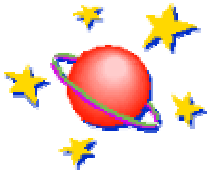

Ενότητα 3η: *Εξαρτήσεις ελέγχου – Αρχές και τεχνικές της παραλληλίας σε επίπεδο βρόχου*

Σκοπός **Ο σκοπός της ενότητας αυτής είναι να εξετάσει τις εξαρτήσεις ελέγχου, καθώς και να παρουσιάσει δύο ιδιότητες που παίζουν σημαντικό ρόλο στη σωστή εκτέλεση του προγράμματος, τη ροή δεδομένων και τη διαχείριση διακοπών. Στη συνέχεια γίνεται εφαρμογή σε παραδείγματα στα οποία η ανάπτυξη σε επίπεδο βρόχου καθορίζει τις εξαρτήσεις μεταξύ των εντολών του κυρίου μέρους του βρόχου και όλων των επαναλήψεών του.**

Προσδοκώμενα Αποτελέσματα **Όταν θα έχετε μελετήσει την ενότητα, θα είστε σε θέση:**

<u>να εντοπίζετε τις εξαρτήσεις ελέγχου σε ένα τμήμα κώδικα,</u>

<u>να περιγράφετε τον τρόπο με τον οποίο οι εξαρτήσεις ελέγχου περιορίζουν το βαθμό της παραλληλίας,</u>

<u>να εφαρμόζετε τις αρχές και τις τεχνικές της παραλληλίας σε επίπεδο βρόχου.</u>

εξάρτηση ελέγχου, διαχείριση διακοπών, ροή δεδομένων, υπόθεση, ζωτικότητα

Εξαρτήσεις ελέγχου

Η **εξάρτηση ελέγχου** καθορίζει τη σειρά εκτέλεσης μιας εντολής, λαμβάνοντας υπόψη την εντολή διακλάδωσης με συνθήκη, έτσι ώστε η εντολή να εκτελείται μόνο όταν είναι απαραίτητο.

Παράδειγμα:

Ένα απλό παράδειγμα εξάρτησης ελέγχου είναι το τμήμα then ενός βρόχου if. Έστω το παρακάτω τμήμα κώδικα :

```
if  p1 {
        S1;
      };
if  p2 {
        S2;
      };
```

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

- **Μπορείτε να αναγνωρίσετε ποιες εντολές έχουν εξάρτηση ελέγχου από συνθήκες στον κώδικα του παραδείγματος;**
- **Να περιγράψετε τους περιορισμούς που αφορούν στις εξαρτήσεις ελέγχου.**
- **Θεωρείτε πως μπορούμε να αλλάξουμε τη σειρά των εντολών; Να αιτιολογήσετε την απάντησή σας.**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1

Η εντολή S1 έχει εξάρτηση ελέγχου από τη συνθήκη p1 και η εντολή S2 εξαρτάται από τη συνθήκη p2, αλλά όχι από την p1.
Οι περιορισμοί που αφορούν στις εξαρτήσεις ελέγχου είναι :

- **Μία εντολή εξαρτώμενη από μία διακλάδωση με συνθήκη δεν πρέπει να μεταφερθεί πριν τη διακλάδωση, γιατί τότε η εκτέλεση της δεν θα ελέγχεται από αυτή. Για παράδειγμα, αν η εντολή S1 μεταφερθεί πριν τη συνθήκη διακλάδωσης p1, τότε δεν θα ελέγχεται από την p1.**
- **Μία εντολή που δεν έχει εξάρτηση ελέγχου από μία διακλάδωση με συνθήκη δεν πρέπει να μεταφερθεί μετά τη διακλάδωση, γιατί τότε η εκτέλεση της θα ελέγχεται από αυτή. Για παράδειγμα, αν η εντολή S1 μεταφερθεί μετά τη συνθήκη διακλάδωσης p2, τότε θα ελέγχεται από την p2.**

Είναι δυνατόν μερικές φορές, αυτοί οι περιορισμοί και οι καθυστερήσεις που αφορούν στις εντολές του παραδείγματος, να παραβιαστούν, χωρίς όμως να έχουμε λάθος εκτέλεση.

