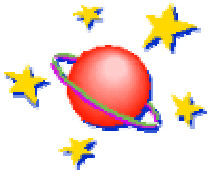


# Ενότητα 4η: *Δυναμικός προγραμματισμός – Η τεχνική του πίνακα αποτελεσμάτων*

**Σκοπός** **Ο σκοπός της ενότητας αυτής είναι να παρουσιάσει την έννοια του δυναμικού προγραμματισμού και την υλοποίησή του, η οποία στηρίζεται στην τεχνική εκτέλεσης των εντολών εκτός σειράς. Για την εκτέλεση των εντολών εκτός σειράς προγράμματος, χρησιμοποιείται η τεχνική του πίνακα αποτελεσμάτων.**

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

**<u>να περιγράφετε την έννοια του δυναμικού προγραμματισμού,</u>**

**<u>να εφαρμόζετε την τεχνική εκτέλεσης εντολών εκτός σειράς για την υλοποίηση του δυναμικού προγραμματισμού,</u>**

**<u>να χρησιμοποιείτε την τεχνική του πίνακα αποτελεσμάτων (SCOREBOARDING).</u>**

**δυναμικός προγραμματισμός, διευθέτηση, πίνακας αποτελεσμάτων (SCOREBOARDING), εντολή προς ολοκλήρωση**

# Η ιδέα του δυναμικού προγραμματισμού

**Στο δυναμικό προγραμματισμό το υλικό του υπολογιστή κατά το χρόνο της εκτέλεσης ρυθμίζει εκ νέου την εκτέλεση των εντολών, ώστε να μειώσει τις καθυστερήσεις στη σωλήνωση.**

Πλεονεκτήματα δυναμικού προγραμματισμού **:**

- Καθιστά ικανή την αντιμετώπιση ορισμένων περιπτώσεων, όπου οι εξαρτήσεις είναι άγνωστες την ώρα της μεταγλώττισης, διευκολύνοντας έτσι το μεταγλωττιστή.
- Επιτρέπει στον κώδικα, ο οποίος έχει μεταγλωττιστεί σε μία σωλήνωση, να τρέχει ικανοποιητικά και σε διαφορετικές σωληνώσεις.

Το βασικό μειονέκτημα του δυναμικού προγραμματισμού είναι ότι αυξάνεται σημαντικά η πολυπλοκότητα του υλικού.

Ένας επεξεργαστής δυναμικά προγραμματισμένος δεν μπορεί να απομακρύνει τις εξαρτήσεις δεδομένων μεταξύ των εντολών. Για να αποφύγει, λοιπόν, τις καθυστερήσεις που θα προκύψουν από τις εξαρτήσεις δεδομένων εκτελεί τις εντολές εκτός σειράς.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

**Στην περίπτωση του δυναμικού προγραμματισμού, για την αποφυγή των εξαρτήσεων, ο επεξεργαστής εκτελεί τις εντολές εκτός σειράς. Μπορείτε να εξηγήσετε τι συμβαίνει στην περίπτωση του στατικού προγραμματισμού;**

## ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1

**Ένας επεξεργαστής που χρησιμοποιεί στατικό προγραμματισμό, προσπαθεί να ελαχιστοποιήσει τις καθυστερήσεις απομακρύνοντας τις εξαρτήσεις μεταξύ των εντολών, ώστε να μην οδηγούν σε κινδύνους.**

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

**Μπορείτε να αναφέρετε ποια είναι τα πλεονεκτήματα του δυναμικού προγραμματισμού; Να επιλέξετε μία ή περισσότερες απαντήσεις από τις παρακάτω:**
**α. Επιτρέπει την ανίχνευση εξαρτήσεων που είναι άγνωστες στο μεταγλωττιστή.**
**β. Επιτρέπει στο μεταγλωττιστή να έχει πολύπλοκο σχεδιασμό.**
**γ. Επιτρέπει κώδικα που έχει μεταγλωττιστεί σε μια σωλήνωση να εκτελεστεί ικανοποιητικά και σε διαφορετική σωλήνωση (μεταφερσιμότητα).**
**δ. Μειώνει τις καθυστερήσεις χωρίζοντας τις εξαρτώμενες εντολές ώστε να μην οδηγούν σε κινδύνους.**

**Να αιτιολογήσετε την απάντησή σας και να τη συγκρίνετε με την παράγραφο που αναφέρεται στα πλεονεκτήματα του δυναμικού προγραμματισμού.**

# Η τεχνική εκτέλεσης εντολών εκτός σειράς

Ένας σημαντικός περιορισμός στις μέχρι τώρα τεχνικές σωλήνωσης είναι ότι όλες χρησιμοποιούν την κατά σειρά εκτέλεση των εντολών. Αν δηλαδή, μια εντολή καθυστερεί, τότε καμιά επόμενη εντολή δεν μπορεί να προχωρήσει. Έτσι, αν υπάρχει εξάρτηση μεταξύ δύο κοντινών εντολών τότε θα προκληθεί καθυστέρηση και αν υπάρχουν πολλαπλές λειτουργικές μονάδες, αυτές θα παραμείνουν ανενεργές.
Εάν η εντολή j εξαρτάται από μία μακρινή εντολή i, τότε όλες οι εντολές μετά την j πρέπει να καθυστερήσουν μέχρι να τελειώσει η i και να εκτελεστεί η j.

## Παράδειγμα:

**Θεωρούμε το τμήμα κώδικα :**

```
DIVD   F0 , F2 , F4
ADDD   F10 , F0 , F8
SUBD   F12 , F8 ,  F14
```

## Απάντηση:

Η εντολή SUBD δεν μπορεί να εκτελεστεί επειδή η εξάρτηση της εντολής ADDD από την εντολή DIVD προκαλεί καθυστέρηση στη σωλήνωση, παρόλο που η εντολή SUBD δεν έχει εξάρτηση δεδομένων από καμία προηγούμενη εντολή. Αυτός ο περιορισμός κατά την εκτέλεση των εντολών μπορεί να αντιμετωπιστεί με την τεχνική εκτέλεση εντολών εκτός σειράς.

