

Αρχιτεκτονική Υπολογιστών



Γιώργος Δημητρίου

Ενότητα 8^η:

Στατική Δρομολόγηση Εντολών
(Παράλληλοι Βρόχοι &
Βεβαιωμένη Εκτέλεση)

Παράλληλοι Βρόχοι

- Εξαρτήσεις εντολών & επαναλήψεων βρόχου:

```
for (i=1; i<=100; i=i+1) {  
    A[i+1] = A[i] + B[i];  
    B[i+1] = C[i] + A[i+1];  
}
```

Σειριακός
βρόχος

```
for (i=1; i<=100; i=i+1) {  
    A[i] = A[i] + B[i+1];  
    B[i+1] = C[i] + D[i];  
}
```

Παράλληλος
βρόχος

Παράλληλοι Βρόχοι

- Μετασχηματισμοί βρόχων που μπορούν να γίνουν παράλληλοι:

```
for (i=1; i<=100; i=i+1) {  
    A[i] = A[i] + B[i];  
    B[i+1] = C[i] + D[i];  
}
```

Σειριακός
βρόχος

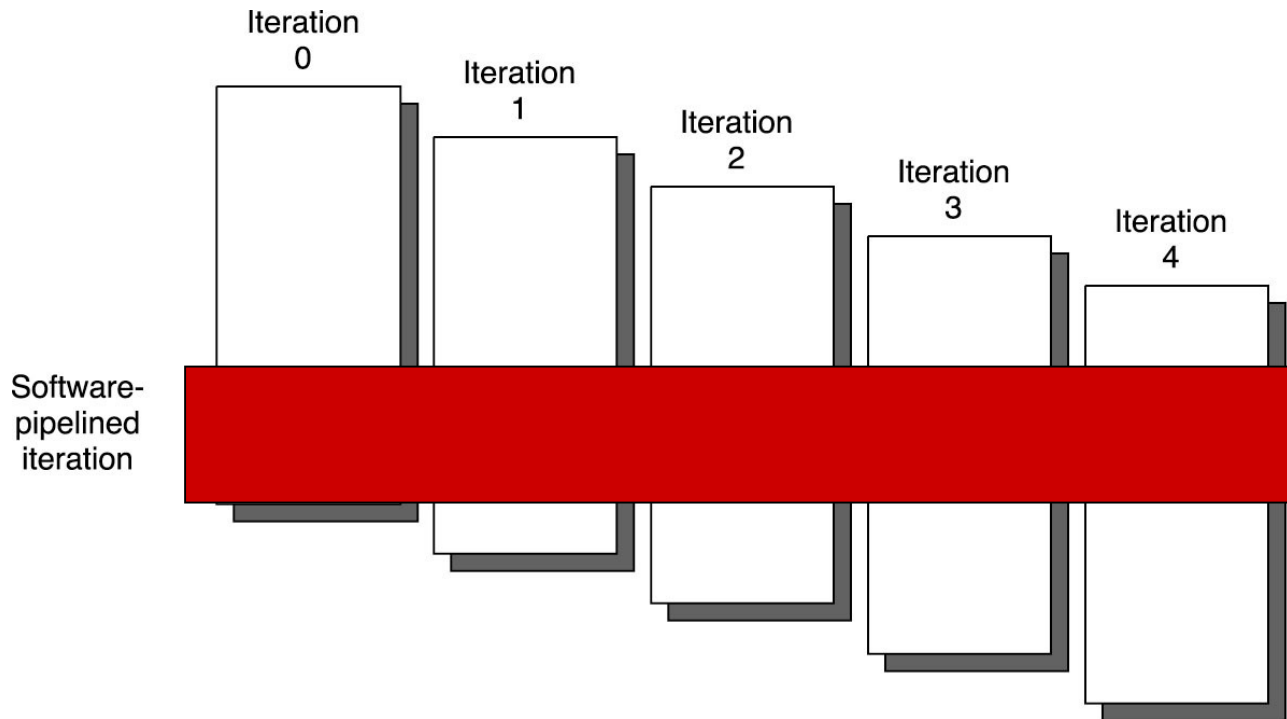
```
A[1] = A[1] + B[1];  
for (i=1; i<=99; i=i+1) {  
    B[i+1] = C[i] + D[i];  
    A[i+1] = A[i+1] + B[i+1];  
}  
B[101] = C[100] + D[100];
```

