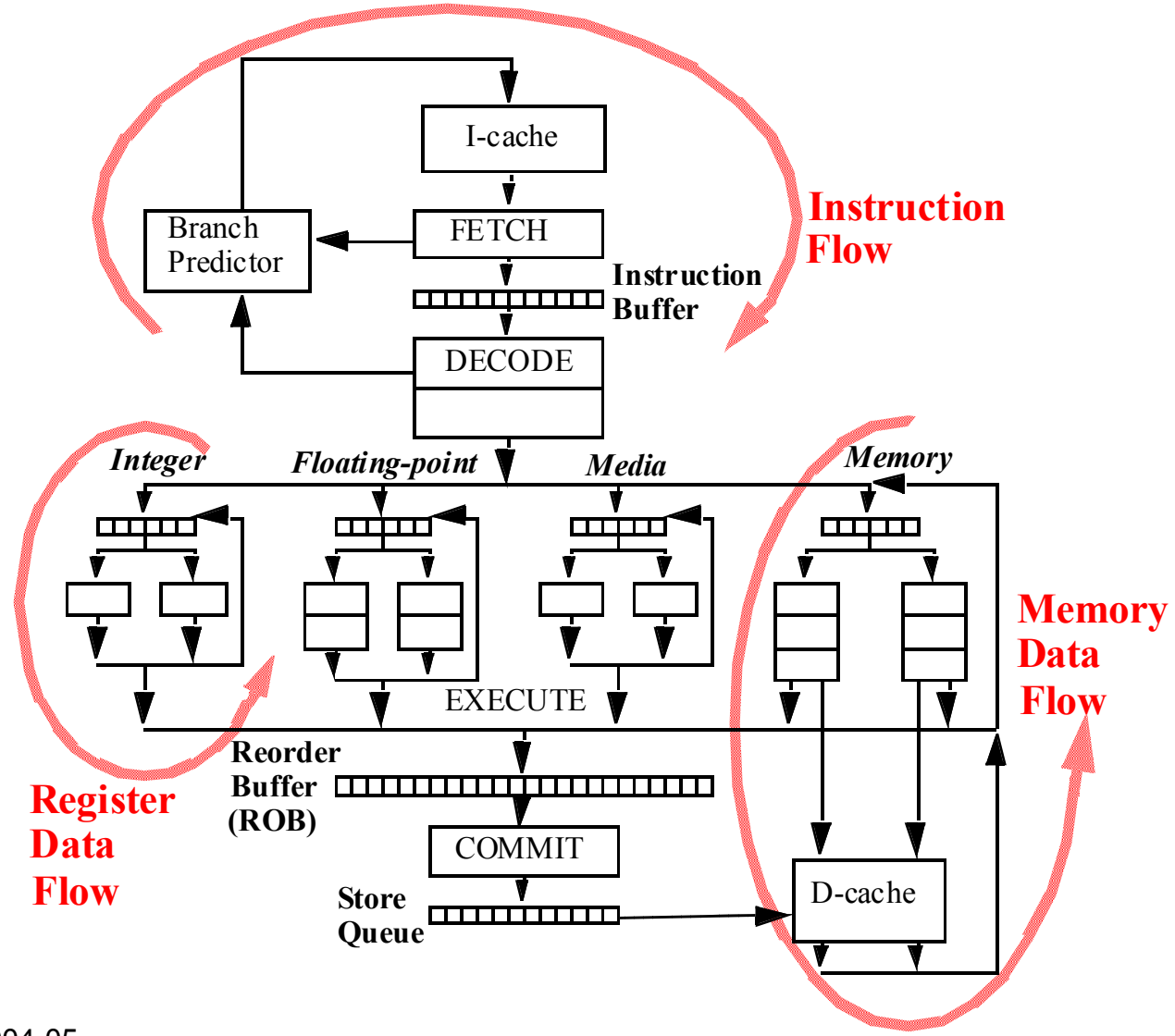
Η ταχύτητα διανομής των εντολών δεν ακολουθεί το ρυθμό επεξεργασίας τους



Superpipelined machine

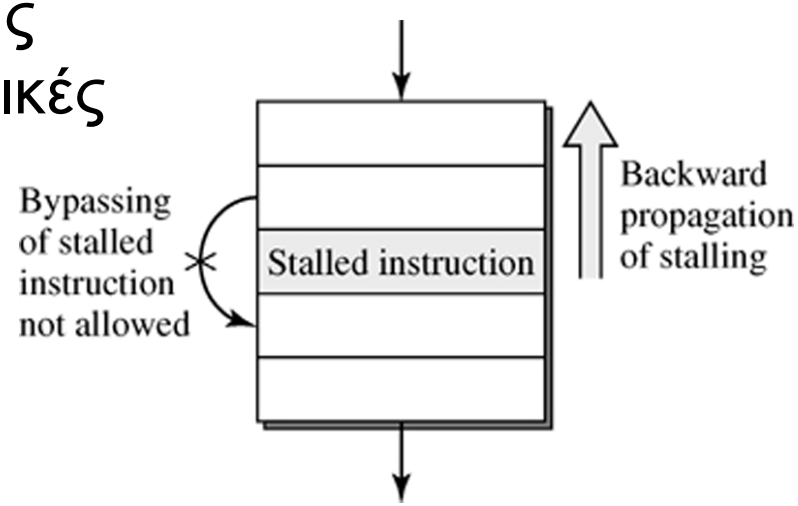
Στα Superpipelined μηχανήματα, τα αποτελέσματα μίας εντολής δεν είναι διαθέσιμα στις επόμενες m-1 διαδοχικές εντολές

Superscalar Challenges



Οι περιορισμοί των βαθμωτών αρχιτεκτονικών

- Μέγιστο throughput: 1 εντολή / κύκλο ρολογιού (IPC≤1)
- Υποχρεωτική ροή όλων των (διαφορετικών) τύπων εντολών μέσα από κοινή σωλήνωση
- Εισαγωγή καθυστερήσεων σε ολόκληρη την ακολουθία εκτέλεσης λόγω stalls μίας εντολής
  (οι απόλυτα βαθμωτές αρχιτεκτονικές υλοποιούν εν σειρά (in order) εκτέλεση των εντολών)



Οι υπερβαθμωτές αρχιτεκτονικές αγωγού

- Επεκτείνονται οι βαθμωτές αρχιτεκτονικές ώστε να ξεπεραστούν τα 3 (παραπάνω) εμπόδια που έθεταν στην επίδοση
- Εκτελούν πολλαπλές εντολές ανά κύκλο μηχανής (πρόκειται για παράλληλες αρχιτεκτονικές)
- Περιέχουν διαφορετικούς αγωγούς ροής δεδομένων, ο καθένας με όμοιες (πολλαπλή εμφάνιση του ίδιου τύπου) ή και ετερογενείς λειτουργικές μονάδες
- Επιτρέπουν εκτός σειράς (out-of-order) εκτέλεση των εντολών (δυναμικές αρχιτεκτονικές)

Παράδειγμα

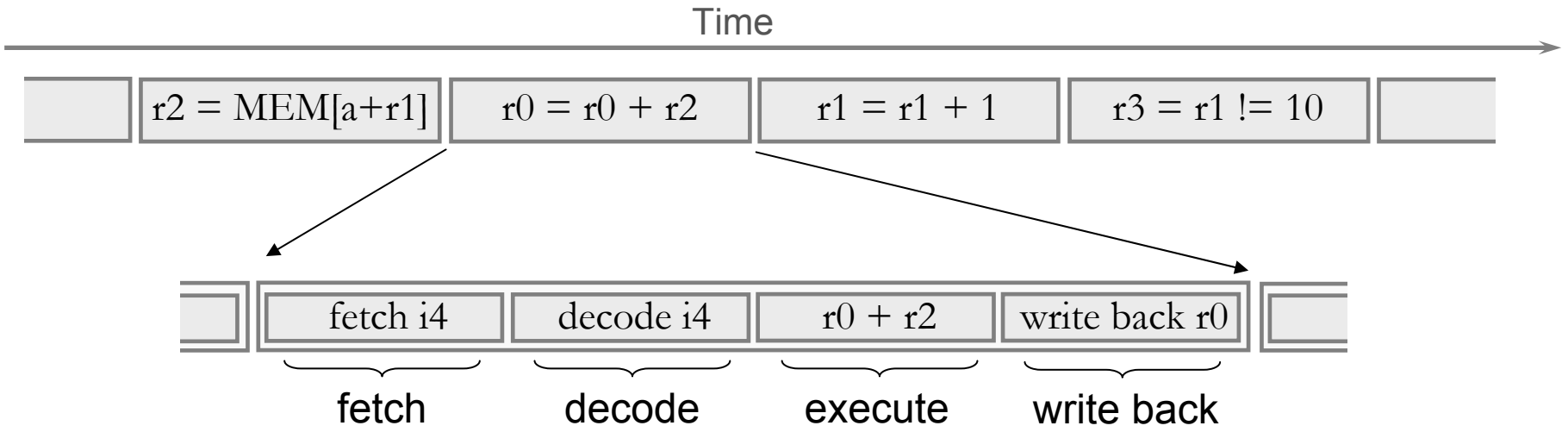
```
sum = 0;
for i=0 to 10
    sum = sum + a[i];
```