Γνωρίζουμε ότι στην DLX σωλήνωση, τόσο οι κίνδυνοι δεδομένων όσο και οι κατασκευαστικοί ανιχνεύονται στο στάδιο ID. Αν δεν εντοπιστούν κίνδυνοι, τότε μια εντολή μπορεί να διευθετηθεί και να εκτελεστεί κανονικά.

**Για να εκτελέσουμε μία ανεξάρτητη εντολή όπως η SUBD, εκτός σειράς, πρέπει να διασπάσουμε το στάδιο ID σε δύο μέρη :**
**Διευθέτηση** - Αποκωδικοποίηση εντολών, έλεγχος κατασκευαστικών κινδύνων.
**Διάβασμα τελεστέων** – Αναμονή μέχρι να μειωθούν οι κίνδυνοι δεδομένων, διάβασμα τελεστέων.

Όταν γίνεται εκτέλεση εκτός σειράς, όλες οι εντολές περνούν από το στάδιο διευθέτησης με τη σειρά. Στο δεύτερο στάδιο του ID μπορεί να καθυστερήσουν ή να παρακάμψουν η μία την άλλη, ανάλογα με το εάν είναι διαθέσιμοι οι τελεστέοι τους, εισάγοντας έτσι εκτέλεση εκτός σειράς.

Η εκτέλεση μιας εντολής εκτός σειράς σημαίνει ότι η εντολή θα αποπερατωθεί εκτός σειράς. Ανάλογα με τη λειτουργία που πρόκειται να γίνει, η εκτέλεση μιας εντολής διαρκεί ορισμένους κύκλους ρολογιού. Έτσι, πρέπει να διακρίνουμε πότε μια εντολή ξεκινάει την εκτέλεσή της και πότε ολοκληρώνει την εκτέλεσή της. Μεταξύ αυτών των δύο χρονικών σημείων η εντολή βρίσκεται σε εκτέλεση. Έτσι λοιπόν, πολλές εντολές μπορούν να εκτελούνται την ίδια χρονική στιγμή.

Επιπλέον, για να μελετήσουμε καλύτερα τις πιο προχωρημένες τεχνικές σωλήνωσης, θα αλλάξουμε το σχεδιασμό της λειτουργικής μονάδας μεταβάλλοντας τον αριθμό των μονάδων, την καθυστέρηση των λειτουργιών και την σωλήνωση των λειτουργικών μονάδων.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

**Μπορείτε να αναγνωρίσετε το είδος των κινδύνων που μπορεί να προκύψουν από την εκτέλεση των εντολών εκτός σειράς και δεν υπήρχαν κατά την εκτέλεση των εντολών σε σειρά; Οι κίνδυνοι αυτοί εμφανίζονταν στις σωληνώσεις κινητής υποδιαστολής ή στις ακέραιες σωληνώσεις;**

## ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3

**Κατά την εκτέλεση των εντολών εκτός σειράς, ίσως προκύψουν WAR και WAW κίνδυνοι, οι οποίοι δεν υπήρχαν στις σωληνώσεις κινητής υποδιαστολής ή στις ακέραιες σωληνώσεις.**

## Παράδειγμα:

**Θεωρούμε το τμήμα κώδικα της δραστηριότητας 4, όπου ο καταχωρητής προορισμού της εντολής SUBD από F8 έχει γίνει F10:**

```
DIVD  F0 , F2 , F4
ADDD  F10 , F0 , F8
SUBD  F10 , F8 , F14
```

## Απάντηση:

Ο καταχωρητής F10 είναι ο προορισμός τόσο της ADDD όσο και της SUBD, έτσι μπορεί να προκύψει WAW κίνδυνος. Και στα δύο προηγούμενα παραδείγματα οι κίνδυνοι μπορούν να αποφευχθούν καθυστερώντας τη διευθέτηση της εντολής SUBD, χρησιμοποιώντας την τεχνική του πίνακα αποτελεσμάτων που θα γνωρίσουμε παρακάτω.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

**Έστω το τμήμα κώδικα:**

**DIVD F0 , F2 , F4**
**ADDD F10 , F0 , F8**
**SUBD F8 , F8 , F14**

**Μπορείτε να περιγράψετε την εκτέλεση των εντολών του κώδικα και να προσδιορίσετε τους κινδύνους που προκύπτουν;**

## ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

**Παρατηρούμε ότι υπάρχει αντι-εξάρτηση μεταξύ των εντολών SUBD και ADDD. Πράγματι, η εντολή SUBD εκτελεί εγγραφή στον καταχωρητή F8 από τον οποίο διαβάζει η εντολή ADDD που εκτελείται πρώτη. Εάν όμως η σωλήνωση εκτελέσει την εντολή SUBD πριν την εντολή ADDD, τότε θα παραβιαστεί η αντι-εξάρτηση και θα έχουμε λανθασμένη εκτέλεση, αφού η εντολή SUBD θα έχει γράψει στον καταχωρητή F8 πριν η εντολή ADDD διαβάσει από αυτόν. Σε αυτή την περίπτωση προκαλείται WAR κίνδυνος. Επιπλέον, για να αποφύγουμε την παραβίαση των εξαρτήσεων εξόδου, πρέπει να ανιχνεύσουμε τους WAW κινδύνους.**

# Η τεχνική του πίνακα αποτελεσμάτων SCOREBOARDING

Η καταγραφή αποτελεσμάτων σε πίνακα, είναι μία τεχνική που επιτρέπει στις εντολές να εκτελούνται εκτός σειράς όταν υπάρχουν επαρκείς πόροι και δεν υπάρχουν εξαρτήσεις δεδομένων. Ο πίνακας αποτελεσμάτων χρησιμοποιήθηκε για πρώτη φορά στον υπολογιστή CDC 6600 από όπου και η τεχνική πήρε το όνομά της.

