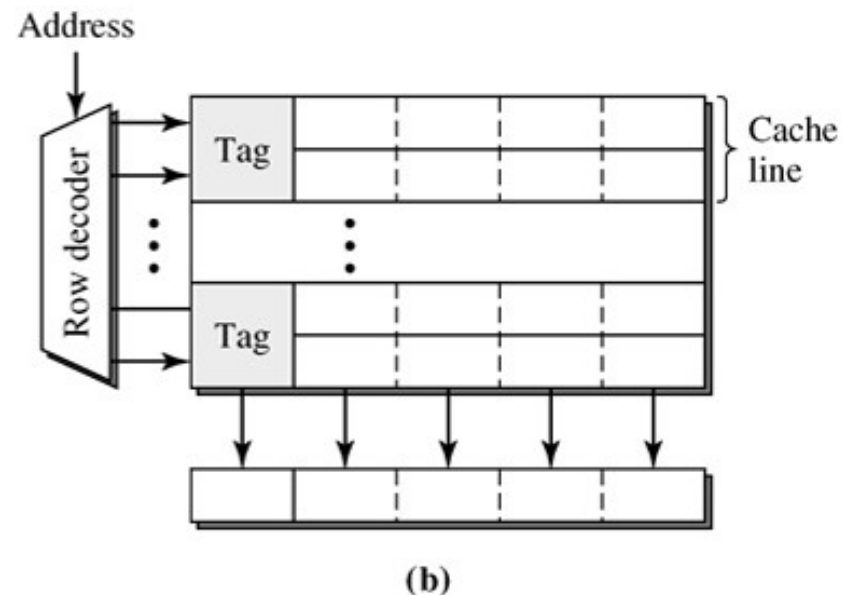
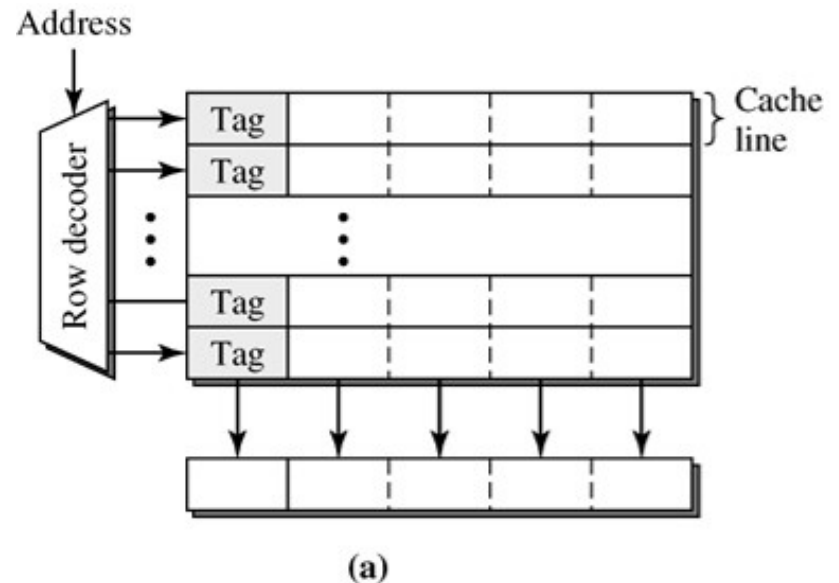


Υπερβαθμωτή Οργάνωση Υπολογιστών

Από τις βαθμωτές στις υπερβαθμωτές
αρχιτεκτονικές αγωγού...

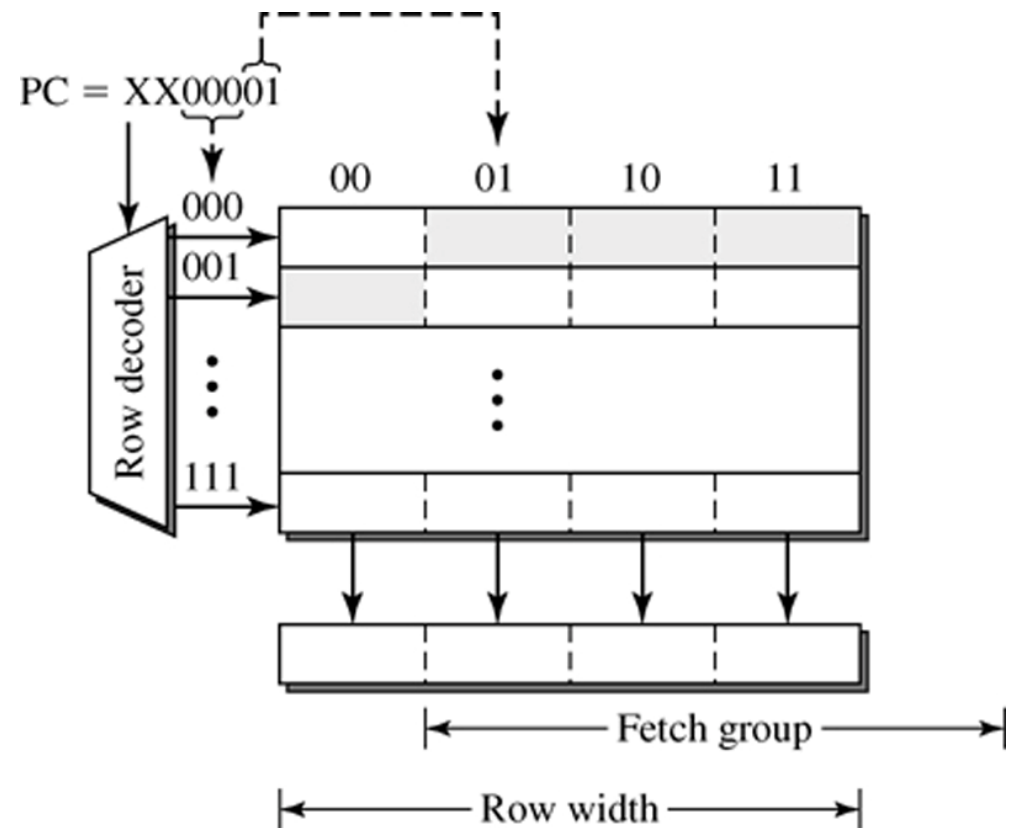
Ανάγνωση εντολής (Instruction Fetch)

- Σε μία αρχιτεκτονική πλάτους s , πρέπει διαβάζονται s εντολές σε κάθε κύκλο μηχανής από την κρυφή μνήμη εντολών (I-cache)
- Σε κάθε προσπέλαση της I-cache διαβάζεται μία ολόκληρη γραμμή (s εντολές ανά γραμμή): 1 cache line ανά γραμμή της I-cache (a). Θα μπορούσε μία cache line να επεκτείνεται σε περισσότερες της μίας γραμμής (b).



Ανάγνωση εντολής (Instruction Fetch)

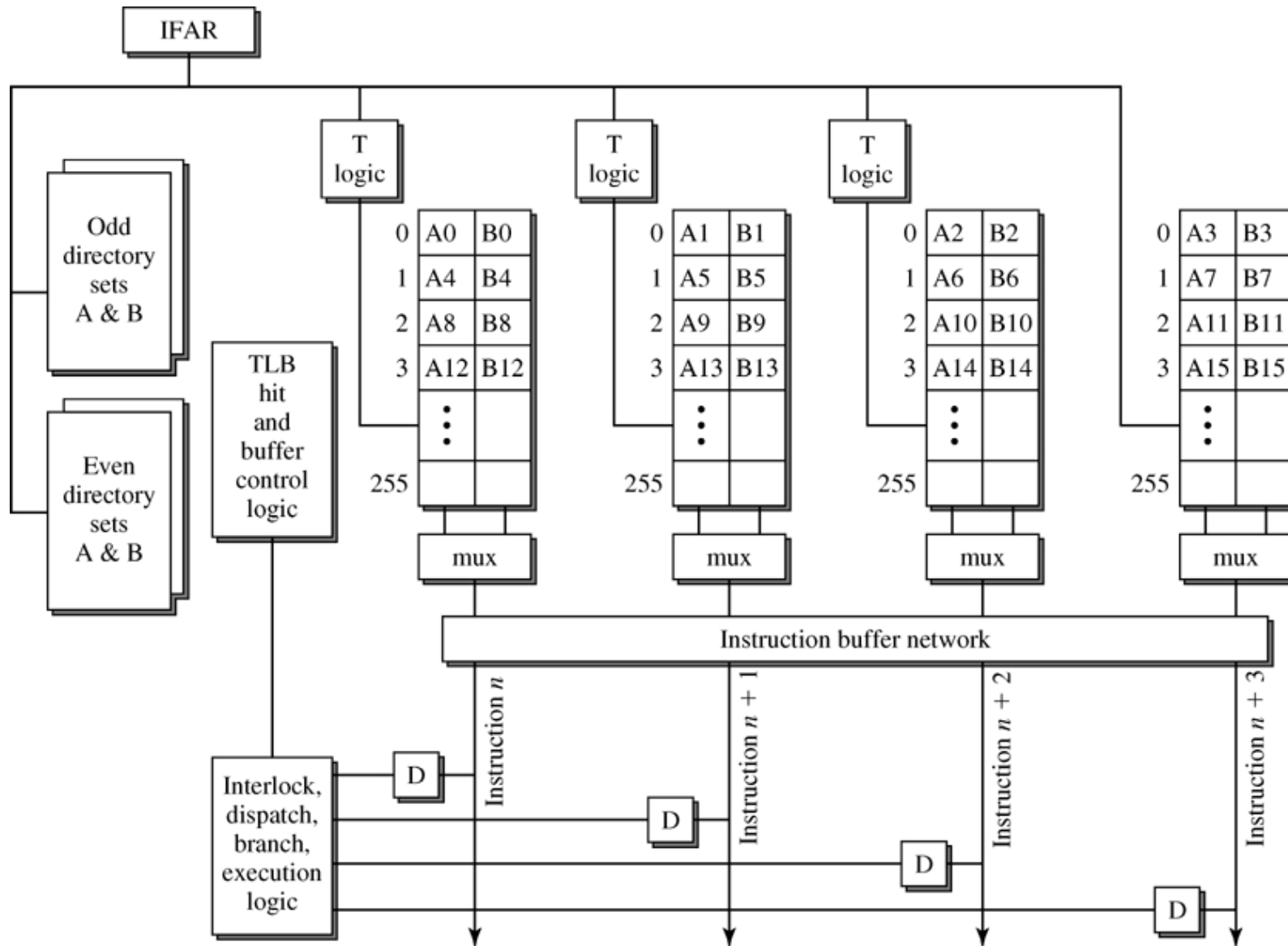
- Για την ανάγνωση s εντολών σε κάθε κύκλο μηχανής θα πρέπει:
 - Οι s εντολές (που ανήκουν στο ίδιο **fetch group**) να είναι ευθυγραμμισμένες (aligned) στην I-cache



Ανάγνωση εντολής (Instruction Fetch)

- Εναλλακτικά, υπάρχουν μηχανισμοί ελέγχου της ροής των εντολών, που μπορούν να φροντίσουν για την ευθυγράμμιση των εντολών που ανήκουν στο ίδιο fetch group
 - Software (compiler static alignment)
 - Hardware (run time technique)

Alignment Hardware (IBM RS/6000)

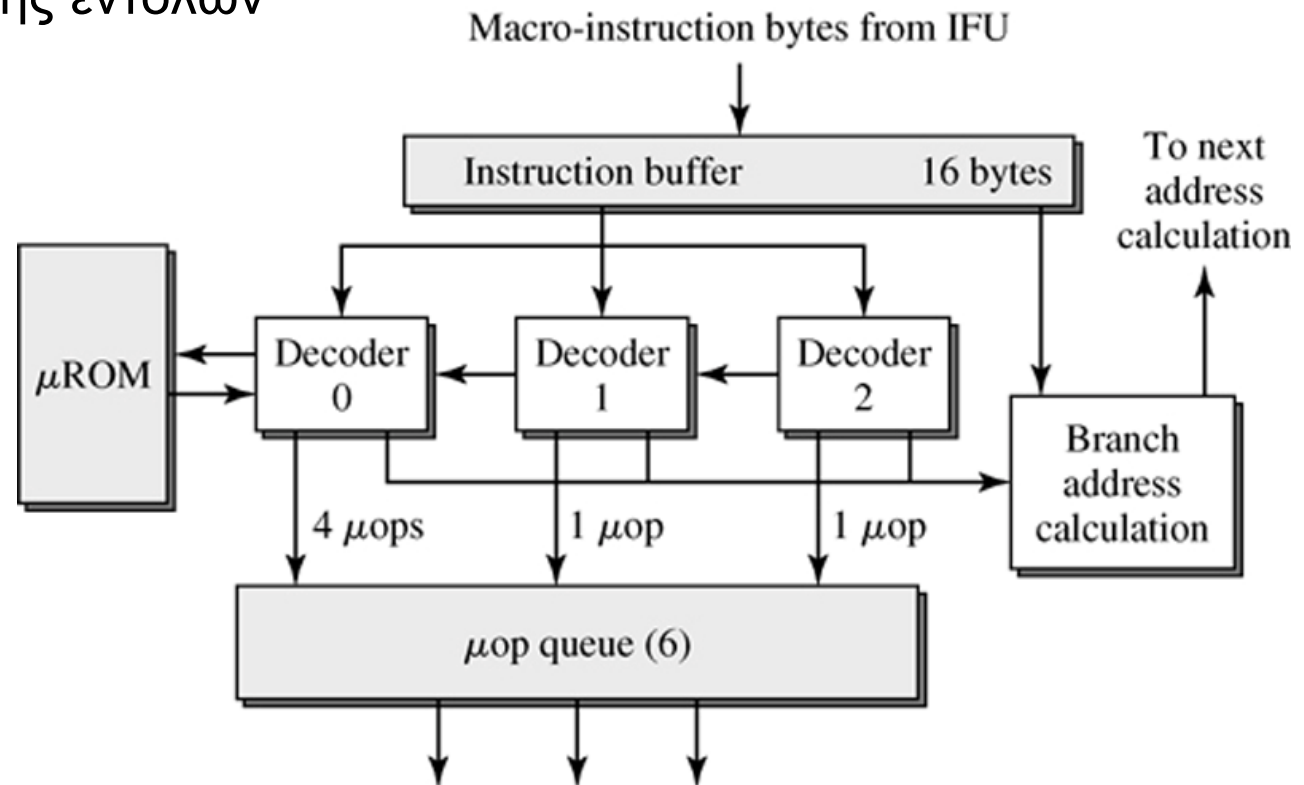


Αποκωδικοποίηση εντολών (Instruction Decode)

- Αναγνώριση των όποιων εξαρτήσεων μεταξύ εντολών – προώθηση των μη εξαρτημένων εντολών στο επόμενο στάδιο
- Αναγνώριση των εντολών άλματος
- Αποτελεί το κρισιμότερο στάδιο για την επίδοση ολόκληρης της αρχιτεκτονικής αγωγού
- Ακόμη μεγαλύτερη δυσκολία στις CISC εντολές: κάθε εντολή μεταφράζεται σε περισσότερες μικρο-εντολές (1,5-2 μops / CISC εντολή για την Intel)

Αποκωδικοποίηση εντολών (Instruction Decode)

- Μονάδα ανάγνωσης-αποκωδικοποίησης εντολών του Intel P6



Αποκωδικοποίηση εντολών (Instruction Decode)

- Η αποκωδικοποίηση των εντολών CISC σε παράλληλες αρχιτεκτονικές αγωγού ή οι πολύ πλατιές σωληνώσεις
 - Απαιτεί πολλαπλά στάδια αγωγού.
 - Αυξάνει το branch penalty

Πώς μπορεί να αποφευχθεί η αύξηση του βάθους του τμήματος αποκωδικοποίησης στη σωλήνωση;

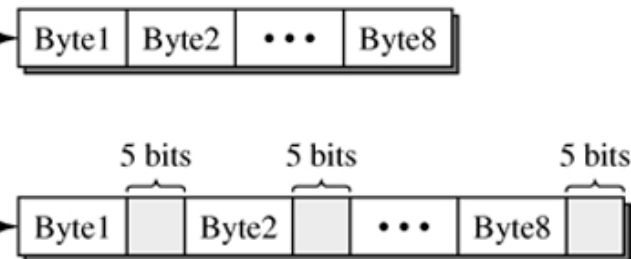
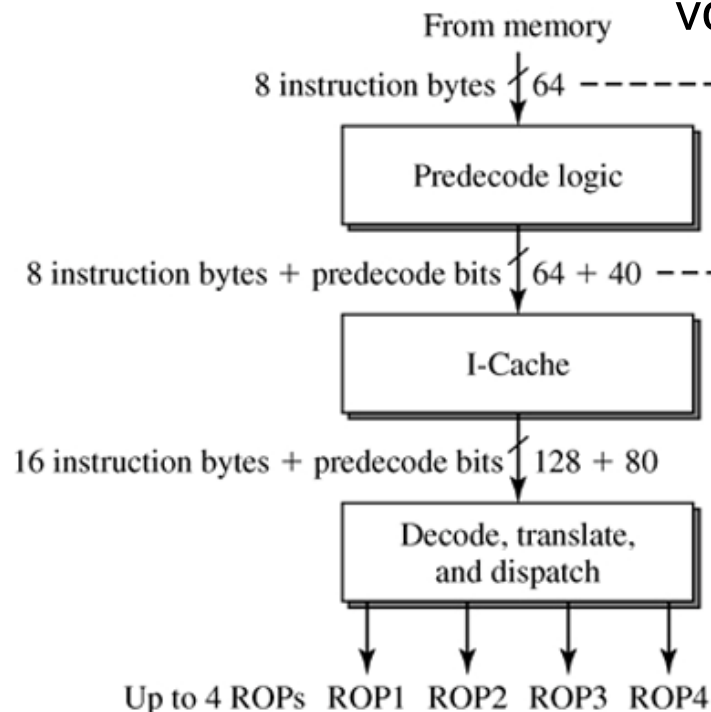
Αποκωδικοποίηση εντολών (Instruction Decode)

- Λύση:

- Μερική αποκωδικοποίηση των εντολών πριν την είσοδό τους στην I-cache (**predecoding**)

- ❖ Αύξηση του I-cache miss penalty

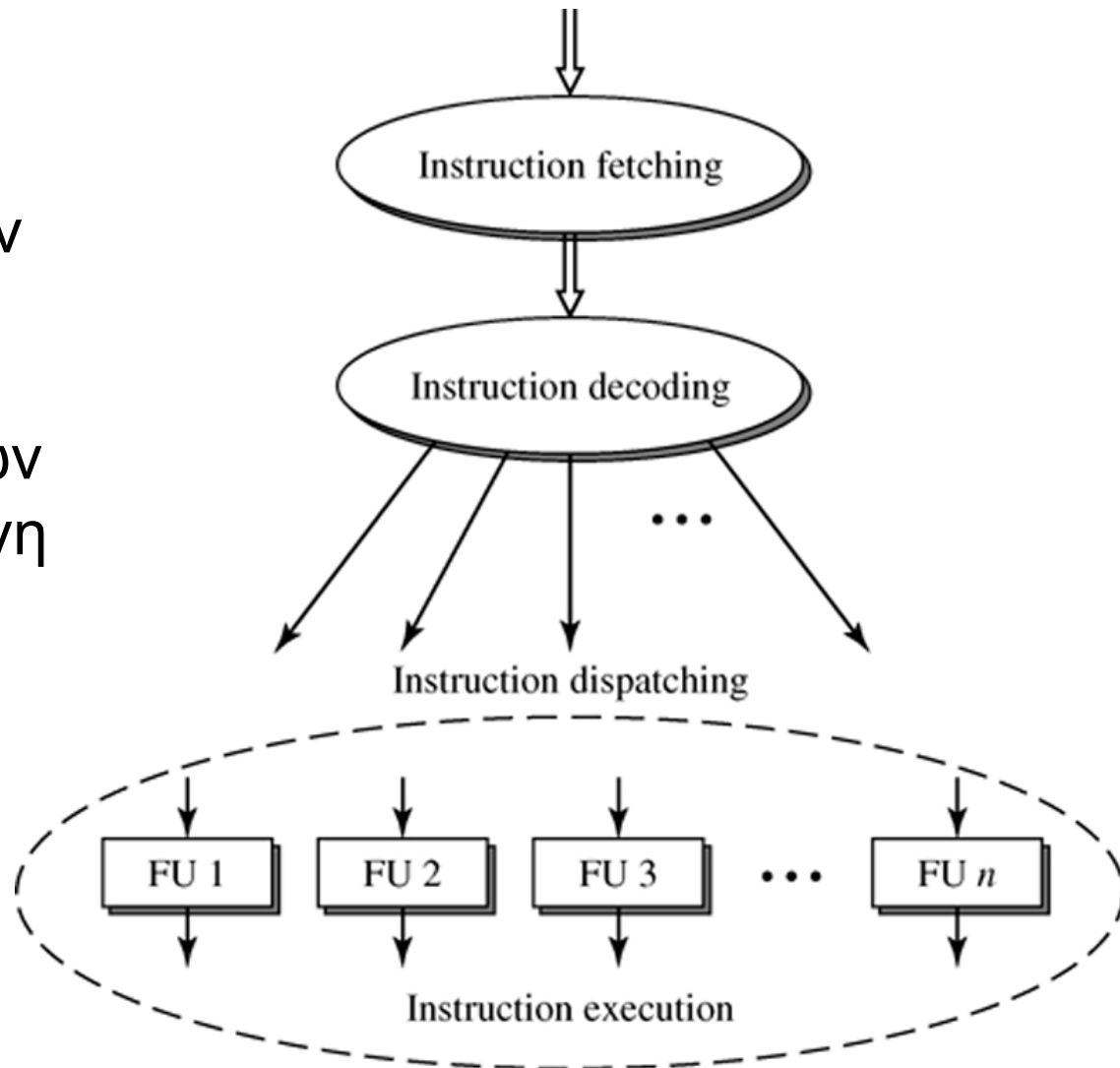
- ❖ Αύξηση του μεγέθους της I-cache (για να συμπεριληφθούν τα predecoded bits)



Ο μηχανισμός πρωθύστερης αποκωδικοποίησης του AMD K5

Διανομή των εντολών – Instruction Dispatch

- Το στάδιο αυτό αναλαμβάνει τη μεταγωγή από την κεντρική (centralized) διαχείριση εντολών στην κατανεμημένη εκτέλεσή τους