Assembly

```
i1          r1 = 0
i2          r2 = 0
i3   loop: r2 = MEM[a+r2]
i4          r0 = r0 + r2
i5          r1 = r1 + 1
i6          r3 = r1 != 10
i7          if (r3==true) goto loop
```

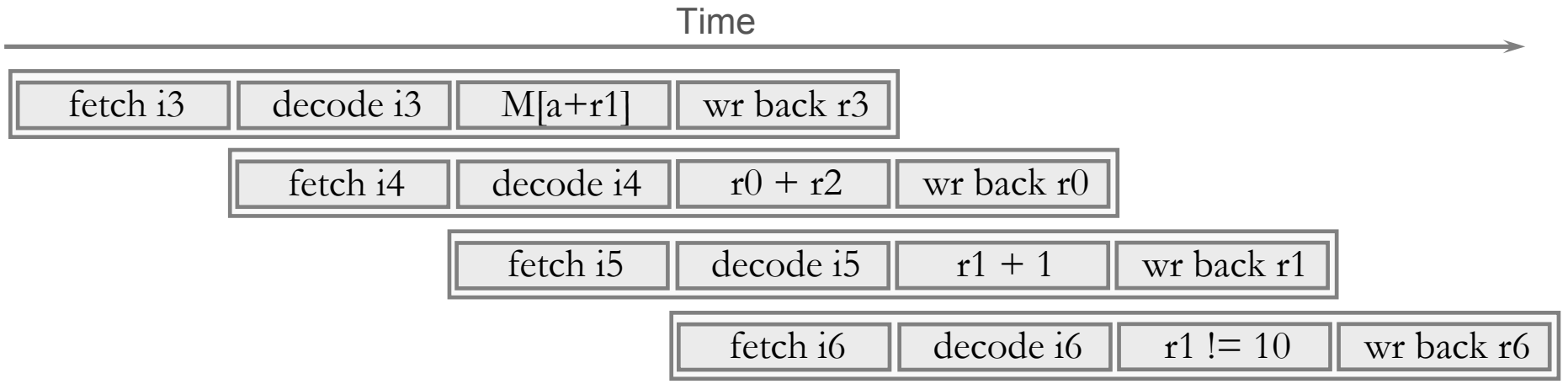
(sum → r0, i → r1)

Παράδειγμα (συνέχεια…)



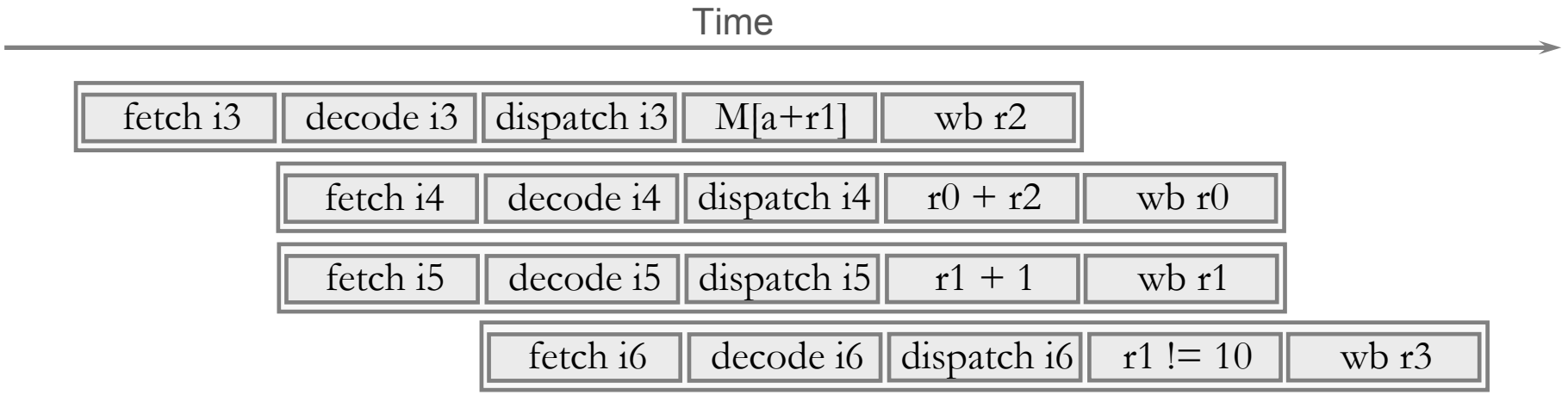
Σειριακή εκτέλεση των εντολών (σύστημα χωρίς αρχιτεκτονική αγωγού)

Παράδειγμα (συνέχεια…)



Εν σειρά εκτέλεση των εντολών (σε αρχιτεκτονική αγωγού)

Παράδειγμα (συνέχεια…)



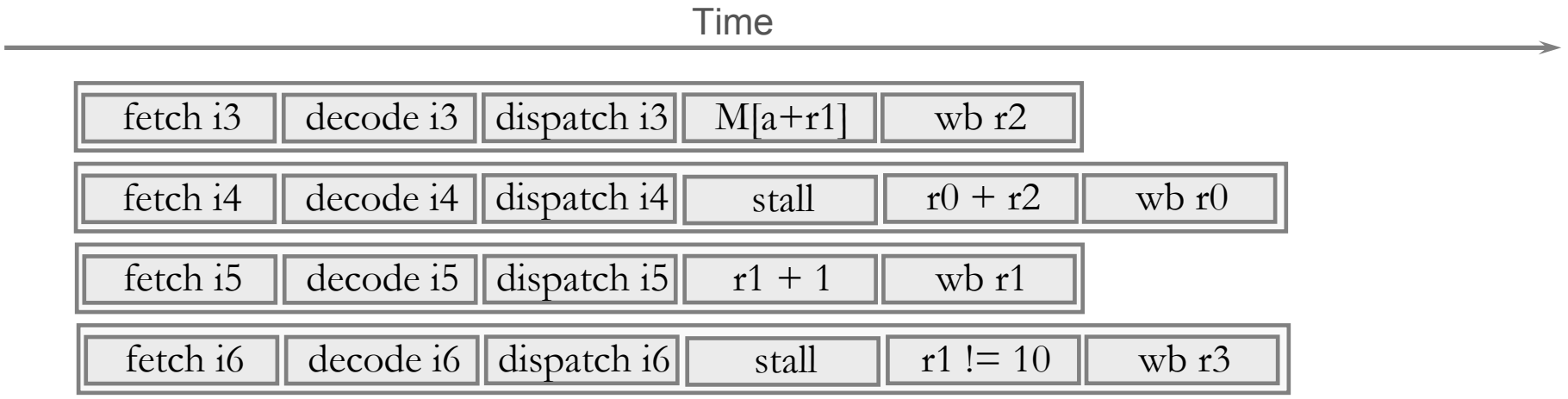
Υπερβαθμωτή (εν σειρά) εκτέλεση των εντολών :

Οι εντολές i4 και i5 μπορούν να εκτελεστούν παράλληλα.