**Στόχος της τεχνικής είναι η διατήρηση του ρυθμού εκτέλεσης μιας εντολής ανά κύκλο ρολογιού, όταν δεν υπάρχουν κατασκευαστικοί κίνδυνοι. Αυτό μπορεί να γίνει με την εκτέλεση μιας εντολής που δεν εξαρτάται από καμιά τρέχουσα ή καθυστερημένη εντολή, όσο το δυνατόν νωρίτερα.**

Ο πίνακας αποτελεσμάτων αναλαμβάνει τη διευθέτηση και εκτέλεση της εντολής καθώς και την ανίχνευση όλων των κινδύνων. Κάθε εντολή διέρχεται μέσα από τον πίνακα αποτελεσμάτων, όπου καταχωρούνται οι εξαρτήσεις δεδομένων. Επίσης καθορίζει πότε η εντολή μπορεί να διαβάσει τους τελεστέους της και να αρχίσει την εκτέλεση. Εάν ο πίνακας αποτελεσμάτων αποφασίσει ότι η εντολή δεν μπορεί να εκτελεστεί, τότε, παρακολουθώντας κάθε αλλαγή στο υλικό, αποφασίζει πότε η εντολή θα μπορέσει να εκτελεστεί. Επιπλέον, ο πίνακας ελέγχει πότε μια εντολή μπορεί να γράψει το αποτέλεσμά της στον καταχωρητή προορισμού. Τέλος επιτρέπει σε πολλαπλές εντολές να βρίσκονται ταυτόχρονα στο στάδιο εκτέλεσης EX, εφόσον εφαρμόζουμε εκτέλεση εκτός σειράς. Αυτό μπορεί να επιτευχθεί είτε με πολλαπλές λειτουργικές μονάδες, είτε με σωληνωμένες λειτουργικές μονάδες ή και με συνδυασμό τους. Θα υποθέσουμε ότι ο επεξεργαστής έχει πολλαπλές λειτουργικές μονάδες.

Έχουμε δύο είδη επεξεργαστών, τον CDC 6600 και τον DLX. Και οι δύο είναι κατασκευασμένοι σύμφωνα με την αρχιτεκτονική φόρτωσης-αποθήκευσης. Έτσι, οι τεχνικές που χρησιμοποιούν είναι σχεδόν ταυτόσημες. Οι λειτουργικές μονάδες που χρησιμοποιούν οι δύο αυτοί επεξεργαστές για την υλοποίηση του δυναμικού προγραμματισμού απεικονίζονται στο σχήμα 2.4.1.

| Υπολογιστής | Λειτουργικές μονάδες |
|---|---|
| CDC6600 | 4 FP: 2 mult, 1 add, 1 div<br>7 μονάδες μνήμης: 5 load, 2 store<br>7 μονάδες ακεραίων: add,shift, logical, κλπ. |
| DLX | 4 FP: 2 mult, 1 add, 1 div<br>1 μονάδα ακεραίων: integer |

**Σχήμα 2.4.1 – Οι λειτουργικές μονάδες των CDC6600 και DLX για την υλοποίηση του δυναμικού προγραμματισμού.**

**Κάθε εντολή, κατά την εκτέλεση της ακολουθεί τέσσερα βήματα, που αντικαθιστούν τα ID, EX και WB στάδια στη βασική DLX σωλήνωση:**

**1. Διευθέτηση** - Εάν μια λειτουργική μονάδα για την εντολή είναι ελεύθερη, τότε ο πίνακας αποτελεσμάτων διευθετεί την εντολή και την αποστέλλει στη λειτουργική μονάδα, με την προϋπόθεση ότι καμιά άλλη ενεργή εντολή δεν έχει τον ίδιο καταχωρητή προορισμού. Εξασφαλίζοντας ότι καμία άλλη ενεργή λειτουργική μονάδα δε θέλει να γράψει το αποτέλεσμά της στον καταχωρητή προορισμού, αποφεύγουμε τους WAW κινδύνους. Εάν υπάρχει ένας κατασκευαστικός ή WAW κίνδυνος, τότε η διευθέτηση της εντολής καθυστερεί και καμιά από τις επόμενες εντολές δε διευθετείται μέχρι οι κίνδυνοι να εξαλειφθούν. Αυτό το βήμα αντικαθιστά μέρος του ID βήματος στη DLX σωλήνωση.

**2. Διάβασμα τελεστέων** - Ο πίνακας αποτελεσμάτων ελέγχει τη διαθεσιμότητα των πηγαίων τελεστέων. Ένας πηγαίος τελεστέος είναι διαθέσιμος σε δύο περιπτώσεις**:**

- Όταν καμία προηγούμενα διευθετημένη ενεργή εντολή δεν πρόκειται να γράψει σε αυτόν.
- Όταν ο καταχωρητής που περιέχει τον τελεστέο γράφεται από μία τρέχουσα ενεργή λειτουργική μονάδα.

Όταν οι πηγαίοι τελεστέοι είναι διαθέσιμοι, ο πίνακας αποτελεσμάτων επιτρέπει στη λειτουργική μονάδα να προχωρήσει στην ανάγνωση των τελεστέων από τους καταχωρητές και να αρχίσει την εκτέλεση. Ο πίνακας επιλύει δυναμικά τους RAW κινδύνους και ίσως τότε οι εντολές να εκτελεστούν εκτός σειράς.