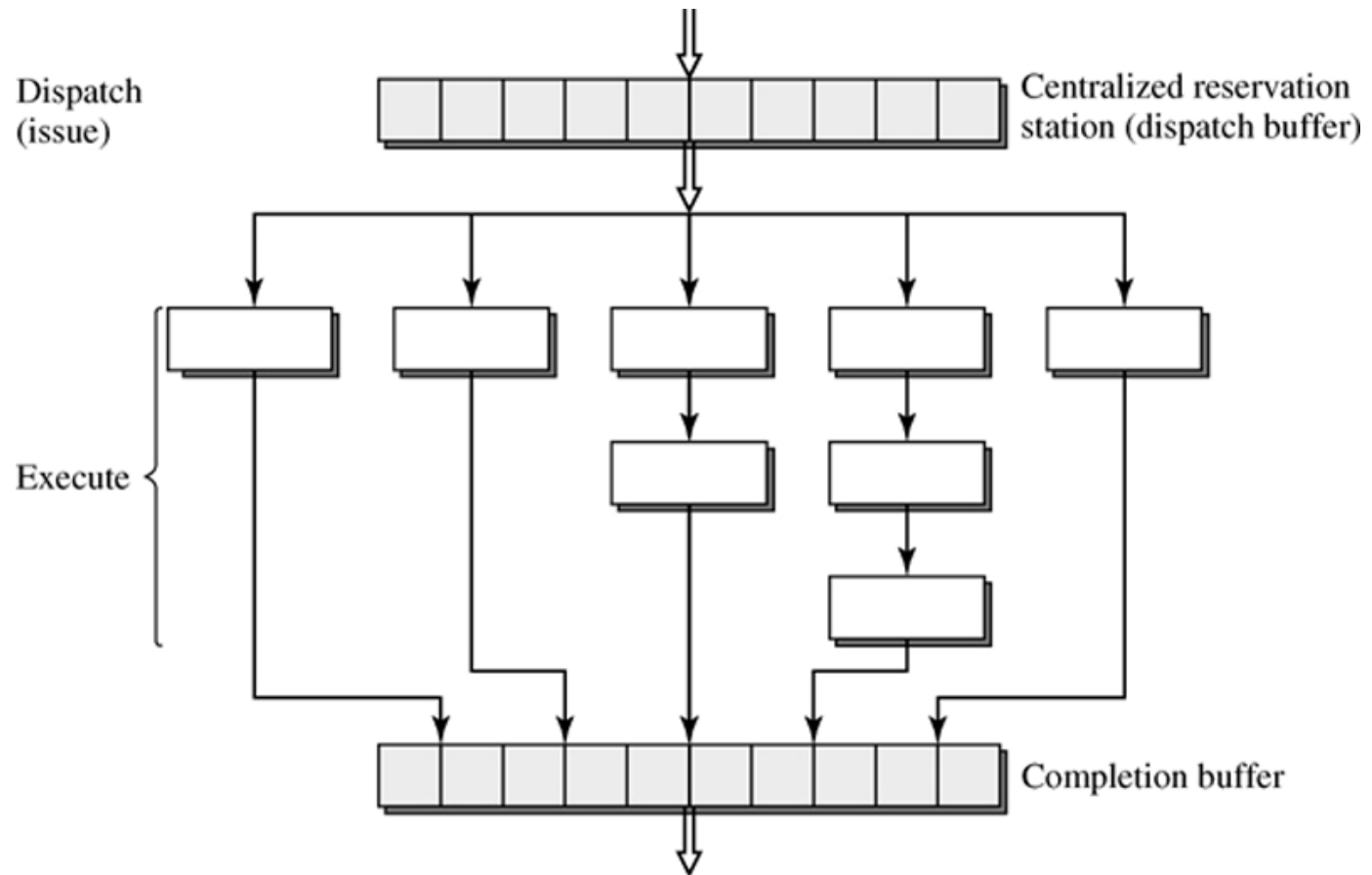


Διανομή των εντολών – Instruction Dispatch

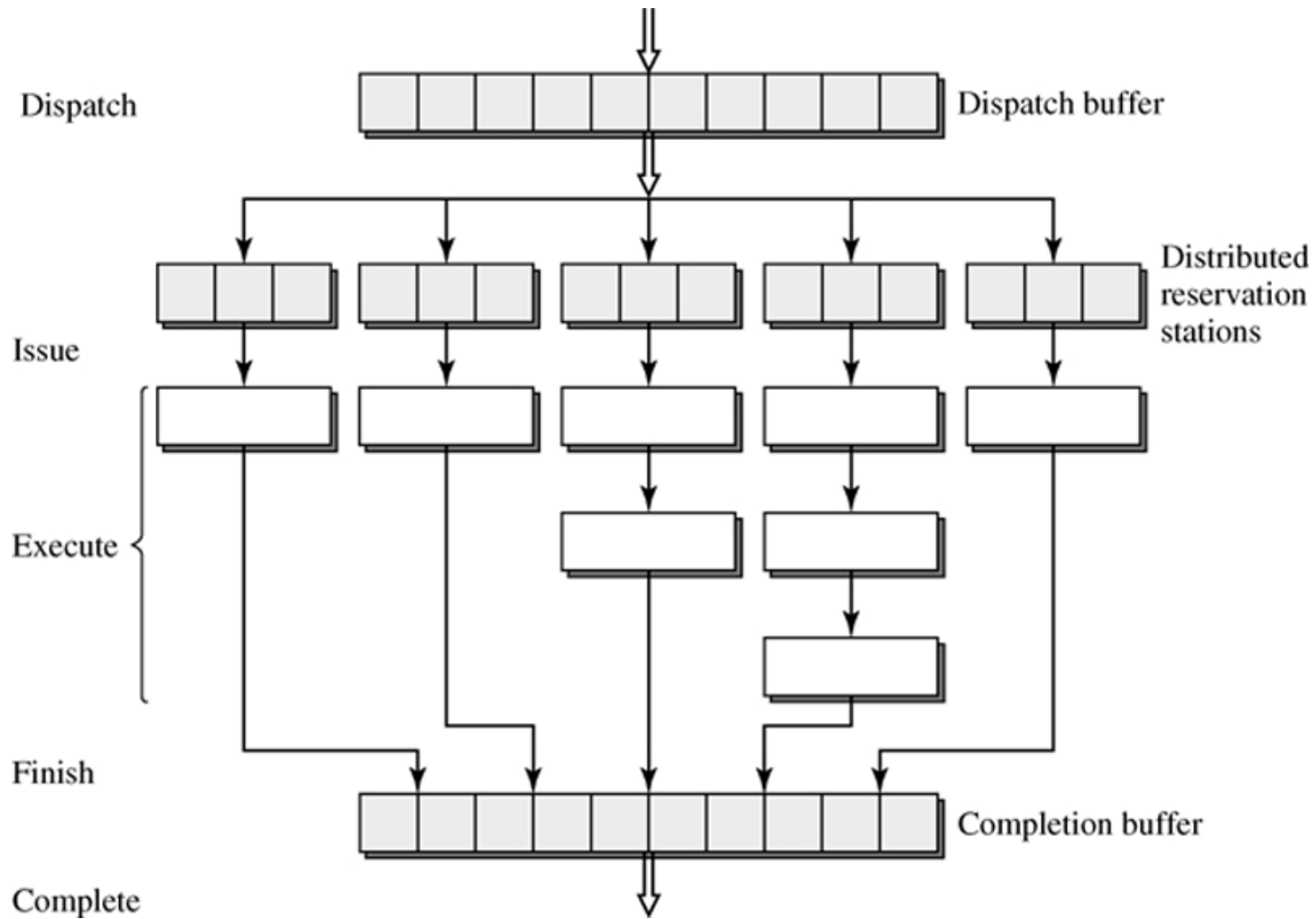
- **Reservation station**: προσωρινός καταχωρητής αποθήκευσης των αποκωδικοποιημένων εντολών που δεν έχουν διαθέσιμα όλα τα ορίσματά τους
 - Κεντρικός καταχωρητής (**centralized reservation station**)
 - Για παράλληλες σωληνώσεις
 - Κατανεμημένοι καταχωρητές (**distributed reservation station**)
 - Για ετερογενείς σωληνώσεις

Centralized reservation station

*Ενοποίηση
των σταδίων
dispatch και
issue*



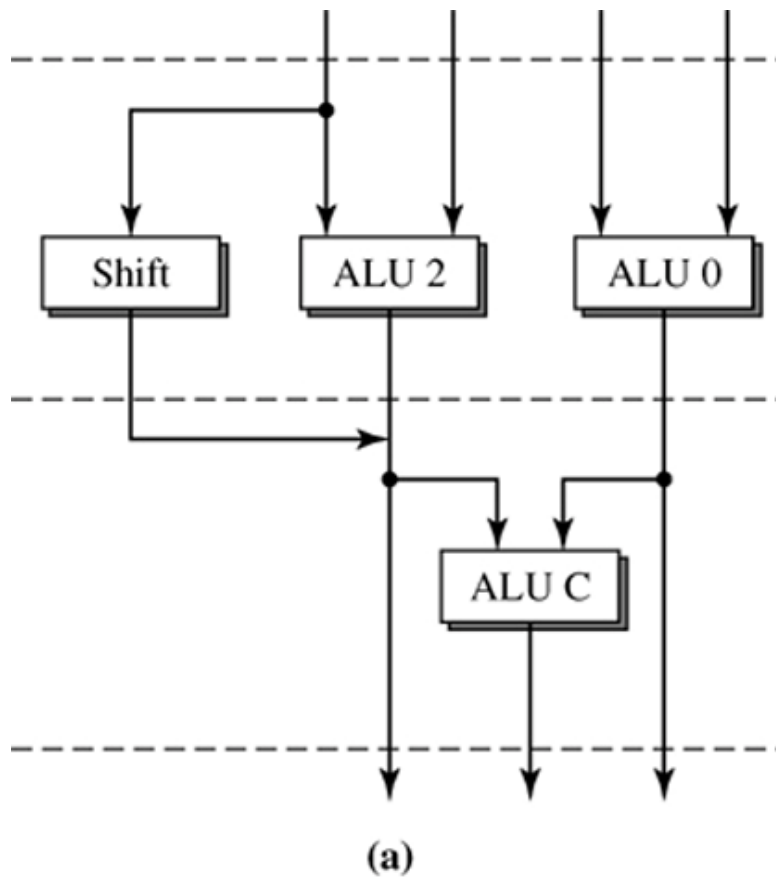
Distributed reservation station



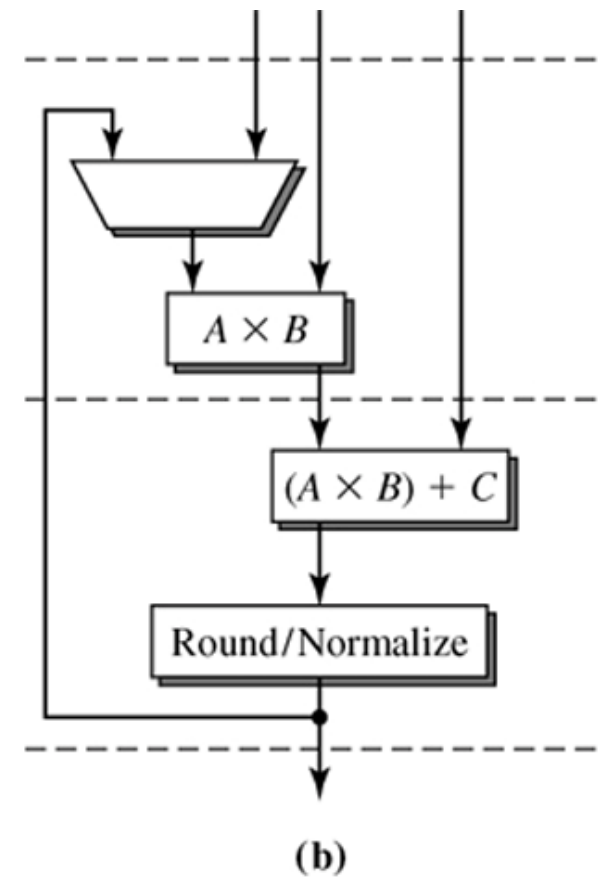
Εκτέλεση εντολών – Instruction Execution

- Σύγχρονες τάσεις:
 - Πολλές παράλληλες σωληνώσεις (δύσκολη η out-of-order εκτέλεση με bypassing εντολών)
 - Διαφοροποιημένες μεταξύ τους σωληνώσεις
 - Βαθιές σωληνώσεις
- Συνήθης καταμερισμός (δεν ακολουθεί τη στατιστική αναλογία των προγραμμάτων σε τύπους εντολών):
 - 4 μονάδες ALU
 - 1 μονάδα διακλάδωσης (μπορεί να εκτελέσει θεωρητικά (speculatively) > 1 εντολές διακλάδωσης)
 - 1 μονάδα ανάγνωσης/εγγραφής στη μνήμη (πολύ πολύπλοκη η υλοποίηση μνημών πολλαπλών εισόδων-εξόδων, μόνο με πολλαπλά banks)
 - Περισσότερες ειδικευμένες (και πιο αποδοτικές στην επίδοση) λειτουργικές μονάδες

Εκτέλεση εντολών – Instruction Execution



TI SuperSPARC



IBM RS/6000

Ολοκλήρωση και Αποδέσμευση εντολών – Instruction Completion and Retiring

- **Instruction Completion**: ενημέρωση της κατάστασης του μηχανήματος (machine state update)
- **Instruction Retiring**: ενημέρωση της μνήμης (memory state update)
- Αν η εντολή δεν περιλαμβάνει ενημέρωση της μνήμης, μετά το στάδιο ολοκλήρωσης, η εντολή αποδεσμεύεται

Ολοκλήρωση και Αποδέσμευση εντολών – Instruction Completion and Retiring

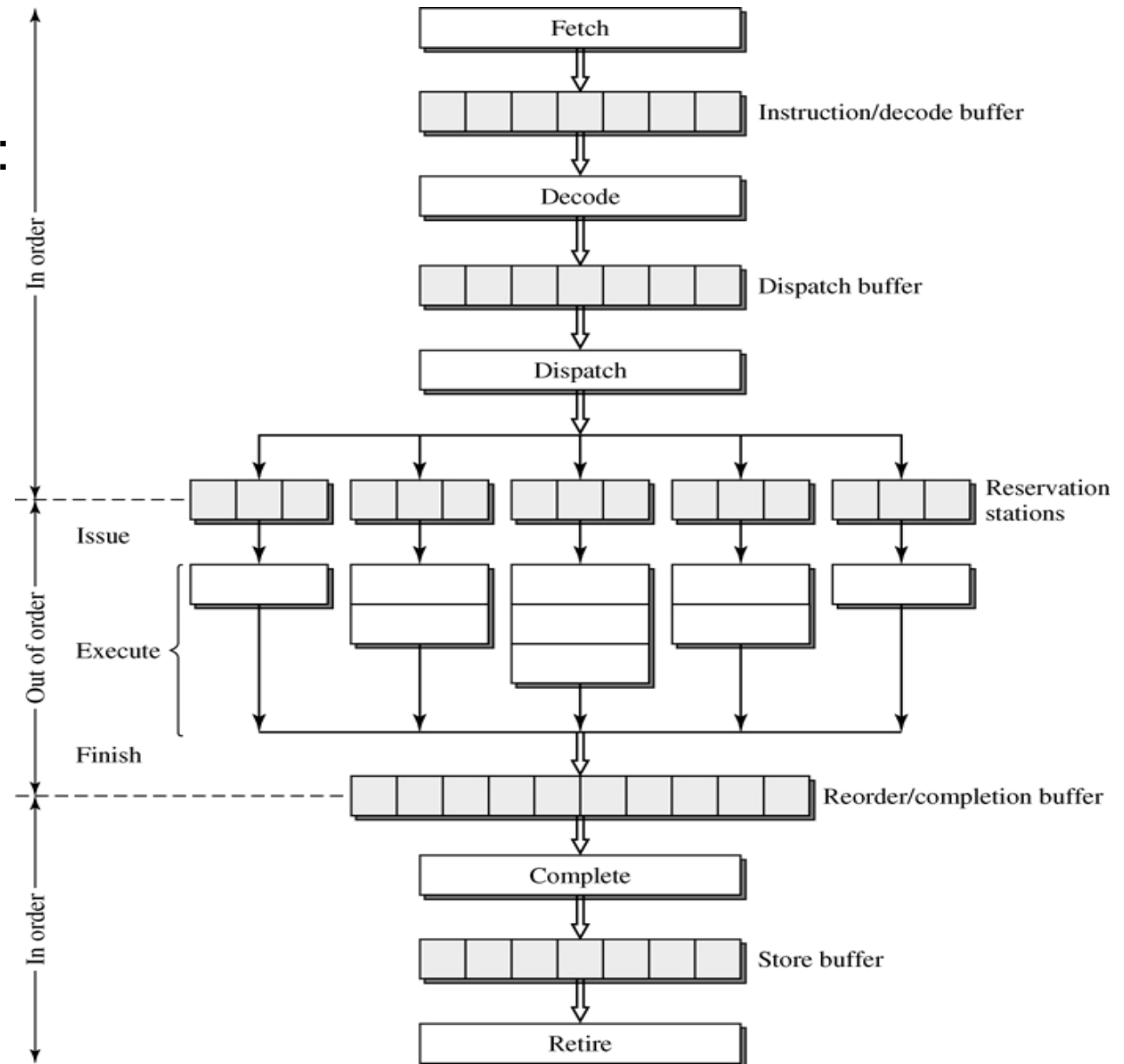
- Τα στάδια ολοκλήρωσης και αποδέσμευσης των εντολών πρέπει να εκτελέσουν τις εντολές εν σειρά
 - coherence και consistency μνήμης

- Λύση:
 - Reorder buffer
 - Store buffer
 - Load queue snooping

Ολοκλήρωση και Αποδέσμευση εντολών – Instruction Completion and Retiring

- Τα στάδια εντολών:

- Fetch
- Decode
- Dispatch
- Issue
- Execute
- Finish
- Complete
- Retire



Τεχνικές Βελτιστοποίησης της Επίδοσης

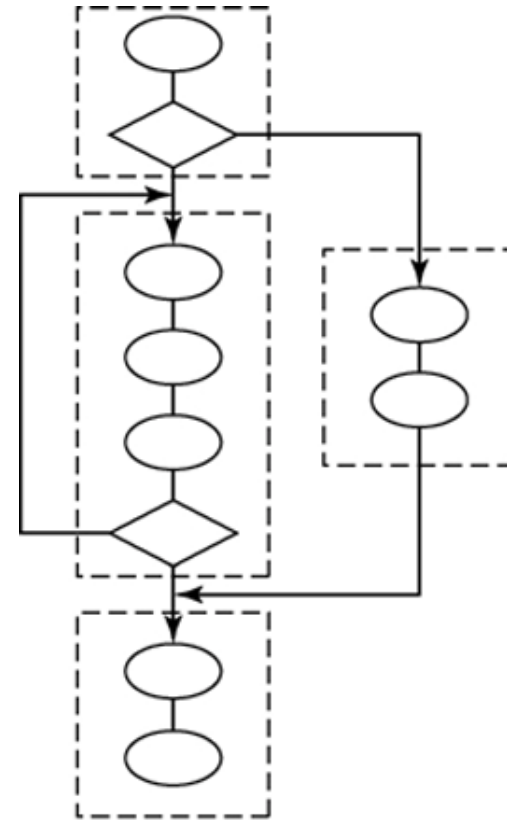
- Τρεις οι βασικοί τύποι εντολών: branch, ALU, load/store
- Η βελτιστοποίηση της επίδοσης απαιτεί την ελαχιστοποίηση των branch, ALU, load penalties. Δηλαδή την εξασφάλιση της συνεχιζόμενης:
 - Ροής εντολών (για την επεξεργασία των εντολών branch)
 - Ροής δεδομένων που βρίσκονται στους καταχωρητές (για την επεξεργασία των εντολών ALU)
 - Ροής δεδομένων μνήμης (για την εξέλιξη των εντολών load/store)

Τεχνικές που επηρεάζουν τη Ροή Εντολών

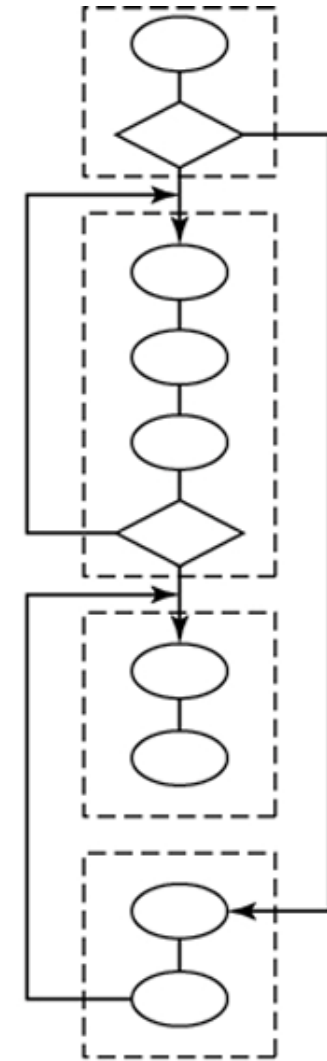
- Αφορούν την τροφοδοσία των αρχικών σταδίων του αγωγού (ανάγνωση-αποκωδικοποίηση) με εντολές και επομένως το throughput αυτό επηρεάζει τη ροή και των επόμενων σταδίων
 - Εξαρτήσεις και ροή ελέγχου προγράμματος
 - Μείωση της επίδοσης λόγω εντολών άλματος
 - Τεχνικές πρόβλεψης άλματος
 - Ανάνηψη από λάθος πρόβλεψη άλματος
 - Προηγμένες τεχνικές πρόβλεψης άλματος

Εξαρτήσεις και ροή ελέγχου προγράμματος

- Η διακοπή της ακολουθιακής ροής εντολών μπορεί να εισάγει καθυστερήσεις (stalls) στον αγωγό, μειώνοντας το ρυθμό ανάγνωσης και επομένως εκτέλεσης των εντολών