Όμως: Η i4 χρειάζεται το αποτέλεσμα της i3

Η i6 χρειάζεται το αποτέλεσμα της i5

Παράδειγμα (συνέχεια…)

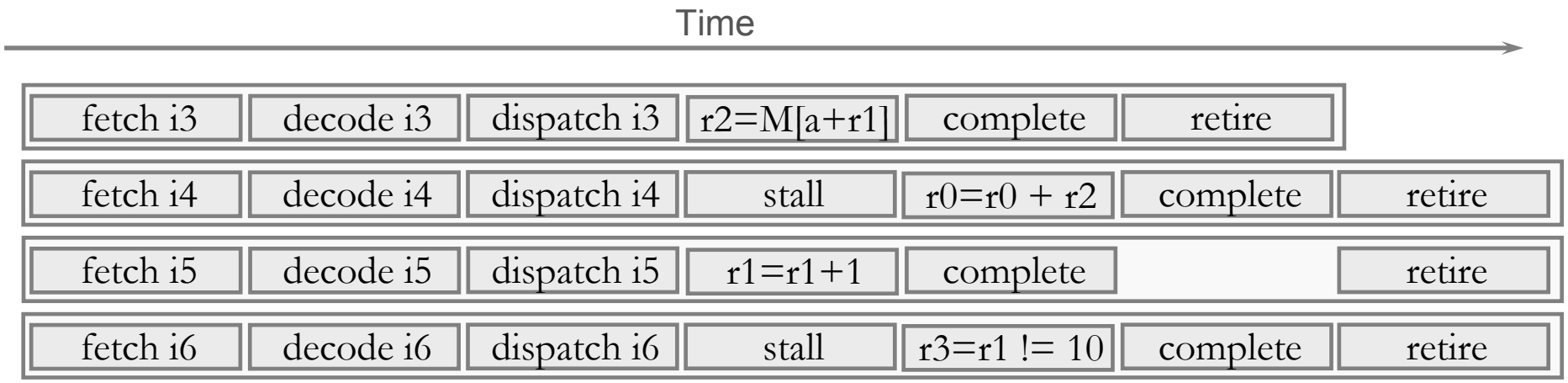


Εκτός σειράς εκτέλεση των εντολών :

Οι εντολές i3 και i5 εκτελούνται παράλληλα.

Οι i4 και i6 πρέπει να περιμένουν το αποτέλεσμα των i3 και i5

Παράδειγμα (συνέχεια…)



Διατήρηση της ορθότητας στην εκτός σειρά εκτέλεση των εντολών :

Εισαγωγή του σταδίου complete

(retire ≡ write back)

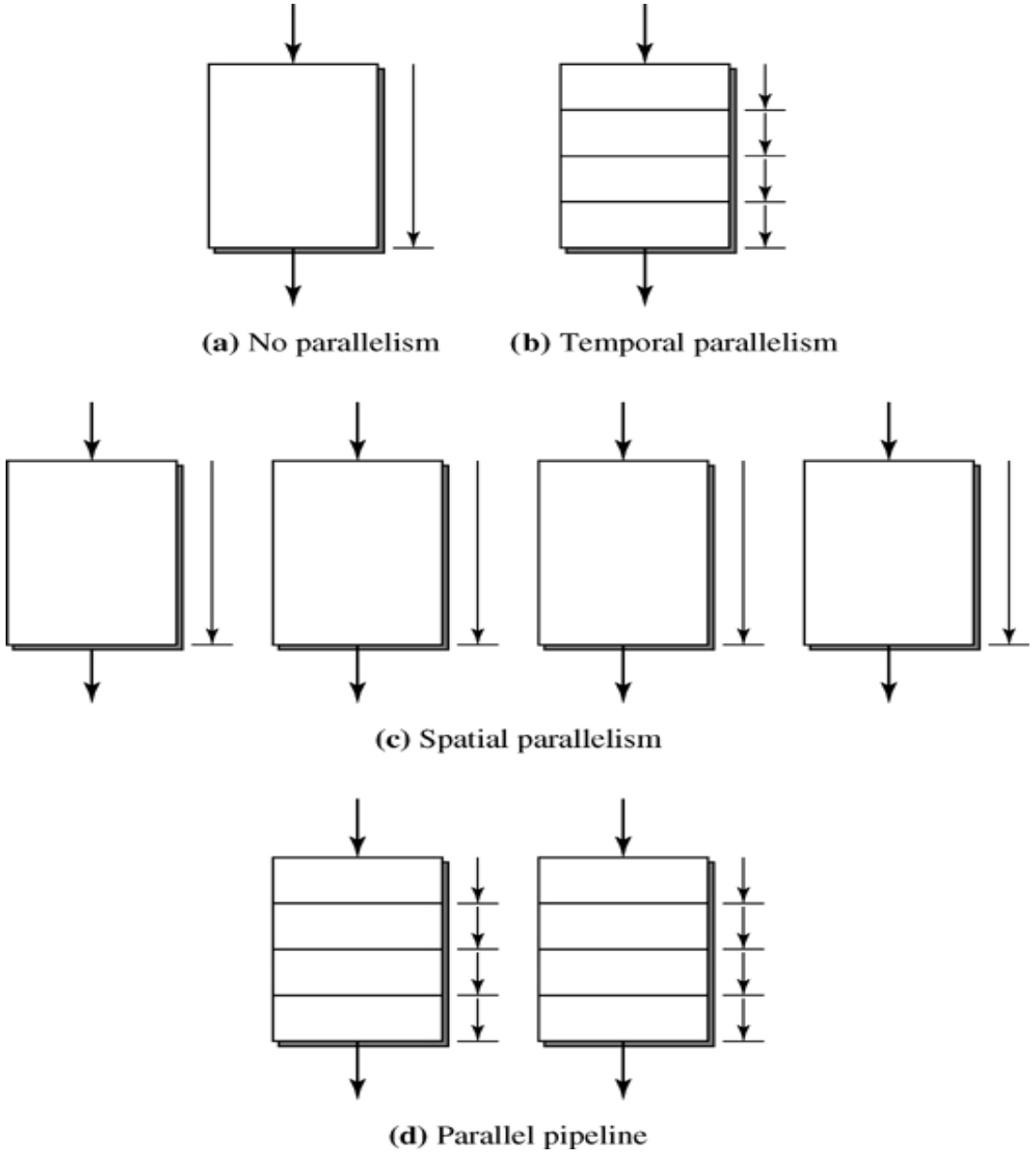
Out of order complete – *in order* retire

Αλλάζει το *speculative state*

Αλλάζει το *architectural state*

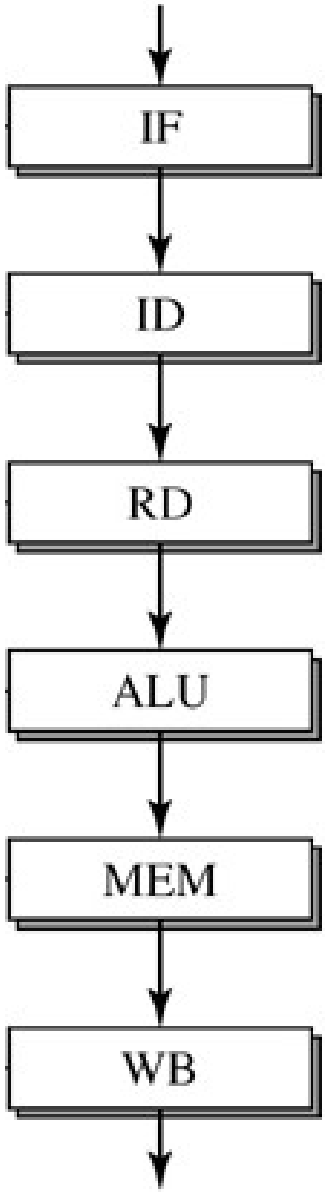
Παράλληλες αρχιτεκτονικές αγωγού

- Βαθμός παραλληλισμού μηχανήματος = ο μέγιστος αριθμός εντολών που μπορούν ταυτόχρονα να είναι σε εξέλιξη (σε μία βαθμωτή αρχιτεκτονική αγωγού ισούται με τον αριθμό τον σταδίων της σωλήνωσης- pipeline depth)



Αρχιτεκτονική αγωγού 6 σταδίων

- Η αρχιτεκτονική αγωγού που χρησιμοποιείται από τους Shen & Lipasti - Modern Processor Design:
  - IF – instruction fetch
  - ID – instruction decode
  - RD – register read
  - ALU – ALU op./addr. generation
  - MEM – read/write mem
  - WB – register write