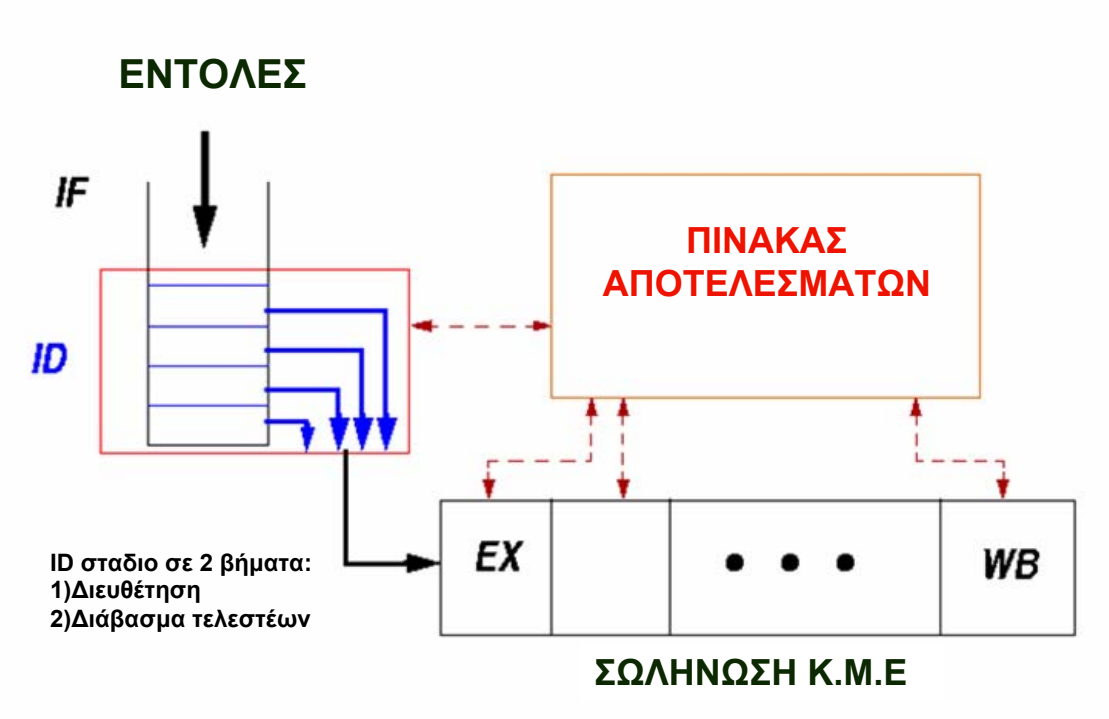
**Τα βήματα ένα και δύο αντικαθιστούν το ID στάδιο της απλής DLX σωλήνωσης.**

**3. Εκτέλεση** - Η λειτουργική μονάδα αρχίζει την εκτέλεση με τη λήψη των τελεστέων. Όταν το αποτέλεσμα είναι έτοιμο, γνωστοποιεί στον πίνακα αποτελεσμάτων ότι έχει ολοκληρώσει την εκτέλεση. Το βήμα αυτό αντικαθιστά το EX στάδιο της απλής DLX σωλήνωσης.

**4. Εγγραφή αποτελέσματος** - Μόλις ο πίνακας αποτελεσμάτων ενημερωθεί ότι η λειτουργική μονάδα έχει ολοκληρώσει την εκτέλεση, ελέγχει για WAR κινδύνους και αν είναι απαραίτητο καθυστερεί την εντολή προς ολοκλήρωση.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

**Μπορείτε να απαριθμήσετε τα βήματα για την εκτέλεση μιας εντολής με την τεχνική του πίνακα αποτελεσμάτων και να περιγράψετε τις λειτουργίες που επιτελούνται σε κάθε ένα;**
**Ποια στάδια της βασικής σωλήνωσης του DLX αντικαθιστούν τα βήματα αυτά;**
**Μπορείτε να σχεδιάσετε ένα σχήμα το οποίο να απεικονίζει την εκτέλεση των εντολών με τη χρήση της τεχνικής του πίνακα αποτελεσμάτων; Να συγκρίνετε την απάντησή σας με το σχήμα που ακολουθεί (σχήμα 2.4.2).**




**Σχήμα 2.4.2 – Η εκτέλεση των εντολών με τη χρήση της τεχνικής του πίνακα αποτελεσμάτων.**

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 6

**Έστω το τμήμα κώδικα:**

**DIVD F0 , F2 , F4**
**ADDD F10 , F0 , F8**
**SUBD F8 , F8 , F14**

- **Μπορείτε να προσδιορίσετε τις εξαρτήσεις μεταξύ των εντολών;**

## ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 6

**Η εντολή ADDD έχει ως πηγαίο τελεστέο τον καταχωρητή F8, ο οποίος είναι και ο καταχωρητής προορισμού της SUBD. Επίσης η εντολή ADDD εξαρτάται από την εντολή DIVD. Ο πίνακας αποτελεσμάτων θα καθυστερήσει επιπλέον την εντολή SUBD, κατά το στάδιο εγγραφής αποτελέσματος, μέχρι η εντολή ADDD να διαβάσει τους τελεστέους της.**

**Εντολές προς ολοκλήρωση ονομάζονται οι εντολές που είναι έτοιμες να γράψουν το αποτέλεσμά τους κατά το στάδιο εγγραφής αποτελέσματος.**

Μία εντολή προς ολοκλήρωση δεν μπορεί να γράψει το αποτέλεσμα της όταν**:**

- Υπάρχει μια εντολή που προηγείται και δεν έχει διαβάσει τους τελεστέους της. και
- ένας από τους τελεστέους είναι και καταχωρητής προορισμού.

Για παράδειγμα, στον κώδικα της δραστηριότητας 6, η εντολή SUBD δεν μπορεί να γράψει το αποτέλεσμά της αν η εντολή ADDD δεν έχει διαβάσει τους τελεστέους της και ο καταχωρητής F8 είναι και πηγαίος καταχωρητής και καταχωρητής προορισμού.

Εάν ο WAR κίνδυνος δεν υπάρχει ή έχει εξαλειφθεί, τότε ο πίνακας αποτελεσμάτων επιτρέπει στη λειτουργική μονάδα να αποθηκεύσει το αποτέλεσμά της στον καταχωρητή προορισμού.
Το βήμα εγγραφής αποτελέσματος αντικαθιστά το WB στάδιο της απλής DLX σωλήνωσης.

## Παράδειγμα:

**Στο παράδειγμα παρουσιάζεται με λεπτομέρεια η εσωτερική δομή του πίνακα αποτελεσμάτων, υποστηριζόμενη από έναν επεξεργαστή DLX με πέντε λειτουργικές μονάδες. Θεωρούμε το παρακάτω τμήμα κώδικα:**

```
LD      F6 , 34(R2)
LD      F2 , 45(R3)
MULTD   F0 , F2 , F4
SUBD    F8 , F6 , F2
DIVD    F10 , F0 , F6
ADDD    F6 , F8 , F2
```

## Απάντηση:

Ο πίνακας αποτελεσμάτων για ένα στιγμιότυπο του παραπάνω κώδικα, όταν η πρώτη εντολή LD έχει γράψει το αποτέλεσμά της, αποτελείται από τρία τμήματα: το τμήμα κατάστασης εντολής, το τμήμα λειτουργικής μονάδας και το τμήμα κατάστασης καταχωρητή αποτελέσματος.