Γράφος Ελέγχου Ροής

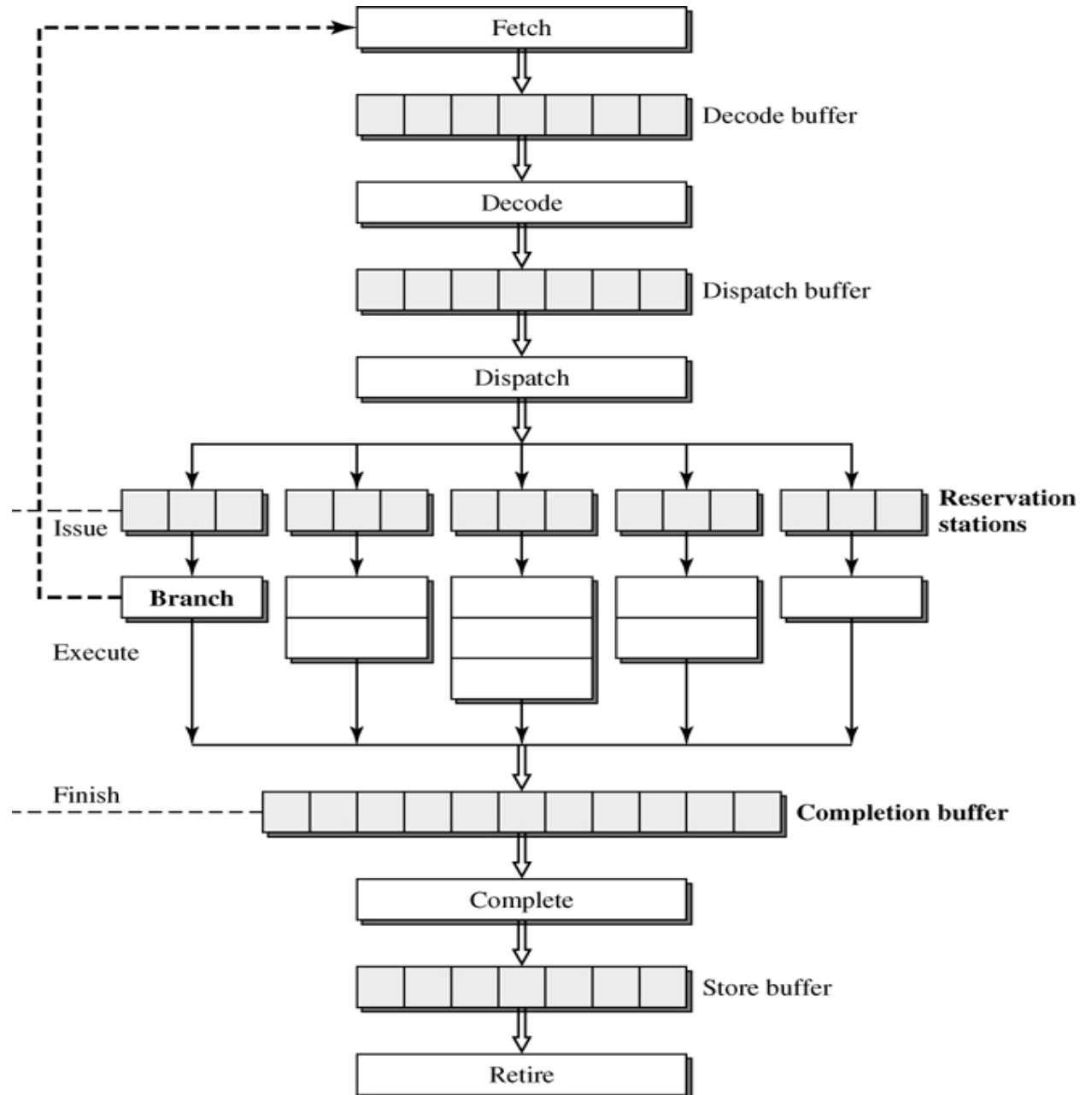


Αντιστοίχιση του γράφου σε σειριακές θέσεις μνήμης

Μείωση της επίδοσης λόγω εντολών άλματος

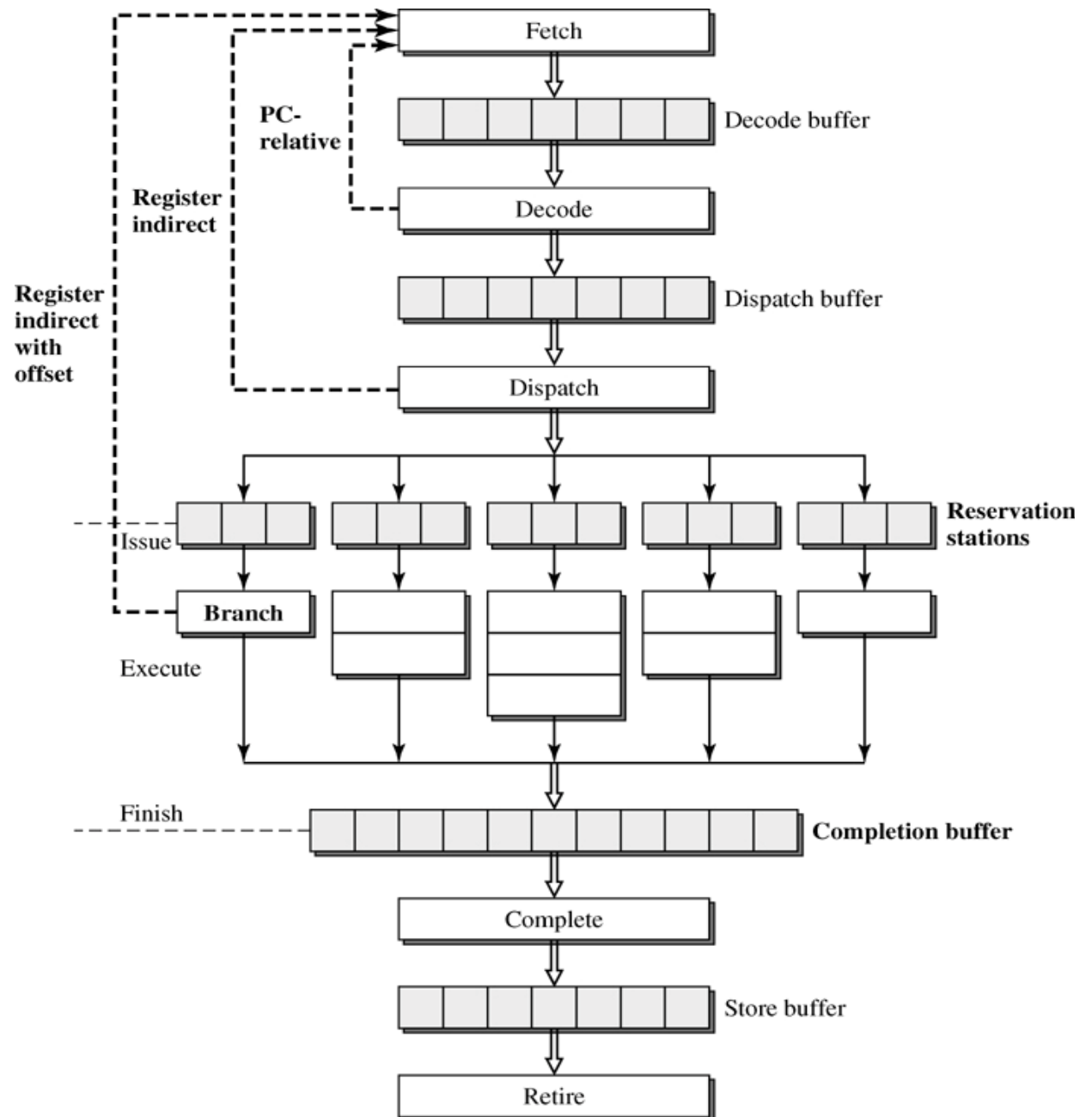
- Μέγιστο throughput όταν φορτώνονται διαδοχικές εντολές από τη μνήμη

Τι συμβαίνει μετά από μία εντολή άλματος;



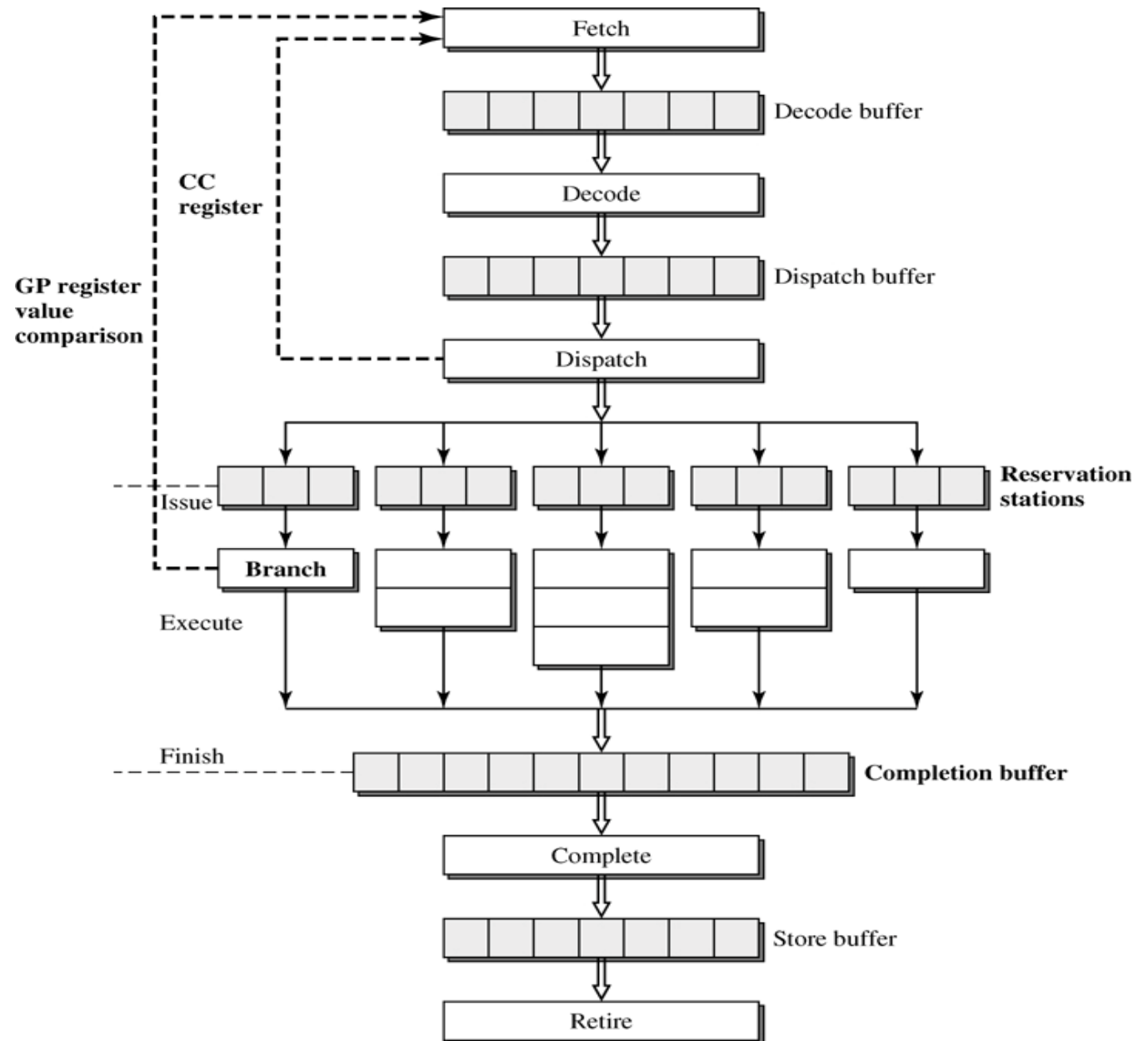
Μείωση της επίδοσης λόγω εντολών άλματος

- Branch penalties ανάλογα με το είδος της εντολής άλματος
- Αριθμός των «bubbles» που εισάγεται στον αγωγό = (βάθος που λαμβάνεται η απόφαση άλματος) × (πλάτος της σωλήνωσης)



Μείωση της επίδοσης λόγω εντολών άλματος

- Διαφορετικές μέθοδοι ανάλυσης των εντολών άλματος



Τεχνικές Πρόβλεψης Άλματος

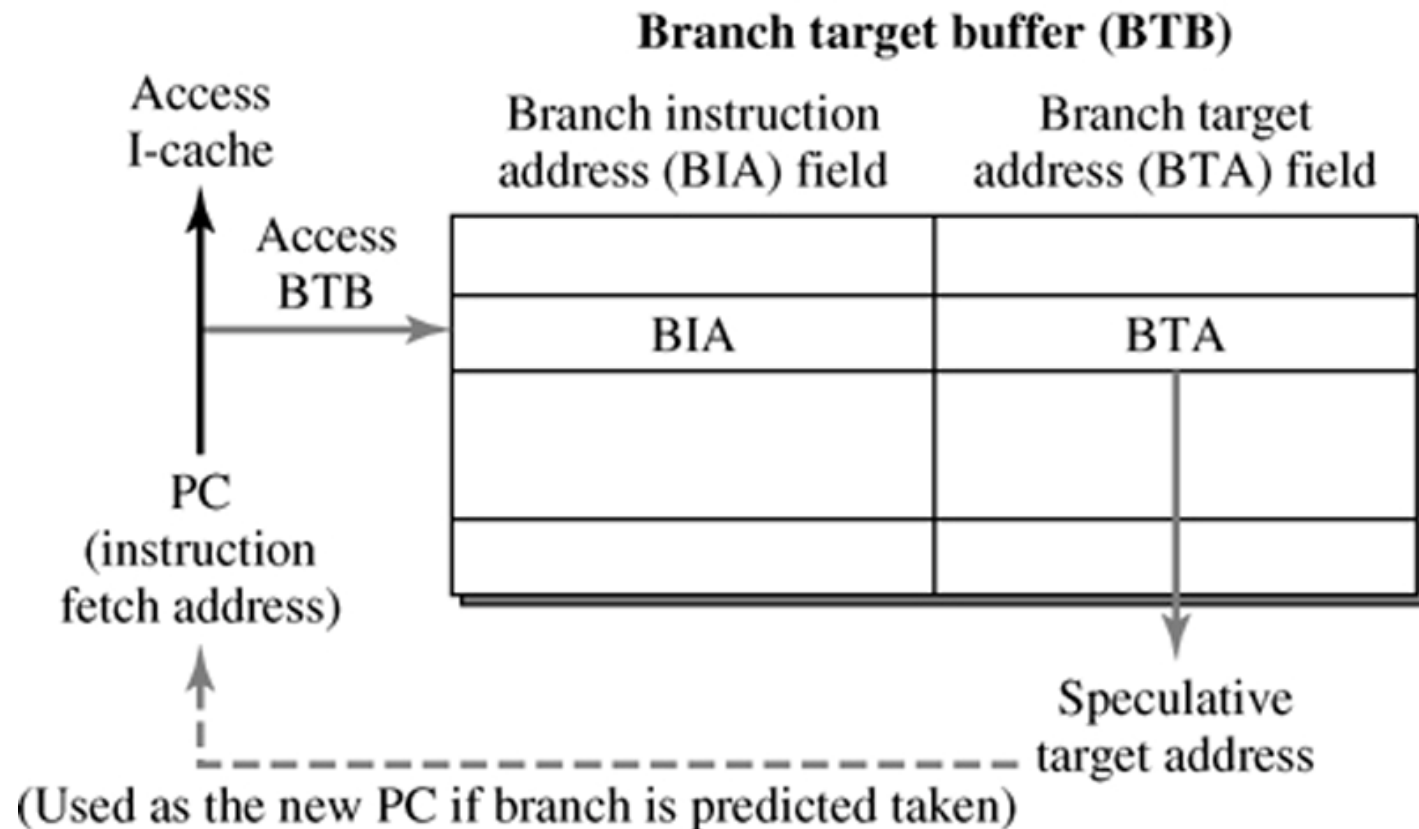
Εύρεση της διεύθυνσης-στόχου του άλματος

- **BTB (branch target buffer)**: μία μικρή (πλήρως συσχετιστική) κρυφή μνήμη που προσπελάζεται κατά τη διάρκεια του σταδίου ανάγνωσης εντολών
- Περιέχει:
 - **BIA (branch instruction address)**
 - **BTA (branch target address)**

Τεχνικές Πρόβλεψης Άλματος

Εύρεση της διεύθυνσης-στόχου του άλματος

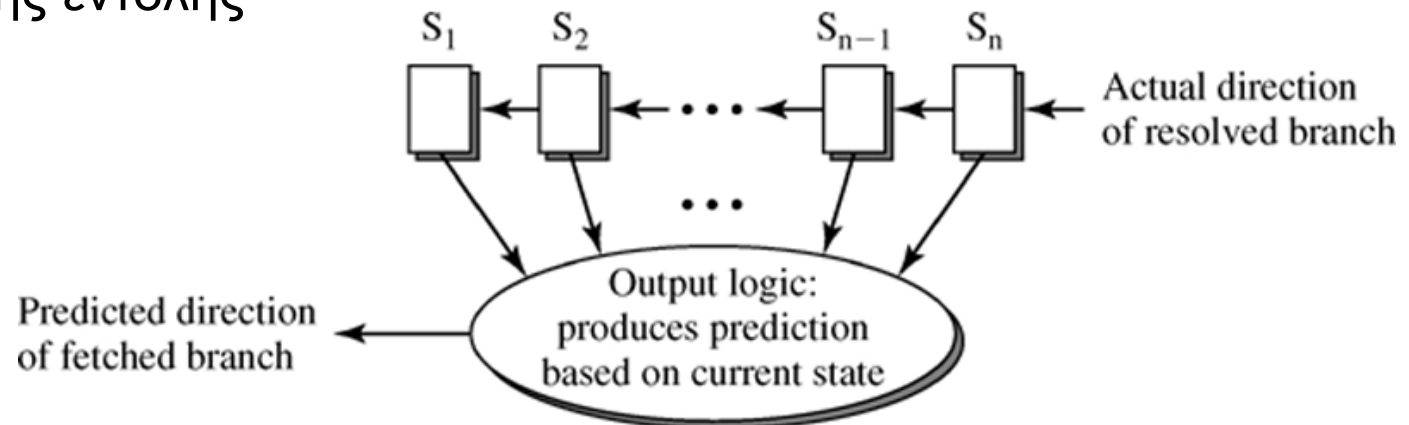
- Με προσπέλαση του BTB, αν η πρόβλεψη άλματος είναι σωστή, δεν θα εισαχθούν καθυστερήσεις στον αγωγό



Τεχνικές Πρόβλεψης Άλματος

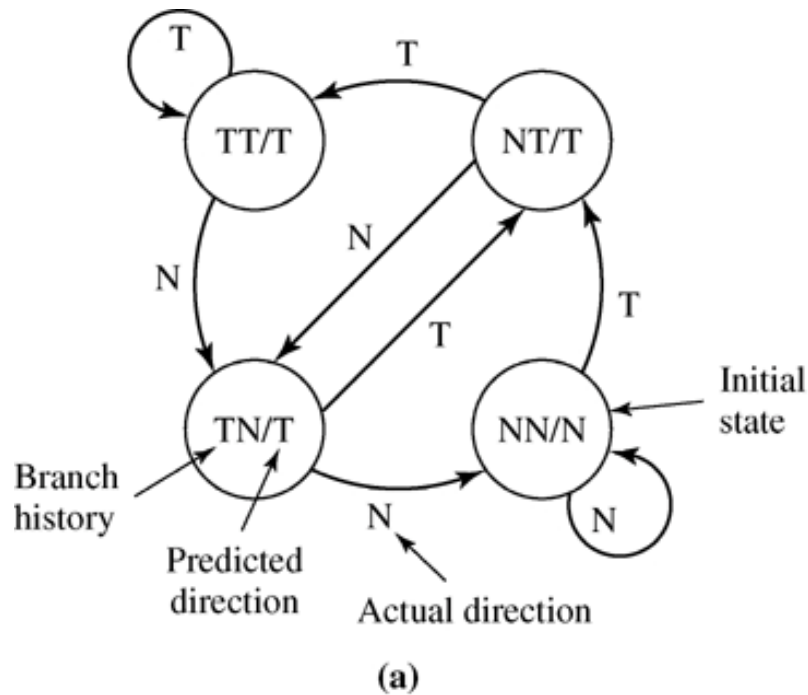
Απόφαση άλματος

- Σταθερή απόφαση: branch not taken
- Επιπλέον bits από τον compiler: hints σχετικά με την απόφαση άλματος (π.χ. στο τέλος των for loops) – στατική πρόβλεψη
- Η σχετική θέση της διεύθυνσης στόχου καθορίζει την πρόβλεψη της απόφασης άλματος (θετικό offset - branch not taken, αρνητικό offset (συνήθως σε loops) – branch taken) – IBM RS/6000
- Πρόβλεψη άλματος με βάση το ιστορικό των προηγούμενων εκτελέσεων της εντολής

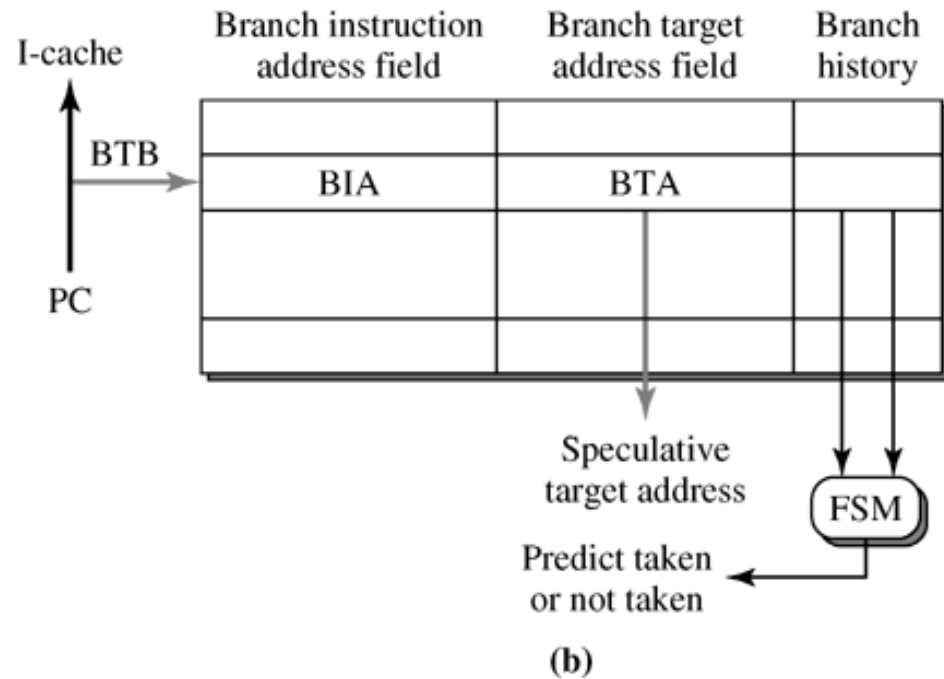


Τεχνικές Πρόβλεψης Άλματος

Απόφαση άλματος



Πρόβλεψη άλματος με αλγόριθμο 2-bits



Επέκταση του BTB για να συμπεριληφθούν και τα bits πρόβλεψης άλματος

Τεχνικές Πρόβλεψης Άλματος

Απόφαση άλματος

- 6 διαφορετικοί αλγόριθμοι πρόβλεψης διακλάδωσης των 2-bit

- τα ποσοστά επιτυχίας τους για τα συγκεκριμένα μετρο-προγράμματα που εκτελέστηκαν