Στη συνέχεια θα μελετήσουμε τον τρόπο με τον οποίο οι εξαρτήσεις ελέγχου περιορίζουν το βαθμό της παραλληλίας.

Παράδειγμα:

Να δείξετε την ανάπτυξη του βρόχου και να καθορίσετε τις εξαρτήσεις ελέγχου. Πως οι εξαρτήσεις ελέγχου μετακινήθηκαν ;

```
loop:  LD      F0 , 0 (R1)
       ADDD    F4 , F0 , F2
       SD      0 (R1) , F4          Αρχικό βασικό μπλοκ
       SUBI    R1 , R1 , #8
       BEQZ    R1 , exit
       LD      F6 , 0 (R1)
       ADDD    F8 , F6 , F2
       SD      0 (R1) , F8
       SUBI    R1 , R1 , #8         ;παράλειψη της BNEZ
       BEQZ    R1 , exit
       LD      F10 , 0 (R1)
       ADDD    F12 , F10 , F2
       SD      0 (R1) , F12
       SUBI    R1 , R1 , #8         ;παράλειψη της BNEZ
       BEQZ    R1 , exit
       LD      F14 , 0 (R1)
       ADDD    F16 , F14 , F2
       SD      0 (R1) , F16
       SUBI    R1 , R1 , #8         ;παράλειψη της BNEZ
       BNEZ    R1 , loop

exit :
```

Απάντηση:

Κάθε εντολή, εκτός από αυτές του αρχικού βασικού μπλοκ, έχει εξάρτηση ελέγχου από τις εντολές διακλάδωσης BEQZ. Οι εξαρτήσεις ελέγχου φαίνονται με τα βέλη. Η συνθήκη BEQZ είναι συμπληρωματική της BNEZ και μας προστατεύει από κακή εκτέλεση του προγράμματος. Συγκεκριμένα η εντολή διακλάδωσης με συνθήκη

BEQZ δεν επιτρέπει το περιεχόμενο του καταχωρητή R1 να γίνει αρνητικό, εφόσον δεν υπάρχουν αρνητικές θέσεις μνήμης.
Η παρουσία των ενδιάμεσων εντολών διακλάδωσης με συνθήκη BEQZ, εμποδίζει **:**

- Τις αλλαγές των εξαρτήσεων ελέγχου που προκύπτουν από τη μετακίνηση των εντολών, αν είχαμε προγραμματισμό του βρόχου.
- Την απομάκρυνση των εντολών SUBI, αφού η νέα τιμή του R1 σε κάθε εντολή SUBI, χρησιμοποιείται σε κάθε εντολή διακλάδωσης BEQZ.

Πρωταρχικός στόχος λοιπόν, είναι να αφαιρέσουμε τις ενδιάμεσες εντολές διακλάδωσης με συνθήκη, με αποτέλεσμα την αλλαγή των εξαρτήσεων ελέγχου και την απομάκρυνση των περιττών εντολών SUBI. Σε αυτή την περίπτωση το περιεχόμενο του R1 είναι πολλαπλάσιο του 32 και επομένως ο αριθμός των επαναλήψεων του βρόχου είναι πολλαπλάσιο του 4. Επειδή η συνθήκη των τριών ενδιάμεσων BEQZ είναι συμπληρωματική, οι εντολές BEQZ δεν εκτελούνται και τις απομακρύνουμε. Έτσι λοιπόν, καμία εντολή δεν έχει εξάρτηση ελέγχου από τις διακλαδώσεις.
Οπότε ο κώδικας μετασχηματίζεται σε:

```
loop: LD      F0 , 0 (R1)
      ADDD    F4 , F0 , F2
      SD      0 (R1) , F4
      LD      F0 , -8 (R1)
      ADDD    F4 , F0 , F2
      SD      -8 (R1) , F4
      LD      F0 , -16 (R1)
      ADDD    F4 , F0 , F2
      SD      -16 (R1) , F4
      LD      F0 , -24 (R1)
      ADDD    F4 , F0 , F2
      SD      -24 (R1) , F4
      SUBI    R1 , R1 , #32
      BNEZ    R1 , loop
```