### Τμήμα κατάστασης εντολής

| Κατάσταση εντολής | | | | |
|---|---|---|---|---|
| Εντολή | Διευθέτηση | Διάβασμα τελεστέων | Ολοκλήρωση εκτέλεσης | Εγγραφή αποτελέσματος |
| LD F6,34(R2) | √ | √ | √ | √ |
| LD F2,45(R3) | √ | √ | √ | |
| MULTD F0 ,F2,F4 | √ | | | |
| SUBD F8,F6,F2 | √ | | | |
| DIVD F10,F0,F6 | √ | | | |
| ADDD F6,F8,F2 | | | | |

Κάθε εντολή που έχει διευθετηθεί ή βρίσκεται σε αναμονή για να διευθετηθεί, έχει μια εγγραφή στον πίνακα κατάστασης εντολής. Το τμήμα κατάστασης εντολής υποδεικνύει σε ποιο από τα τέσσερα βήματα βρίσκεται η εντολή. Στο παράδειγμά μας, ο καταχωρητής κατάστασης εντολής λέει ότι η πρώτη εντολή LD έχει ολοκληρωθεί και γράψει το αποτέλεσμά της, ενώ η δεύτερη εντολή LD έχει ολοκληρώσει την εκτέλεση της αλλά δεν έχει γράψει ακόμη το αποτέλεσμά της. Οι εντολές MULTD, SUBD και DIVD έχουν όλες διευθετηθεί αλλά παρουσιάζουν καθυστέρηση, περιμένοντας τους τελεστέους τους.

### Τμήμα κατάστασης λειτουργικής μονάδας

| Κατάσταση λειτουργικής μονάδας | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Busy** | **Op** | **Fi** | **Fj** | **Fk** | **Qj** | **Qk** | **Rj** | **Rk** |
| Integer | Yes | Load | F2 | R3 | | | | No | No |
| Mult1 | Yes | Mult | F0 | F2 | F4 | Integer | | No | Yes |
| Mult2 | No | | | | | | | | |
| Add | Yes | Sub | F8 | F6 | F2 | | Integer | Yes | No |
| Divide | Yes | Div | F10 | F0 | F6 | Mult1 | | No | Yes |

Υποδεικνύει την κατάσταση της λειτουργικής μονάδας (FU). Στον πίνακα υπάρχει μια εγγραφή για κάθε λειτουργική μονάδα. Μόλις μια εντολή εκτελεστεί, οι τελεστέοι καταχωρούνται στον πίνακα κατάστασης λειτουργικής μονάδας.
Υπάρχουν εννέα πεδία για κάθε λειτουργική μονάδα **:**

Busy **:** Υποδεικνύει αν η λειτουργική μονάδα είναι απασχολημένη ή όχι.
Op **:** Λειτουργία που εκτελείται στη λειτουργική μονάδα (π.χ πρόσθεση ή αφαίρεση).
Fi **:** Καταχωρητής προορισμού.
Fj,Fk **:** Πηγαίοι καταχωρητές.
Qj,Qk **:** Λειτουργικές μονάδες που χρησιμοποιούν τους πηγαίους

καταχωρητές Fj, Fk.

Rj,Rk **:** Σημαίες που υποδεικνύουν πότε οι καταχωρητές Fj, Fk είναι έτοιμοι.

Τα πεδία μηδενίζονται όταν νέες τιμές διαβάζονται έτσι ώστε ο πίνακας αποτελεσμάτων να γνωρίζει ότι έχει εκτελεστεί ανάγνωση του πηγαίου τελεστέου (αυτό απαιτείται για την αντιμετώπιση των WAR κινδύνων).

Στο παράδειγμα μας, η κατάσταση λειτουργικής μονάδας λέει ότι η πρώτη μονάδα πολλαπλασιασμού περιμένει τη μονάδα ακέραιων, η μονάδα πρόσθεσης περιμένει τη μονάδα ακέραιων και η μονάδα διαίρεσης περιμένει την πρώτη μονάδα πολλαπλασιασμού. Επίσης, όταν ένας τελεστέος έχει διαβαστεί, τα πεδία Rj και Rk παίρνουν την τιμή No.

- **Τμήμα κατάστασης καταχωρητή αποτελέσματος**

| **Κατάσταση καταχωρητή αποτελέσματος** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **F0** | **F2** | **F4** | **F6** | **F8** | **F10** | **F12** | **…** | **F30** |
| Mult | integer | | | Sub | Divide | | | |

Αν μια ενεργή εντολή έχει τον καταχωρητή ως προορισμό της, τότε το τμήμα αυτό υποδεικνύει ποια λειτουργική μονάδα θα γράψει σε κάθε καταχωρητή. Ο αριθμός των εγγραφών στον πίνακα είναι ίσος με τον αριθμό των καταχωρητών.

## Παράδειγμα:

**Στις λειτουργικές μονάδες κινητής υποδιαστολής έχουμε τις παρακάτω καθυστερήσεις, μετρημένες σε κύκλους ρολογιού, για το στάδιο EX. Η πρόσθεση έχει καθυστέρηση 2 κύκλους ρολογιού, ο πολλαπλασιασμός 10 κύκλους ρολογιού και η διαίρεση 40 κύκλους ρολογιού. Χρησιμοποιώντας το τμήμα κώδικα του προηγούμενου παραδείγματος και ξεκινώντας από το σημείο που υποδεικνύεται από το τμήμα κατάστασης εντολής, θα δείξουμε πώς γίνονται οι πίνακες καταστάσεων όταν οι εντολές MULTD και DIVD είναι έτοιμες να προχωρήσουν στο βήμα εγγραφή αποτελέσματος.**

```
LD      F6 , 34(R2)
LD      F2 , 45(R3)
MULTD   F0 , F2 , F4
SUBD    F8 , F6 , F2
DIVD    F10 , F0 , F6
ADDD    F6 , F8 , F2
```