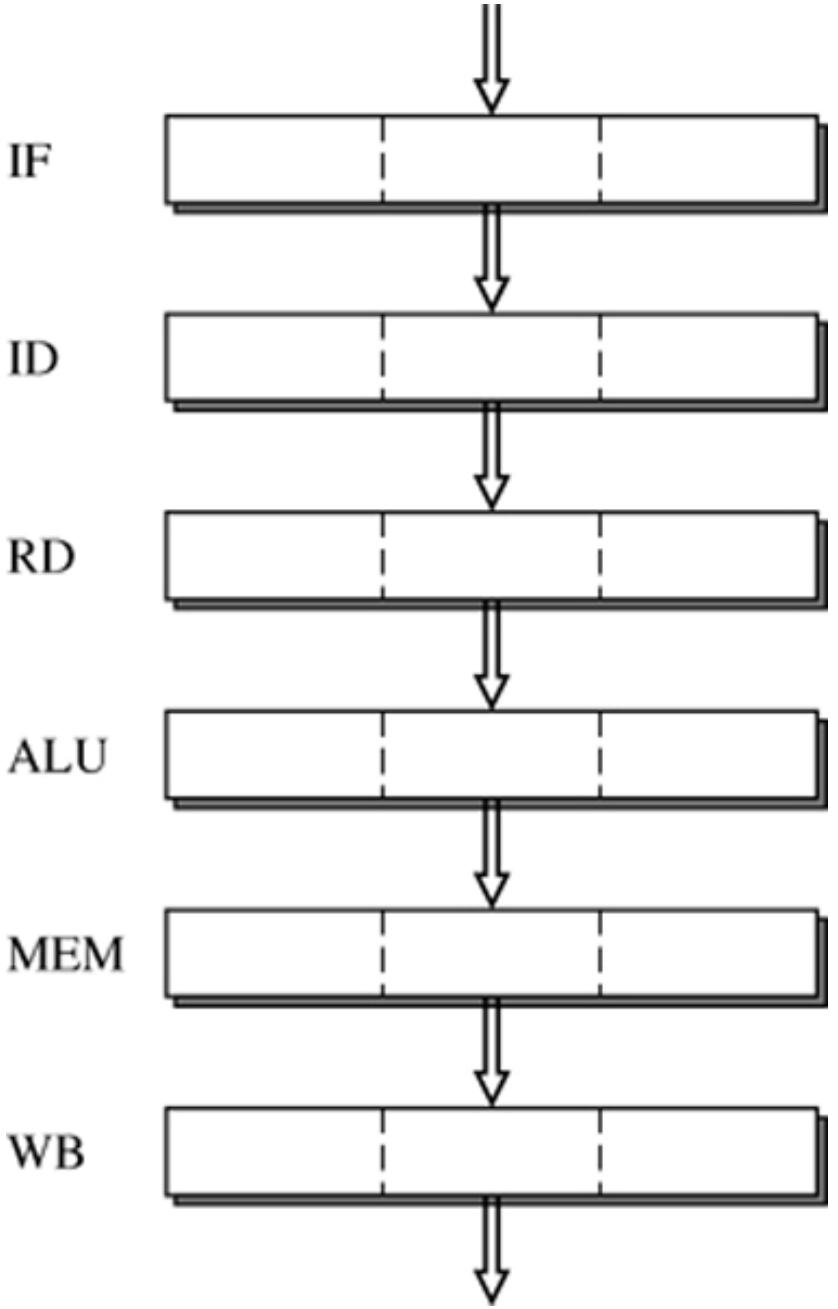


Επιτυγχάνεται επιτάχυνση ίση με 6 (όσα τα στάδια της σωλήνωσης) σε σύγκριση με μία αρχιτεκτονική χωρίς pipeline

Pipeline πλάτους 3

- Πολλαπλά δομικά στοιχεία (functional units) στο hardware
- Αυξάνεται η λογική πολυπλοκότητα των σταδίων του pipeline
- Απαιτούνται πολλαπλές θύρες εγγραφής/ανάγνωσης του Register File για την ταυτόχρονη προσπέλασή του από όλους τους αγωγούς της αρχιτεκτονικής

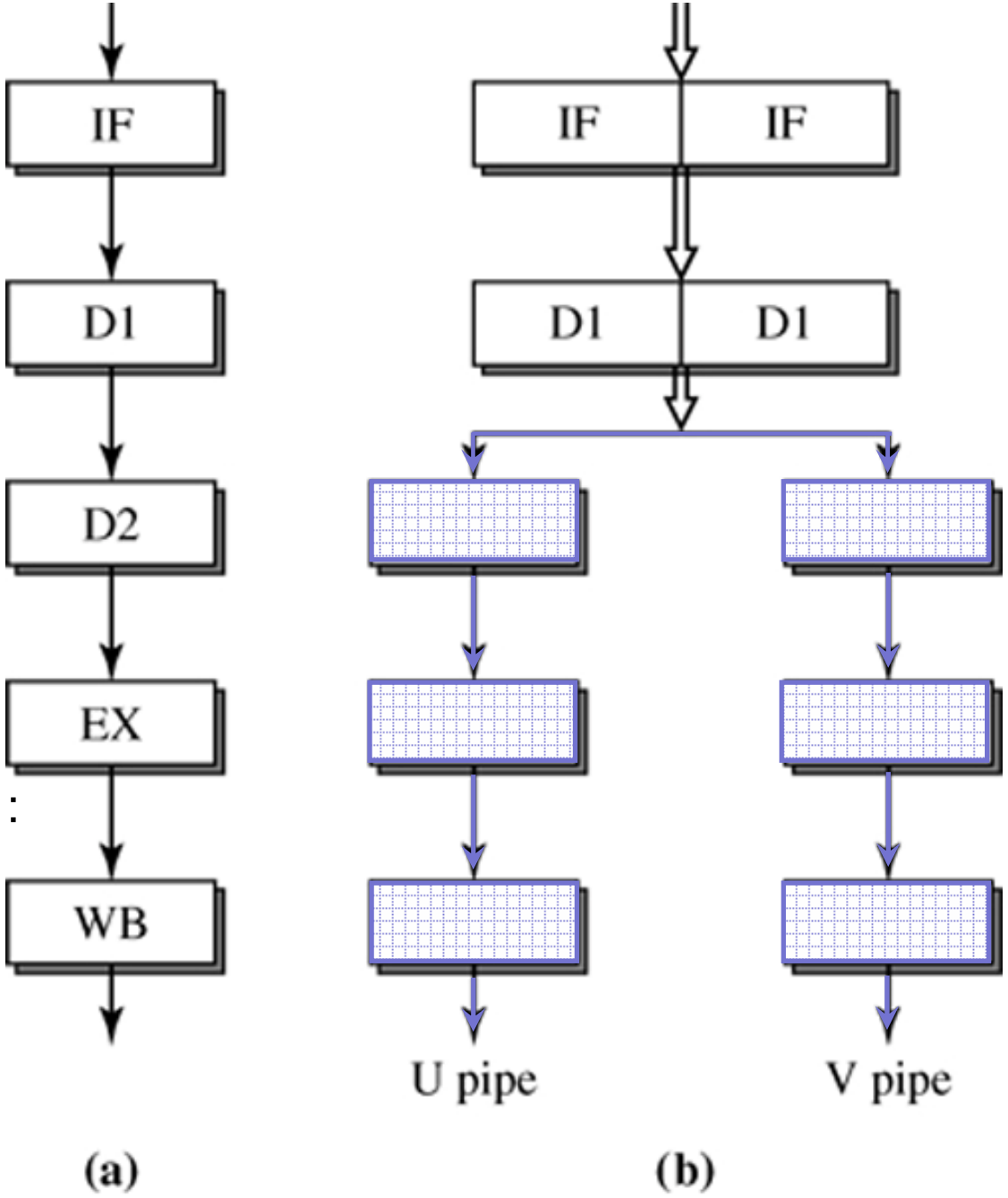
Επιτυγχάνεται επιτάχυνση ίση με 3 (όσο το πλάτος της σωλήνωσης) σε σύγκριση με μία απλή βαθμωτή αρχιτεκτονική αγωγού



