

ΗΥ425 - Αρχιτεκτονική Υπολογιστών  
Μικροαρχιτεκτονική Επεξεργαστών Υπερχλίμακας

Βασισμένο στην δημοσίευση:

**J. E. Smith and G. S. Sohi, The Microarchitecture of Superscalar Processors,  
Proceedings of the IEEE, vol. 83, pp. 1609–1624, Dec. 1995.**

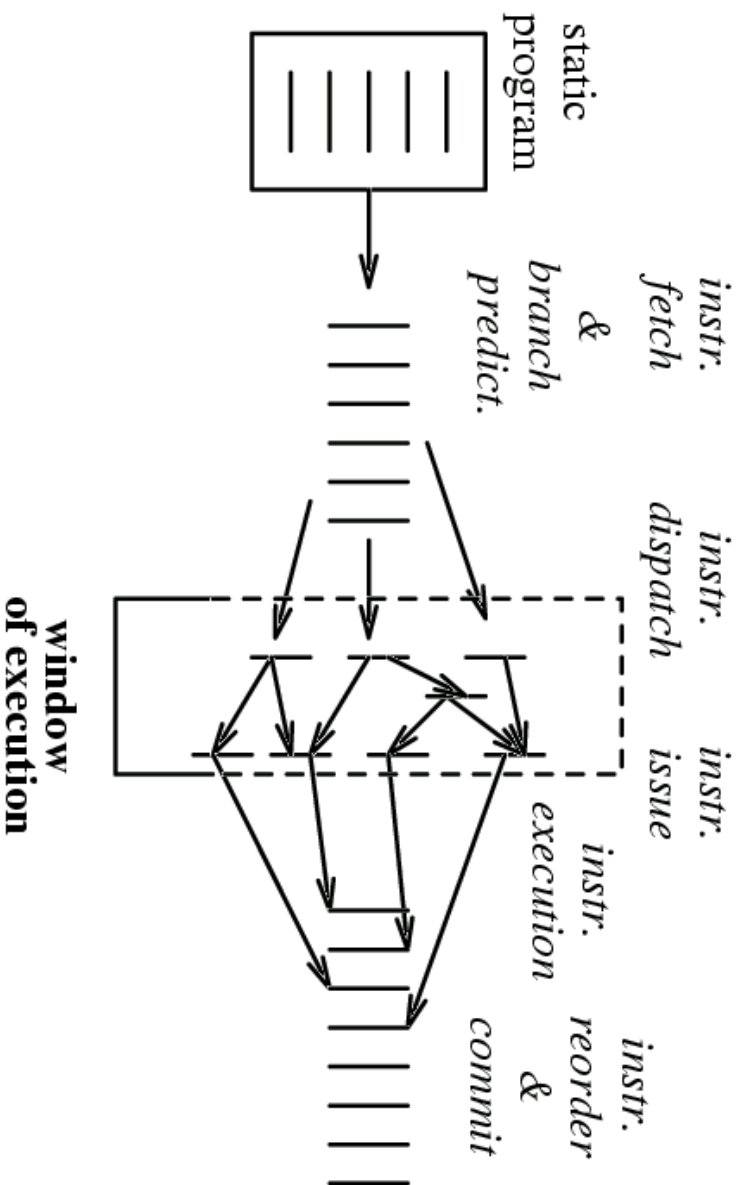
X. Σωτηρίου

11 Δεκεμβρίου 2001

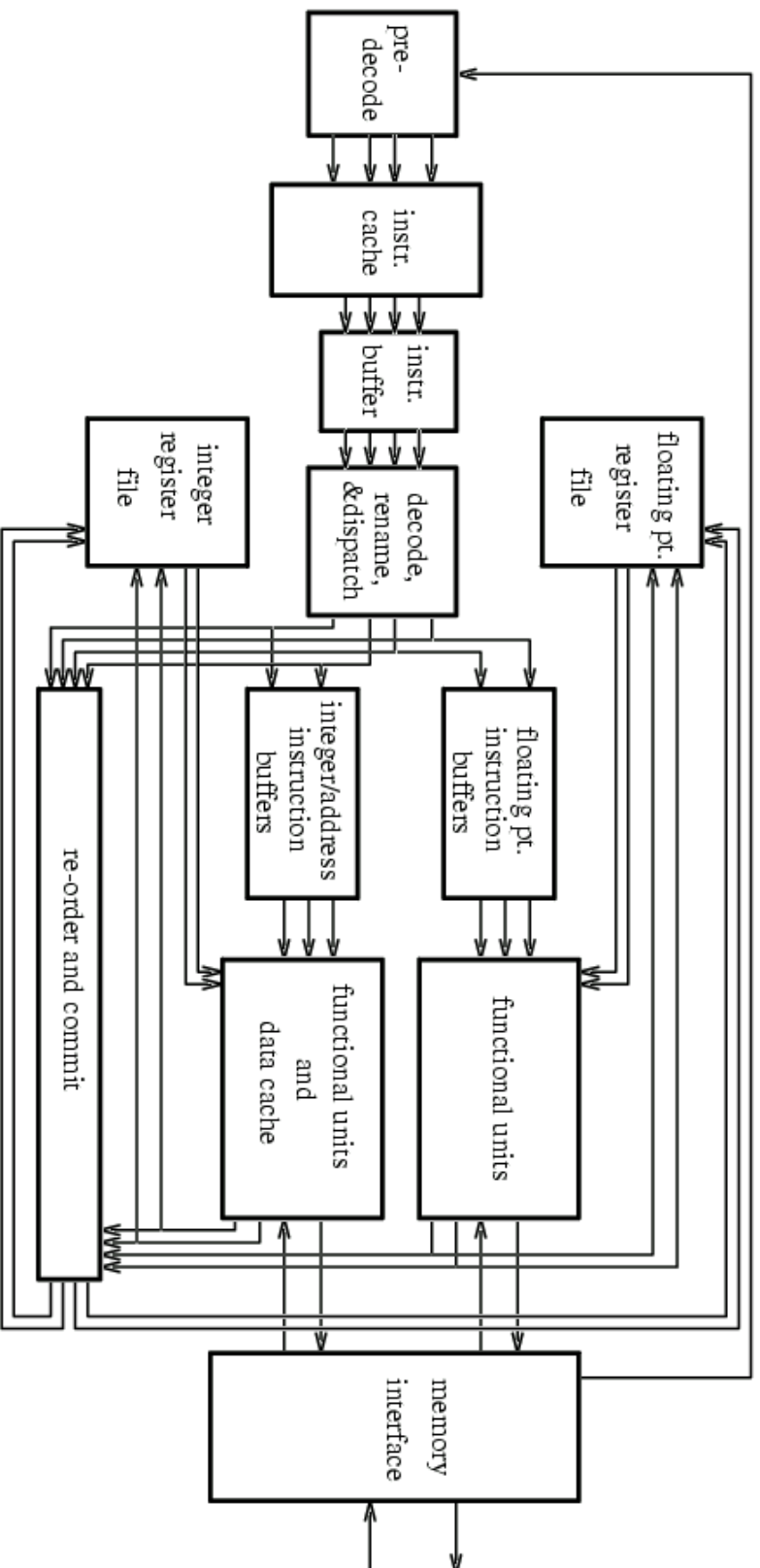
## Στοιχεία Επεξεργαστών Υψηλής Απόδοσης

- στρατηγική ανάγνωσης (*IF*) πολλαπλών εντολών που συνδιάζεται με πρόβλεψη βρόχων.
- μέθοδοι ανάλυσης αληθινών εξαρτήσεων μεταξύ καταχωρητών και μηχανισμοί επικοινωνίας αυτών.
- μέθοδοι έκδοσης πολλαπλών εντολών παράλληλα.
- υποστηρίξη παράλληλης εκτέλεσης πολλαπλών εντολών· πολλαπλές λειτουργικές μονάδες και ιερ-αρχίες μνήμης πολλαπλών ταυτόχρονων προσβάσεων.
- μέθοδοι χειρισμού εξαρτήσεων μνήμης - μεταξύ εντολών *load* και *store*.
- μέθοδοι για σωστή δέσμευση των αποτελεσμάτων των εντολών· μηχανισμοί που δίνουν την εξωτερική εντύπωση της ακολουθιακής εκτέλεσης.