Παράλληλος
βρόχος

Έλεγχος ΜΚΔ (GCD)

- Έστω δύο αναφορές στο ίδιο διάνυσμα X :
 - $X[a*i+b]$, $X[c*i+d]$
- Αν υπάρχει εξάρτηση μεταξύ των δύο αναφορών, τότε πρέπει:
 - $(d-b) \bmod \text{GCD}(c,a) = 0$
- Αναγκαία συνθήκη, αλλά όχι και ικανή: μπορούμε να συμπεράνουμε ότι δεν υπάρχει εξάρτηση, αλλά όχι το ανάποδο!

Συμβολικό Ξεδίπλωμα Βρόχων



© 2003 Elsevier Science (USA). All rights reserved.

Παράδειγμα

- Αρχικός βρόχος:

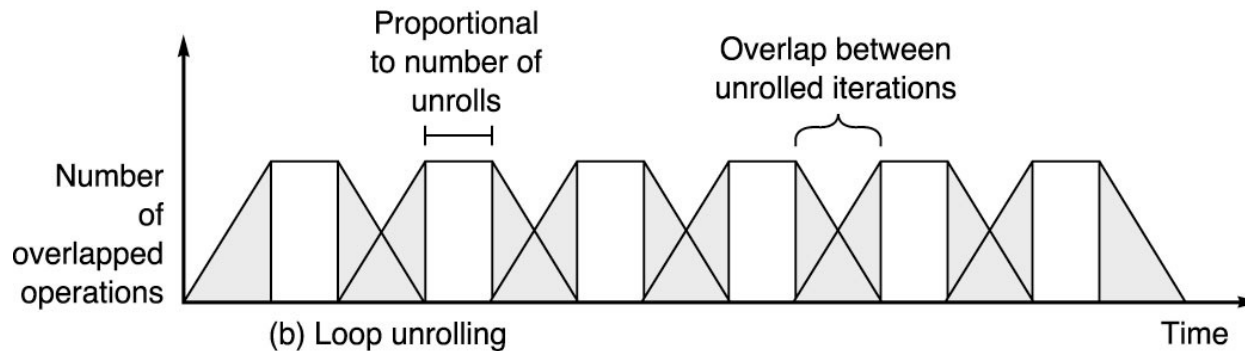
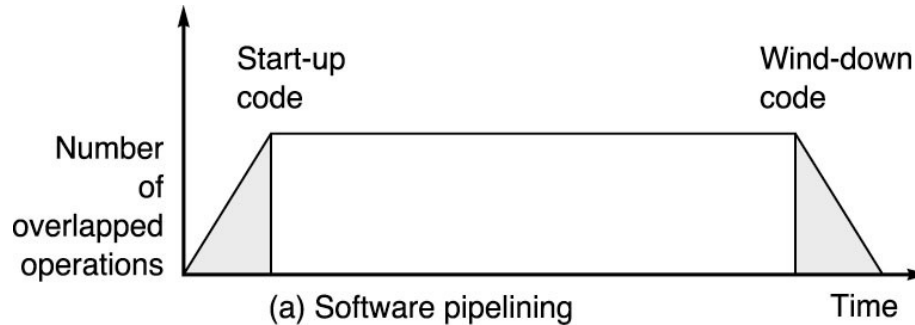
```
Loop: ldc1    $f0, 0($1)
      add.d  $f4, $f0, $f2
      sdc1   $f4, 0($1)
      addiu  $1, $1, -8
      bne   $1, $2, Loop
```

- Νέος βρόχος:

```
<prologue>
Loop: sdc1   $f4, 0($1)
      add.d  $f4, $f0, $f2
      ldc1   $f0, -16($1)
      addiu  $1, $1, -8
      bne   $1, $2, Loop
<epilogue>
```

