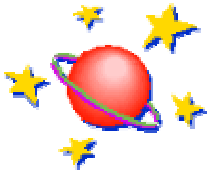

Ενότητα 7η: *Διακοπές στη Σωλήνωση*

Σκοπός **Ο σκοπός της ενότητας αυτής είναι να εξετάσει τους τύπους των διακοπών στη σωλήνωση, τις απαιτήσεις τους σε υλικό και τον τρόπο με τον οποίο αυτές αντιμετωπίζονται. Στη συνέχεια αναφερόμαστε στις διακοπές του υπολογιστή DLX και στις προκλήσεις που προκύπτουν από την αντιμετώπιση των διακοπών σε υπολογιστές που χρησιμοποιούν διαφορετικά σύνολα εντολών.**

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

<u>να αναγνωρίζετε τους τύπους των διακοπών,</u>

<u>να προσδιορίζετε το υλικό υπολογιστή που απαιτείται για την υποστήριξη των διακοπών,</u>

<u>να περιγράφετε τον τρόπο με τον οποίο σταματά και ξαναρχίζει η εκτέλεση των εντολών κατά τη διαχείριση των διακοπών,</u>

<u>να αναγνωρίζετε τις διακοπές στον υπολογιστή DLX,</u>

<u>να προσδιορίζετε τις επιπλοκές για το σύνολο εντολών.</u>

σύγχρονες /ασύγχρονες διακοπές, αναγκαστικές διακοπές, διακοπές απαιτούμενες από το χρήστη, διακοπές που απενεργοποιούνται /δεν απενεργοποιούνται από το χρήστη, διακοπές μέσα /ενδιάμεσα στις εντολές, συνέχιση /τερματισμός, επανεκκινήσιμη σωλήνωση,

διακοπές ακριβείας, προβληματικές διακοπές, διάνυσμα κατάστασης διακοπής, δέσμευση, μικροεντολή

Πώς αντιμετωπίζουμε τις διακοπές

Σε ένα σωληνωμένο υπολογιστή μια εντολή εκτελείται κατά στάδια και δεν ολοκληρώνεται παρά μόνο σε μερικούς κύκλους ρολογιού. Άλλες εντολές όμως, μπορεί να προκαλέσουν διακοπές κατά την εκτέλεση του προγράμματος και ίσως αναγκάσουν τον υπολογιστή να σταματήσει την εκτέλεση των εντολών που βρίσκονται στη σωλήνωση, πριν αυτές ολοκληρωθούν.

Στη σωλήνωση είναι πιο δύσκολο να γνωρίζουμε αν μια εντολή μπορεί να αλλάξει με ασφάλεια την κατάσταση του υπολογιστή, με αποτέλεσμα να γίνεται δύσκολη η αντιμετώπιση των διακοπών.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Μπορείτε να εξηγήσετε το λόγο για τον οποίο στη σωλήνωση είναι πιο δύσκολο να αντιμετωπίσουμε τις διακοπές;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1

Σε ένα κύκλο ρολογιού στη σωλήνωση μπορούν εκτελούνται παραπάνω από μία εντολές. Η επικάλυψη αυτή και η επίδραση που έχει η εκτέλεση μιας εντολής, πάνω σε κάποια άλλη εντολή, είναι αυτά που καθιστούν δυσκολότερη την ανίχνευση και την αντιμετώπιση των διακοπών στη σωλήνωση.

Τύποι διακοπών και απαιτήσεις

Η ορολογία που χρησιμοποιείται για την περιγραφή των καταστάσεων στις οποίες αλλάζει η κανονική σειρά εκτέλεσης των εντολών, είναι διαφορετική από υπολογιστή σε υπολογιστή. Χρησιμοποιούνται οι όροι **διακοπή**, **σφάλμα** και **εξαίρεση**. Χρησιμοποιούμε τον όρο διακοπή για την κάλυψη όλων αυτών των όρων, συμπεριλαμβανομένων των παρακάτω:

- **Αίτηση συσκευής εισόδου – εξόδου.**
- **Κλήση υπηρεσίας λειτουργικού συστήματος απ' ένα πρόγραμμα χρήστη.**
- **Παρακολούθηση (tracing) εκτέλεσης εντολής.**
- **Σημείο διακοπής (διακοπή από τον προγραμματιστή).**

- **Αριθμητική υπερχείλιση ή υποχείλιση ακεραίων.**
- **Ανωμαλία αριθμών κινητής υποδιαστολής (FP).**
- **Σφάλμα σελίδας (όχι στην κύρια μνήμη).**
- **Μη ευθυγραμμισμένη πρόσβαση στη μνήμη (αν απαιτείται ευθυγράμμιση).**
- **Παραβίαση προστασίας μνήμης.**
- **Χρήση απροσδιόριστης εντολής.**
- **Δυσλειτουργία υλικού.**
- **Πτώση τάσης**

Οι απαιτήσεις των διακοπών μπορούν να ταξινομηθούν στις επόμενες κατηγορίες:

1. **Σύγχρονες έναντι ασύγχρονων:** Αν το γεγονός συμβαίνει στο ίδιο σημείο κάθε φορά που το πρόγραμμα εκτελείται με τα ίδια δεδομένα και παραχώρηση μνήμης, τότε το γεγονός είναι **σύγχρονο**. Με εξαίρεση τις δυσλειτουργίες υλικού, τα **ασύγχρονα** συμβάντα προκαλούνται από εξωτερικές, ως προς τον επεξεργαστή και τη μνήμη, συσκευές. Συνήθως, τα ασύγχρονα συμβάντα μπορούν να αντιμετωπιστούν μετά από την ολοκλήρωση της τρέχουσας εντολής, γεγονός που καθιστά πιο εύκολο τον χειρισμό τους.