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

- **Να εκτελέσετε των κώδικα του παραδείγματος χρησιμοποιώντας το λογισμικό προσομοίωσης του DLX, Windlx, το οποίο παρατίθεται στις ιστοσελίδες:**

  http://www.gup.uni-linz.ac.at/downloads/ti3/windlx

  http://www.mkp.com/pub/dlx
- **Να συγκρίνετε την απάντηση του παραδείγματος με τα αποτελέσματα από τη χρήση του λογισμικού προσομοίωσης Windlx.**
- **Για περισσότερες λεπτομέρειες καλό θα ήταν να συμβουλευόσασταν το παράρτημα Α: «Εργαλεία Προσομοίωσης του DLX».**

Η εξάρτηση ελέγχου διατηρείται λόγω δύο ιδιοτήτων των απλών σωληνώσεων, χωρίς όμως να παίζει καθοριστικό ρόλο.

- Οι εντολές εκτελούνται με τη σειρά, εξασφαλίζοντας ότι μία εντολή που βρίσκεται πριν από μία διακλάδωση θα εκτελεστεί πρώτη.

- Η ανίχνευση των εξαρτήσεων ή των κινδύνων ελέγχου εξασφαλίζει ότι μια εντολή που έχει εξάρτηση ελέγχου από μια διακλάδωση με συνθήκη δεν θα εκτελεστεί, μέχρι να γίνει γνωστός ο στόχος της διακλάδωσης.

Υπάρχουν δύο ιδιότητες που παίζουν σημαντικό ρόλο στην σωστή εκτέλεση του προγράμματος. Αυτές είναι, η διαχείριση διακοπών (ή εξαιρέσεων) και η ροή δεδομένων οι οποίες διαφυλάσσονται από την εξάρτηση ελέγχου.

Με τον όρο διαχείριση διακοπών, συνήθως εννοούμε ότι οποιεσδήποτε αλλαγές στη σειρά εκτέλεσης των εντολών δεν πρέπει να προκαλούν την εμφάνιση νέων διακοπών στο πρόγραμμα.

Στη συνέχεια θα δείξουμε πως μπορούμε να αποφύγουμε τέτοιες περιπτώσεις διατηρώντας τις εξαρτήσεις ελέγχου.

Παράδειγμα:

Θεωρούμε το παρακάτω τμήμα κώδικα ακολουθίας εντολών, όπου οι διακλαδώσεις εκτελούνται χωρίς καθυστέρηση.

```
BEQZ   R2 , L1
LW     R1 , 0 (R2)
L1:
```

Απάντηση:

Η LW όταν εκτελεστεί πιθανώς να προκαλέσει παραβίαση προστασίας μνήμης. Εάν η εντολή BEQZ ακολουθείται η LW δεν θα εκτελεστεί. Η εξάρτηση δεδομένων δεν μας εμποδίζει να μεταφέρουμε την εντολή LW πριν την εντολή BEQZ, παρά μόνο η εξάρτηση ελέγχου. Αγνοώντας λοιπόν, την εξάρτηση ελέγχου μεταφέρουμε την εντολή LW πριν την εντολή BEQZ. Σε αυτή την περίπτωση η εντολή LW θα εκτελείται πάντα προκαλώντας πιθανώς διακοπή (ή εξαίρεση).

Η ροή δεδομένων είναι η πραγματική ροή δεδομένων μεταξύ των εντολών.

Οι διακλαδώσεις επιτρέπουν στα δεδομένα μιας εντολής να προέρχονται από πολλά σημεία, με αποτέλεσμα να καθιστούν δυναμική τη ροή δεδομένων.

Παράδειγμα:

Θεωρούμε το παρακάτω τμήμα κώδικα.

```
     ADD    R1 , R2 , R3
     BEQZ   R4 , L
     SUB    R1 , R5 , R6
L:   OR     R7 , R1 , R8
```

Απάντηση:

Η τιμή του R1, που χρησιμοποιείται από την εντολή OR, εξαρτάται από το εάν η συνθήκη διακλάδωσης αληθεύει ή όχι. Η εξάρτηση δεδομένων μόνο δεν επαρκεί για την εξασφάλιση της ορθότητας του προγράμματος. Αν η συνθήκη διακλάδωσης δεν αληθεύει, τότε εκτελείται η εντολή SUB και η τιμή του R1 που υπολογίστηκε στην εντολή SUB, θα χρησιμοποιηθεί από την εντολή OR. Αν αληθεύει, τότε εκτελείται η εντολή OR και η τιμή του R1 που υπολογίστηκε στην εντολή ADD θα χρησιμοποιηθεί από την εντολή OR. Διατηρώντας την εξάρτηση ελέγχου της εντολής SUB από την εντολή διακλάδωσης (δηλαδή δεν μετακινείται η εντολή SUB), αποτρέπεται κάποια παράνομη αλλαγή στη ροή των δεδομένων.