## Μοντέλο Εκτέλεσης Πολλαπλής Εκδόσης



# Οργάνωση Επεξεργαστή Υπερχλίμακας



## Ανάγνωση Εντολής - *Instruction Fetch*

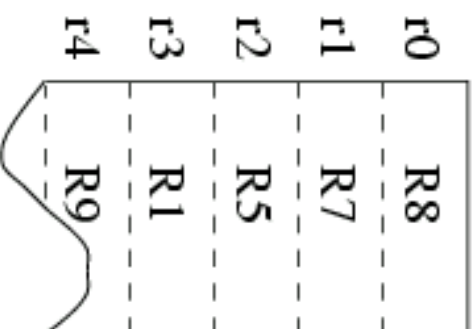
- ύπαρξη *cache* εντολών.
- ανάγνωση πολλαπλών εντολών ανα κύκλο απο το *cache*.
- μείωση καθυστερήσεων λόγω αστοχίας στο *cache* - *instruction buffer*.
- γρήγορη αναγνώριση εντολών αλλαγής *PC*.
- πρόγνωση αποτελέσματος και υπολογισμός διεύθυνσης βρόχου:
  - στατική πρόγνωση - βοήθεια απο *compiler*.
  - δυναμική πρόγνωση - *branch prediction table*.

## Αποκωδικοποίηση, Μετονομασία και Αποστολή

- κατασκευή τριάδας: λειτουργία, δρώμενα, αποτέλεσμα.
- ανάλυση κινδύνων και αποστολή σε σταθμούς για αργότερη έκδοση σε  $A.M.$  και εκτέλεση.
- μετονομασία αφαιρεί εξαρτήσεις  $WAR$  και  $WAW$ .
- αντιστοιχία απο λογικούς σε φυσικούς καταχωρητές.
- η τιμή ενός καταχωρητή βρίσκεται στο  $A.K.$  ή στην ουρά σειράς.