2. **Απαιτούμενες από το χρήστη έναντι αναγκαστικών:** Αν η διεργασία του χρήστη απαιτεί απευθείας παρέμβαση, τότε είναι ένα συμβάν **απαιτούμενο από το χρήστη**. Οι απαιτούμενες από το χρήστη διακοπές δεν είναι πραγματικά διακοπές γιατί είναι προβλέψιμες. Όμως τις χειριζόμαστε και αυτές σαν διακοπές, αφού χρησιμοποιούνται και για αυτές οι ίδιοι μηχανισμοί αποθήκευσης και επαναφοράς της κατάστασης που χρησιμοποιούνται για τις διακοπές. Αυτού του είδους οι διακοπές μπορούν να αντιμετωπιστούν πάντα μετά την ολοκλήρωση της εντολής. Από την άλλη μεριά οι **αναγκαστικές** διακοπές προκαλούνται από κάποιο συμβάν στο υλικό που δε βρίσκεται υπό τον έλεγχο του προγράμματος του χρήστη. Οι διακοπές αυτές είναι πιο δύσκολο να υλοποιηθούν, γιατί είναι απρόβλεπτες.

3. **Διακοπές που απενεργοποιούνται από το χρήστη έναντι διακοπών που δεν απενεργοποιούνται από το χρήστη:**
   Αν ένα συμβάν μπορεί να πάρει μάσκα ή να απενεργοποιηθεί από το χρήστη, τότε είναι maskable από τον χρήστη. Η δουλειά της μάσκας είναι να ελέγχει πότε το υλικό ανταποκρίνεται στη διακοπή ή όχι.

4. **Μέσα στις εντολές έναντι ενδιάμεσα στις εντολές:** Αν η διακοπή συμβαίνει στο ενδιάμεσο διαφορετικών εντολών τότε έχουμε διακοπή που συμβαίνει **ενδιάμεσα στις εντολές**. Αντιθέτως, αν η διακοπή εμποδίζει την ολοκλήρωση της εντολής, εμφανιζόμενη στη μέση της εκτέλεσης –όσο σύντομη και αν είναι – τότε έχουμε διακοπή που συμβαίνει **μέσα στις εντολές.** Οι διακοπές μέσα στις εντολές είναι πάντοτε σύγχρονες, καθώς η εντολή προκαλεί τη διακοπή.

5. **Συνέχιση έναντι τερματισμού:** Αν η εκτέλεση του προγράμματος σταματά πάντα μετά τη διακοπή τότε έχουμε ένα συμβάν που **τερματίζει**. Αν όμως η εκτέλεση συνεχίζεται μετά τη διακοπή τότε λέμε ότι έχουμε ένα συμβάν με **συνέχιση**.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Να συγκρίνετε τις διακοπές μέσα στις εντολές, με αυτές ενδιάμεσα στις εντολές. Ποια από τις δύο περιπτώσεις θεωρείτε πως είναι ευκολότερη στην υλοποίηση;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2

Πιο δύσκολα υλοποιούνται οι διακοπές που συμβαίνουν μέσα στις εντολές, αφού η εντολή πρέπει να σταματήσει και να ξαναρχίσει. Απ' τη στιγμή που σταματά η εκτέλεση της εντολής, χρειάζεται κάπου να αποθηκευτεί η κατάσταση του προγράμματος μέχρι εκείνη τη στιγμή, μέχρι να χειριστεί η διακοπή.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Να συγκρίνετε ένα συμβάν που τερματίζει με ένα συμβάν με συνέχιση. Ποια από τις δύο περιπτώσεις θεωρείτε πως είναι ευκολότερη στην υλοποίηση;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3

Είναι πιο εύκολο να υλοποιήσουμε διακοπές που τερματίζουν την εκτέλεση από ότι τις διακοπές που συνεχίζουν, αφού η μηχανή δε χρειάζεται να είναι σε θέση να ξαναρχίσει την εκτέλεση του ιδίου προγράμματος, μετά από το χειρισμό της διακοπής.