Παράδειγμα: Intel Pentium parallel pipeline

(a) Intel i486 :
pipeline 5 σταδίων

(b) Intel Pentium microprocessor :
περιέχει 2 pipeline του i486



Ετερογενείς σωληνώσεις

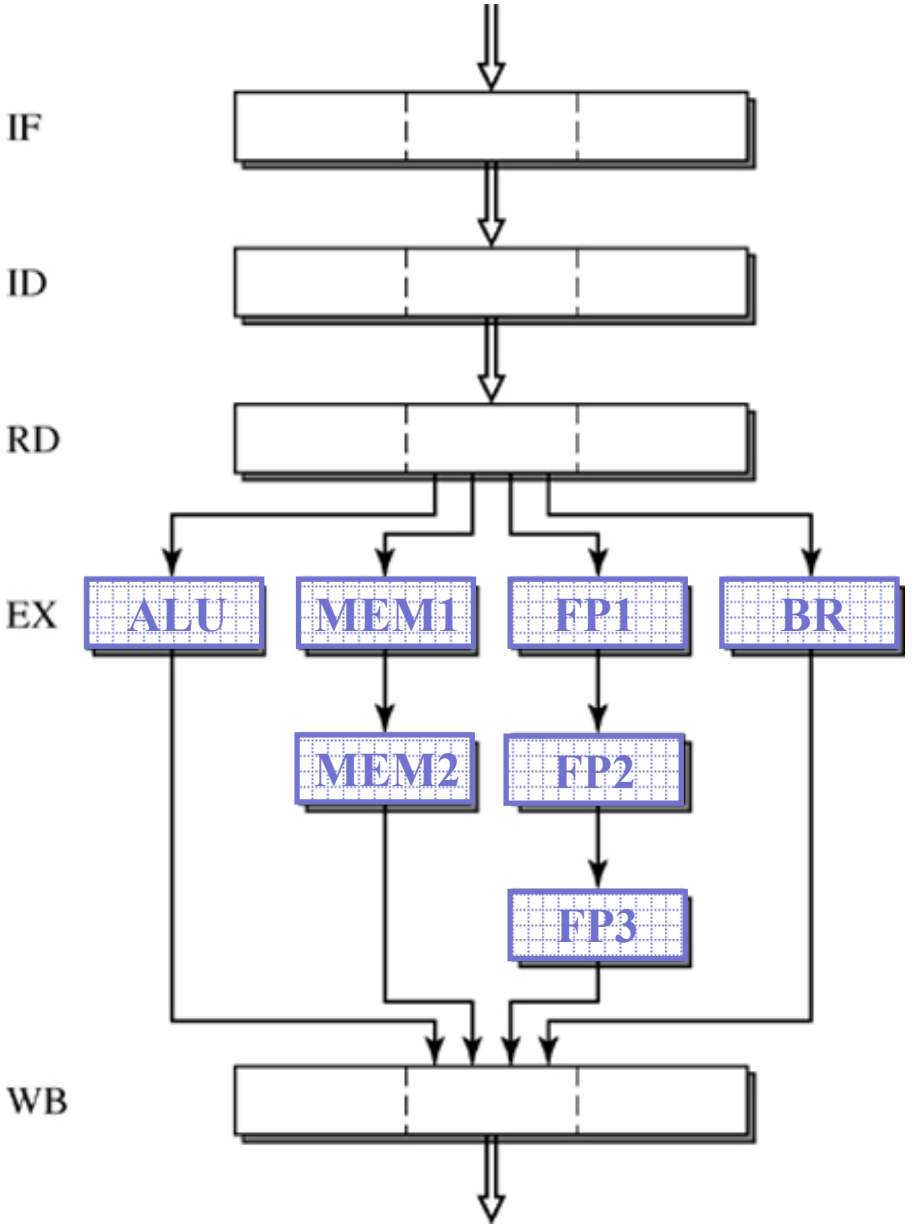
- Διαφορετικοί τύποι εντολών απαιτούν διαφορετικά στάδια εκτέλεσης. Οι βαθμωτές αρχιτεκτονικές αγωγού διαθέτουν μία μόνο σωλήνωση μέσα από την οποία πρέπει να περάσουν όλες οι εντολές... (συμβιβασμός όλων των διαφορετικών απαιτήσεων)

  Μπορεί η ροή εκτέλεσης να διαφοροποιηθεί για κάθε τύπο εντολής;

Ετερογενείς σωληνώσεις

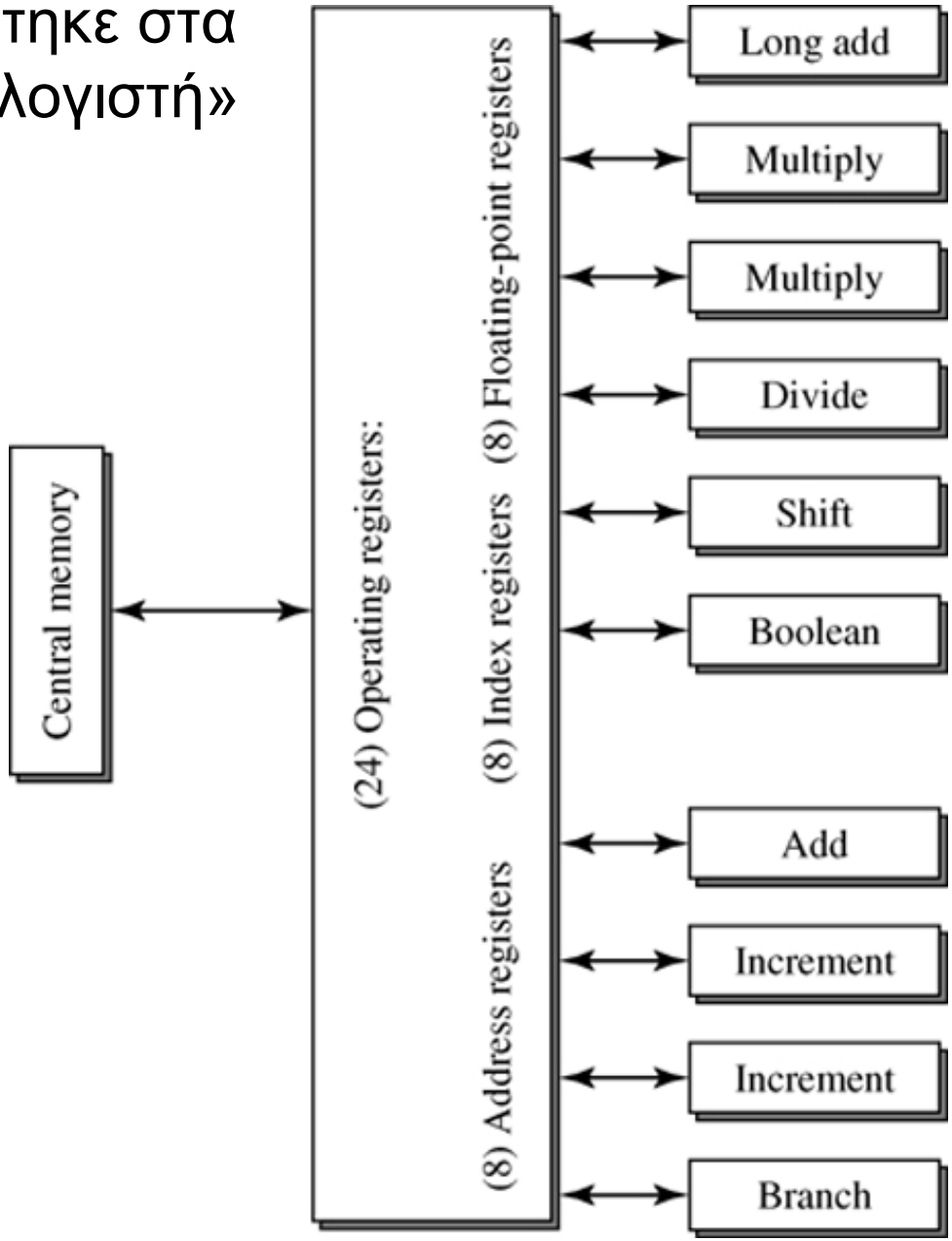
- Στις υπερβαθμωτές αρχιτεκτονικές, αντί πολλαπλών πανομοιότυπων αγωγών, χρησιμοποιούνται διαφορετικές λειτουργικές μονάδες σε κάθε αγωγό εκτέλεσης εντολών

Στο διπλανό σχήμα: Μετά το στάδιο RD κάθε εντολή κατευθύνεται μέσα από τον αγωγό που αντιστοιχεί στο συγκεκριμένο τύπο εντολής που ανήκει