Μερικές φορές, η παραβίαση της εξάρτησης ελέγχου δεν έχει επίδραση ούτε στη διαχείριση διακοπών ούτε στη ροή δεδομένων.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Έστω το παρακάτω τμήμα κώδικα:

```
            ADD    R1 , R2 , R3
            BEQZ   R12 , skipnext
            SUB    R4 , R5 , R6
            ADD    R5 , R4 , R9
skipnext:   OR     R7 , R8 , R9
```

Μπορείτε να εξηγήσετε τον τρόπο με τον οποίο θα εκτελεστούν οι εντολές;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3

Εάν η συνθήκη διακλάδωσης αληθεύει τότε εκτελείται η εντολή με ετικέτα skipnext, δηλαδή η εντολή OR. Στην περίπτωση αυτή η εντολή SUB δεν εκτελείται. Εάν όμως, η συνθήκη διακλάδωσης δεν αληθεύει, τότε εκτελείται η εντολή SUB. Σε αυτή την περίπτωση (η εντολή διακλάδωσης BEQZ δεν ακολουθείται), ο υπολογιστής υποθέτει ότι ο καταχωρητής R4, της SUB εντολής, δεν πρόκειται να χρησιμοποιηθεί μετά την εντολή με ετικέτα skipnext. 'Ετσι λοιπόν, η εντολή SUB μπορεί να μετακινηθεί πριν την πρώτη εντολή ADD, οπότε θα εκτελείται πάντα, ανεξάρτητα από τη συνθήκη διακλάδωσης και δεν θα επηρεάζει το τμήμα κώδικα μετά την εντολή με ετικέτα skipnext.

Η μορφή προγραμματισμού που χρησιμοποιήθηκε για την απάντηση της δραστηριότητας 3 λέγεται **υπόθεση**, αφού ουσιαστικά ο μεταγλωττιστής ποντάρει στο αποτέλεσμα της διακλάδωσης.

Η ιδιότητα που καθορίζει κατά πόσον η τιμή ενός καταχωρητή θα χρησιμοποιηθεί από επόμενη εντολή λέγεται <u>ζωτικότητα.</u>

Η εξάρτηση ελέγχου διατηρείται πραγματοποιώντας ανίχνευση των κινδύνων ελέγχου, οι οποίοι προκαλούν καθυστερήσεις ελέγχου. Οι καθυστερήσεις ελέγχου εξαλείφονται ή περιορίζονται με διάφορες τεχνικές.

Αρχές και τεχνικές της παραλληλίας σε επίπεδο βρόχου.

Η ανάλυση σε επίπεδο βρόχου καθορίζει τις εξαρτήσεις μεταξύ των εντολών του κυρίου μέρους του βρόχου και όλων των επαναλήψεων του. Επίσης καθορίζει κατά πόσο τα δεδομένα των μεταγενέστερων επαναλήψεων έχουν εξάρτηση δεδομένων με τα δεδομένα που παράχθηκαν από τις αρχικές επαναλήψεις. Στη συνέχεια θα ασχοληθούμε μόνο με τις εξαρτήσεις δεδομένων, ενώ οι εξαρτήσεις ονομάτων που υπάρχουν θα αντιμετωπιστούν με τη μέθοδο της μετονομασίας. Οι υπολογισμοί σε κάθε επανάληψη είναι ανεξάρτητοι των προηγούμενων επαναλήψεων.