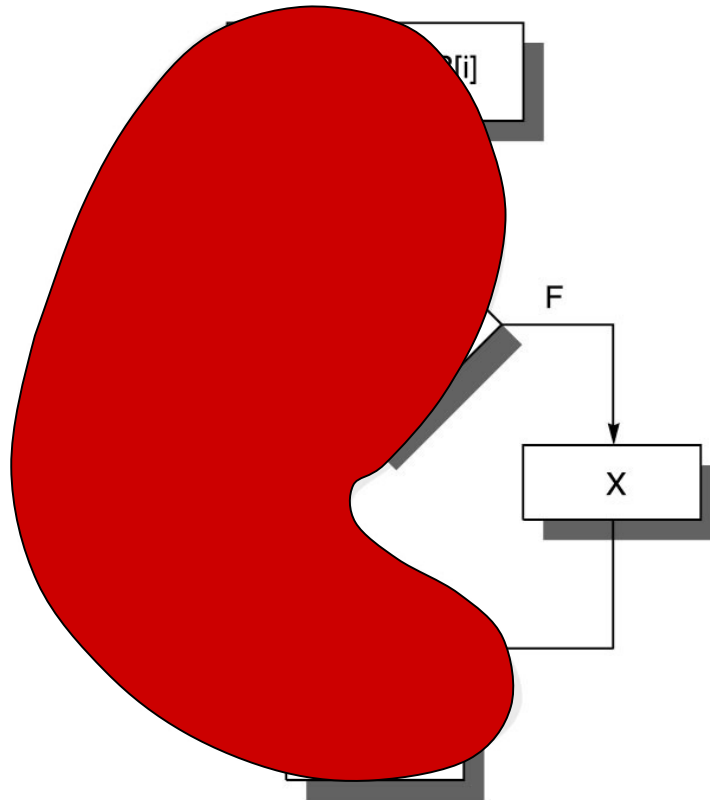
- Πώς προκύπτει μεγαλύτερο ILP;

Πραγματικό & Συμβολικό Ξεδίπλωμα



© 2003 Elsevier Science (USA). All rights reserved.