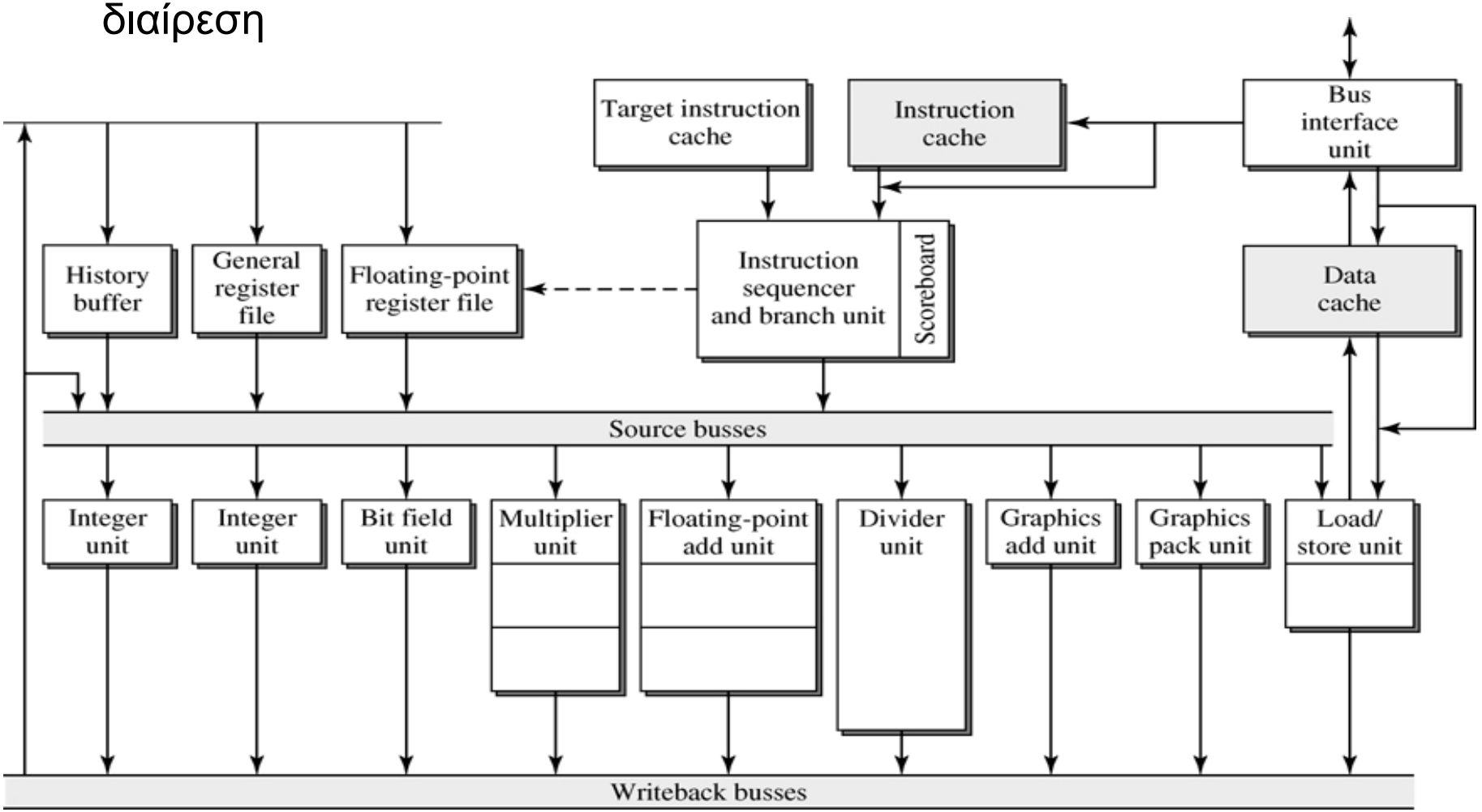
CDC 6600 : ο πρώτος υπολογιστής που σχεδιάστηκε στα πρότυπα ενός «υπερ-υπολογιστή»

- 1964 (πριν από τους επεξεργαστές RISC) : περιείχε 10 διαφορετικές λειτουργικές μονάδες έξω από τη σωλήνωση, με διαφορετικό latency η κάθε μία
- Στόχος η διεκπεραίωση 1 εντολής ανά κύκλο μηχανής

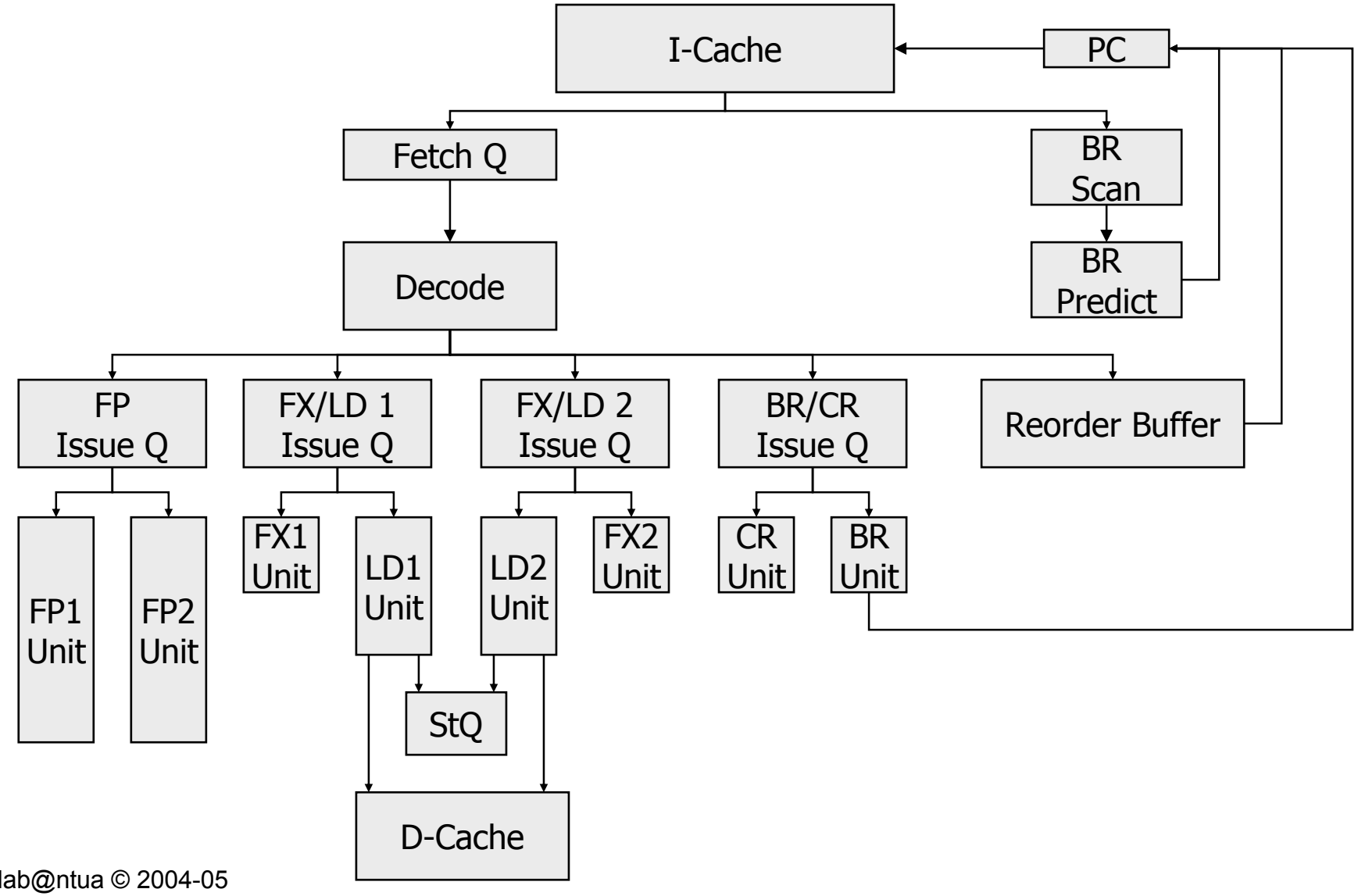


Motorola 88110 : έναν από τους πλατύτερους αγωγούς υπερ-υπολογιστών

- Περιέχει 10 λειτουργικές μονάδες, στην πλειοψηφία τους με latency ενός κύκλου. Όλες ενσωματωμένες στη σωλήνωση, εκτός από τη μονάδα για διαίρεση



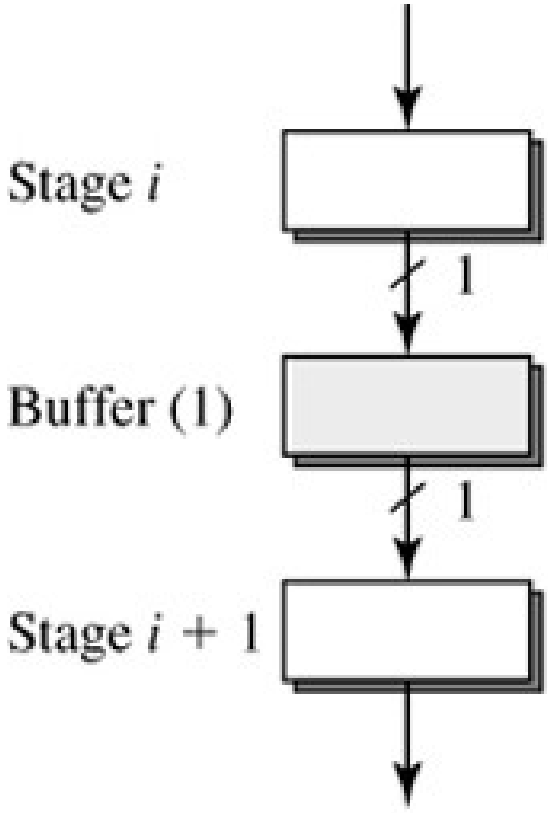
Ετερογενής σωλήνωση του Power4



Βαθμωτή αρχιτεκτονική αγωγού – εκτέλεση εν σειρά (in order)