Το δύσκολο είναι η υλοποίηση διακοπών που συμβαίνουν μέσα στις εντολές. Η υλοποίηση τέτοιου είδους διακοπών απαιτεί να κληθεί ένα άλλο πρόγραμμα για να αποθηκεύσει την κατάσταση του εκτελούμενου προγράμματος, να διορθώσει τα αίτια της διακοπής και τέλος να αποκαταστήσει την κατάσταση του προγράμματος, πριν η εντολή που προκάλεσε τη διακοπή προσπαθήσει ξανά να εκτελεστεί. Αν μια σωλήνωση προσφέρει τη δυνατότητα σε έναν υπολογιστή να χειριστεί τη διακοπή, να αποθηκεύσει την κατάσταση και να ξαναρχίσει, χωρίς να επηρεάσει την εκτέλεση του προγράμματος, τότε η σωλήνωση ή ο υπολογιστής λέγεται ότι είναι επανεκκινήσιμη/ος. Σχεδόν όλοι οι σημερινοί υπολογιστές υποστηρίζουν αυτή τη δυνατότητα.

Πώς σταματάμε και ξαναρχίζουμε την εκτέλεση

Όπως συμβαίνει στις μη σωληνωμένες υλοποιήσεις, οι δυσκολότερες διακοπές έχουν δύο ιδιότητες:
1. Συμβαίνουν μέσα στις εντολές (δηλαδή, στη μέση της εκτέλεσης της εντολής, στα στάδια EX και MEM).
2. Πρέπει οι εντολές να μπορούν να ξαναρχίσουν.

Στη σωλήνωση του υπολογιστή DLX, για παράδειγμα, ένα σφάλμα σελίδας ιδεατής μνήμης δεν μπορεί να συμβεί πριν από κάποιο σημείο στο στάδιο MEM της εντολής. Μέχρι να γίνει αντιληπτό το σφάλμα έχει αρχίσει η εκτέλεση άλλων εντολών. Ένα σφάλμα σελίδας υλοποιείται συνήθως με την αποθήκευση της τιμής του PC της εντολής με την οποία θα ξαναρχίσει. Τότε έχουμε δύο περιπτώσεις:

- Αν η εντολή που ξαναρχίζει δεν είναι εντολή διακλάδωσης τότε θα συνεχίσουμε να καλούμε τη σειρά επόμενων εντολών και θα ξεκινήσουμε την εκτέλεση τους με τον κανονικό τρόπο.
- Αν η εντολή επανεκκίνησης είναι εντολή διακλάδωσης, τότε θα επανεκτιμήσουμε τη συνθήκη διακλάδωσης και θα ξεκινήσουμε τη ροή εντολών, είτε από το στόχο είτε από το σημείο στο οποίο η εντολή διακόπηκε.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Θυμάστε τι είναι η ιδεατή μνήμη και πότε αυτή χρησιμοποιείται; Να περιγράψετε συνοπτικά την τεχνική της ιδεατής μνήμης και τον τρόπο λειτουργίας της. Για περισσότερες λεπτομέρειες καλό θα ήταν να ανατρέξετε στις σημειώσεις του μαθήματος «Αρχιτεκτονική Υπολογιστών Ι» και συγκεκριμένα στο 4^{ο} κεφάλαιο: «Η Μνήμη των Υπολογιστικών Συστημάτων».

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

Στην περίπτωση που ένα πρόγραμμα είναι μεγαλύτερο από την κύρια μνήμη ή από το τμήμα της κύριας μνήμης που διατίθεται γι' αυτό, χρησιμοποιείται η τεχνική της ιδεατής μνήμης. Η βασική ιδέα πάνω στην οποία βασίζεται η ιδεατή μνήμη είναι η χρήση ενός τμήματος του σκληρού δίσκου σαν μια λογική επέκταση της κύριας μνήμης.
Το λειτουργικό σύστημα σε συνεργασία με το κατάλληλο υλικό κρατάει στην κύρια μνήμη εκείνα τα τμήματα του προγράμματος που χρησιμοποιούνται και τα υπόλοιπα τα διατηρεί στο δίσκο, τα οποία μεταφέρονται στην κύρια μνήμη όταν ζητηθούν. Τα προγράμματα χρησιμοποιούν ιδεατές διευθύνσεις και όταν αυτά εκτελούνται, το λειτουργικό σύστημα σε συνεργασία με το υλικό αναλαμβάνει να μετατρέψει τις ιδεατές διευθύνσεις σε πραγματικές.

Όταν συμβεί μια διακοπή, μπορούμε να εκτελεστούν τα ακόλουθα για να σώσουμε με ασφάλεια την κατάσταση της σωλήνωσης:

1. Να αναγκαστεί μια εντολή παγίδας να μπει στη σωλήνωση στο επόμενο στάδιο IF.
2. Μέχρι να αρχίσει η εκτέλεση της παγίδας, πρέπει να καταργηθούν όλες οι εγγραφές για τη λανθάνουσα εντολή και για όλες τις εντολές που ακολουθούν τη σωλήνωση. Αυτό μπορεί να γίνει θέτοντας μηδενικά στους καταχωρητές της σωλήνωσης για όλες τις εντολές της σωλήνωσης, ξεκινώντας από αυτή που παράγει τη διακοπή, αλλά όχι για τις προηγούμενες. Αυτό εμποδίζει κάθε αλλαγή κατάστασης για εντολές που δε θα ολοκληρωθούν πριν η διακοπή αντιμετωπιστεί.
3. Αφού η ρουτίνα εξυπηρέτησης της διακοπής στο λειτουργικό σύστημα αναλάβει τον έλεγχο, αποθηκεύεται αμέσως στον PC η διεύθυνση της λανθάνουσας εντολής. Η τιμή αυτή θα χρησιμοποιηθεί αργότερα για την επιστροφή από τη διακοπή.