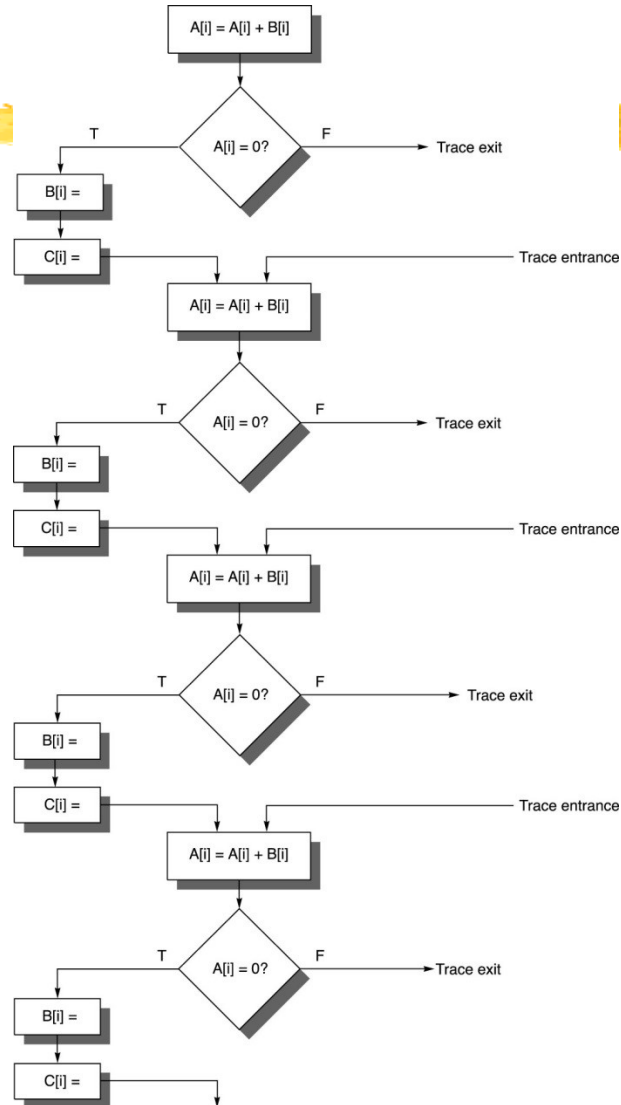
Καθολική Δρομολόγηση



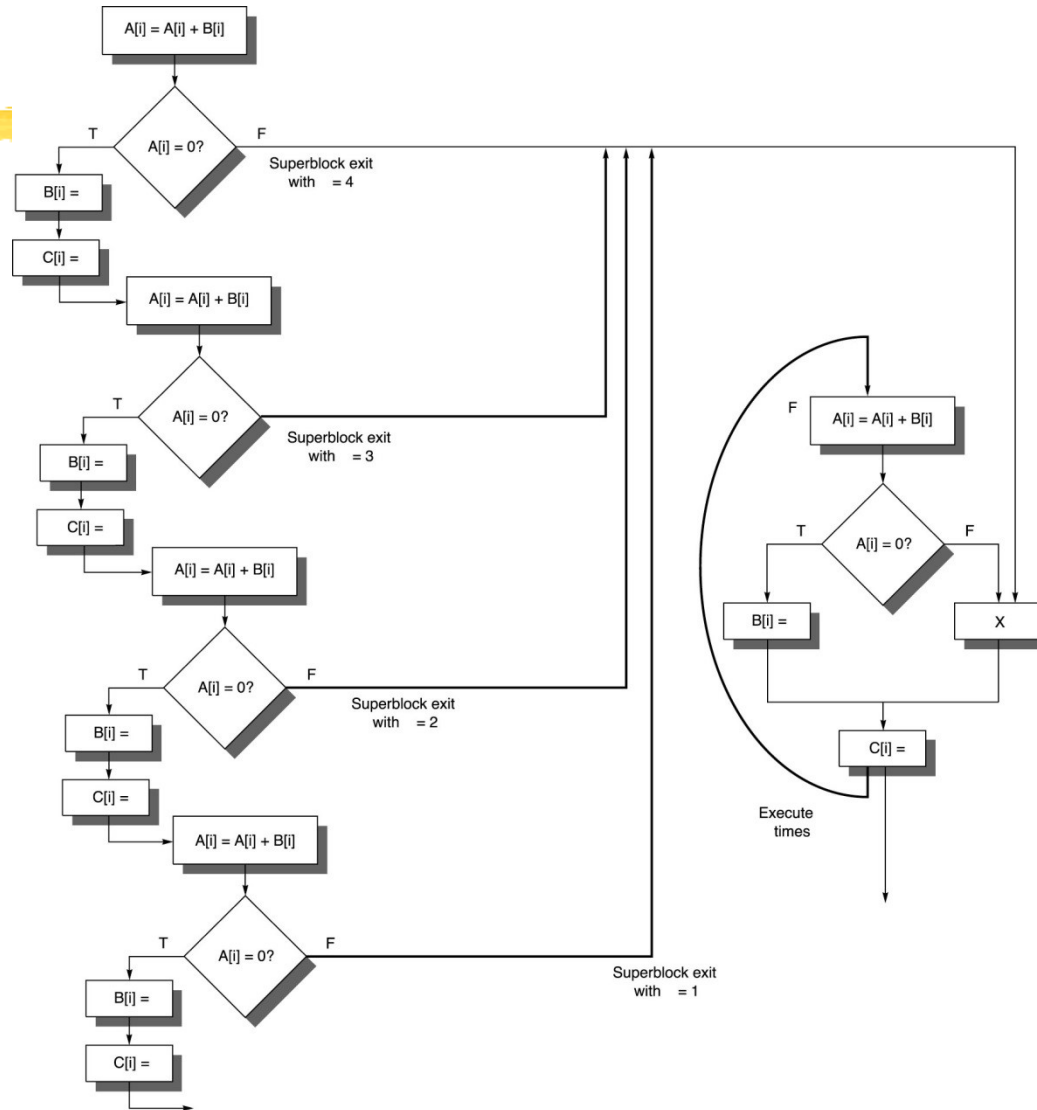
© 2003 Elsevier Science (USA). All rights reserved.