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 7

- **Να εξηγήσετε τους κινδύνους που προκύπτουν από την εκτέλεση των εντολών.**
- **Μπορείτε να σχεδιάσετε το τμήμα κατάστασης εντολής, το τμήμα λειτουργικής μονάδας και το τμήμα κατάστασης καταχωρητή αποτελέσματος για την εκτέλεση του παραπάνω τμήμα κώδικα;**
- **Να συγκρίνετε την απάντησή σας με την απάντηση που ακολουθεί.**

## Απάντηση:

Υπάρχουν RAW κίνδυνοι δεδομένων μεταξύ της δεύτερης εντολής LD με την εντολή MULTD και την SUBD, της εντολής MULTD με την DIVD και της εντολής SUBD με την ADDD. Υπάρχει ένας WAR κίνδυνος δεδομένων μεταξύ της εντολής DIVD και ADDD. Τέλος, υπάρχει ένας κατασκευαστικός κίνδυνος στην λειτουργική μονάδα πρόσθεσης για την εντολή ADDD. Ο πίνακας αποτελεσμάτων για ένα στιγμιότυπο του παραπάνω κώδικα, όταν η εντολή MULTD είναι έτοιμη να γράψει το αποτέλεσμά της, αποτελείται από τρία τμήματα **:**

**Τμήμα κατάστασης εντολής**

| **Κατάσταση εντολής** | | | | |
|---|---|---|---|---|
| Εντολή | Διευθέτηση | Διάβασμα τελεστέων | Ολοκλήρωση εκτέλεσης | Εγγραφή Αποτελέσματος |
| LD F6,34(R2) | √ | √ | √ | √ |
| LD F2,45(R3) | √ | √ | √ | √ |
| MULTD F0,F2,F4 | √ | √ | √ | |
| SUBD F8,F6,F2 | √ | √ | √ | √ |
| DIVD F10,F0,F6 | √ | | | |
| ADDD F6,F8,F2 | √ | √ | √ | |

Από το τμήμα κατάστασης εντολής φαίνεται ότι η εντολή DIVD δεν έχει ακόμα διαβάσει τους τελεστέους της, αφού έχει εξάρτηση από το αποτέλεσμα της εντολής MULTD. Η εντολή ADDD έχει διαβάσει τους τελεστέους της και εκτελείται, αν και περίμενε μέχρι να τελειώσει η SUBD ώστε να αποκτήσει τη λειτουργική μονάδα. Η εντολή ADDD δεν μπορεί να γράψει το αποτέλεσμα στον καταχωρητή F6, γιατί αυτός χρησιμοποιείται από την DIVD.

**Τμήμα κατάστασης λειτουργικής μονάδας**

| **Κατάσταση λειτουργικής μονάδας** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Name** | **Busy** | **Op** | **Fi** | **Fj** | **Fk** | **Qj** | **Qk** | **Rj** | **Rk** |
| Integer | No | | | | | | | | |
| Mult1 | Yes | Mult | F0 | F2 | F4 | | | No | No |
| Mult2 | No | | | | | | | | |
| Add | Yes | Add | F6 | F8 | F2 | | | No | No |
| Divide | Yes | Div | F10 | F0 | F6 | Mult1 | | No | Yes |

**Τμήμα κατάστασης καταχωρητή αποτελέσματος**

| Κατάσταση καταχωρητή αποτελέσματος | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **F0** | **F2** | **F4** | **F6** | **F8** | **F10** | **F12** | **. .** | **F30** |
| Mult1 | | | Add | | Divide | | | |

Ο πίνακας αποτελεσμάτων για ένα στιγμιότυπο του παραπάνω κώδικα, όταν η εντολή DIVD είναι έτοιμη να γράψει το αποτέλεσμά της, αποτελείται από τα παρακάτω τρία τμήματα **:**

**Τμήμα κατάστασης εντολής**

| Κατάσταση εντολής | | | | |
|---|---|---|---|---|
| Εντολή | Διευθέτηση | Διάβασμα τελεστέων | Ολοκλήρωση εκτέλεσης | Εγγραφή Αποτελέσματος |
| LD F6,34(R2) | √ | √ | √ | √ |
| LD F2,45(R3) | √ | √ | √ | √ |
| MULTD F0,F2,F4 | √ | √ | √ | √ |
| SUBD F8,F6,F2 | √ | √ | √ | √ |
| DIVD F10,F0,F6 | √ | √ | √ | |
| ADDD F6,F8,F2 | √ | √ | √ | √ |

Από το τμήμα κατάστασης εντολής φαίνεται ότι η εντολή ADDD είναι ικανή να ολοκληρωθεί μόλις η DIVD περάσει από το βήμα διάβασμα τελεστέων και πάρει ένα αντίγραφο του F6. Μόνο η DIVD απομένει να τελειώσει.

**Τμήμα κατάστασης λειτουργικής μονάδας**

| Κατάσταση λειτουργικής μονάδας | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Name** | **Busy** | **Op** | **Fi** | **Fj** | **Fk** | **Qj** | **Qk** | **Rj** | **Rk** |
| Integer | No | | | | | | | | |
| Mult1 | No | | | | | | | | |
| Mult2 | No | | | | | | | | |
| Add | No | | | | | | | | |
| Divide | Yes | Div | F10 | F0 | F6 | | | No | No |

**Τμήμα κατάστασης καταχωρητή αποτελέσματος**

| Κατάσταση καταχωρητή αποτελέσματος | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **F0** | **F2** | **F4** | **F6** | **F8** | **F10** | **F12** | **. . .** | **F30** |
| **FU** | | | | | Divide | | | |

Στο παρακάτω πίνακα παρουσιάζονται οι απαιτούμενοι έλεγχοι και ενέργειες για κάθε βήμα κατά την εκτέλεση μιας εντολής.