Αφού η διακοπή έχει αντιμετωπισθεί, ειδικές εντολές επιστρέφουν τον υπολογιστή από τη διακοπή, ξαναφορτώνοντας τους PC's και ξαναρχίζοντας την ακολουθία εντολών.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Θυμάστε ποια είναι η εντολή ελέγχου του υπολογιστή DLX η οποία προκαλεί την επιστροφή στον κώδικα μετά από μία διακοπή; Για περισσότερες λεπτομέρειες καλό θα ήταν να ανατρέξετε στις σημειώσεις του μαθήματος «Αρχιτεκτονική Υπολογιστών Ι» και συγκεκριμένα στο 2^{ο} κεφάλαιο: «Αρχιτεκτονικές Συνόλου Εντολών».

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Η εντολή ελέγχου που προκαλεί την επιστροφή στον κώδικα μετά από μία διακοπή είναι η RFE.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 6

Έστω ότι έχουμε μια σχισμή διακλάδωσης και ότι η διακοπή συμβαίνει στην εντολή που βρίσκεται στη σχισμή. Σε αυτή την περίπτωση δεν είναι πλέον δυνατό να ξαναδημιουργήσουμε την κατάσταση του υπολογιστή με τη διεύθυνση από ένα μόνο PC γιατί οι εντολές στη σωλήνωση μπορεί να μην είναι στην κανονική σειρά τους, λόγω της σχισμής καθυστέρησης.
Μπορείτε να εξηγήσετε τι πρέπει να γίνει σε αυτή την περίπτωση;

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 6

Σε αυτή την περίπτωση πρέπει να σώσουμε όλες τις διευθύνσεις σε τόσους PC's όσο και το μήκος της σχισμής καθυστέρησης συν ένα, ο οποίος θα περιέχει τη διεύθυνση της λανθάνουσας εντολής. Επομένως οι τιμές θα ανακληθούν από τους PC's και θα χρησιμοποιηθούν αργότερα, μετά την επιστροφή από τη διακοπή.

Αν η σωλήνωση μπορεί να σταματήσει, έτσι ώστε οι εντολές που βρίσκονται ακριβώς πριν από τη λανθάνουσα εντολή να έχουν ολοκληρωθεί και αυτές που βρίσκονται μετά να μπορούν να ξεκινήσουν από το σημείο διακοπής, λέμε ότι η εντολή έχει **διακοπές ακριβείας.**

Στις διακοπές των εντολών κινητής υποδιαστολής, η λανθάνουσα εντολή σε μερικές μηχανές, γράφει το αποτέλεσμα της πριν αντιμετωπισθεί η διακοπή. Σε αυτές τις περιπτώσεις, είναι δουλειά του υλικού να είναι προετοιμασμένο να ανακαλέσει τους πηγαίους τελεστέους, ακόμα και αν ο προορισμός είναι ίδιος με έναν από αυτούς. Επειδή όμως οι λειτουργίες κινητής υποδιαστολής μπορούν να εκτελούνται για πολλούς κύκλους ρολογιού, είναι πολύ πιθανό κάποια άλλη εντολή να έχει εκτελέσει εγγραφή των πηγαίων τελεστέων. Για να μπορέσουν να αντιμετωπίσουν αυτό το φαινόμενο, πολλοί υπολογιστές υψηλών επιδόσεων έχουν εισάγει δύο ρυθμούς λειτουργίας:

1. ρυθμός λειτουργίας του οποίου οι εντολές να υποστηρίζουν διακοπές ακριβείας,
2. ρυθμός λειτουργίας (γρήγορος ή ρυθμός επιδόσεων) του οποίου οι εντολές δεν υποστηρίζουν διακοπές ακριβείας.

Ο ρυθμός λειτουργίας με διακοπές ακρίβειας είναι πιο αργός επειδή επιτρέπει λιγότερες επικαλύψεις μεταξύ εντολών κινητής υποδιαστολής.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 7

Θυμάστε ποιες είναι οι εντολές κινητής υποδιαστολής του υπολογιστή DLX και τι καθορίζουν; Να αναφέρετε ενδεικτικά κάποιες από αυτές. Για περισσότερες λεπτομέρειες καλό θα ήταν να ανατρέξετε στις σημειώσεις του μαθήματος «Αρχιτεκτονική Υπολογιστών Ι» και συγκεκριμένα στο 2^{ο} κεφάλαιο: «Αρχιτεκτονικές Συνόλου Εντολών».