Αρχιτεκτονική Υπολογιστών

Δρομολόγηση ενός Δρόμου



Superblock



Υποστήριξη από το Υλικό για Αύξηση ILP

- Βεβαιωμένη εκτέλεση
 - Μετατροπή διαδικασιακά εξαρτημένων εντολών σε εντολές που εκτελούνται προαιρετικά
 - Οι διαδικασιακές εξαρτήσεις γίνονται εξαρτήσεις από δεδομένα
- Στατική υποθετική εκτέλεση
 - Ο μεταγλωττιστής παράγει ειδικές εντολές υποθετικής εκτέλεσης
 - Το υλικό εξασφαλίζει την ορθότητα εκτέλεσης

Βεβαιωμένη Εκτέλεση

- Η συνθήκη διακλάδωσης αποθηκεύεται σε κάποιον καταχωρητή:

- Ειδικοί καταχωρητές βεβαίωσης

```
cmp.ne p1=$8,$0
```

```
(p1) add $7,$8,$2
```

```
sw $7,0($1)
```

- Καταχωρητές γενικού σκοπού

```
slti $5,$9,8
```

```
cmovz $7,$1,$5
```

Στατική Υποθετική Εκτέλεση

- Ο μεταγλωττιστής μετακινεί εντολές για εκτέλεση πριν από τη διακλάδωση από την οποία εξαρτώνται
 - Ειδικές εντολές αγνοούν ή καθυστερούν τις ειδικές περιπτώσεις
- Ο μεταγλωττιστής μετακινεί εντολές φόρτωσης πριν από εντολές αποθήκευσης
 - Ειδικές εντολές φόρτωσης οδηγούν σε έλεγχο επικάλυψης όταν γίνεται η αποθήκευση

Παράδειγμα (speculative load)

- Πηγαίος κώδικας C:

```
if (A == 0) A = B; else A = A+4;
```

- Αρχική μετάφραση:

```
lw      $1, 0($3)    ;load A
bnez    $1, L1       ;test A
lw      $1, 0($2)    ;then clause (load B)
j       L2           ;skip else
L1: addiu $1, $1, 4   ;else clause (add 4 to A)
L2: sw   $1, 0($3)    ;store A
```

Παράδειγμα (speculative load)

- Απλή μετακίνηση εντολής φόρτωσης:

```
lw      $1, 0($3) ; load A
```

```
lw      $14, 0($2) ; speculative load B
```

```
beqz    $1, L2 ; test A
```

```
L1: addiu $14, $1, 4 ; else clause
```

```
L2: sw    $14, 0($3) ; store A
```

- Ειδική περίπτωση χωρίς επάνοδο δεν πρέπει να τερματίζει το πρόγραμμα, αλλά να επιστρέφει κατάλληλη τιμή

Παράδειγμα (speculative load)

- Υλοποίηση με εντολή υποθετικής φόρτωσης και ειδική εντολή ελέγχου:

```
lw      $1, 0($3) ; load A
slw     $14, 0($2) ; speculative, no exception
bnez    $1, L1     ; test A
specck  0($2)     ; speculation check
j       L2         ; skip else
L1: addiu $14, $1, 4 ; else clause
L2: sw   $14, 0($3) ; store A
```

- Ο έλεγχος μπορεί να υπονοηθεί με poison bits

Παράδειγμα (advanced load)

- Αρχικός κώδικας:

```
sw      $12, 0 ($4) ; store
lw      $6, 0 ($8)  ; load
sw      $6, 0 ($18) ; store
```

- Εντολή πρώιμης φόρτωσης και ελέγχου:

```
lw.a   $6, 0 ($8)  ; advanced load
sw      $12, 0 ($4) ; store
lw.c    $6, 0 ($8)  ; check load
sw      $6, 0 ($18) ; store
```

- Έλεγχος με κώδικα επαναφοράς