| **Κατάσταση εντολής** | **Wait until** | **Bookkeeping** |
|---|---|---|
| **Διευθέτηση** | Not busy (FU) and not result (D) | Busy (FU)←yes;<br>Op(FU)←op;<br>Fi(FU)←('D');<br>Fj(FU)←'S1' ;<br>Fk(FU)←'S2' ;<br>Qj←Result (S1);<br>Qk←Result(S2);<br>Rj←not Qj;<br>Rk← not Qk;<br>Result(D)←FU; |
| **Διάβασμα τελεστέων** | Rj and Rk | Rj ←No; Rk ←No; Qj ← 0; Qk ← 0 |
| **Ολοκλήρωση εκτέλεσης** | Functional unit done | |
| **Εγγραφή αποτελέσματος** | ∀*f* ((Fj(f)≠ Fi(FU) or Rj(*f*) = No) &(Fk(*f*) ≠ Fi(FU) or Rk(*f*) =No)) | ∀*f* (if Qj(*f* )=FU then Rj(*f* )←Yes);<br>∀*f* (if Qk(*f* )=FU then Rk(*f* )←Yes);<br>Result(Fi(FU)) ← 0 ; Busy(FU)←No |

Το FU συμβολίζει τη λειτουργική μονάδα που χρησιμοποιείται από την εντολή.Ο καταχωρητής D είναι ο καταχωρητής προορισμού, ενώ οι S1 και S2 είναι οι πηγαίοι καταχωρητές. Η Op είναι η λειτουργία που θα γίνει. Όταν η καταχώρηση με όνομα Fj χρησιμοποιείται από τη λειτουργική μονάδα FU χρησιμοποιούμε το συμβολισμό Fj(FU). Η Result(D) είναι η τιμή του πεδίου αποτελέσματος καταχωρητή για τον καταχωρητή D. Η προσπάθεια εγγραφής αποτελέσματος αποτυγχάνει όταν υπάρχει ένας WAR κίνδυνος. Ο συμβολισμός Ri αφορά το όνομα του καταχωρητή Ri και όχι το περιεχόμενό του. Για απλότητα υποθέτουμε ότι όλες οι ενέργειες γίνονται σε ένα κύκλο ρολογιού.

Ο πίνακας αποτελεσμάτων αξιοποιεί το διαθέσιμο βαθμό της παραλληλίας σε επίπεδο εντολής, με σκοπό να μειώσει τις καθυστερήσεις που προκύπτουν από τις εξαρτήσεις δεδομένων. Στη προσπάθεια εξάλειψης των καθυστερήσεων, ο πίνακας περιορίζεται από ορισμένους παράγοντες **:**

**1. Το διαθέσιμο βαθμό παραλληλίας μεταξύ των εντολών**.
Ο διαθέσιμος βαθμός παραλληλίας καθορίζει κατά πόσο ανεξάρτητες εντολές μπορούν να εντοπιστούν και να εκτελεστούν. Αν κάθε εντολή εξαρτάται από προηγούμενες, κανένα σχήμα δυναμικού προγραμματισμού δεν μπορεί να μειώσει τις καθυστερήσεις.

**2. Τον αριθμό των καταχωρήσεων στον πίνακα αποτελεσμάτων.**
Ο αριθμός των καταχωρήσεων καθορίζει πόσο μακριά μπορεί να ψάξει η σωλήνωση για ανεξάρτητες εντολές. Το σύνολο των εντολών που εξετάζεται ως υποψήφιο για πιθανή εκτέλεση ονομάζεται **παράθυρο**. Το μέγεθος (αριθμός καταχωρήσεων) του πίνακα αποτελεσμάτων καθορίζει το μέγεθος του παραθύρου. Θεωρούμε ότι ένα παράθυρο δεν επεκτείνεται πέρα από μία διακλάδωση κι έτσι το παράθυρο και ο πίνακας αποτελεσμάτων περιέχουν πάντα κώδικα που εκτελείται σειριακά περιορισμένο σε ένα μονό βασικό μπλοκ.

**3. Το πλήθος και τους τύπους των λειτουργικών μονάδων.**
Καθορίζουν τη σπουδαιότητα των κατασκευαστικών κινδύνων, οι οποίοι μπορεί να αυξηθούν όταν χρησιμοποιείται δυναμικός προγραμματισμός.

**4. Την παρουσία αντιεξαρτήσεων και εξαρτήσεων εξόδου.**
Αυτές προκαλούν WAR και WAW κινδύνους, που με τη σειρά τους προκαλούν τις αντίστοιχες καθυστερήσεις.

Ο δεύτερος και ο τρίτος παράγοντας μπορούν να αντιμετωπιστούν αυξάνοντας το μέγεθος του πίνακα αποτελεσμάτων και το πλήθος των λειτουργικών μονάδων. Αυτές όμως οι αλλαγές, αυξάνουν το κόστος και επηρεάζουν τη διάρκεια του κύκλου. Οι WAW και WAR κίνδυνοι είναι περισσότερο σημαντικοί σε δυναμικά προγραμματισμένους επεξεργαστές, επειδή στη σωλήνωση εμφανίζονται περισσότερες εξαρτήσεις ονομάτων. Οι WAW κίνδυνοι είναι επίσης σημαντικοί στη περίπτωση που χρησιμοποιούμε δυναμικό προγραμματισμό με σχήμα πρόβλεψης διακλάδωσης, το οποίο επιτρέπει πολλαπλές επαναλήψεις ενός βρόχου να επικαλύπτονται.