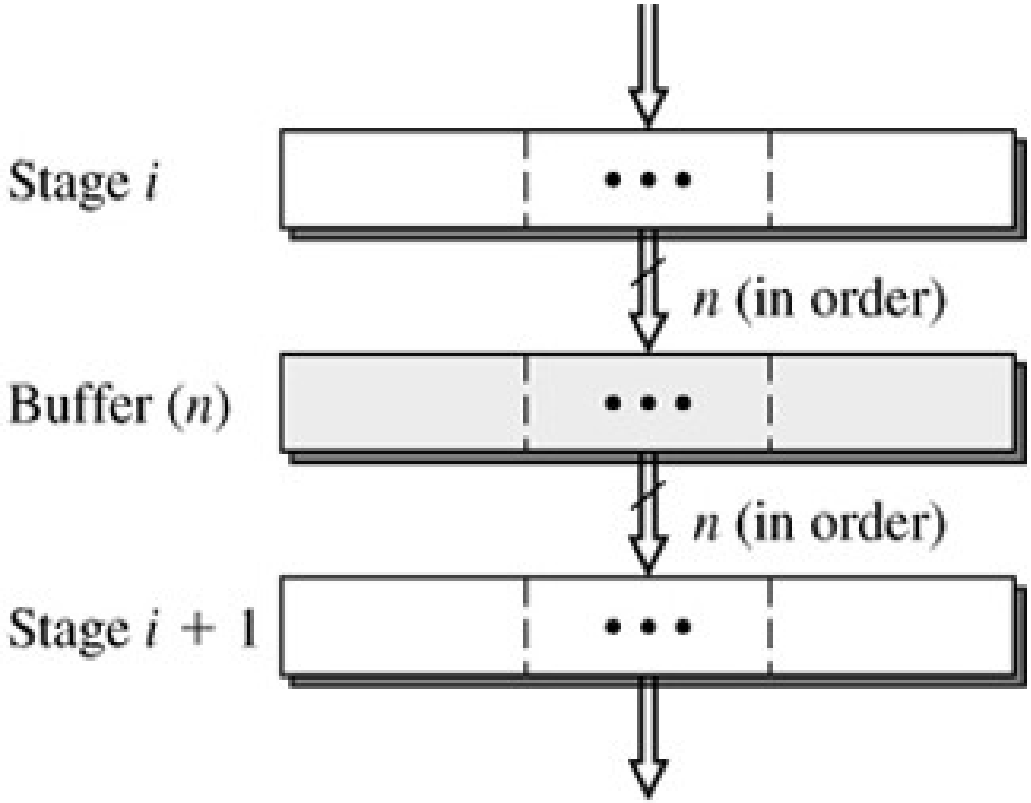
- Μεταξύ δύο σταδίων της σωλήνωσης βρίσκεται πάντα ένας προσωρινός καταχωρητής δεδομένων (buffer)

Σε βαθμωτή αρχιτεκτονική αγωγού (με in order εκτέλεση των εντολών) κάθε buffer έχει μία μόνο θύρα εισόδου και μία εξόδου



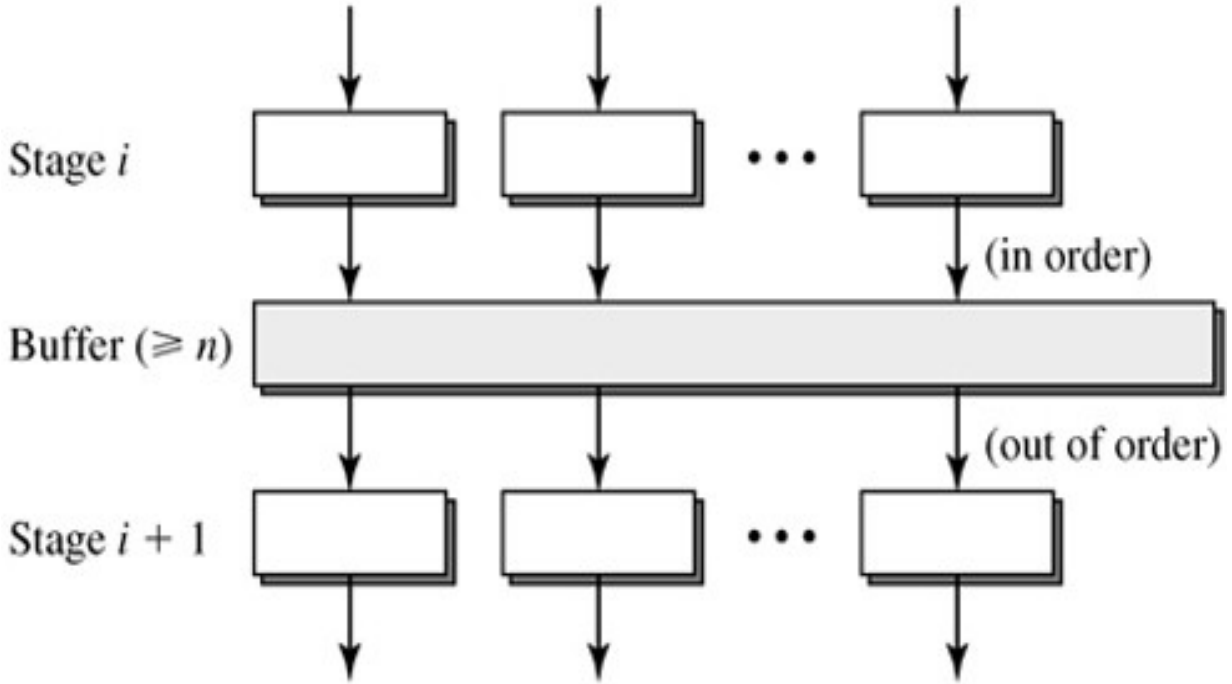
Υπερβαθμωτή αρχιτεκτονική αγωγού – εκτέλεση εν σειρά (in order)

- Κάθε buffer έχει πολλαπλές θύρες εισόδου και εξόδου
- Κάθε απλός καταχωρητής μπορεί να λάβει δεδομένα μόνο από μία θύρα. Δεν υπάρχει δυνατότητα διακίνησης των δεδομένων μεταξύ των καταχωρητών του ίδιου buffer



Υπερβαθμωτή αρχιτεκτονική αγωγού – εκτέλεση εκτός σειράς (out of order)

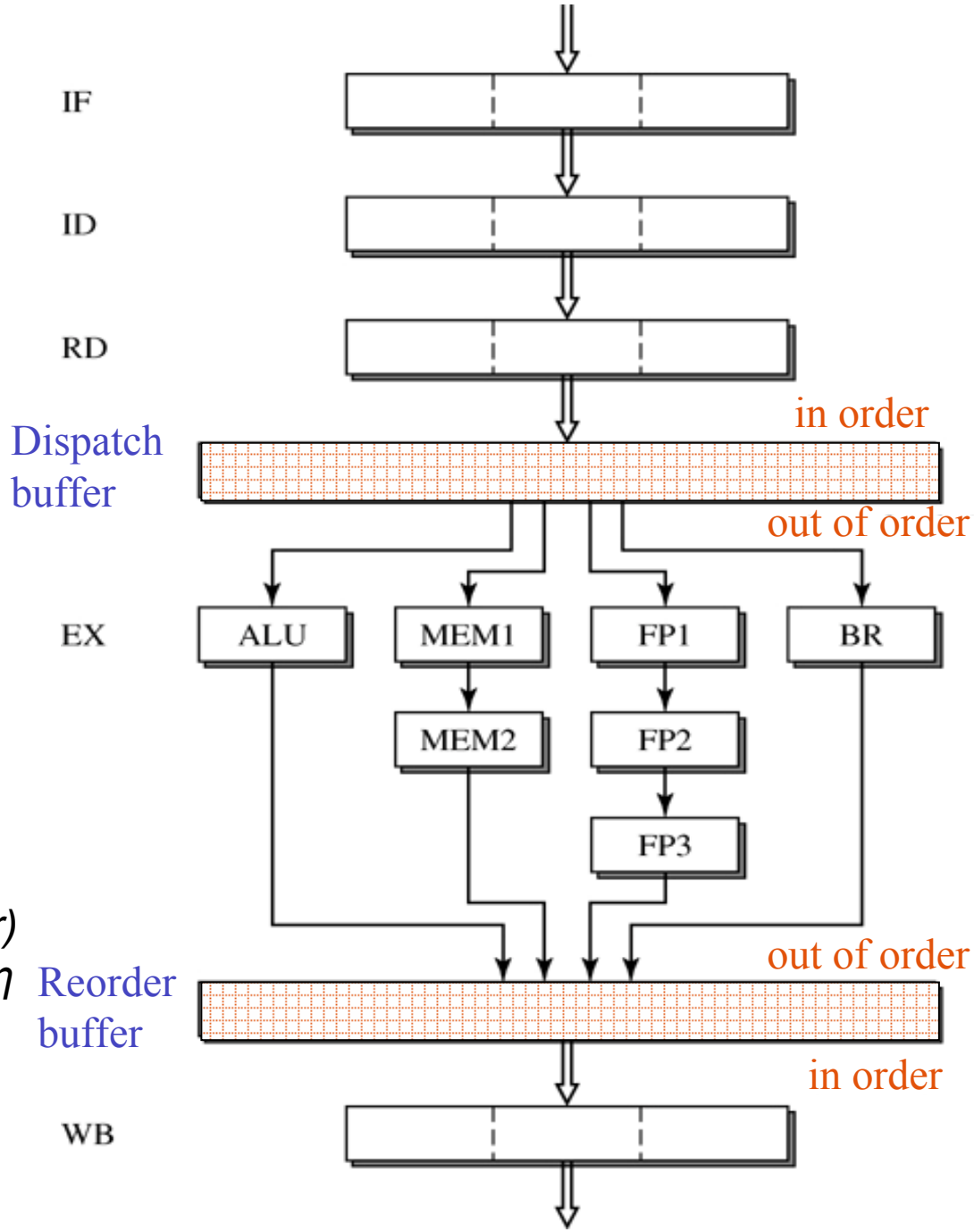
- Οι εντολές που καθυστερούν μέσα στη σωλήνωση (stall) παρακάμπτονται και συνεχίζεται η ροή εκτέλεσης με τις επόμενες εντολές (εκτέλεση εντολών εκτός σειράς – out of order)