- η βέλτιστη δυνατή πρόβλεψη

<u>Benchmark</u>	<u>Optimal</u>	<u>“Counter”</u>
------------------	----------------	------------------

spice2g6	97.2	97.0
----------	------	------

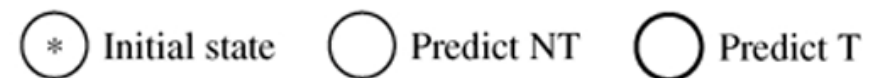
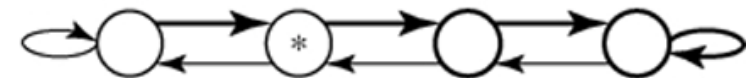
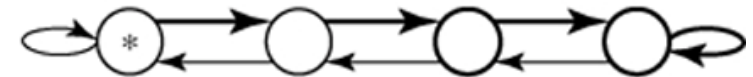
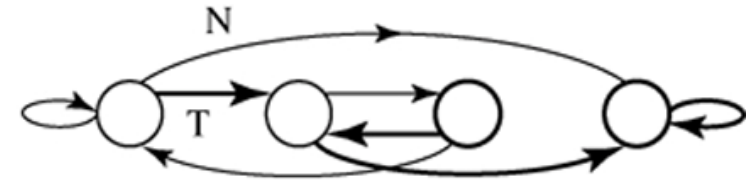
doduc	94.3	94.3
-------	------	------

gcc	89.1	89.1
-----	------	------

espresso	89.1	89.1
----------	------	------

li	87.1	86.8
----	------	------

eqntott	87.9	87.2
---------	------	------

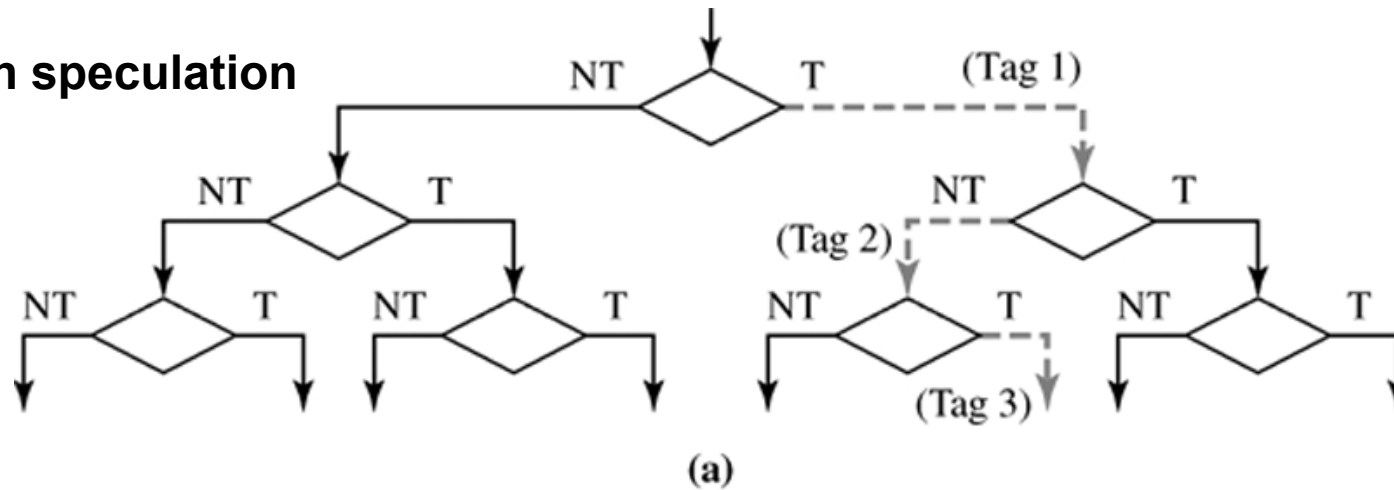


Ανάληψη από λάθος πρόβλεψη άλματος

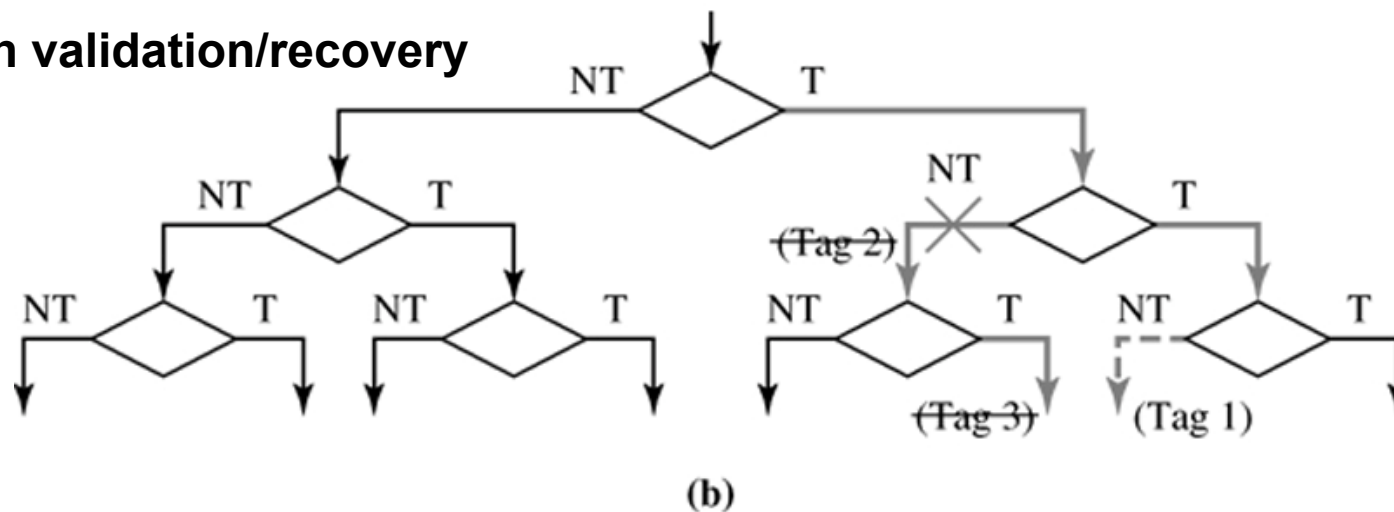
- **Branch speculation** (υποθετική εκτέλεση άλματος): πρόβλεψη της κατεύθυνσης του άλματος και ανάγνωση των εντολών που βρίσκονται στο προβλεπόμενο μονοπάτι εκτέλεσης (a)
 - Κάθε ομάδα εντολών που ακολουθεί μία υπόθεση άλματος, φέρει την ίδια επικεφαλίδα (tag)
- **Branch validation** (επικύρωση υπόθεσης άλματος): όταν ληφθεί η απόφαση άλματος, μπορεί πλέον να καθοριστεί η ορθότητα της υποθετικής εκτέλεσης (b)
 - Αν μία υπόθεση αποδειχθεί λανθασμένη, τερματίζεται η λανθασμένη ροή εκτέλεσης εντολών (με βάση την ένδειξη των επικεφαλίδων διαγράφονται οι εντολές που έχουν ήδη φορτωθεί, καθαρίζονται τα περιεχόμενα των buffers και αποδесμεύονται οι αντίστοιχοι καταχωρητές)
 - Ξεκινάει νέο μονοπάτι εκτέλεσης
 - Ενημερώνεται το περιεχόμενο του BTB

Ανάληψη από λάθος πρόβλεψη άλματος

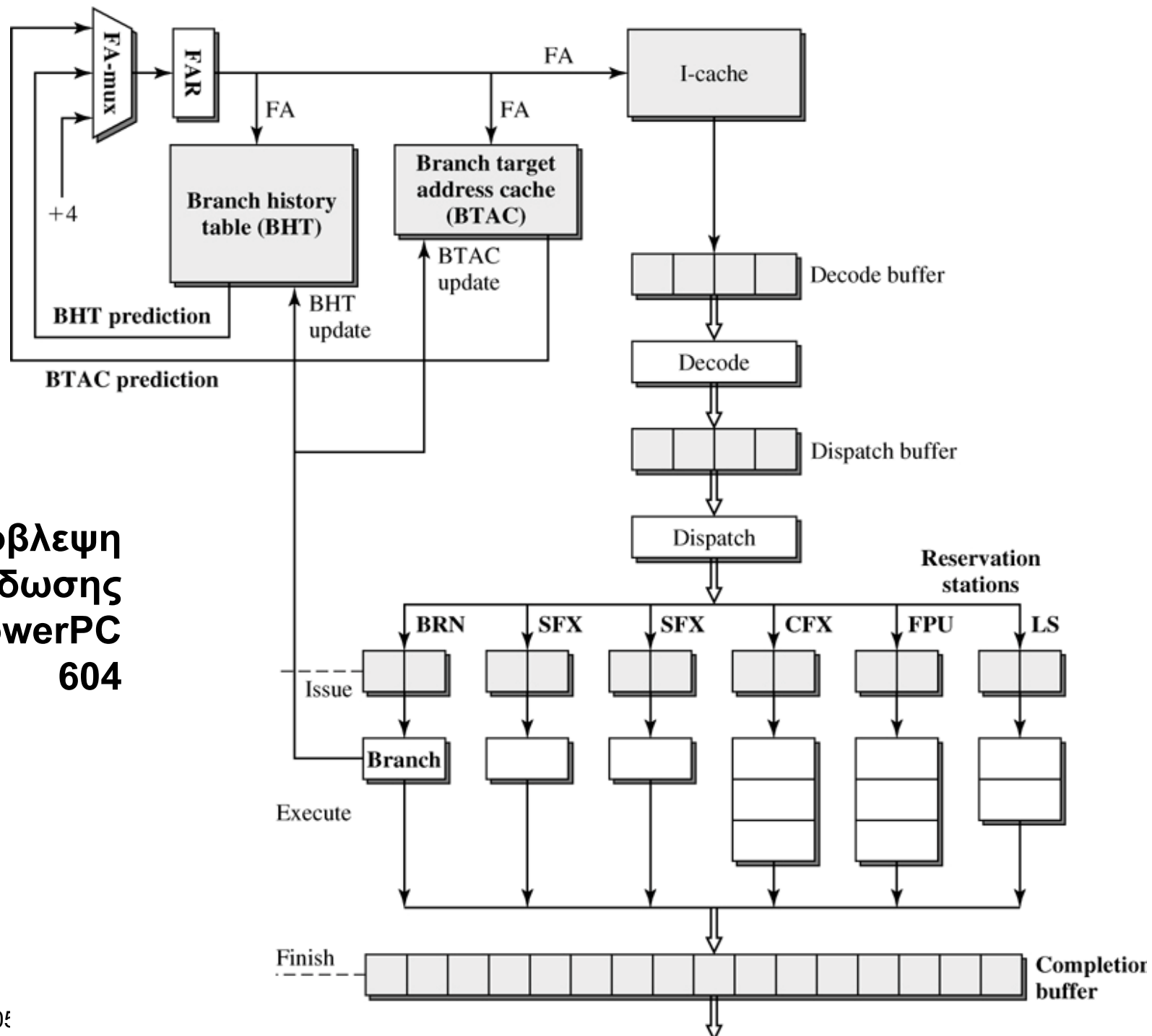
Branch speculation



Branch validation/recovery



Πρόβλεψη διακλάδωσης στον PowerPC 604

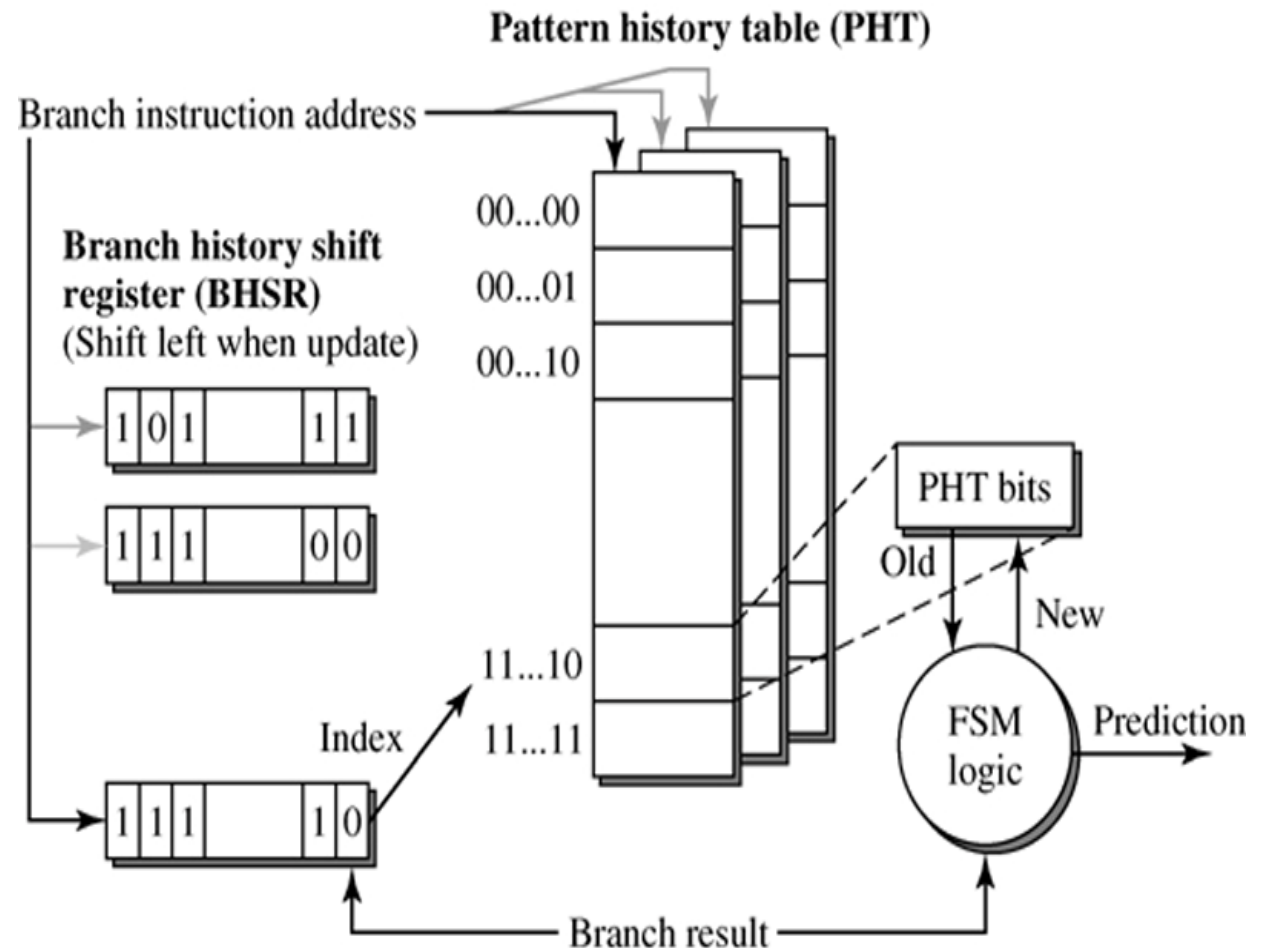


Προηγμένες τεχνικές πρόβλεψης άλματος

- Πιο έγκυρη πρόβλεψη επιτυγχάνουν οι αλγόριθμοι που λαμβάνουν υπόψη το ιστορικό και των άλλων συσχετιζόμενων εντολών διακλάδωσης

- **Pattern history table (PHT)** -πίνακας ιστορικού με αντίγραφα- 2 επιπέδων

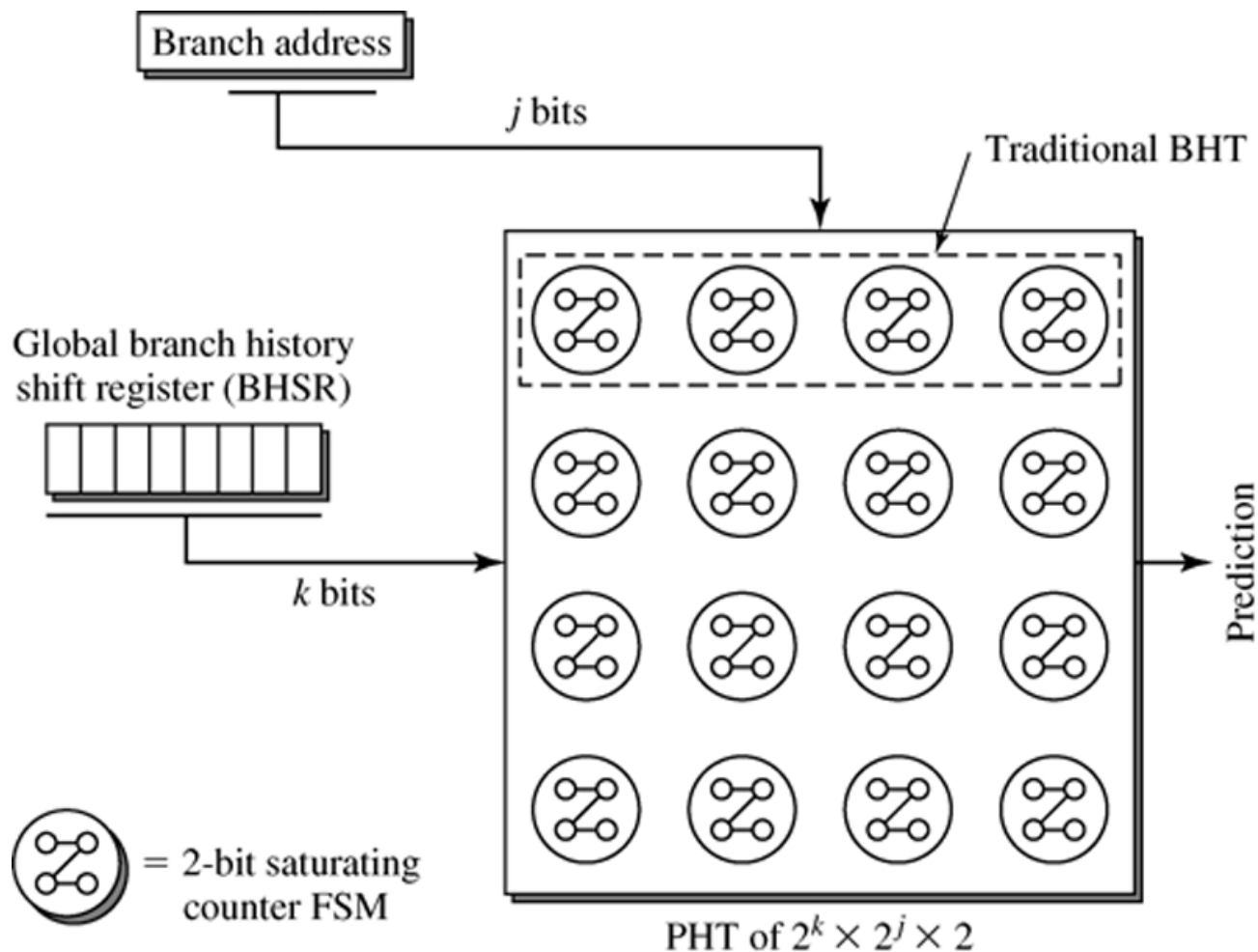
- **Branch History Shift Register (BHSR)** – κάθε register δείχνει σε μία από τις σχετικές εγγραφές του PHT



Προηγμένες τεχνικές πρόβλεψης άλματος

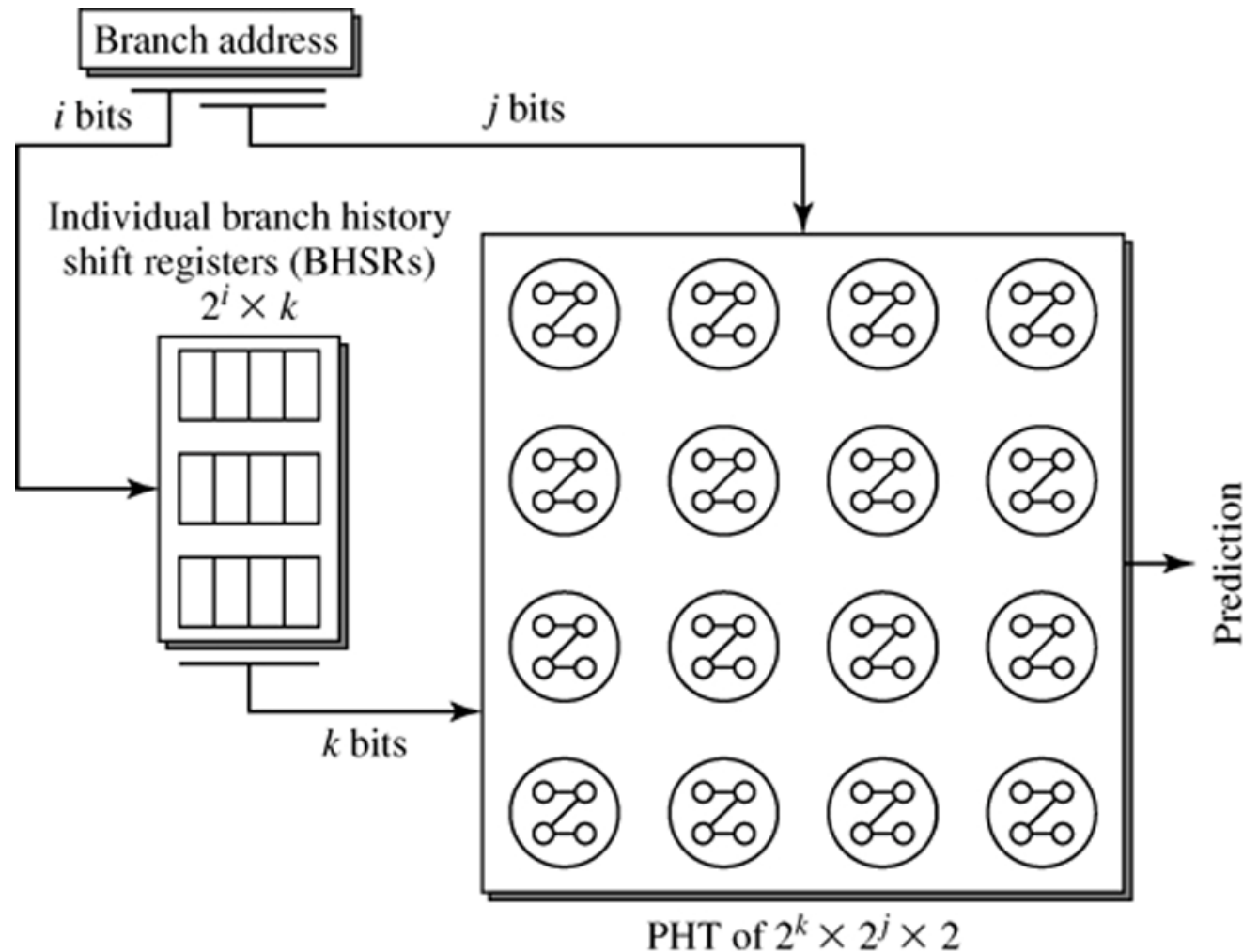
Πρόβλεψη άλματος με συσχέτιση
(1 global BHSR και 1 shared PHT)

- BHSR: καταγράφει την κατεύθυνση άλματος των k τελευταίων εντολών
- PHT: επιλέγεται μία εγγραφή του πίνακα, βάση του BHSR



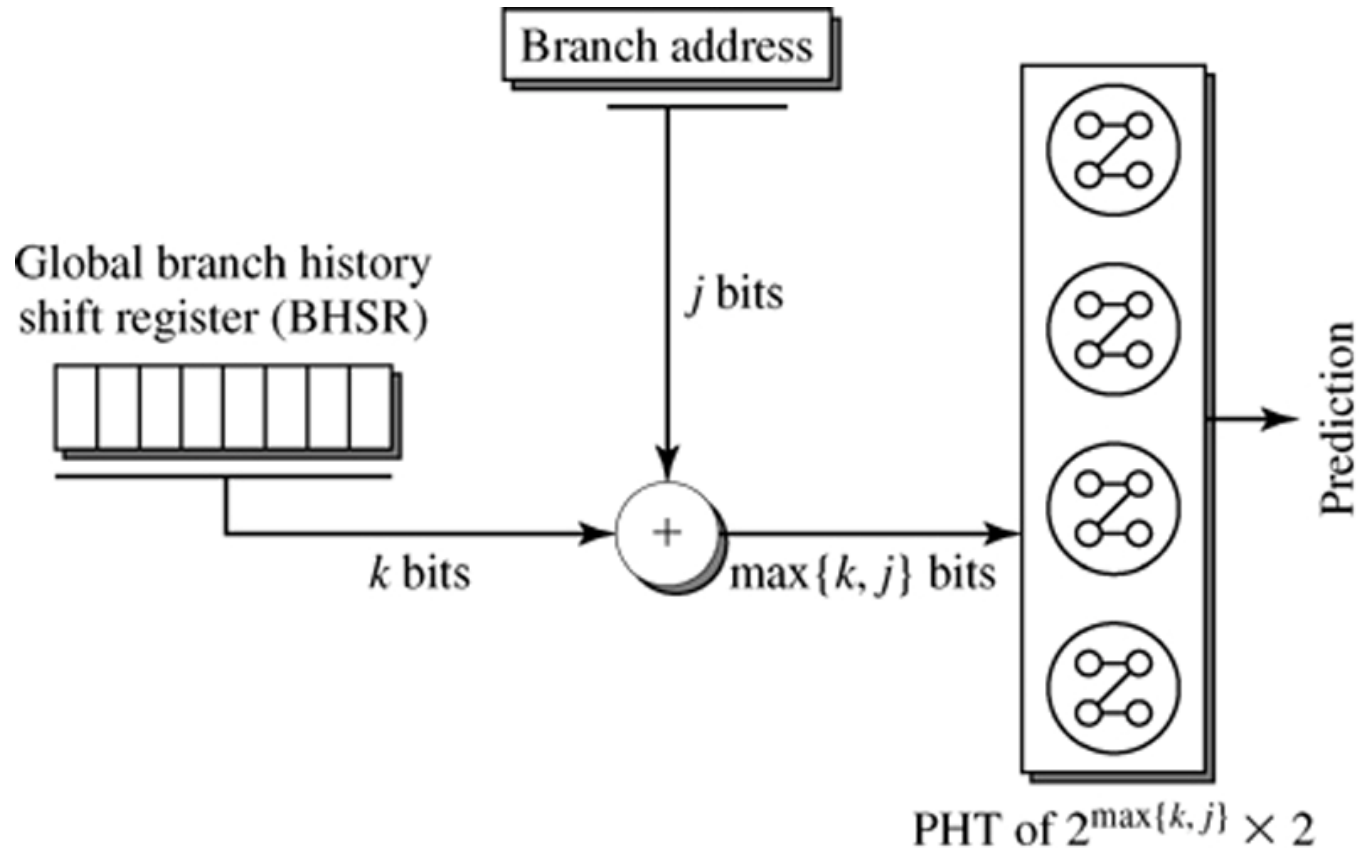
Προηγμένες τεχνικές πρόβλεψης άλματος

Πρόβλεψη άλματος με συσχέτιση
(1 per branch BHSR και 1 shared PHT)