Παράδειγμα:

Έστω το παρακάτω τμήμα κώδικα

```
for   (i=1; i<=1000; i++)
        x[i] = x[i] + s;
```

Απάντηση:

Στο σώμα του βρόχου υπάρχει εξάρτηση μεταξύ της παλιάς και νέας τιμής του x[i], η οποία βρίσκεται, κάθε φορά, μέσα σε μία επανάληψη. Αντιθέτως δεν υπάρχει εξάρτηση μεταξύ εντολών διαφορετικών επαναλήψεων. Επομένως ο βρόχος είναι παράλληλος. Συνεπώς αφού μετατρέψουμε τον πηγαίο κώδικα σε γλώσσα assembly, αναπτύσσουμε το βρόγχο **:**

```
loop:  LD      F0 , 0 (R1)
       ADDD    F4 , F0 , F2
       SD       0 (R1) , F4     ; παράλειψη των SUBI και BNEZ

       LD      F6 , -8 (R1)
       ADDD    F8 , F6 , F2
       SD      -8 (R1) , F8    ; παράλειψη των SUBI και BNEZ

       LD      F10 , -16 (R1)
       ADDD    F12 , F10 , F2
       SD      -16 (R1) , F12 ; παράλειψη των SUBI και BNEZ

       LD      F14 , -24 (R1)
       ADDD    F16 , F14 , F2
       SD      -24 (R1),  F16
       SUBI     R1 , R1 , #32
       BNEZ     R1 , loop
```

Στον κώδικα, μετά την ανάπτυξη, δημιουργείται εξάρτηση μεταξύ των επαναλήψεων λόγω του καταχωρητή R1.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

- **Να εκτελέσετε των κώδικα του παραδείγματος χρησιμοποιώντας το λογισμικό προσομοίωσης του DLX, Windlx, το οποίο παρατίθεται στις ιστοσελίδες:**

  http://www.gup.uni-linz.ac.at/downloads/ti3/windlx

  http://www.mkp.com/pub/dlx
- **Να συγκρίνετε την απάντηση του παραδείγματος με τα αποτελέσματα από τη χρήση του λογισμικού προσομοίωσης Windlx.**
- **Για περισσότερες λεπτομέρειες καλό θα ήταν να συμβουλευόσασταν το παράρτημα Α: «Εργαλεία Προσομοίωσης του DLX».**

Παράδειγμα:

Θεωρούμε το βρόχο :

```
for   (i=1; i<=100; i = i+1) {
      A[i+1] = A[i] + C[i];   /* S1 */
      B[i+1] = B[i] + A[i+1]; }  /* S2 */
```

Υποθέτουμε ότι οι πίνακες A, B και C είναι διαφορετικοί χωρίς επικαλύψεις. Ποιες είναι οι εξαρτήσεις δεδομένων μεταξύ των εντολών S1 και S2 στο βρόχο ;

Απάντηση:

Υπάρχουν δύο διαφορετικές εξαρτήσεις **:**

- Η S1 χρησιμοποιεί την τιμή που υπολογίστηκε από την S1 σε προηγούμενη επανάληψη , αφού η επανάληψη i υπολογίζει το A[i+1] το οποίο διαβάζεται στην i+1 επανάληψη. Το ίδιο συμβαίνει και με την S2 για τα B[i] και B[i+1].

- Η S2 χρησιμοποιεί την τιμή A[i+1] που υπολογίστηκε από την S1 στην ίδια επανάληψη.

Θα εξετάσουμε κάθε περίπτωση χωριστά, αγνοώντας κάθε φορά την ύπαρξη της άλλης.

Η πρώτη περίπτωση εξάρτησης, όπου η εντολή S1 (ή S2) εξαρτάται από την εντολή S1 (ή S2) της αμέσως προηγούμενης επανάληψης, ονομάζεται εξάρτηση που μεταφέρεται με τον βρόχο. Η εξάρτηση αυτή, παρουσιάζεται μεταξύ δύο διαδοχικών επαναλήψεων του βρόχου. Επειδή η εντολή S1 εξαρτάται από τον εαυτό της, πρέπει

οι επαναλήψεις της εντολής S1 να εκτελούνται με τη σειρά προκειμένου να είναι επιτυχείς. Το ίδιο ισχύει και για την S2.