## Διαδικασία Μετονομασίας

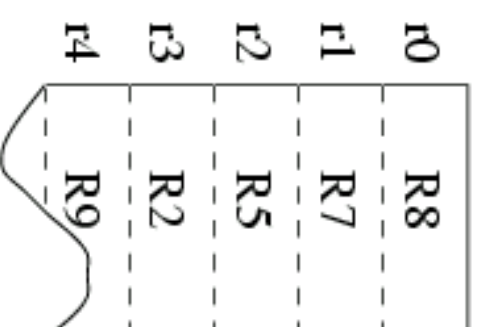
*before:*      **add r3,r3,4**



*free list:*



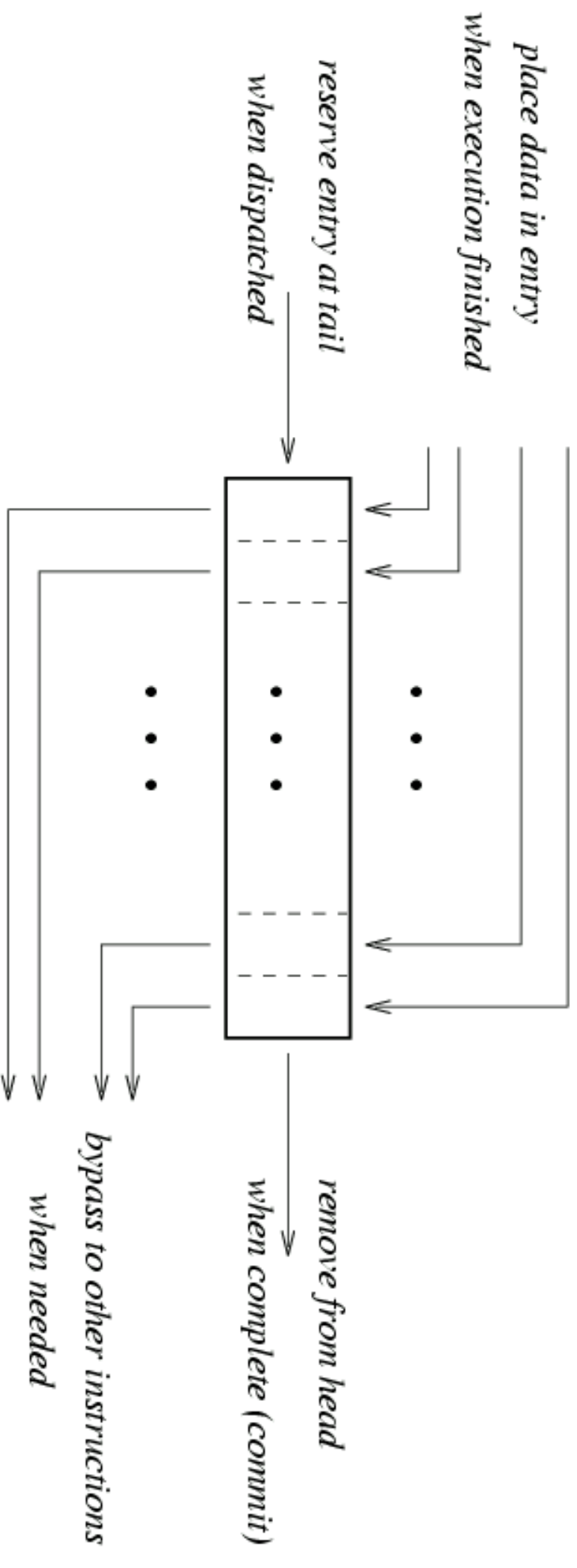
*after:*      **add R2,R1,4**



*free list:*



## Ουρά Σειράς - Reorder Buffer

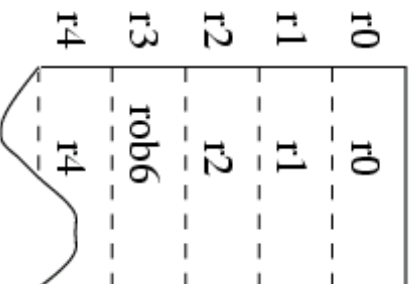




## Διαδικασία Μετονομασίας με Ουρά Σειράς

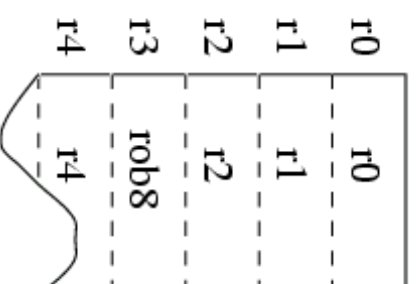
*before:*    `add r3,r3,4`

*mapping table:*



*after:*    `add r3,    rob6, 4`  
              `(rob8)`

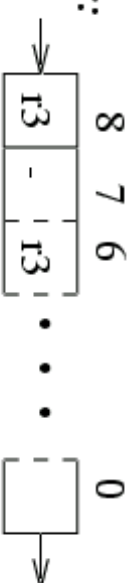
*mapping table:*



*reorder buffer:*  
*(partial)*



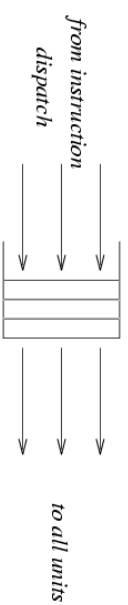
*reorder buffer:*  
*(partial)*



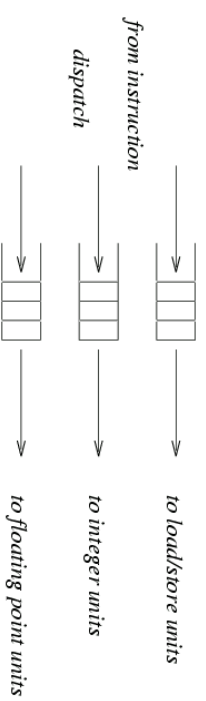
## Έκδοση και Παράλλαξη Εκτέλεση

- η έκδοση εντολών ελέγχει την διαθεσιμότητα των δρώντων.
- αυτή είναι το επίκεντρο των αρχιτεκτονικών υπερκλίμακας και περιλαμβάνει το ‘παράθυρο’ εκτέλεσης.
- η έκδοση εξαρτάται και από την διαθεσιμότητα χώρου στην εκτέλεση.
- πριν τις μονάδες εκτέλεσης υπάρχουν ουρές εντολών.