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 7

Οι εντολές κινητής υποδιαστολής χρησιμοποιούν τους καταχωρητές κινητής υποδιαστολής και καθορίζουν το αν μία πράξη που θα πραγματοποιηθεί είναι απλής ή διπλής ακρίβειας. Οι εντολές αυτές είναι:

- **MOVD, MOVF: μετακινούν το περιεχόμενο ενός καταχωρητή κινητής υποδιαστολής απλής ή διπλής ακρίβειας αντίστοιχα σε έναν καταχωρητή του ίδιου τύπου.**
- **MOVFP2I, MOVI2FP: μετακινούν δεδομένα ανάμεσα σε έναν καταχωρητή κινητής υποδιαστολής απλής ακρίβειας και σε έναν καταχωρητή ακεραίου.**
- **ADDD, ADDF, SUBD, SUBF, MULTD, MULTF, DIVD, DIVF: προσθέτουν, αφαιρούν, πολλαπλασιάζουν και διαιρούν αριθμούς κινητής υποδιαστολής απλής και διπλής ακρίβειας αντίστοιχα.**

Διακοπές στον υπολογιστή DLX

Το σχήμα 1.7.1 παρουσιάζει τα στάδια σωλήνωσης του υπολογιστή DLX και ποιες 'προβληματικές' διακοπές συμβαίνουν σε κάθε στάδιο. Εξαιτίας της σωλήνωσης στον ίδιο κύκλο ρολογιού μπορούν να συμβούν πολλαπλές διακοπές, αφού υπάρχουν πολλές εντολές σε εκτέλεση.

Στάδιο σωλήνωσης	**Προβληματικές διακοπές που συμβαίνουν**
IF	Σφάλμα σελίδας κατά την ανάκληση εντολής, μη ευθυγραμμισμένη πρόσβαση στη μνήμη, παραβίαση προστασίας μνήμης
ID	Ακαθόριστο ή παράνομο τμήμα κωδικού λειτουργίας εντολής (opcode)
EX	Αριθμητική διακοπή
MEM	Σφάλμα σελίδας κατά την ανάκληση δεδομένων, μη ευθυγραμμισμένη πρόσβαση στη μνήμη, παραβίαση προστασίας μνήμης
WB	Καμία

Σχήμα 1.7.1 - Οι διακοπές που μπορούν να συμβούν στη σωλήνωση του υπολογιστή DLX.

ΠΑΡΑΔΕΙΓΜΑ

Έστω η παρακάτω ακολουθία εντολών:

LW	IF	ID	EX	MEM	WB	
ADD		IF	ID	EX	MEM	WB

Οι δύο αυτές εντολές μπορούν να προκαλέσουν ένα σφάλμα σελίδας δεδομένων και ταυτόχρονα μια αριθμητική διακοπή, αφού η εντολή LW βρίσκεται στο στάδιο MEM, ενώ η εντολή ADD στο στάδιο EX. Αυτή η περίπτωση αντιμετωπίζεται με το χειρισμό μόνο του σφάλματος σελίδας δεδομένων και με την επανεκκίνηση, στη συνέχεια της εκτέλεσης. Βέβαια, η δεύτερη διακοπή θα ξανασυμβεί, όχι όμως και η πρώτη (αν το λογισμικό είναι σωστό) και τότε μπορεί να αντιμετωπιστεί ανεξάρτητα.

Στην πραγματικότητα, οι διακοπές μπορεί να συμβούν εκτός σειράς των εντολών του προγράμματος. Αυτό μπορεί να συμβεί όταν μια εντολή προκαλέσει μια διακοπή προτού μια εντολή που βρίσκεται πριν από αυτή, προκαλέσει τη δικιά της διακοπή, γιατί αυτό θα οδηγούσε τις διακοπές να προκύπτουν εκτός της μη σωληνωμένης σειράς.

ΠΑΡΑΔΕΙΓΜΑ

Έστω η ίδια ακολουθία εντολών:

LW	IF	ID	EX	MEM	WB	
ADD		IF	ID	EX	MEM	WB

Η εντολή LW μπορεί να πάρει ένα σφάλμα σελίδας δεδομένων που φαίνεται όταν η εντολή φτάσει στο στάδιο MEM. Επίσης, η εντολή ADD προκαλεί ένα σφάλμα σελίδας εντολών, το οποίο φαίνεται στο στάδιο IF της εντολής ADD. Έτσι το σφάλμα σελίδας εντολών θα συμβεί πρώτα, παρά το γεγονός ότι προκαλείται από μια μεταγενέστερη εντολή.

Στην περίπτωση που μια μεταγενέστερη διακοπή προκαλεί πρώτη διακοπή όπως στο προηγούμενο παράδειγμα, το υλικό τοποθετεί όλες τις διακοπές που προκαλεί η δοθείσα εντολή σε ένα **διάνυσμα κατάστασης διακοπής**, που σχετίζεται με αυτή την εντολή. Το διάνυσμα κατάστασης διακοπής μεταφέρεται μαζί, καθώς η εντολή προχωράει στη σωλήνωση. Όταν ένας δείκτης διακοπής τεθεί στο διάνυσμα κατάστασης διακοπής, κάθε σήμα ελέγχου, που μπορεί να προκαλέσει την εγγραφή μιας τιμής δεδομένων, απενεργοποιείται.