Η δεύτερη περίπτωση εξάρτησης, όπου η S2 εξαρτάται από την S1, βρίσκεται μέσα σε κάθε επανάληψη, δηλαδή στο εσωτερικό του βρόχου και άρα δεν είναι εξάρτηση που μεταφέρεται με τον βρόχο. Συνεπώς, πολλαπλές επαναλήψεις του βρόχου εκτελούνται παράλληλα, εφόσον διατηρούμε τη σειρά για κάθε ζευγάρι εντολών σε κάθε επανάληψη.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Έστω το παρακάτω τμήμα κώδικα.

```
for   (i=1; i<=100; i = i+1)
      {
        A[i] = A[i] + B[i];   /* S1 */
        B[i+1] = C[i] + D[i]; /* S2 */
      }
```

- **Μπορείτε να εντοπίστε τις εξαρτήσεις δεδομένων μεταξύ των εντολών S1 και S2;**
- **Θεωρείτε πως είναι ο βρόχος αυτός παράλληλος; Αν θεωρείτε πως δεν είναι, να δείξετε τον τρόπο με τον οποίο θα γίνει παράλληλος.**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Η εντολή S1 χρησιμοποιεί την τιμή που καθορίστηκε στην προηγούμενη επανάληψη από τη εντολή S2 και άρα υπάρχει μια εξάρτηση που μεταφέρεται με τον βρόχο. Σε αντιδιαστολή με το βρόχο του προηγούμενου παραδείγματος, αυτή η εξάρτηση δεν είναι κυκλική γιατί καμία εντολή δεν εξαρτάται από τον εαυτό της και ενώ η S1 εξαρτάται από την S2, η S2 δεν εξαρτάται από την S1. Ο βρόχος μπορεί να γίνει παράλληλος, εκτός αν υπάρχει κύκλος εξαρτήσεων. Η απουσία κύκλου σημαίνει ότι οι εξαρτήσεις προϋποθέτουν ένα μερικό καθορισμό σειράς εκτέλεσης των εντολών.

Αν και δεν υπάρχουν κυκλικές εξαρτήσεις στον παραπάνω βρόχο, πρέπει να μετατραπεί έτσι ώστε να είναι προσαρμοσμένος σε μια μερική διάταξη των εντολών και να εμφανίζεται η παραλληλία.

Δύο παρατηρήσεις είναι σημαντικές για την υλοποίηση αυτής της μετατροπής :

1. **Η S2 δεν εξαρτάται από την S1 οπότε η εναλλαγή των δύο εντολών δεν θα έχει επίδραση στην εκτέλεση της S2. Αν είχε εξάρτηση, τότε θα είχαμε κυκλική εξάρτηση και ο βρόχος δεν θα ήταν παράλληλος.**

2. **Κατά τη διάρκεια της πρώτης επανάληψης του βρόχου, η εντολή S1 εξαρτάται από την αρχική τιμή που δόθηκε στον B[1].**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Αυτές οι δύο παρατηρήσεις μας επιτρέπουν να αντικαταστήσουμε τον παραπάνω βρόχο με το επόμενο τμήμα κώδικα :

```
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];
```

Η εξάρτηση μεταξύ των δύο εντολών δεν είναι εξάρτηση που μεταφέρεται με τον βρόχο, οπότε οι επαναλήψεις του βρόχου μπορούν να επικαλυφθούν, με την προϋπόθεση όμως, ότι οι εντολές σε κάθε επανάληψη διατηρούν την κατάλληλη σειρά.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 6

Έστω ότι έχουμε δύο αρχεία: το second1.s και το second2.s, τα οποία περιέχουν τα αντίστοιχα προγράμματα:

Ο κώδικας του second1.s	Ο κώδικας του second1.s

.data
.word 4,4
.text
main: LW r1,$DATA(r0)
LW r2,$DATA +4(r0)
SUB r3,r1,#2
BNEZ r3,L1
ADD r1,r0,r0
L1: SUB r3,r2,#2
BNEZ r3,L2
ADD r1,r0,r0
L2: SUB r3,r1,r2
	.data
	.word 4,4
	.text
	main: LW r1,$DATA(r0)
	LW r2,$DATA +4(r0)
	SUB r3,r1,#4
	BNEZ r3,L1
	ADD r1,r0,r0
	L1: SUB r3,r2,#4
	BNEZ r3,L2
	ADD r2,r0,r0
	L2: SUB r3,r1,r2