Προηγμένες τεχνικές πρόβλεψης άλματος

Πρόβλεψη άλματος με συσχέτιση (gshare – McFarling, 1993)



Τεχνικές βελτίωσης της ροής δεδομένων στους προσωρινούς καταχωρητές δεδομένων (registers)

- Για την εκτέλεση μίας εντολής τύπου ALU απαιτούνται:
 - Η λειτουργική μονάδα ALU
 - Να είναι διαθέσιμα τα περιεχόμενα των 2 προσωρινών καταχωρητών προς ανάγνωση (source registers)
 - Να είναι αδέσμευτος ο προσωρινός καταχωρητής προς εγγραφή (destination register)

•

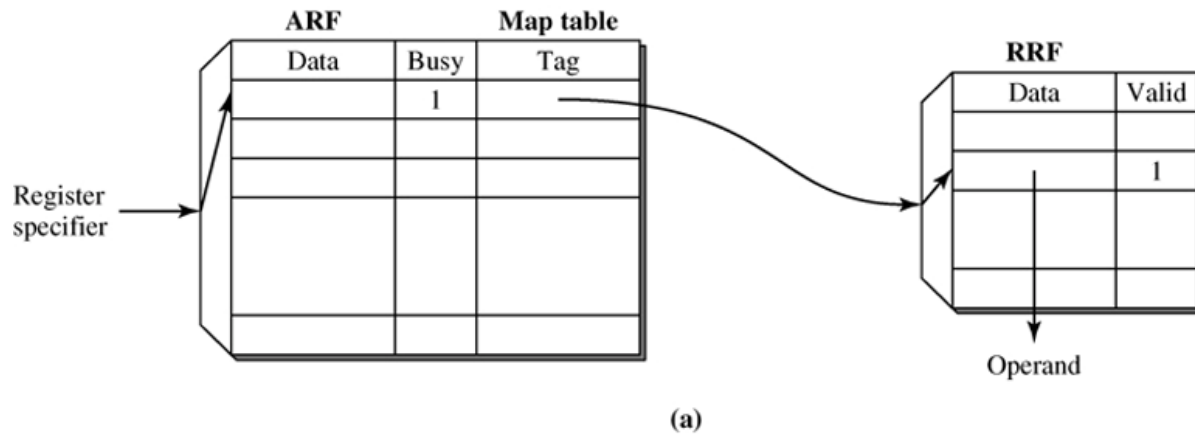
Τεχνικές βελτίωσης της ροής δεδομένων στους προσωρινούς καταχωρητές δεδομένων (registers)

- **Register recycling** (για την αντιστοίχιση των άπειρων -κατά τον compiler- καταχωρητών στον περιορισμένο αριθμό των φυσικών γίνεται ανακύκλωση των ίδιων φυσικών καταχωρητών)
- **Register renaming** (μετονομασία-χρήση πολλαπλών ονομάτων για τους ίδιους φυσικούς καταχωρητές ώστε να αποφευχθούν εξαρτήσεις τύπου WAR ή WAW)

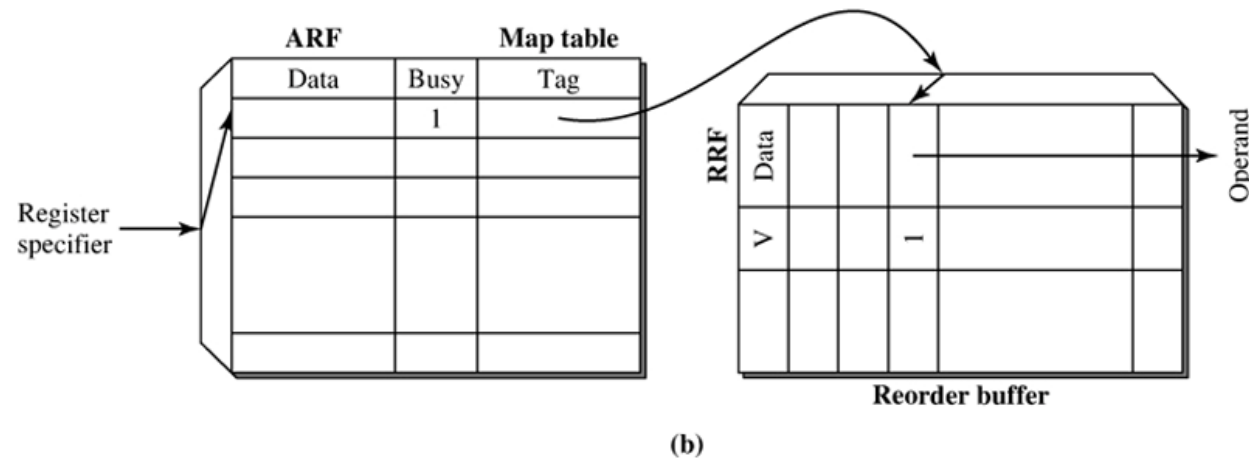
Register renaming

- Rename register file (RRF) : αρχείο των μετονομασμένων προσωρινών καταχωρητών
- Architected register file (ARF) : αρχείο των φυσικών προσωρινών καταχωρητών (το register file, όπως το εννοούσαμε μέχρι τώρα)
 - RRF απλό διπλότυπο του ARF
 - RRF με λιγότερες εγγραφές (χρειάζεται κάποιος πίνακας αντιστοίχισης)

Register renaming



Το RRF ως δομή ανεξάρτητη του ARF



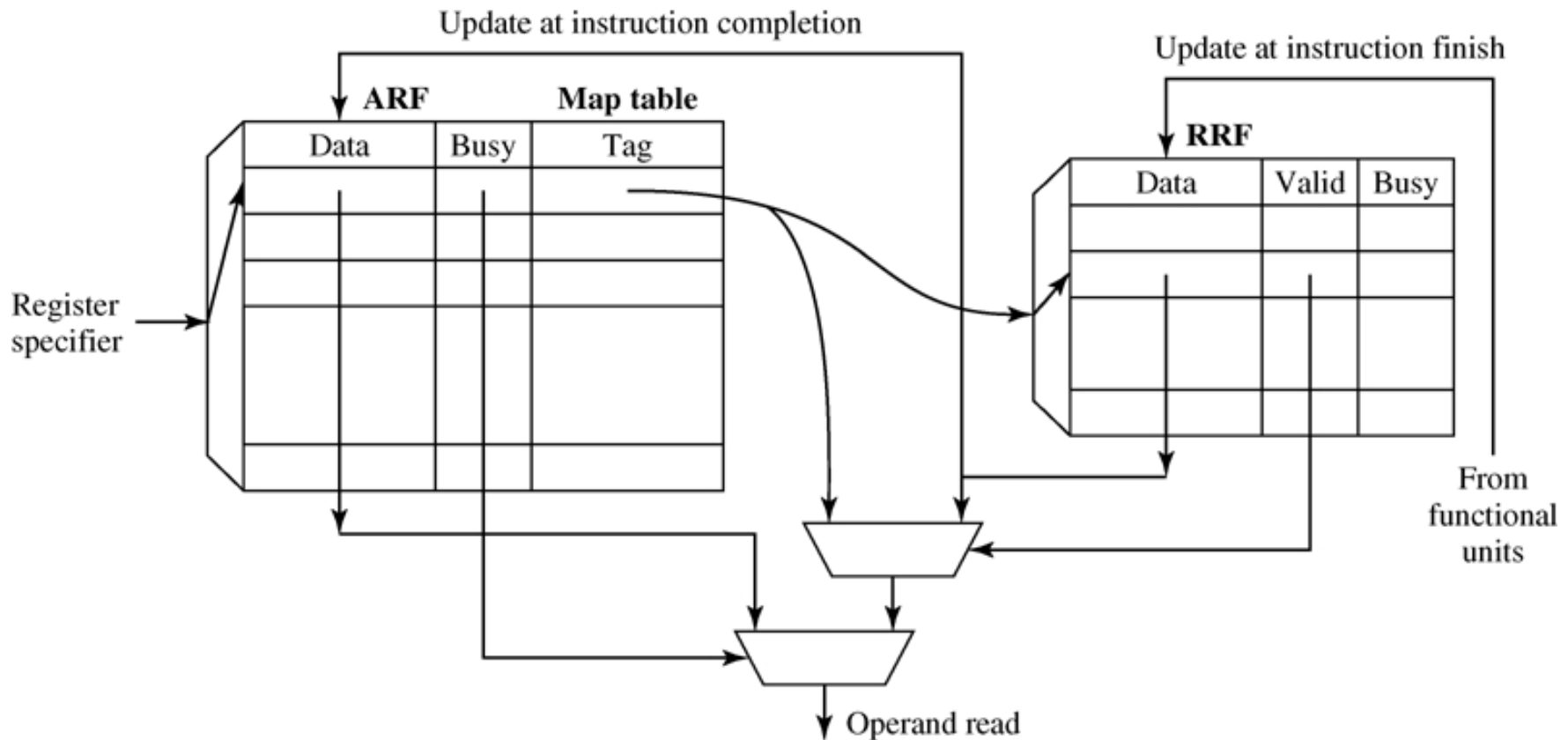
*Ενσωμάτωση του RRF στον reorder buffer (ένας rename register δεσμεύεται για κάθε εντολή που είναι **in flight**)*

Register renaming

- Τρεις διαδικασίες πραγματοποιούνται κατά την μετονομασία καταχωρητών:
 - Ανάγνωση από τους καταχωρητές πηγής (κατά την αποκωδικοποίηση -decode- ή διανομή -dispatch- της εντολής)
 - Δέσμευση του καταχωρητή προορισμού (κατά την αποκωδικοποίηση -decode- ή διανομή -dispatch- της εντολής) : θέτει το busy bit (για τη δήλωση της εκκρεμούσας εγγραφής), επιλέγει tag, ενημερώνει τον πίνακα αντιστοίχισης (για χρήση από επόμενες εντολές εξαρτώμενες από το αποτέλεσμα της εκκρεμούσας εγγραφής)
 - Ενημέρωση του καταχωρητή : πραγματοποιείται σε δύο στάδια – πρώτα η ενημέρωση του RRF και τέλος (μετά την in-order ολοκλήρωση της εντολής) η ενημέρωση του ARF

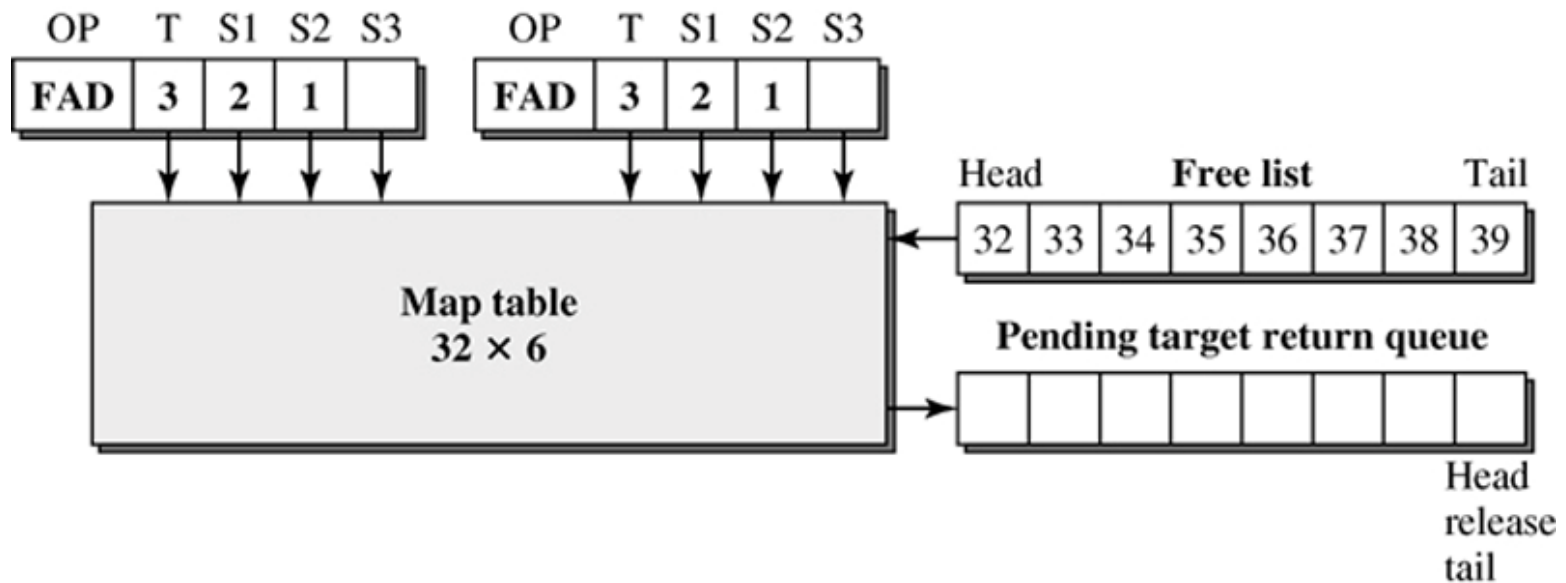
Register renaming

ανάγνωση, δέσμευση καταχωρητή προορισμού, ενημέρωση καταχωρητή



Register renaming

- IBM RS/6000
 - Η μονάδα κινητής υποδιαστολής περιέχει 40 φυσικούς καταχωρητές, από τους οποίους οι 32 χρησιμοποιούνται από το ISA



Register Renaming

- Παράδειγμα

```
i1:   add r1, r2, 10
i2:   mul r3, r1, 10
i3:   add r1, r4, 20
i4:   sub r5, r1, 20
```

i1 → i2 πραγματική εξάρτηση δεδομένων (read-after-write dependence)
i1 → i3 output dependence (write-after-write)
i2 → i3 anti-dependence (write-after-read) } artificial dependences

Register Renaming - Παράδειγμα

- Για να επιτρέψουμε την out-of-order εκτέλεση στην περίπτωση των *artificial dependencies*, πραγματοποιούμε register renaming

			<u>renamed registers</u>
i1:	add r1, r2, 10	—————→	add ra, rb, 10
i2:	mul r3, r1, 10	—————→	mul rc, ra, 10
i3:	add r1, r4, 20	—————→	add rf, rd, 20
i4:	sub r5, r1, 20	—————→	sub re, rf, 20

Διατήρηση των true dependencies:

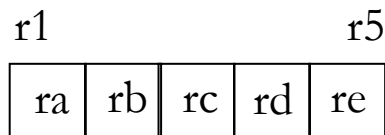
In-order εκτέλεση στην περίπτωση αυτή

Register Renaming - Παράδειγμα

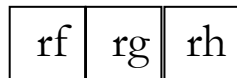
Physical Register File



Physical Renaming Table

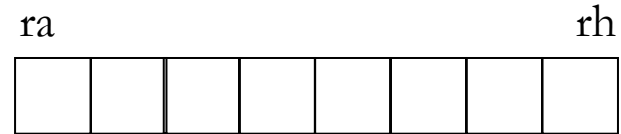


Free List

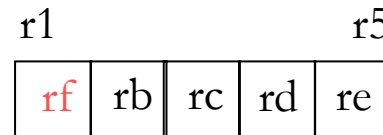


Μετά την αποκωδικοποίηση των εντολών i1 και i2, οι καταχωρητές r1 έως r5 αντιστοιχίζονται στους φυσικούς καταχωρητές ra έως rh, αντίστοιχα

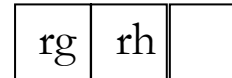
Physical Register File



Physical Renaming Table



Free List



Αποκωδικοποίηση της εντολής i3: ο καταχωρητής r1 αντιστοιχίζεται τώρα στον φυσικό καταχωρητή rf.

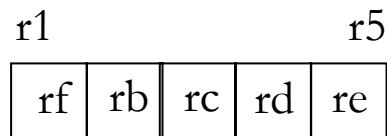
Η επόμενη εντολή i4 θα διαβάσει το αποτέλεσμα της i3 (από τον rf).

Register Renaming - Παράδειγμα

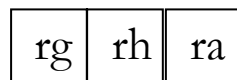
Physical Register File



Physical Renaming Table



Free List



Μετά την αποδέσμευση (retire) της εντολής i2, ο καταχωρητής ra επιστρέφει στη λίστα των ελεύθερων καταχωρητών (free list)

Πραγματικές Εξαρτήσεις Δεδομένων

- RAW: υποχρεωτική σειριοποίηση των εντολών
 - Παράδειγμα: fast fourier transform

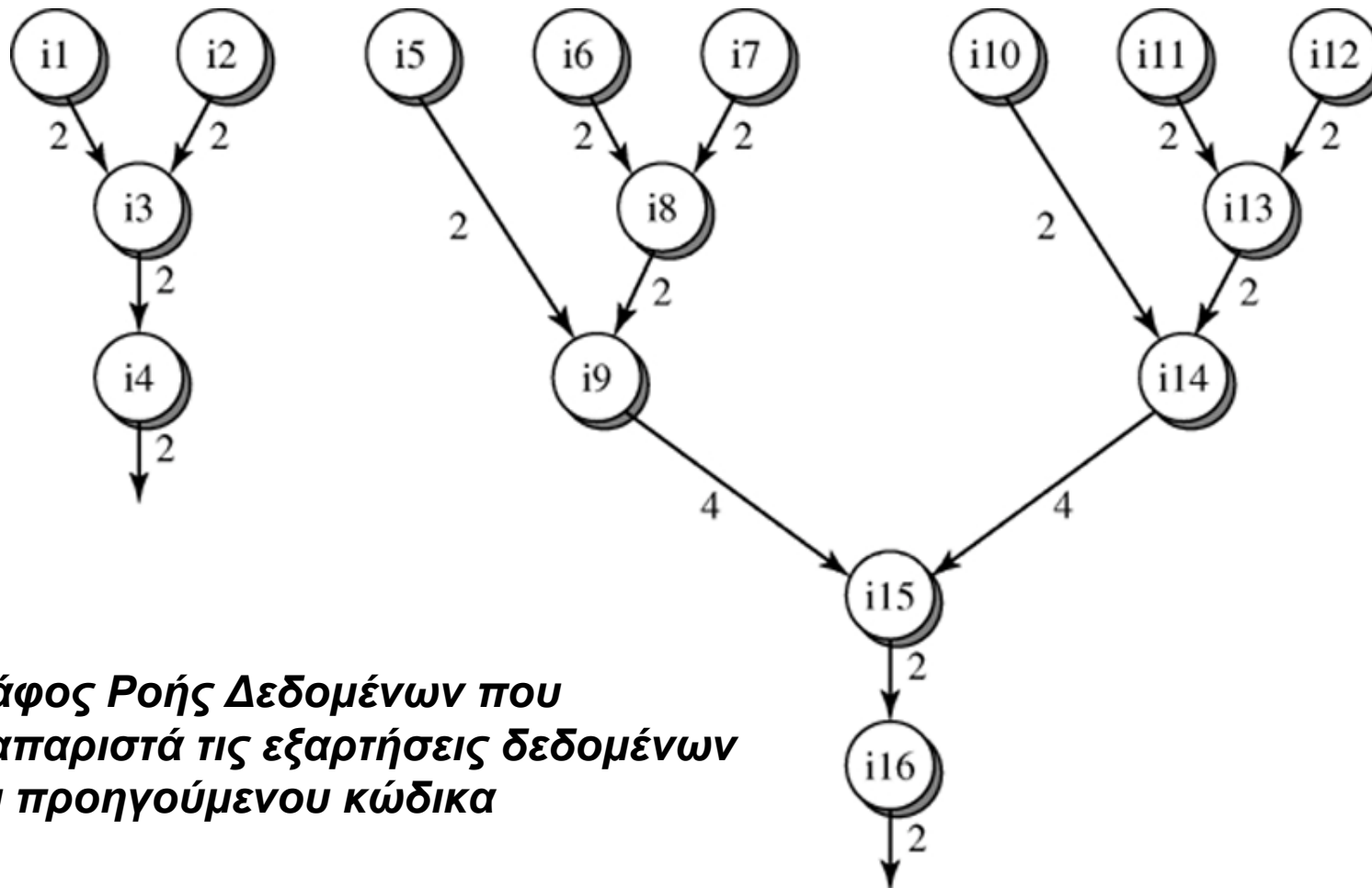
```
w[i+k].ip = z[i].rp + z[m+i].rp;  
w[i+j].rp = e[k+1].rp * (z[i].rp - z[m+i].rp) - e[k+1].ip * (z[i].ip - z[m+i].ip);
```

(a)

```
i1: f2 ← load,4(r2)  
i2: f0 ← load,4(r5)  
i3: f0 ← fadd,f2,f0  
i4: 4(r6) ← store,f0  
i5: f14 ← load,8(r7)  
i6: f6 ← load,0(r2)  
i7: f5 ← load,0(r3)  
i8: f5 ← fsub,f6,f5  
i9: f4 ← fmul,f14,f5  
i10: f15 ← load,12(r7)  
i11: f7 ← load,4(r2)  
i12: f8 ← load,4(r3)  
i13: f8 ← fsub,f7,f8  
i14: f8 ← fmul,f15,f8  
i15: f8 ← fsub,f4,f8  
i16: 0(r8) ← store,f8
```

(b)