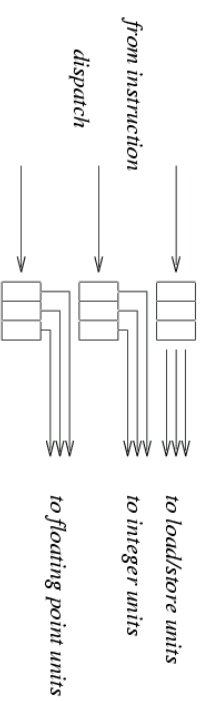
## Οργάνωση έκδοσης με Ουρές



a) Single, shared queue



b) Multiple queues; one per instruction type



c) Multiple reservation stations; one per instruction type

## Παράδειγμα Προγράμματος

σε C:

```
for (i=0; i<last; i++) {  
    if (a[i] > a[i+1]) {  
        temp = a[i];  
        a[i] = a[i+1];  
        a[i+1] = temp;  
        change++;  
    }  
}
```

## Παράδειγμα Προγράμματος

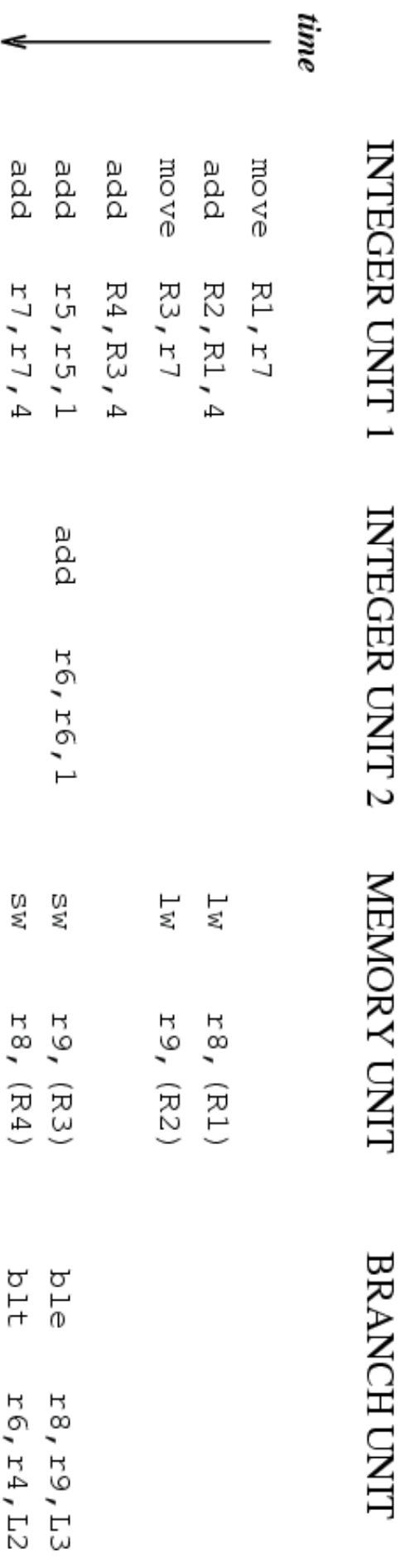
σε *Assembly*:

```
L2:
    move    r3,r7          #r3->a[i]
    lw      r8,(r3)        #load a[i]
    add     r3,r3,4        #r3->a[i+1]
    lw      r9,(r3)        #load a[i+1]
    ble     r8,r9,L3      #branch a[i]>a[i+1]

    move    r3,r7          #r3->a[i]
    sw      r9,(r3)        #store a[i]
    add     r3,r3,4        #r3->a[i+1]
    sw      r8,(r3)        #store a[i+1]
    add     r5,r5,1        #change++

L3:
    add     r6,r6,1        #i++
    add     r7,r7,4        #r4->a[i]
    blt     r6,r4,L2      #branch i<last
```