Επειδή μια εντολή αποθήκευσης μπορεί να προκαλέσει διακοπή κατά τη διάρκεια του σταδίου MEM, το υλικό πρέπει να είναι προετοιμασμένο, έτσι ώστε να αποτρέψει την ολοκλήρωση της αποθήκευσης, αν αυτή προκαλέσει διακοπή. Τη στιγμή που μία εντολή εισέρχεται στο στάδιο WB ελέγχεται το διάνυσμα κατάστασης διακοπής. Αν περιέχει διακοπές, τότε αυτές αντιμετωπίζονται με τη σειρά την οποία θα είχαν αν συνέβαιναν χρονικά σε έναν υπολογιστή με σωλήνωση. Δηλαδή η διακοπή που αντιστοιχεί στην προγενέστερη εντολή αντιμετωπίζεται πρώτη.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 8

Έστω I και I+1 δύο διαδοχικές εντολές και προκαλείται διακοπή στην εντολή I+1 πριν από τη διακοπή στην εντολή I.

- **Μπορείτε να εξηγήσετε τι θα συμβεί σε αυτή την περίπτωση;**
- **Αν η I είναι μια εντολή LW και η I+1 είναι μια εντολή SUB, θεωρείται ότι υπάρχει πρόβλημα με τις εγγραφές στο αρχείο καταχωρητών; Να αιτιολογήσετε την απάντησή σας.**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 8

- **Όλες οι διακοπές θα εμφανιστούν στην εντολή Ι πριν εμφανιστεί καμία στην εντολή Ι+1. Οι διακοπές της εντολής Ι+1 θα τοποθετηθούν από το υλικό σε ένα διάνυσμα κατάστασης διακοπής, που σχετίζεται με αυτή την εντολή. Το διάνυσμα κατάστασης διακοπής μεταφέρεται μαζί, καθώς η εντολή προχωράει στη σωλήνωση.**
- **Κάθε ενέργεια που είχε γίνει σε προγενέστερα στάδια σωλήνωσης για λογαριασμό της εντολής Ι+1 μπορεί να είναι άκυρη, αλλά εφόσον οι εγγραφές στο αρχείο καταχωρητών και τη μνήμη είχαν απενεργοποιηθεί, η κατάσταση δε θα έχει αλλάξει και δεν προκύπτει κανένα πρόβλημα.**

Επιπλοκές συνόλου εντολών

Η εγγύηση ότι μια εντολή θα ολοκληρωθεί ονομάζεται **δέσμευση**.
Στον υπολογιστή DLX, όλες οι εντολές στη σωλήνωση είναι δεσμευμένες όταν φτάσουν στο τέλος του σταδίου MEM (ή στην αρχή του σταδίου WB) και καμία εντολή δεν ενημερώνει την κατάσταση πριν από αυτό το στάδιο. Έτσι οι διακοπές ακριβείας είναι άμεσες.

Σε μερικές μηχανές, υπάρχουν εντολές που αλλάζουν την κατάσταση στη μέση της εκτέλεσης, προτού η εντολή και οι προηγούμενές της έχουν εγγυημένη την ολοκλήρωσή τους.

ΠΑΡΑΔΕΙΓΜΑ

Στον υπολογιστή VAX, οι τρόποι διευθυνσιοδότησης με αυτοαύξηση προκαλούν την ενημέρωση των καταχωρητών στη μέση της εκτέλεσης μιας εντολής. Σε μία τέτοια περίπτωση, αν η εντολή εγκαταλειφθεί εξαιτίας μίας διακοπής, θα έχει μεταβάλλει την κατάσταση του υπολογιστή. Παρότι γνωρίζουμε ποια εντολή προκάλεσε τη διακοπή, χωρίς πρόσθετη υποστήριξη από το υλικό, η διακοπή θα είναι ανακριβής, αφού η εντολή θα είναι μισοτελειωμένη. Η επανεκκίνηση της ροής εντολών μετά από μια τέτοια ανακριβή διακοπή είναι δύσκολη. Θα μπορούσαμε βέβαια να αποφύγουμε την ενημέρωση της κατάστασης πριν δεσμευτεί η εντολή. Όμως αυτό μπορεί να είναι δύσκολο ή δαπανηρό, καθώς είναι δυνατό να υπάρχουν εξαρτήσεις από την ενημερωμένη κατάσταση.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 9

- **Θυμάστε τους τρόπους διευθυνσιοδότησης αυτοαύξησης του υπολογιστή VAX; Να αναφέρετε τους δύο αυτούς τρόπους, τον τρόπο με τον οποίο συμβολίζονται, καθώς επίσης και τη σημασία τους.**
- **Να περιγράψετε τι θα συμβεί αν εκτελεστεί η παρακάτω εντολή του υπολογιστή VAX: CLRL (R9)+**
- **Για περισσότερες λεπτομέρειες καλό θα ήταν να ανατρέξετε στις σημειώσεις του μαθήματος «Αρχιτεκτονική Υπολογιστών Ι» και συγκεκριμένα στο 2ο κεφάλαιο: «Αρχιτεκτονικές Συνόλου Εντολών».**