Δυναμική αρχιτεκτονική αγωγού

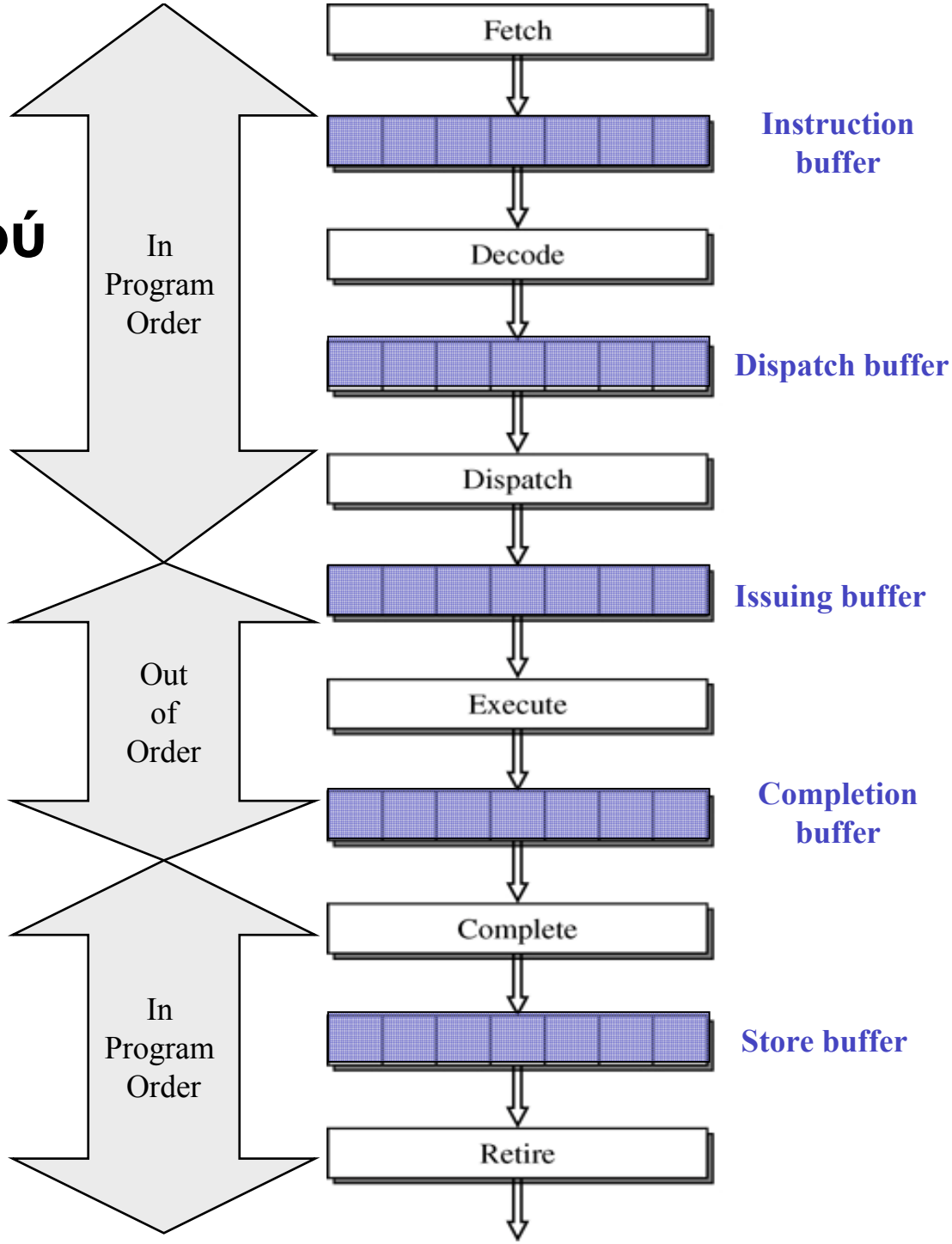
Δυναμική αρχιτεκτονική αγωγού με πολλαπλές, διαφοροποιημένες σωληνώσεις

Ο πρώτος καταχωρητής πολλαπλών εισόδων (dispatch buffer) λαμβάνει τις εντολές εν σειρά και τις κατανείμει στις λειτουργικές μονάδες (ενδεχομένως) εκτός σειράς. Όταν οι εντολές ολοκληρώσουν της εκτέλεσή τους, ο δεύτερος καταχωρητής (completion buffer) τις αναδιατάσσει, σύμφωνα με τη σειρά που υπαγορεύει το πρόγραμμα, προκειμένου να πιστοποιήσει τη σωστή ολοκλήρωσή τους.



Γενική μορφή υπερβαθμωτής αρχιτεκτονικής αγωγού

- Το στάδιο εκτέλεσης (Execute) μπορεί να περιλαμβάνει πολλαπλές διαφορετικού τύπου σωληνώσεις, με διαφορετικό latency η κάθε μία.
- Αναγκαία τα στάδια Dispatch και Complete, για την αναδιάταξη και επαναφορά των εντολών σε σειρά
- Παρεμβολή προσωρινών καταχωρητών μεταξύ των διαδοχικών σταδίων (interstage buffers)



Οι περιορισμοί των βαθμωτών αρχιτεκτονικών

- Οι βαθμωτές αρχιτεκτονικές περιορίζονται από το throughput (IPC≤1)
  - *Λύση: αρχιτεκτονικές μεγαλύτερου εύρους (wide pipelines: superscalar)*
- Υποχρεωτική ροή όλων των (διαφορετικών) τύπων εντολών μέσα από κοινή σωλήνωση (όλες οι εντολές έχουν υποχρεωτικά το ίδιο latency)
  - *Λύση: ετερογενείς, εξειδικευμένες σωληνώσεις*
- Εισαγωγή καθυστερήσεων σε ολόκληρη την ακολουθία εκτέλεσης λόγω stalls μίας εντολής (οι απόλυτα βαθμωτές αρχιτεκτονικές υλοποιούν εν σειρά (in order) εκτέλεση των εντολών)
  - *Λύση: Εκτέλεση εκτός σειράς (out-of-order execution), κατανεμημένες σωληνώσεις για το τμήμα εκτέλεσης των εντολών*

Ροή εντολών

- Στόχος: Η ανάγνωση πολλαπλών εντολών ανά κύκλο ρολογιού
- Προβλήματα
  - Διακλαδώσεις: εξαρτήσεις απόφασης διακλάδωσης (control dependencies)
  - Κακή ευθυγράμμιση της διεύθυνσης-στόχου
  - Cache misses εντολών
- *Λύσεις:*
  - *Ευθυγράμμιση κώδικα*
  - *Πρόβλεψη διακλάδωσης (Prediction) / Θεωρητική εκτέλεση εντολών (speculation)*