## Απομολόγηση Εκτέλεσης Παράδειγματος



## Χειρισμός Λειτουργιών Μνήμης

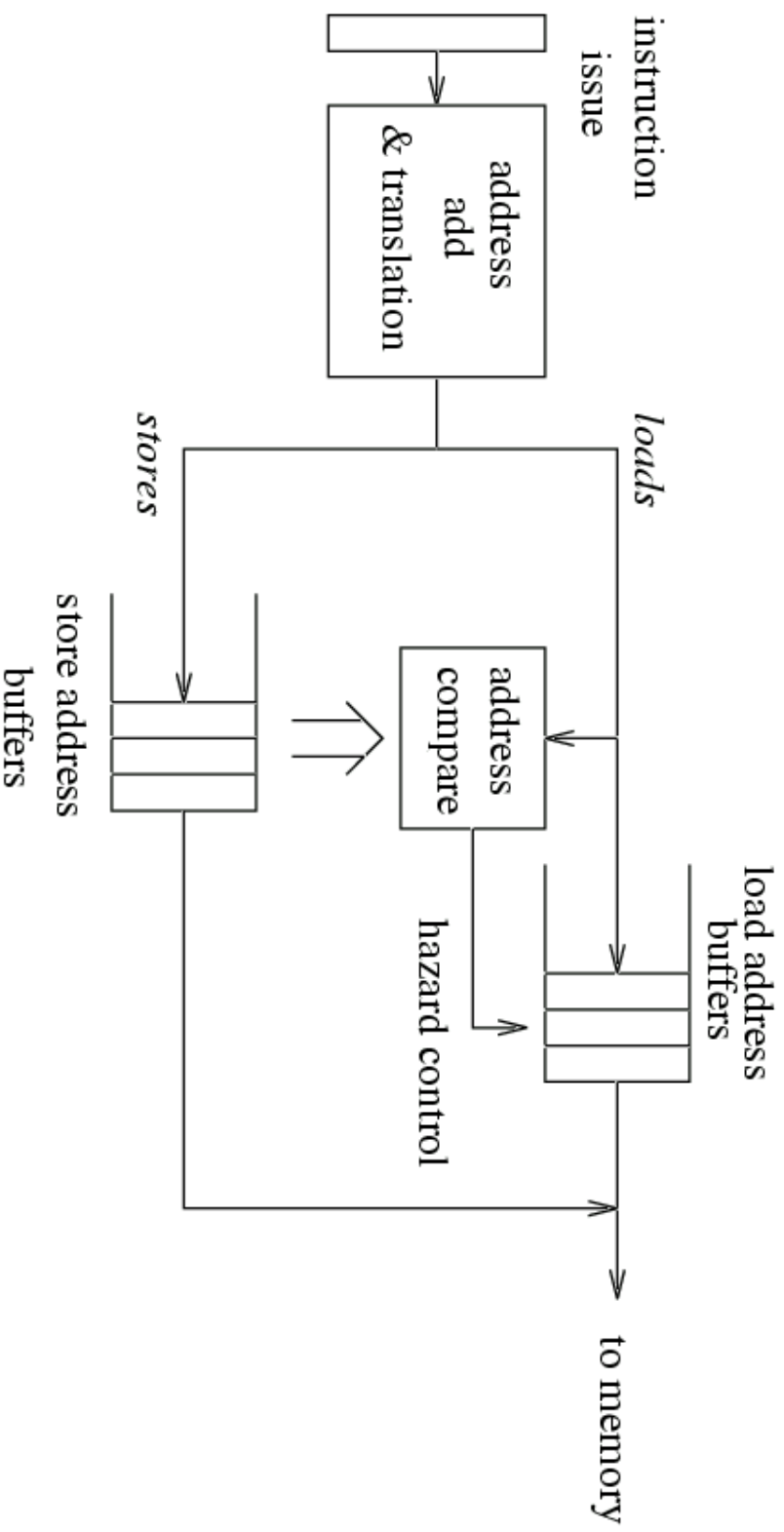
- σε μοντέρνα *RISC* σύνολα εντολών υπάρχουν ειδικές εντολές πρόσβασης μνήμης - *load, store*.
- επεξεργαστές χρησιμοποιούν ιεραρχία μνήμης, συνήθως 2 *cache*, ένα *on-chip* και ένα *off-chip*.
- για κάθε λειτουργία μνήμης η διεύθυνση πρέπει να υπολογιστεί.
- επιπλέον, η διεύθυνση που παράγεται πιθανώς πρέπει να μεταφραστεί σε φυσική διεύθυνση μέσω ενός *TLB*.