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 9

- **Αυτοαύξησης: (Rn)+**
  Η διεύθυνση για την εκτέλεση της εντολής είναι το περιεχόμενο του καταχωρητή Rn (Διεύθυνση = [Rn]). Γίνεται αύξηση του περιεχομένου του Rn.
  Έμμεσος αυτοαύξησης: @(Rn)
  Η διεύθυνση για την εκτέλεση της εντολής είναι το περιεχόμενο του περιεχομένου του καταχωρητή Rn (Διεύθυνση = [[Rn]]). Γίνεται αύξηση του περιεχομένου του Rn.
- **Με την εντολή αυτή μηδενίζεται το περιεχόμενο που βρίσκεται στη θέση μνήμης, η διεύθυνση της οποίας είναι καταχωρημένη στον καταχωρητή R9. Στη συνέχεια το περιεχόμενο του R9 αυξάνεται κατά 4 γιατί ο τελεστέος έχει μέγεθος 4 bytes (λόγω του L).**

Για να καταφέρουν οι περισσότερες μηχανές με εντολές που αλλάζουν την κατάσταση στη μέση της εκτέλεσης, να διατηρήσουν ένα μοντέλο διακοπών ακριβείας, έχουν τη δυνατότητα να ανακαλούν όποιες αλλαγές κατάστασης έγιναν προτού δεσμευθεί η εντολή. Αν συμβεί μια διακοπή, η μηχανή χρησιμοποιεί την ικανότητα να επαναφέρει την κατάστασή της στην τιμή που είχε πριν ξεκινήσει η διακοπείσα εντολή.

Υπάρχουν δυσκολίες οι οποίες προκύπτουν από εντολές που ενημερώνουν την κατάσταση της μνήμης κατά τη διάρκεια της εκτέλεσης.
Για να μπορέσει να γίνει δυνατή η διακοπή και η επανεκκίνηση αυτών των εντολών, οι εντολές χρησιμοποιούν εξ' ορισμού τους καταχωρητές γενικής χρήσης ως καταχωρητές εργασίας. Με τον τρόπο αυτό, η κατάσταση των μερικώς ολοκληρωμένων εντολών, βρίσκεται πάντα στους καταχωρητές, οι οποίοι αποθηκεύονται σε μία διακοπή και επαναφέρονται μέσα από αυτή, επιτρέποντας στην εντολή να συνεχίσει.

ΠΑΡΑΔΕΙΓΜΑ

Στον υπολογιστή VAX μπορεί να προκύψει μια ανάλογη δυσκολία στις εντολές αντιγραφής συμβολοσειράς. Για να αποφευχθούν τέτοιου είδους δυσκολίες, ένα επιπρόσθετο bit κατάστασης καταγράφει το αν μια εντολή έχει αρχίσει να ενημερώνει τη μνήμη. Έτσι, όταν γίνει επανεκκίνηση της σωλήνωσης, ο υπολογιστής ανάλογα με αυτό το bit θα επανεκκινήσει την εντολή από την αρχή ή από τη μέση της εντολής.

Ένα άλλο είδος δυσκολιών εμφανίζεται από τα ψηφία ή σημαίες κατάστασης, τα οποία μπορούν να δημιουργούν επιπρόσθετους κινδύνους σωλήνωσης ή να απαιτήσουν επιπλέον υλικό για την αποθήκευση και την επαναφορά.
Ένα τέτοιο παράδειγμα είναι οι καταχωρητές κατάστασης. Υπάρχουν υπολογιστές που καθορίζουν τους καταχωρητές κατάστασης, ως μέρος της εντολής. Αυτή η προσέγγιση έχει πλεονεκτήματα, αφού οι καταχωρητές κατάστασης διαχωρίζουν την αποτίμηση της συνθήκης από την πραγματική διακλάδωση. Όμως έμμεσα σύνολα καταχωρητών κατάστασης μπορούν να προκαλέσουν δυσκολίες στον προγραμματισμό κάποιας καθυστέρησης σωλήνωσης ανάμεσα στον καθορισμό του καταχωρητή κατάστασης και της διακλάδωσης, καθώς οι περισσότερες εντολές τροποποιούν τον καταχωρητή κατάστασης και δεν μπορούν να χρησιμοποιηθούν στις σχισμές καθυστέρησης μεταξύ της αποτίμησης της συνθήκης και της διακλάδωσης. Επίσης, υπάρχουν υπολογιστές που υποστηρίζουν καταχωρητές κατάστασης, στους οποίους ο επεξεργαστής πρέπει να αποφασίσει πότε η συνθήκη διακλάδωσης ικανοποιείται. Αυτό περιέχει την αναζήτηση του πότε ο καταχωρητής κατάστασης καθορίστηκε για τελευταία φορά, πριν τη διακλάδωση.