- **Να προσδιορίσετε τους κινδύνους που προκύπτουν κατά την εκτέλεση των προγραμμάτων και να προτείνετε τρόπους αντιμετώπισης των κινδύνων.**
- **Να εκτελέσετε τα προγράμματα χρησιμοποιώντας το λογισμικό προσομοίωσης του DLX, Windlx, το οποίο παρατίθεται στην ιστοσελίδα: http://www.gup.uni-linz.ac.at/downloads/ti3/windlx**
- **Να συγκρίνετε την απάντηση που δώσατε με τα αποτελέσματα από την εκτέλεση των προγραμμάτων από τον προσομοιωτή Windlx.**
- **Για περισσότερες λεπτομέρειες καλό θα ήταν να συμβουλευτείτε το παράρτημα Α: «Εργαλεία Προσομοίωσης του DLX».**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 6

Στο πρόγραμμα second1.s προκύπτει κίνδυνος δεδομένων RAW επειδή στις εντολές LW r1,$DATA(r0) και SUB r3,r1,#2 η SUB χρειάζεται την τιμή του r1 στο στάδιο ID, ενώ η LW βρίσκεται στο στάδιο MEM. Ο κίνδυνος αντιμετωπίζεται με προώθηση από το στάδιο MEM της LW στο στάδιο EX της SUB.

Από την εκτέλεση του προγράμματος second1.s παρατηρούμε:

Οι τιμές των r1, r2 είναι τέτοιες ώστε να ακολουθούνται και οι δύο διακλαδώσεις.

Οι εντολές ADD δεν εκτελούνται ποτέ.

Έχουμε δύο καθυστερήσεις ενός κύκλου λόγω των εντολών διακλάδωσης:

BNEZ r3,L1 και BNEZ r3,L2

για κάθε φορά που εκτελείται η ακολουθία εντολών.

Από την εκτέλεση του προγράμματος second2.s παρατηρούμε:

Στο second2.s δεν ακολουθείται καμία από τις δύο συνθήκες διακλάδωσης. Γι' αυτό το λόγο και οι καθυστερήσεις συνολικά είναι δύο λιγότερες.

Ανακεφαλαιώνοντας λοιπόν…

Η εξάρτηση ελέγχου καθορίζει τη σειρά εκτέλεσης μιας εντολής, λαμβάνοντας υπόψη την εντολή διακλάδωσης, έτσι ώστε η εντολή να εκτελείται μόνο όταν είναι απαραίτητο.

Οι περιορισμοί που αφορούν τις εξαρτήσεις ελέγχου είναι :

- **Μία εντολή εξαρτώμενη από μία διακλάδωση δεν πρέπει να μεταφερθεί πριν τη διακλάδωση, γιατί τότε η εκτέλεση της δεν θα ελέγχεται από αυτή.**
- **Μία εντολή που δεν έχει εξάρτηση ελέγχου από μία διακλάδωση δεν πρέπει να μεταφερθεί μετά τη διακλάδωση, γιατί τότε η εκτέλεση της θα ελέγχεται από αυτή.**

Υπάρχουν δύο ιδιότητες που παίζουν σημαντικό ρόλο στην σωστή εκτέλεση του προγράμματος. Αυτές είναι, η διαχείριση διακοπών (εξαιρέσεων) και η ροή δεδομένων οι οποίες διαφυλάσσονται από την εξάρτηση ελέγχου.

Με τον όρο διαχείριση διακοπών, συνήθως εννοούμε ότι οποιεσδήποτε αλλαγές στη σειρά εκτέλεσης των εντολών δεν πρέπει να προκαλούν την εμφάνιση νέων διακοπών στο πρόγραμμα.

Η ροή δεδομένων είναι η πραγματική ροή δεδομένων μεταξύ των εντολών.

Η ανάλυση σε επίπεδο βρόχου καθορίζει τις εξαρτήσεις μεταξύ των εντολών του κυρίου μέρους του βρόχου και όλων των επαναλήψεων του. Επίσης καθορίζει κατά πόσο τα δεδομένα των μεταγενέστερων επαναλήψεων έχουν εξάρτηση δεδομένων με τα δεδομένα που παράχθηκαν από τις αρχικές επαναλήψεις.