Πραγματικές Εξαρτήσεις Δεδομένων

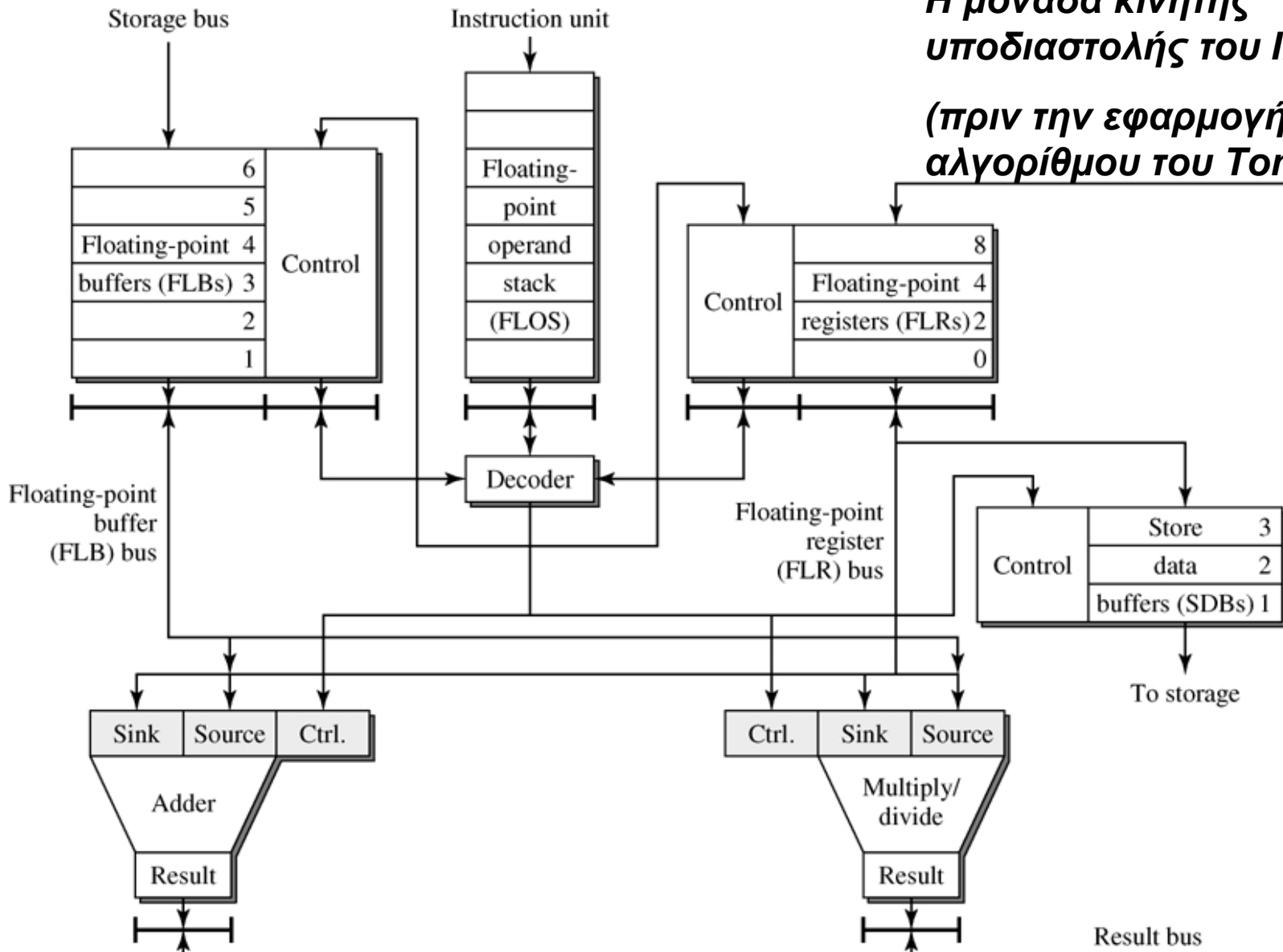


Γράφος Ροής Δεδομένων που αναπαριστά τις εξαρτήσεις δεδομένων του προηγούμενου κώδικα

Ο αλγόριθμος Tomasulo

Η μονάδα κινητής υποδιαστολής του IBM 360

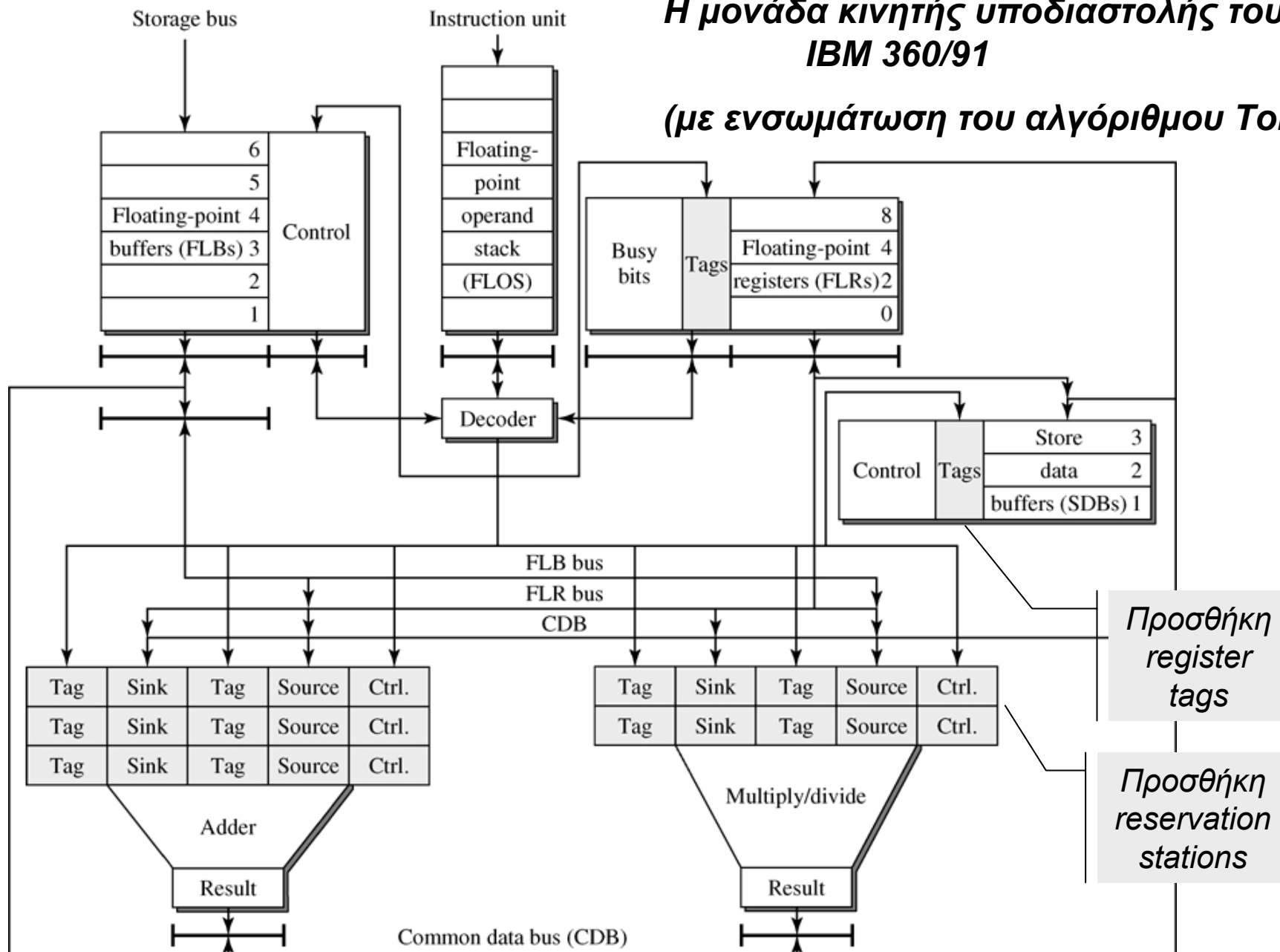
(πριν την εφαρμογή του αλγορίθμου του Tomasulo)



Ο αλγόριθμος Tomasulo

Η μονάδα κινητής υποδιαστολής του
IBM 360/91

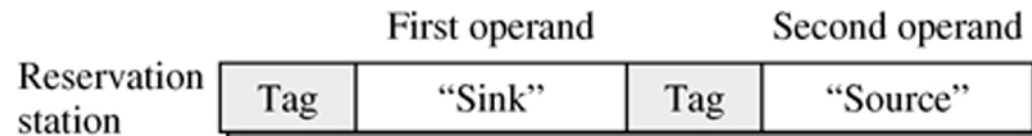
(με ενσωμάτωση του αλγόριθμου Tomasulo)



Ο αλγόριθμος Tomasulo

Πεδία Tag:

για την αναγνώριση των ζητούμενων δεδομένων και την απευθείας αποστολή τους στο reservation station



(a)



(b)



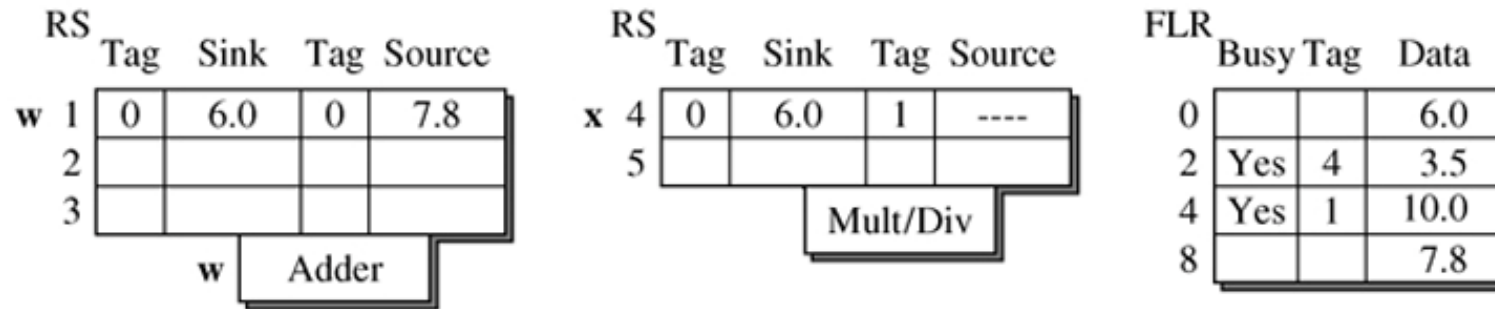
(c)

Ο αλγόριθμος Tomasulo

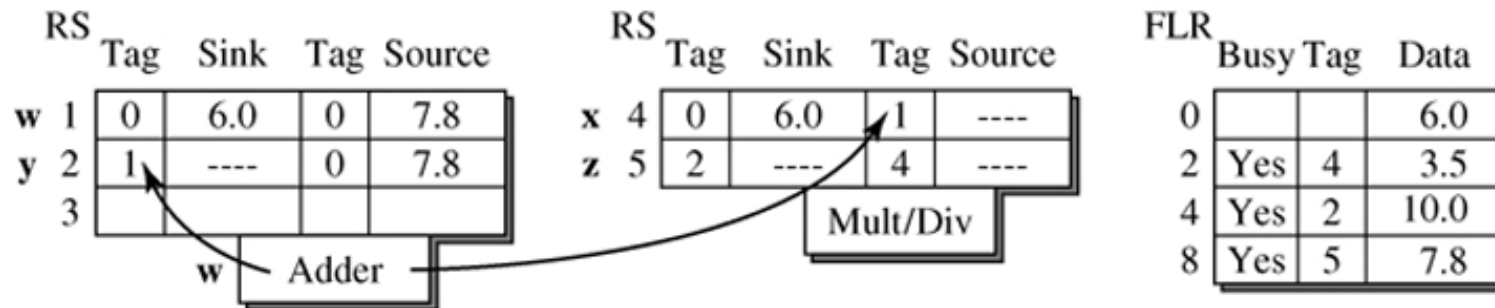
- Παράδειγμα (4 εντολές register-to-register):
 - w: $R4 \rightarrow R0 + R8$
 - x: $R2 \rightarrow R0 * R4$
 - y: $R4 \rightarrow R4 + R8$
 - z: $R8 \rightarrow R4 * R2$

Ο αλγόριθμος Tomasulo

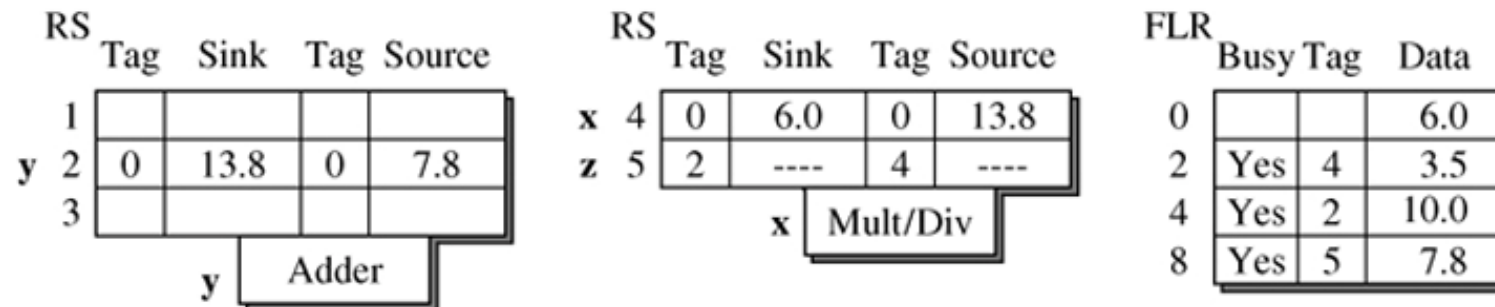
CYCLE 1 Dispatched instruction(s): w, x (in order)



CYCLE 2 Dispatched instruction(s): y, z (in order)

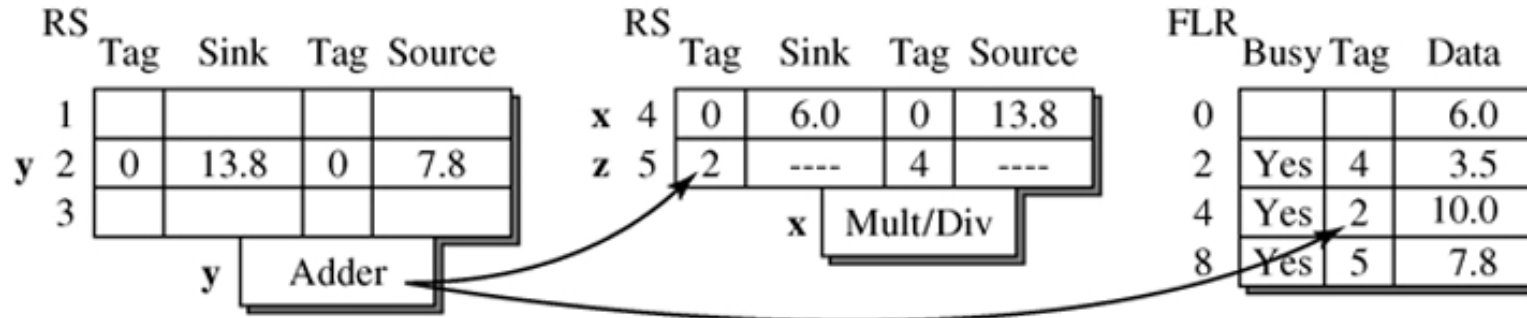


CYCLE 3 Dispatched instruction(s): _____

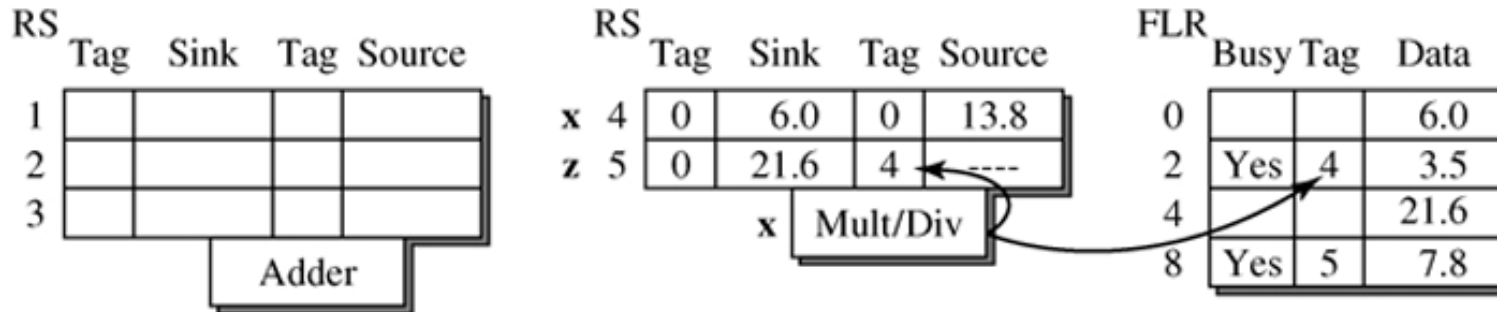


Ο αλγόριθμος Tomasulo

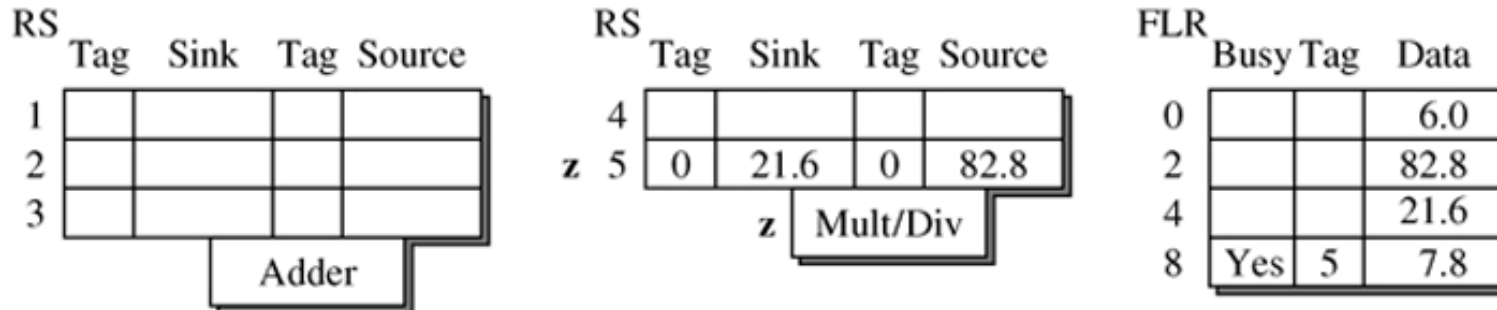
CYCLE 4 Dispatched instruction(s): _____



CYCLE 5 Dispatched instruction(s): _____



CYCLE 6 Dispatched instruction(s): _____

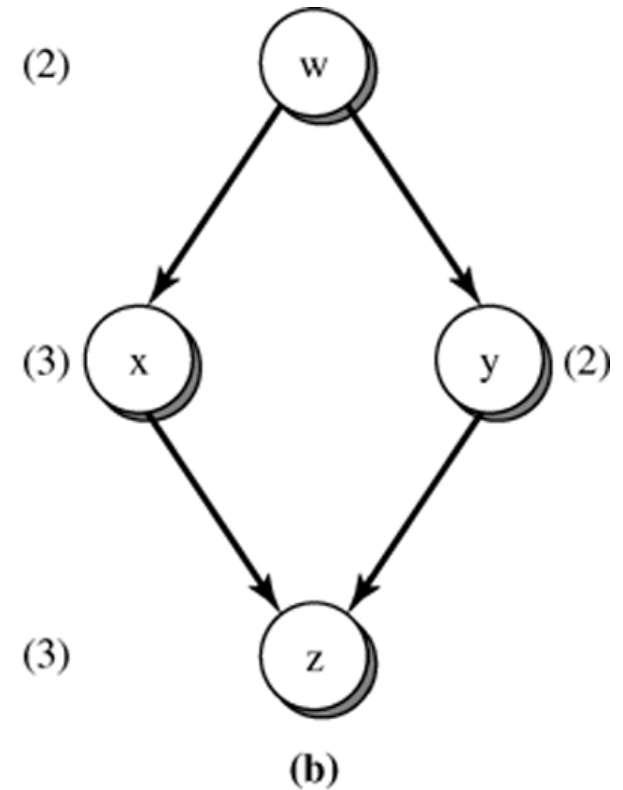
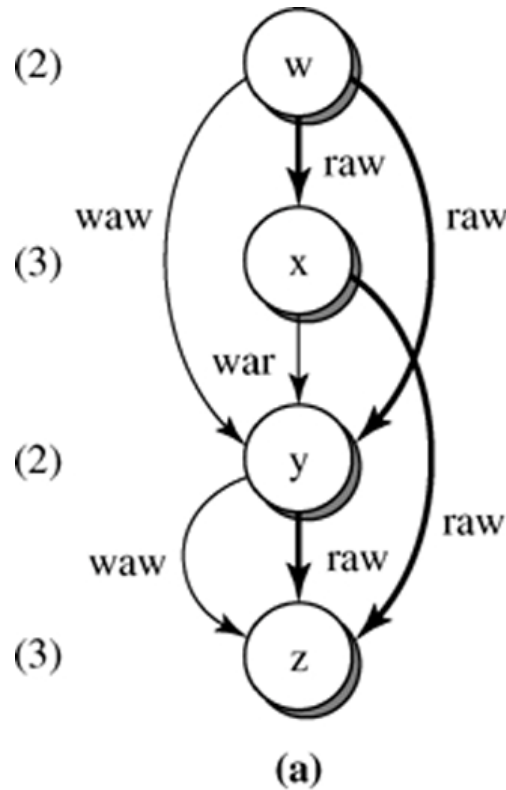