## Χειρισμός Δειτουργιών Μνήμης

- οι ενεργές εντολές μνήμης κρατώνται σε κάποιο χώρο.
- η πρόσβαση στα χαμηλά επίπεδα (πρώτο *cache*) συνήθως έχει πολλαπλές πόρτες.
- επίσεις μοντέρνοι επεξεργαστές υποστηρίζουν *non-blocking*, δηλαδή την εξυπηρέτηση εντολών μνήμης μετά από *miss*.
- κίνδυνοι μνήμης (για εκτέλεση εκτός σειράς) αναγνωρίζονται από τις ουρές μνήμης.

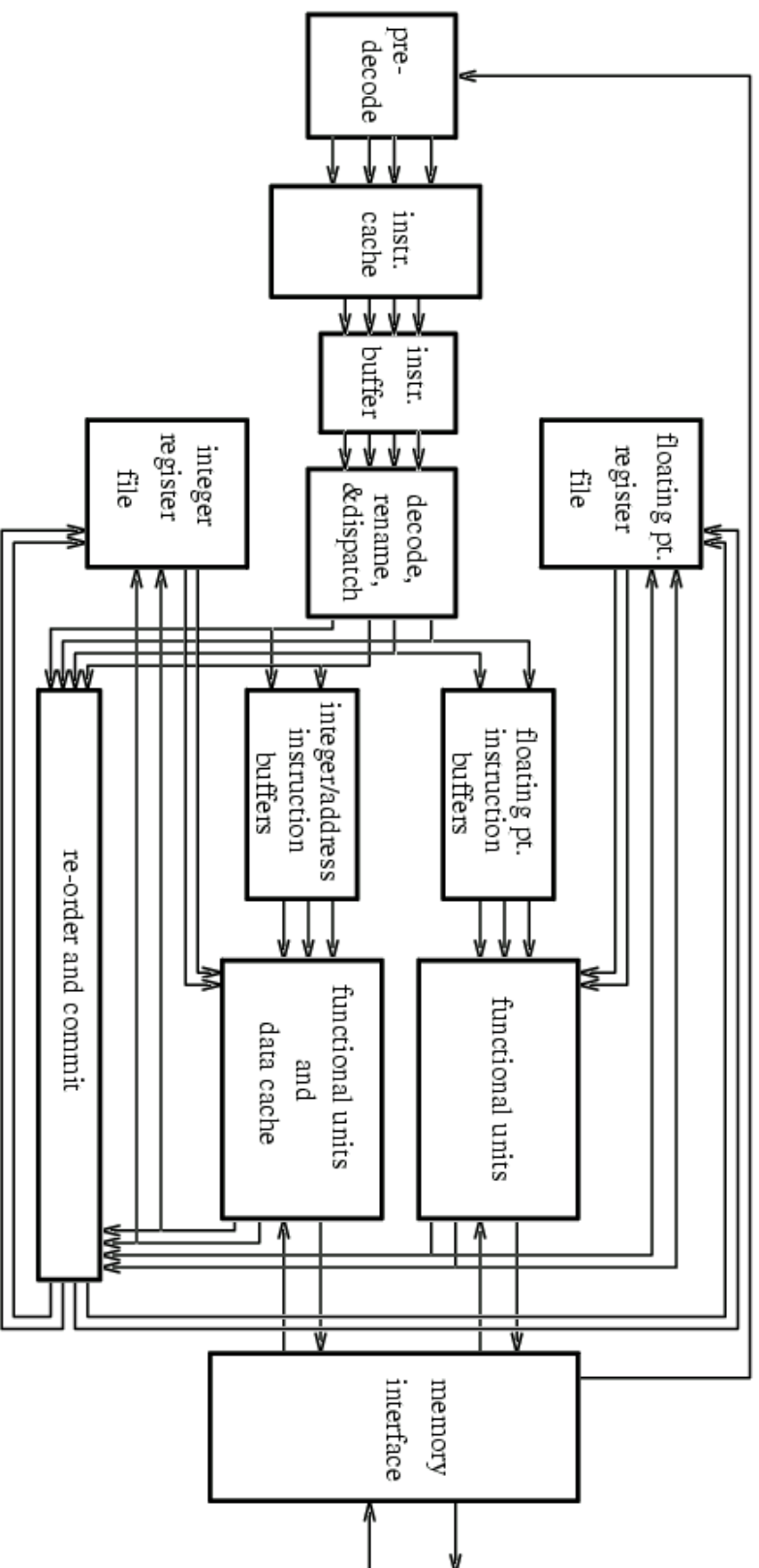


## Ουρές Μνήμης - *Store Buffers*

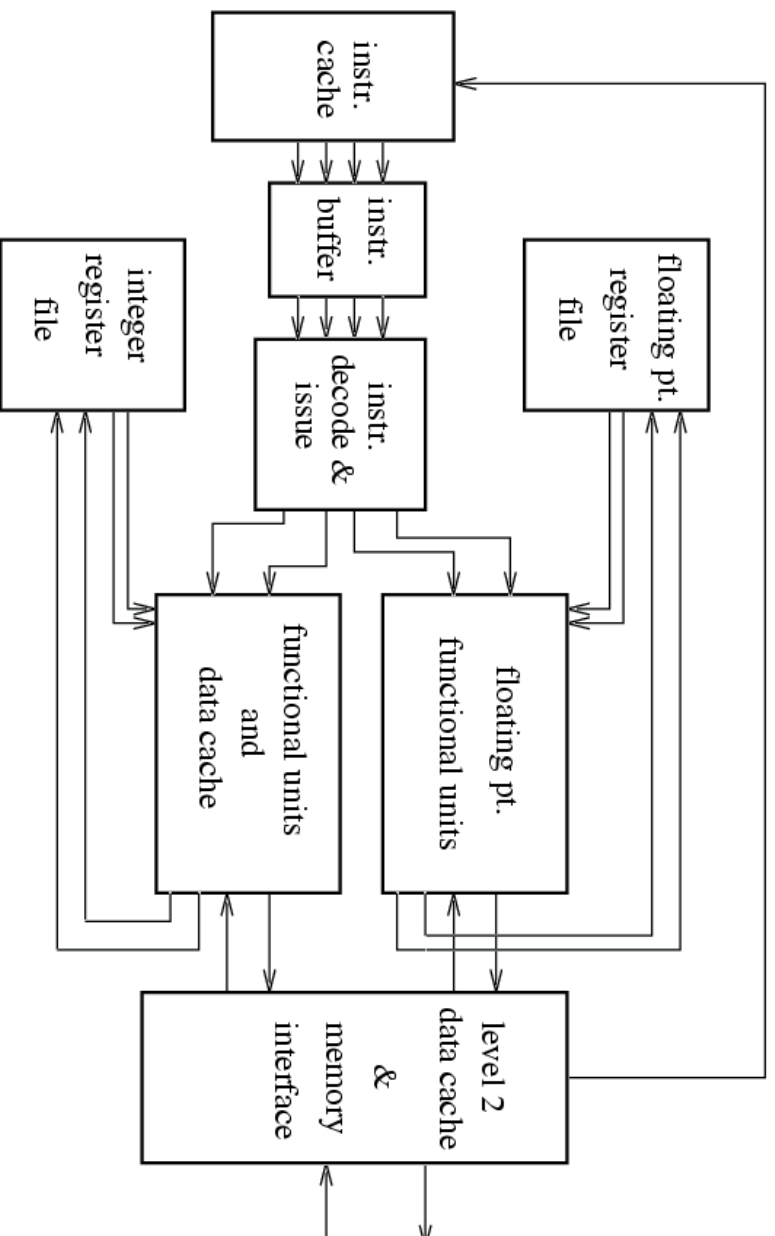


## Δέσμευση Κατάστασης Εντολής - *Instruction Commitment*

- το τελευταίο στάδιο.
- η εντολή αποσύρεται και το αποτέλεσμα της επιτρέπεται να αλλάξει την κατάσταση του επεξεργαστή.
- αυτή η φάση υλοποιεί την ψευδή αίσθηση της σειριακής εκτέλεσης.
- υλοποίηση μέσω της ουράς σειράς - *reorder buffer*.
- οι εντολές που είναι εγγυημένες ότι εκτελούνται (π.χ. πρόβλεψη) στέλνουν τα αποτελέσματα τους απο την ουρά σειράς στο A.K. (και στην μνήμη).

*MIPS R1000*

# Alpha 21264



# AMD K5