Είναι απαραίτητο ο έλεγχος της σωλήνωσης να εντοπίσει την τελευταία εντολή που θέτει τον καταχωρητή κατάστασης για να γνωρίζει πότε η συνθήκη διακλάδωσης αποφασίζεται. Έτσι, ο καταχωρητής κατάστασης πρέπει να αντιμετωπίζεται σαν τελεστέος που απαιτεί ανίχνευση κινδύνων για κινδύνους RAW με διακλαδώσεις.

Τέλος δεν πρέπει να παραλείψουμε να αναφερθούμε στις εντολές πολλαπλών κύκλων. Έστω ότι έχουμε την ακόλουθη ροή εντολών του υπολογιστή VAX:

```
MOVL     R1 , R2
ADDL3    42 (R1) , 56 (R1) + , @ (R1)
SUBL2    R2 , R3
MOVC3    @ (R1) [R2] , 74 (R2) , R3
```

Οι κίνδυνοι δεδομένων είναι πολύ περίπλοκοι και συμβαίνουν τόσο μεταξύ όσο και μέσα στις εντολές.
Μια απλή, αλλά όχι καλή και αποδεκτή λύση, είναι να κάνουμε όλες τις εντολές να εκτελούνται για τον ίδιο αριθμό κύκλων ρολογιού. Όμως αυτό εισάγει ένα τεράστιο αριθμό κινδύνων και συνθηκών παράκαμψης και δημιουργεί μια απείρως μακριά σωλήνωση.
Μια άλλη καλύτερη λύση, είναι να σωληνωθεί η εκτέλεση των **μικροεντολών**. Με τον όρο μικροεντολή εννοούμε μια απλή εντολή που χρησιμοποιείται σε ακολουθίες για την υλοποίηση ενός πολύπλοκου συνόλου εντολών. Την παραπάνω λύση, πρότειναν οι σχεδιαστές του VAX 8800.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ 10

Να συγκρίνετε τη χρήση μικροεντολών στο σωληνωμένο υπολογιστή VAX όταν το προς εκτέλεση σύνολο εντολών είναι πολύπλοκο, με την τεχνική της εκτέλεσης των εντολών στον ίδιο αριθμό κύκλων ρολογιού, για την αντιμετώπιση κινδύνων δεδομένων. Ποια θεωρείτε πως είναι η καλύτερη μέθοδος; Να αιτιολογήσετε την απάντησή σας.

ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 10

Η λύση με τις μικροεντολές στον υπολογιστή VAX είναι καλύτερη από αυτή με την εκτέλεση των εντολών στον ίδιο αριθμό κύκλων ρολογιού, στην αντιμετώπιση κινδύνων δεδομένων. Αυτό συμβαίνει επειδή οι μικροεντολές είναι απλές εντολές με αποτέλεσμα ο έλεγχος της σωλήνωσης να γίνεται πιο εύκολα. Σε αντίθεση, η λύση με την οποία όλες οι εντολές εκτελούνται για τον ίδιο αριθμό κύκλων ρολογιού εισάγει ένα τεράστιο αριθμό κινδύνων και συνθηκών παράκαμψης και συγχρόνως δημιουργεί μια απείρως μακριά σωλήνωση.

Ανακεφαλαιώνοντας λοιπόν …

Οι διακοπές στη σωλήνωση παίζουν βασικό ρόλο αφού η σειρά εκτέλεσης των εντολών αλλάζει κατά απροσδιόριστο τρόπο. Είναι εξαιρετικά σημαντικό να μπορέσουμε να αντιμετωπίσουμε τις διακοπές.

Οι απαιτήσεις των διακοπών ταξινομούνται στις επόμενες κατηγορίες: σύγχρονες-ασύγχρονες, απαιτούμενες από το χρήστη-αναγκαστικές, διακοπές που απενεργοποιούνται από το χρήστη -διακοπές που δεν απενεργοποιούνται από το χρήστη, διακοπές που συμβαίνουν μέσα στις εντολές- διακοπές που συμβαίνουν μεταξύ των εντολών, διακοπές που τερματίζουν την εκτέλεση του προγράμματος, διακοπές που μετά από αυτές συνεχίζει η εκτέλεση του προγράμματος.

Η υλοποίηση των διακοπών που συμβαίνουν μέσα στις εντολές απαιτεί να κληθεί ένα άλλο πρόγραμμα για να αποθηκεύσει την κατάσταση του εκτελούμενου προγράμματος, να διορθώσει τα αίτια της διακοπής και τέλος να αποκαταστήσει την κατάσταση του προγράμματος, πριν η εντολή που προκάλεσε τη διακοπή προσπαθήσει ξανά να εκτελεστεί.

Οι διακοπές στον υπολογιστή DLX αντιμετωπίζονται με την τοποθέτηση από το υλικό όλων των διακοπών που προκαλεί η δοθείσα εντολή σε ένα διάνυσμα κατάστασης διακοπής, που σχετίζεται με αυτή την εντολή. Το διάνυσμα κατάστασης διακοπής μεταφέρεται μαζί, καθώς η εντολή προχωράει στη σωλήνωση.

Στο σύνολο των εντολών του σωληνωμένου υπολογιστή VAX δημιουργούνται διαφόρων ειδών επιπλοκές.