Ο αλγόριθμος Tomasulo

- Διάγραμμα Ροής

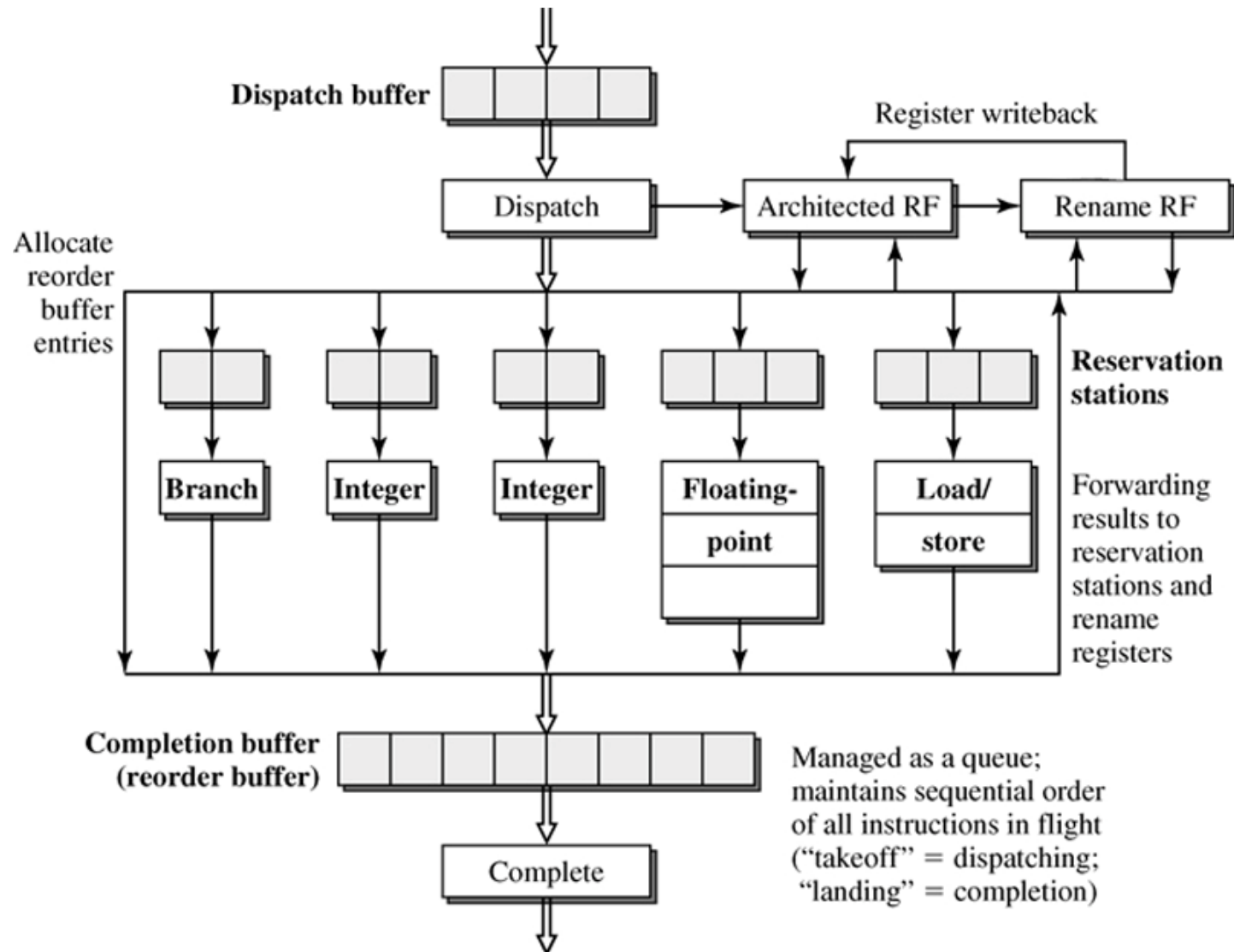
- (a) Όλες οι εξαρτήσεις δεδομένων

- (b) Μόνο οι πραγματικές εξαρτήσεις δεδομένων



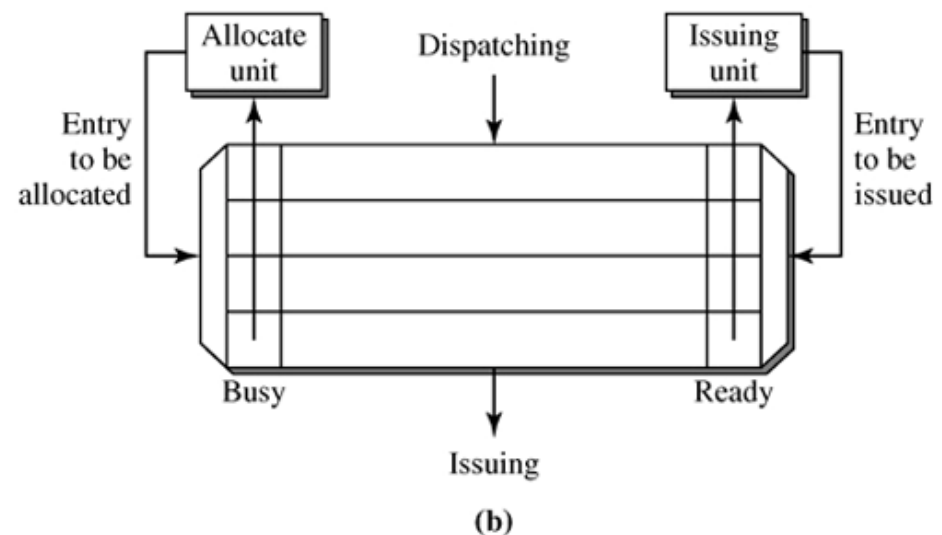
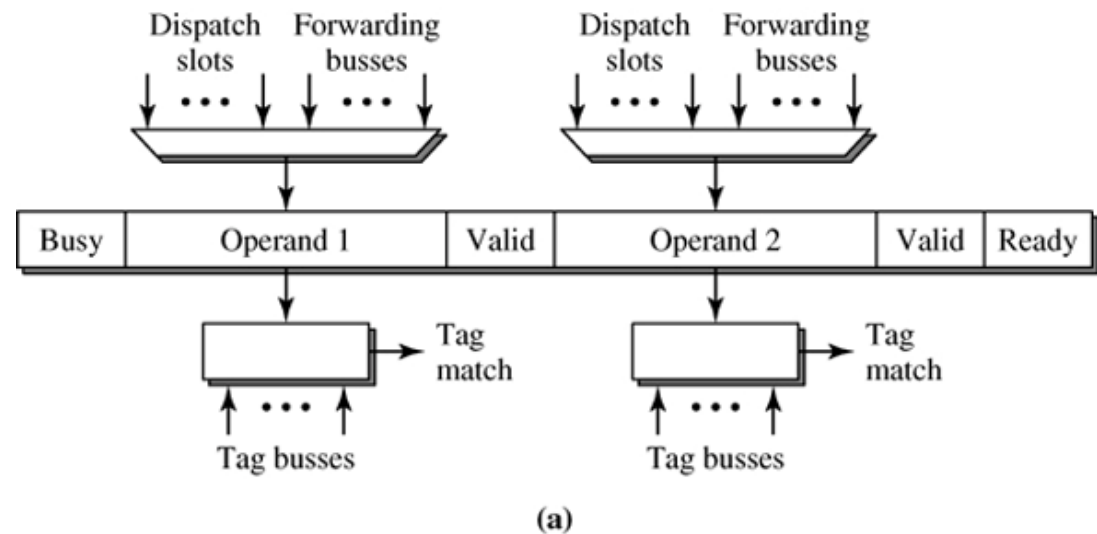
Δυναμική εκτέλεση πυρήνα

- Μηχανή ροής **μικρο-δεδομένων** δυναμική εκτέλεση



Σταθμοί κράτησης (reservation stations) και προσωρινός καταχωρητής αναδιάρταξης

- Μηχανισμοί των σταθμών κράτησης
 - (a) Με μία εγγραφή σταθμού κράτησης
 - (b) Διανομή από τους σταθμούς κράτησης



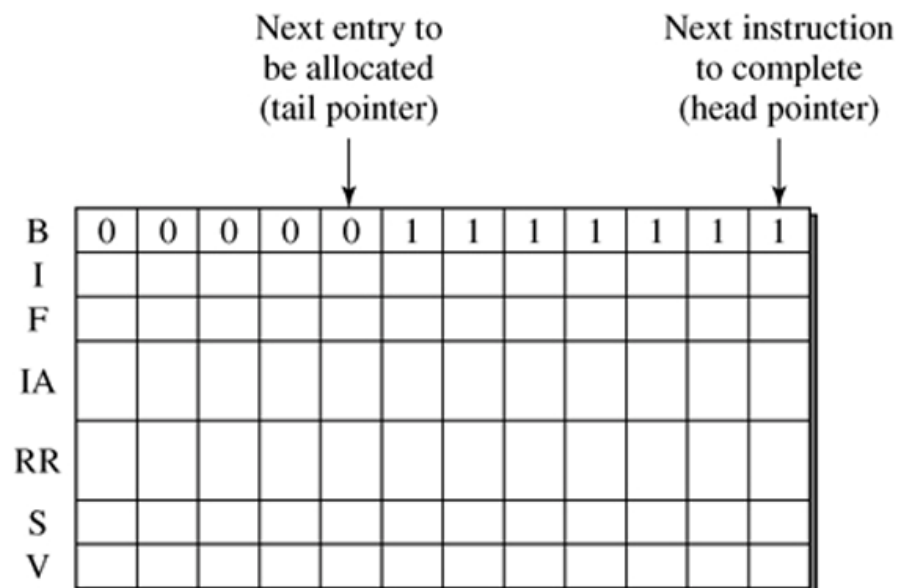
Σταθμοί κράτησης (reservation stations) και προσωρινός καταχωρητής αναδιάταξης

- (a) Εγγραφή ενός προσωρινού καταχωρητή αναδιάταξης

Busy	Issued	Finished	Instruction address	Rename register	Speculative	Valid
------	--------	----------	---------------------	-----------------	-------------	-------

(a)

- (b) Οργάνωση του προσωρινού καταχωρητή αναδιάταξης

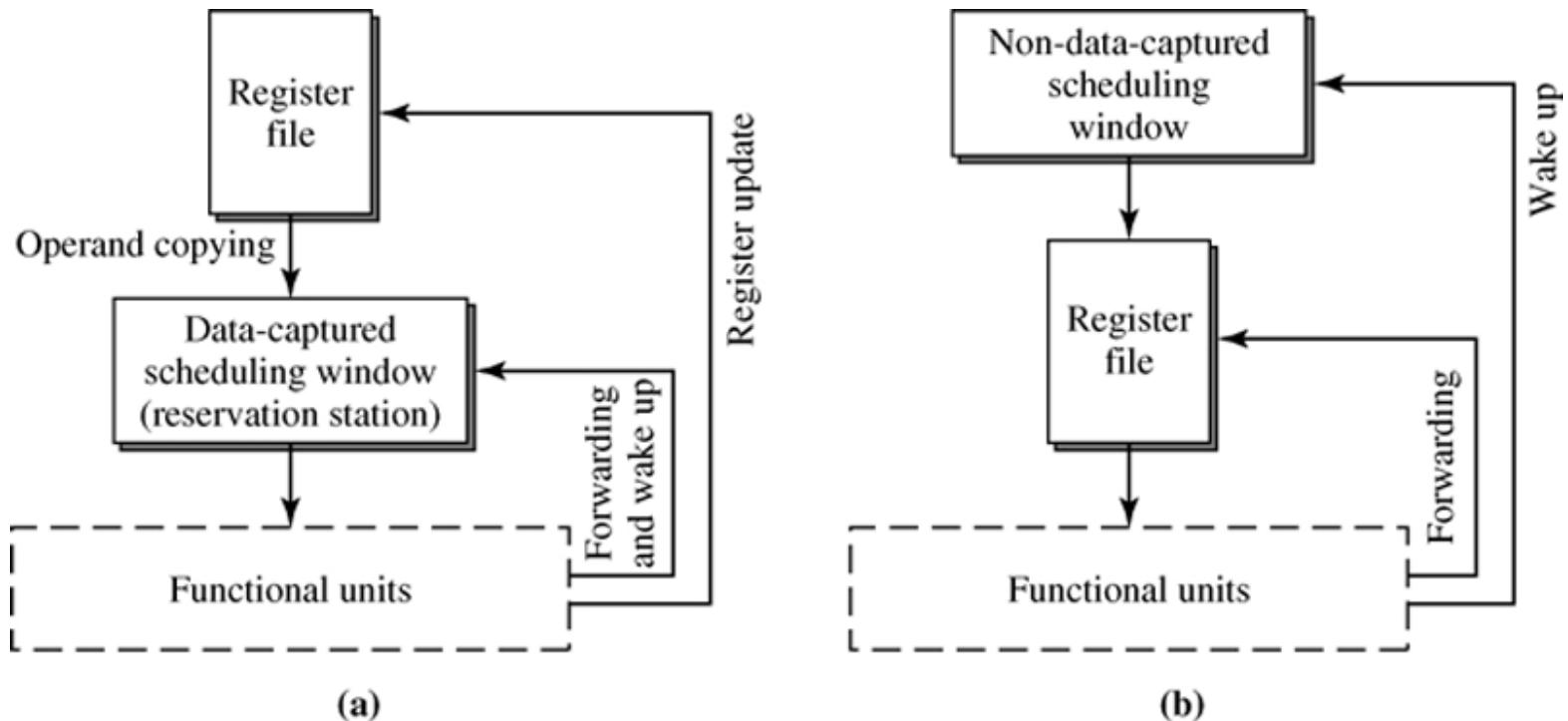


Reorder buffer

(b)

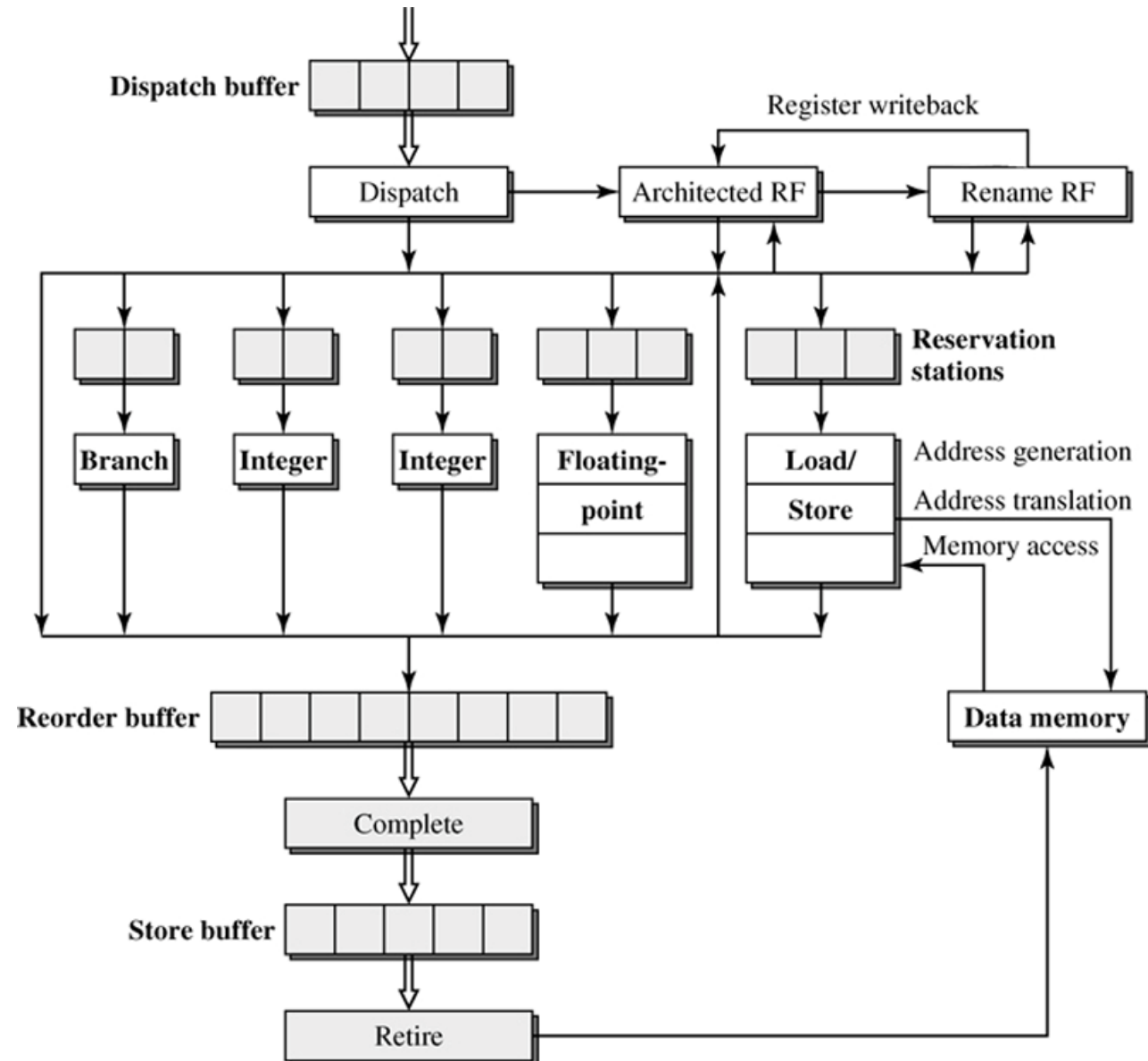
Δρομολογητής δυναμικών εντολών

- Σχέδιο ενός Δρομολογητή δυναμικών εντολών
 - (a) Με «αιχμαλώτιση» δεδομένων
 - (b) Χωρίς «αιχμαλώτιση» δεδομένων



Εντολές Προσπέλασης της μνήμης

- Επεξεργασία των εντολών Φόρτωσης / Αποθήκευσης



Σειριοποίηση των προσπελάσεων της μνήμης

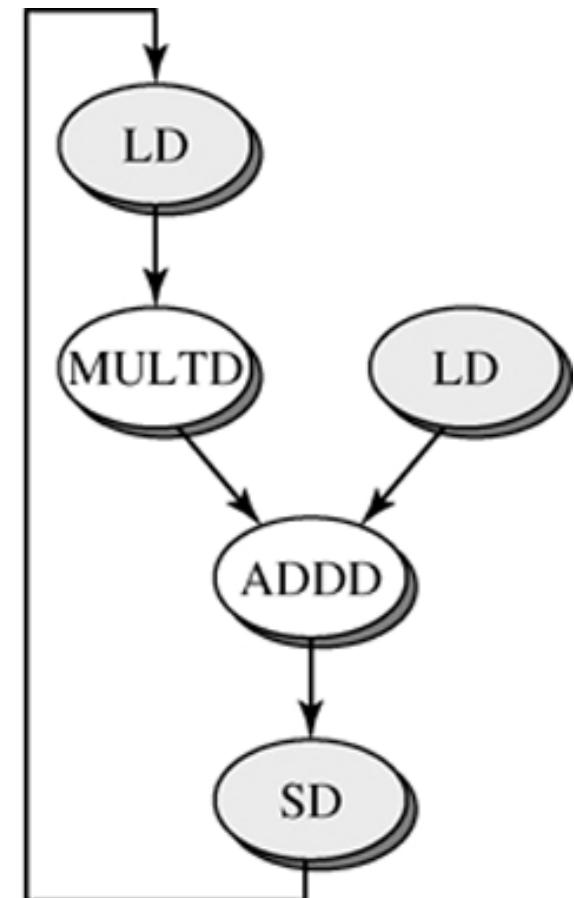
- Παράδειγμα

$Y(i) = A * X(i) + Y(i)$

F0 ← LD, a
R4 ← ADDI, Rx, #512 ;last address

Loop:

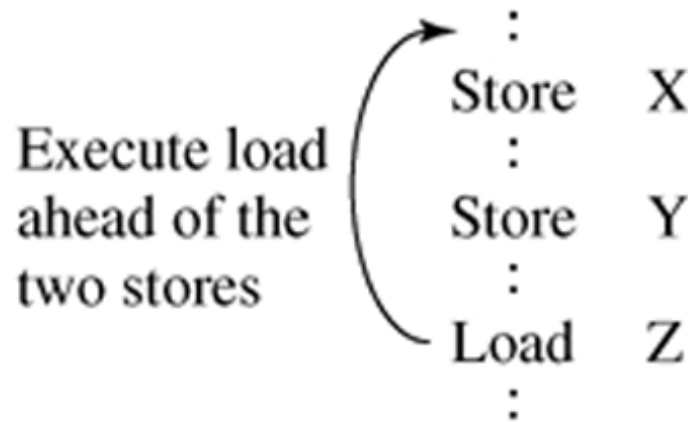
F2 ← LD, 0(Rx) ;load X(i)
F2 ← MULTD, F0, F2 ;A*X(i)
F4 ← LD, 0(Ry) ;load Y(i)
F4 ← ADDD, F2, F4 ;A*X(i)+Y(i)
0(Ry) ← SD, F4 ;store into Y(i)
Rx ← ADDI, Rx, #8 ;inc. index to X
Ry ← ADDI, Ry, #8 ;inc. index to Y
R20 ← SUB, R4, Rx ;compute bound
BNZ, R20, Loop ;check if done



Προσπέραση και Προώθηση των εντολών Φόρτωσης

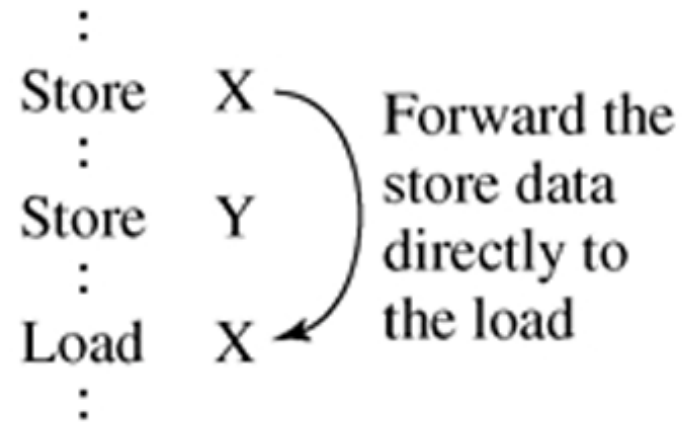
- Εκτέλεση των εντολών φόρτωσης νωρίς:
 - (a) Προσπέραση άλλων εντολών
 - (b) Προώθηση δεδομένων από άλλες εντολές σε εντολές φόρτωσης

Dynamic instruction sequence:



(a)

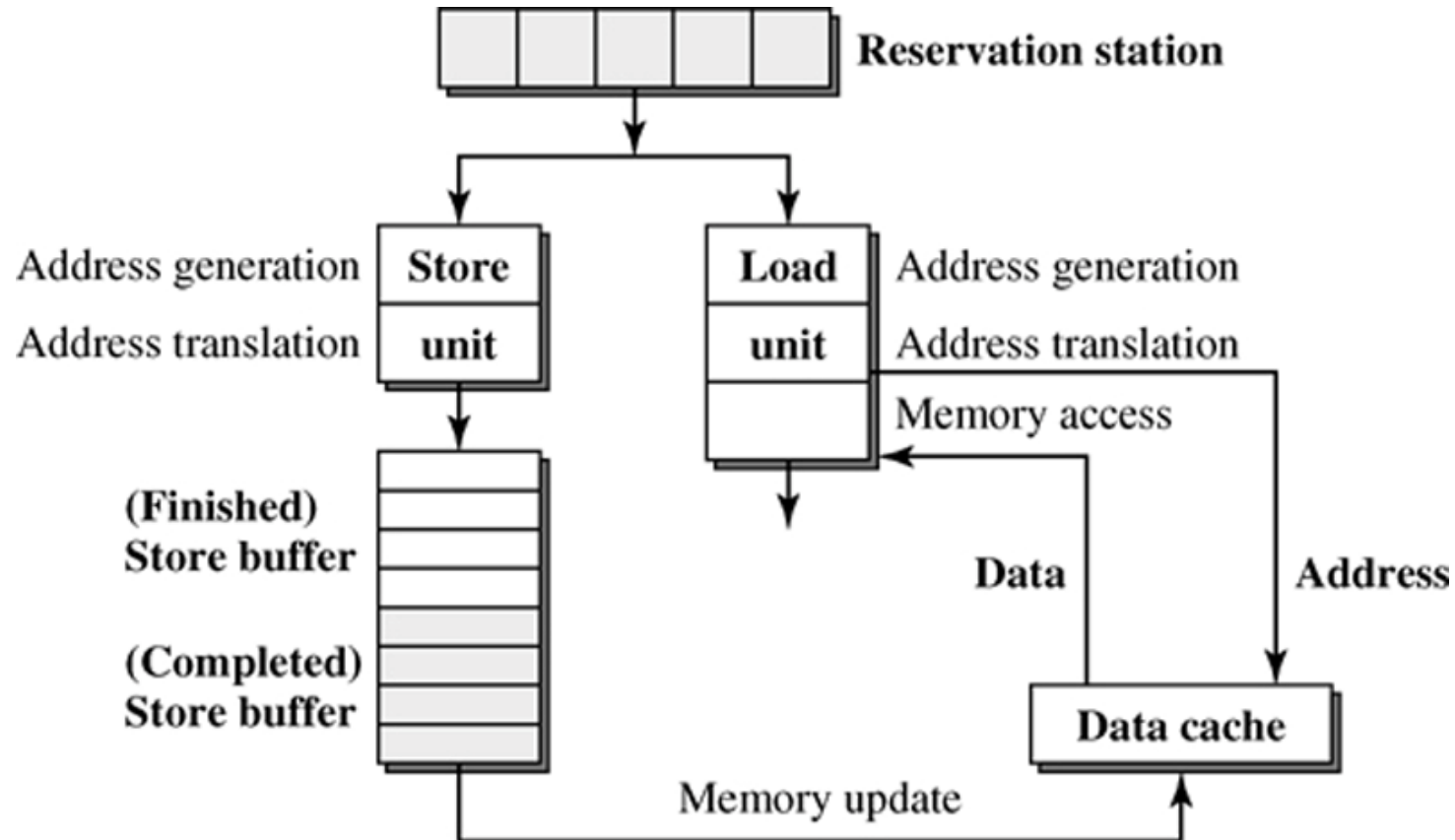
Dynamic instruction sequence:



(b)

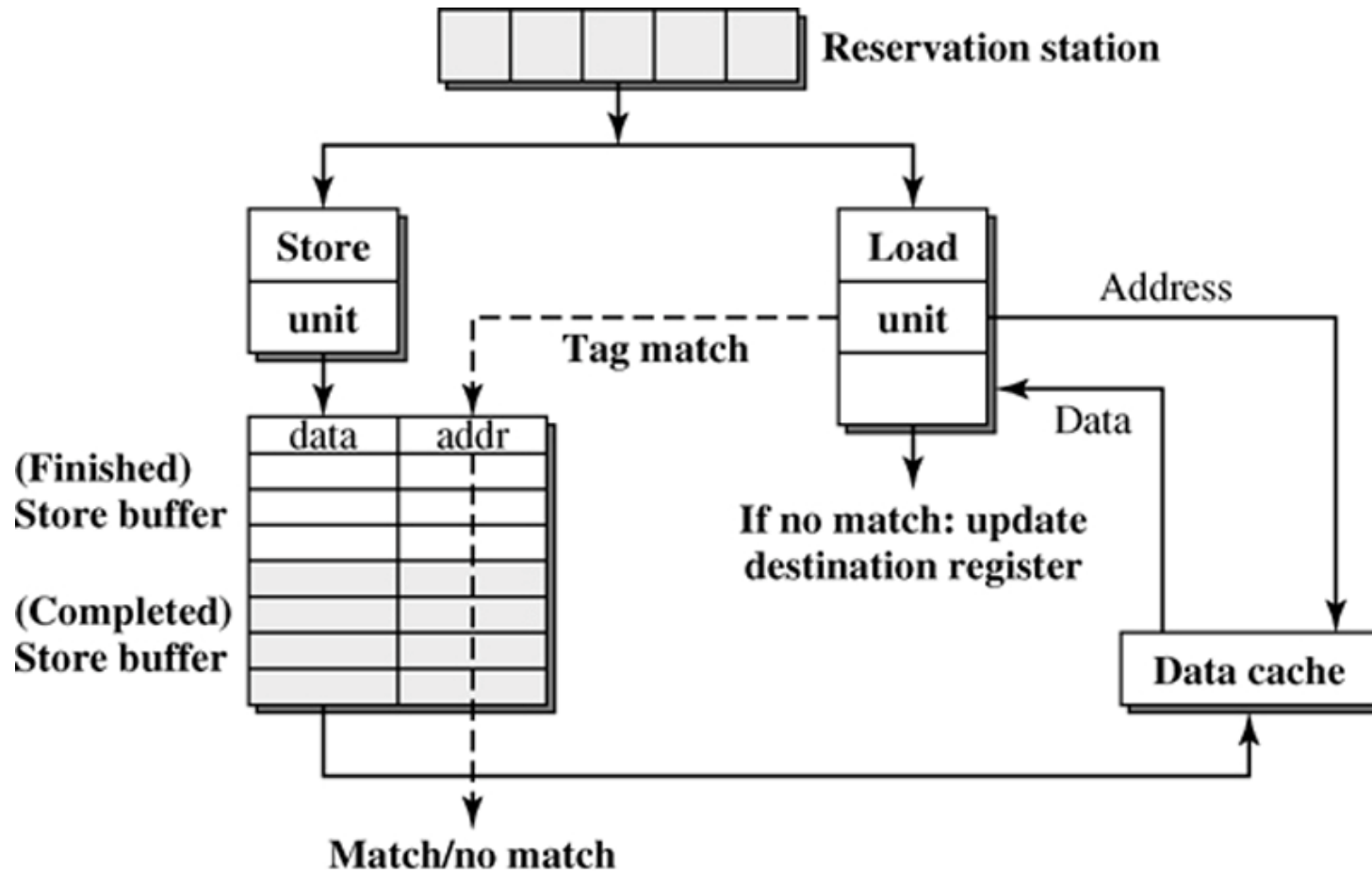
Προσπέραση και Προώθηση των εντολών Φόρτωσης

- Μηχανισμοί επεξεργασίας των εντολών Φόρτωσης / Αποθήκευσης:
Ξεχωριστές Μονάδες φόρτωσης και Αποθήκευσης, με διανομή από ένα κοινό σταθμό κράτησης εν σειρά



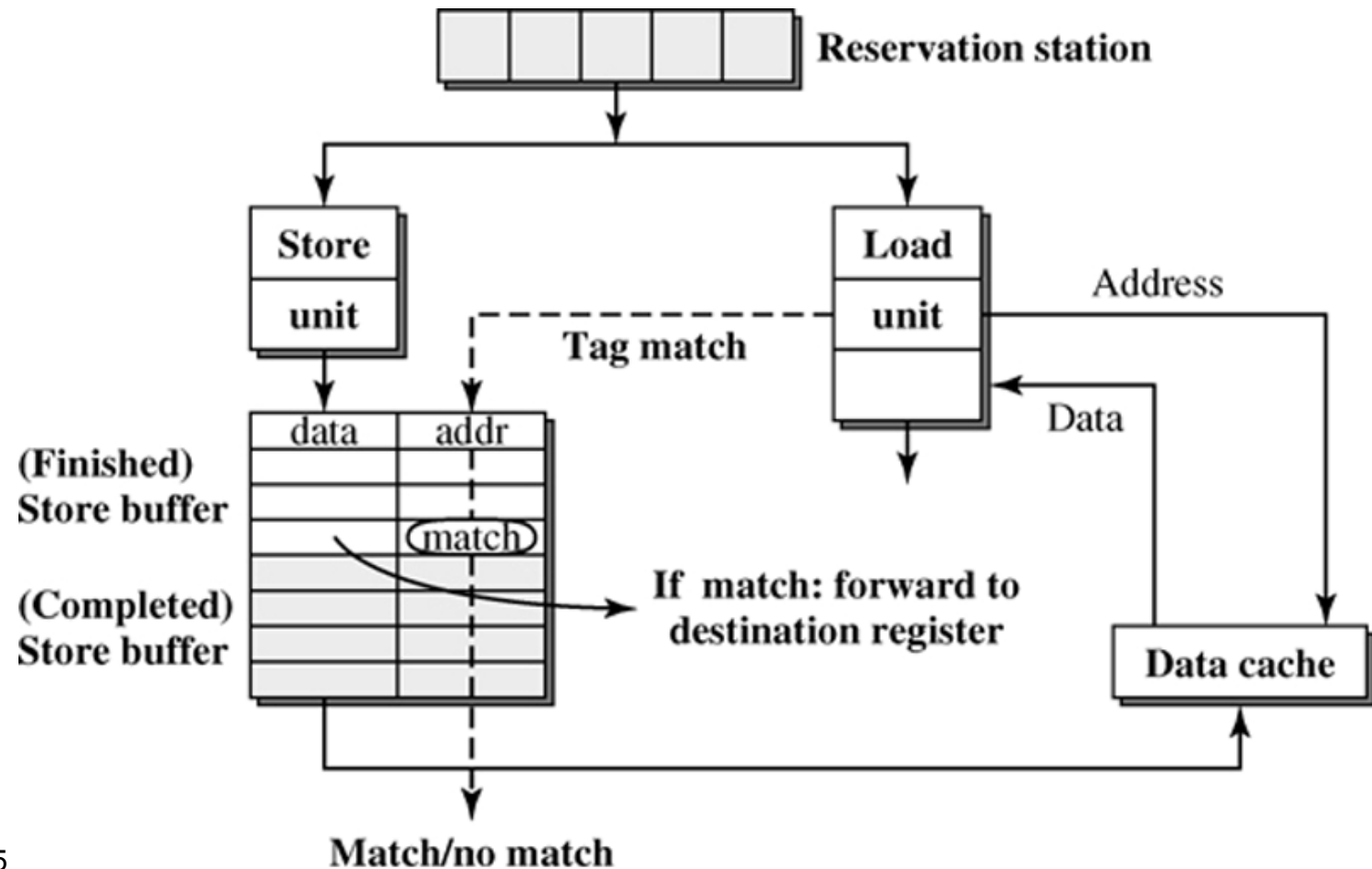
Προσπέραση και Προώθηση των εντολών Φόρτωσης

- Προσπέραση από εντολή Φόρτωσης



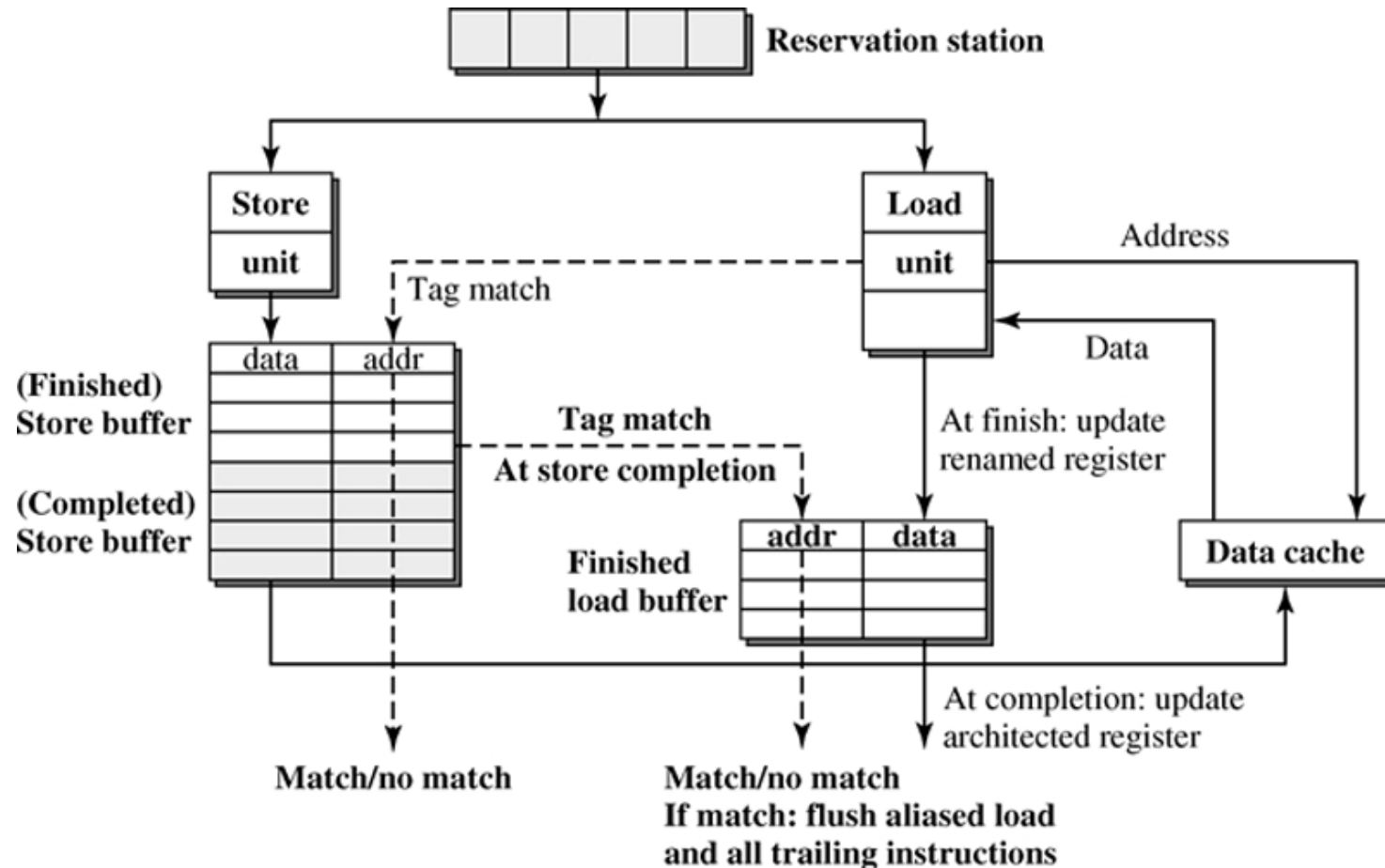
Προσπέραση και Προώθηση των εντολών Φόρτωσης

- Προώθηση σε εντολή Φόρτωσης



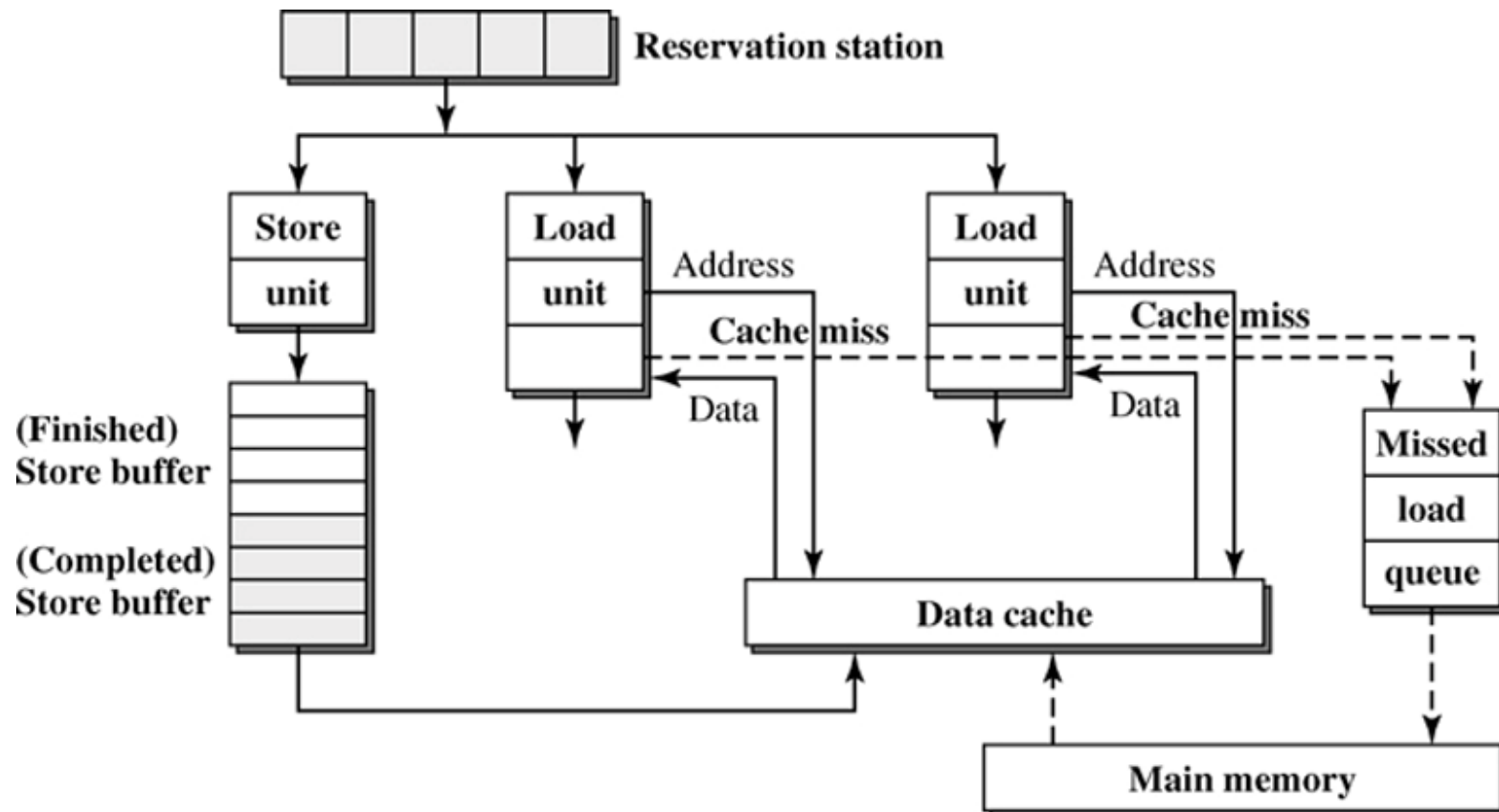
Προσπέραση και Προώθηση των εντολών Φόρτωσης

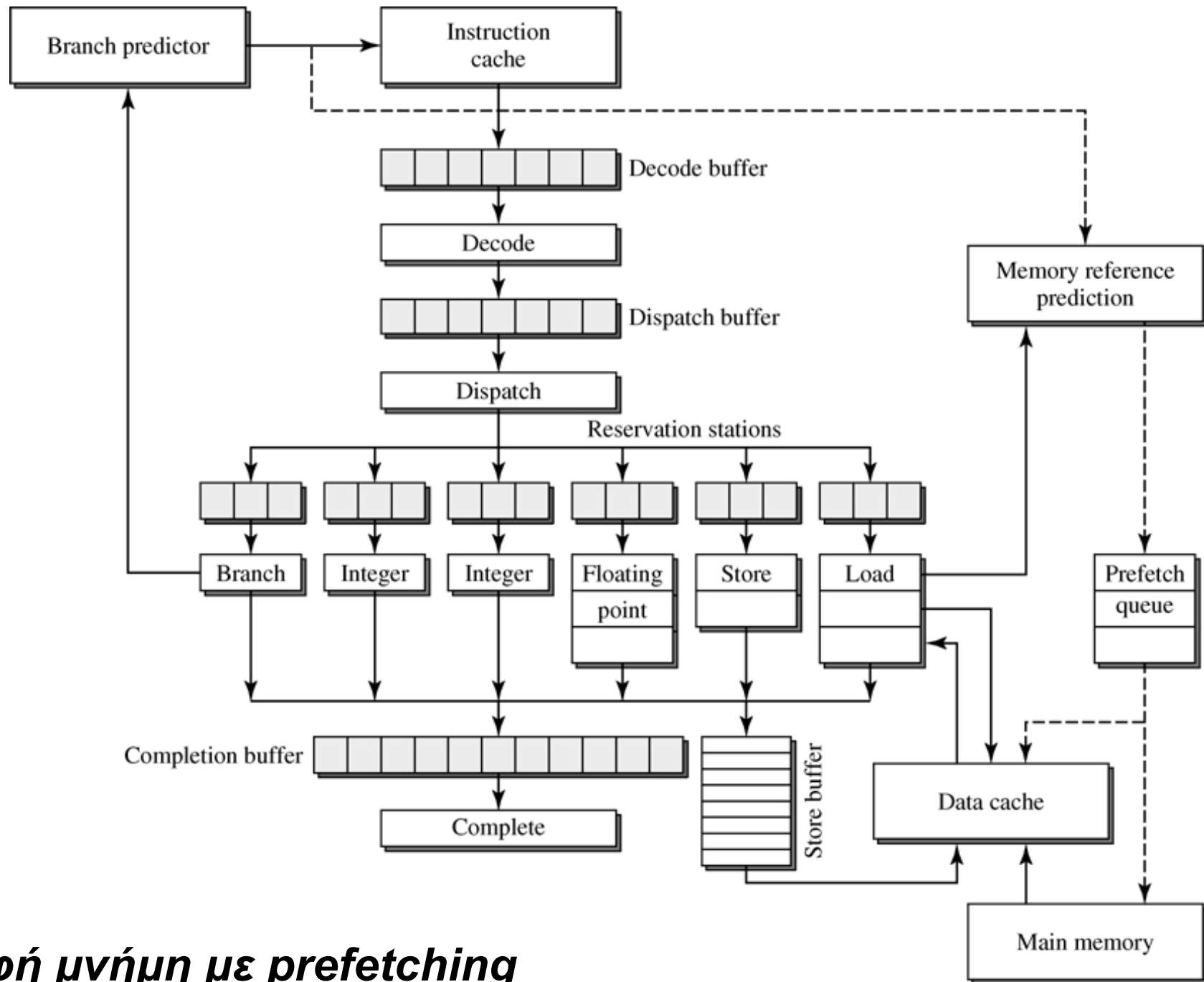
- Διανομή και εκτέλεση των εντολών Φόρτωσης και Αποθήκευσης πλήρως εκτός σειράς



Άλλες Τεχνικές Ροής Δεδομένων της μνήμης

- Κρυφή μνήμη δεδομένων, με διπλή πόρτα ανάγνωσης/εγγραφής που δεν μπλοκάρει (dual-ported, nonblocking)

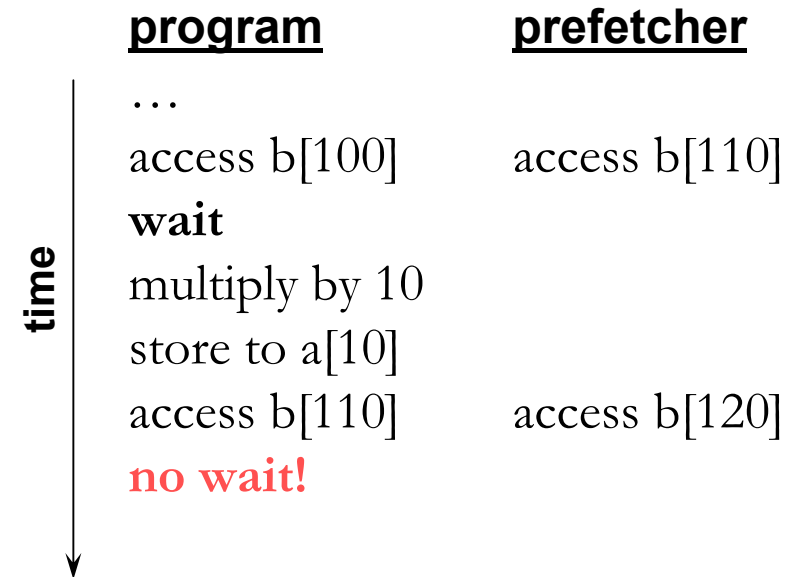
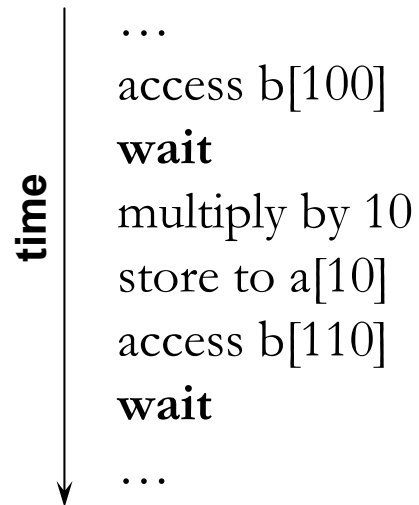




Κρυφή μνήμη με prefetching

Prefetching - Παράδειγμα

```
for i=0 to 1000000  
  a[i] = b[i*10]*10;
```



Trace caches

- **Trace cache**: Δομή παρόμοια της κρυφής μνήμης, στην οποία αποθηκεύεται μία ακολουθία (trace) εντολών, με τη σειρά που προσπελάστηκαν (fetch) στο παρελθόν
- Δίνει λύση στην πρόβλεψη και ανάγνωση πολλαπλών εντολών, συμπεριλαμβανομένου των εντολών διακλάδωσης