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 8

**Δίνεται το παρακάτω τμήμα κατάσταση εντολής, από τον πίνακα αποτελεσμάτων.**

| Κατάσταση εντολής | | | | |
|---|---|---|---|---|
| Εντολή | Διευθέτηση | Διάβασμα τελεστέων | Ολοκλήρωση εκτέλεσης | Εγγραφή Αποτελέσματος |
| LD F6,34(R2) | √ | √ | √ | √ |
| LD F2,45(R3) | √ | √ | √ | √ |
| MULTD F0,F2,F4 | √ | √ | √ | √ |
| SUBD F8,F6,F2 | √ | √ | √ | √ |
| DIVD F10,F0,F6 | √ | √ | √ | |
| ADDD F6,F8,F2 | √ | √ | √ | √ |

**Να εξηγήσετε σε τι κατάσταση βρίσκεται η κάθε εντολή που περιέχεται στον πίνακα για το συγκεκριμένο στιγμιότυπο.**

## ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 8

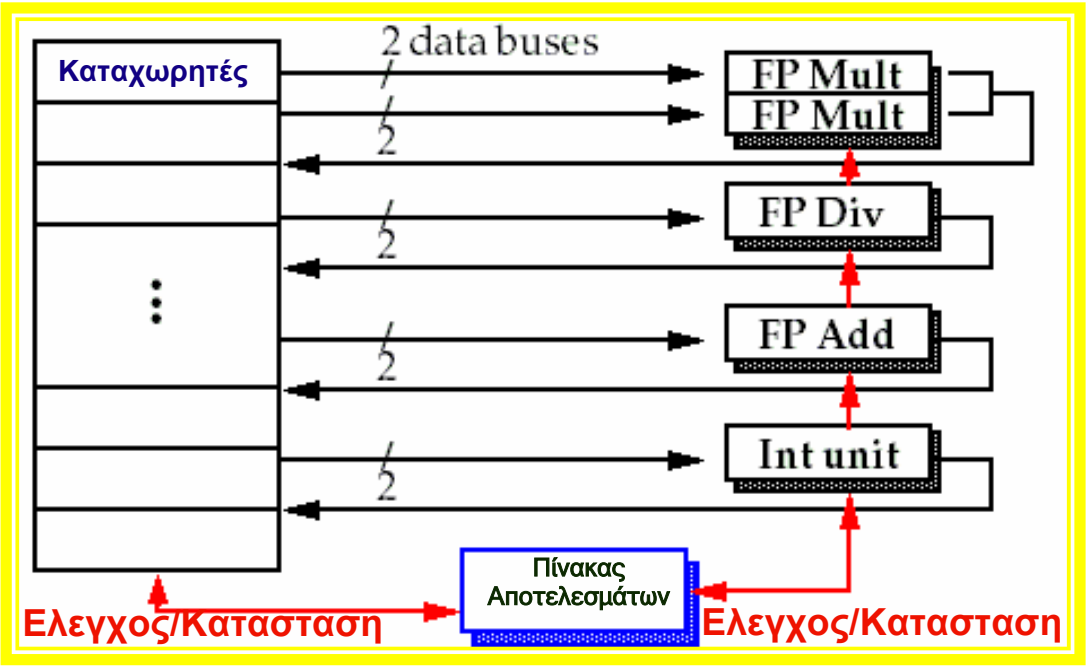
**Η πρώτη εντολή LD έχει διευθετηθεί, έχει διαβάσει τους τελεστέους της, έχει ολοκληρώσει την εκτέλεση της και έχει γράψει το αποτέλεσμα της στο καταχωρητή F6. Ομοίως η δεύτερη εντολή LD, καθώς και οι εντολές MULTD και SUBD. Η εντολή DIVD έχει ολοκληρώσει τα τρία πρώτα βήματα αλλά δεν έχει γράψει το αποτέλεσμά της. Τέλος, η εντολή ADDD ολοκληρώθηκε αφού πήρε ένα αντίγραφο του F6 μόλις η DIVD πέρασε από το βήμα διάβασμα τελεστέων. Μόνο η DIVD απομένει να τελειώσει.**

## Υλικό Επεξεργαστή στον Πίνακα Αποτελεσμάτων

Κάθε εντολή περνά από τον πίνακα αποτελεσμάτων όπου καταγράφονται οι εξαρτήσεις δεδομένων. Αυτό το βήμα αντιστοιχεί στην διευθέτηση εντολών και αντικαθιστά μέρος του ID βήματος στην σωλήνωση για τον DLX. Ο πίνακας αποτελεσμάτων προσδιορίζει πότε μια εντολή μπορεί να διαβάσει τους τελεστέους της για να αρχίσει να εκτελείται. Αν αποφασίσει ότι η εντολή δεν μπορεί να εκτελεστεί άμεσα, ελέγχοντας κάθε αλλαγή στο υλικό αποφασίζει πότε γίνεται δυνατή η εκτέλεση. Ο πίνακας αποτελεσμάτων ακόμη ελέγχει πότε μια εντολή μπορεί να γράψει το αποτέλεσμά της στον καταχωρητή προορισμού. Έτσι η ανίχνευση και η επίλυση όλων των κινδύνων επικεντρώνεται στον πίνακα αποτελεσμάτων. Χρησιμοποιώντας δική της δομή δεδομένων η τεχνική SCOREBOARDING ελέγχει την πρόοδο των εντολών από το ένα βήμα στο άλλο, έχοντας συνεχή επικοινωνία με τις λειτουργικές μονάδες. Παρ' όλα αυτά υπάρχει μια μικρή επιπλοκή. Υπάρχει περιορισμένος αριθμός διαύλων εισόδου και διαύλων αποτελέσματος προς το αρχείο καταχωρητών, πράγμα που αποτελεί κατασκευαστικό κίνδυνο.

Η τεχνική του πίνακα αποτελεσμάτων πρέπει να εγγυηθεί ότι ο αριθμός των λειτουργικών μονάδων που τους επιτρέπεται να προχωρήσουν στα βήματα 2 και 4 δεν πρέπει να υπερβαίνουν τον αριθμό των διόδων που είναι διαθέσιμοι. Αυτό επιτυγχάνεται χωρίζοντας σε ομάδες τις λειτουργικές μονάδες. Επιπλέον μόνο μια μονάδα από κάθε ομάδα μπορεί να διαβάσει ή να γράψει κατά την διάρκεια ενός κύκλου ρολογιού. Η πολυπλοκότητα της λογικής που χρησιμοποιεί ο πίνακας αποτελεσμάτων είναι τις ίδιας τάξης μεγέθους με αυτή των λειτουργικών μονάδων και είναι απρόσμενα μικρή . Το βασικό κόστος έγκειται στο μεγάλο αριθμό διαύλων –

περίπου τέσσερις φορές περισσότερες από όσες θα χρειάζονταν εάν ο επεξεργαστής εκτελούσε εντολές σε σειρά (ή εάν εκτελούσε μόνο μια εντολή για κάθε κύκλο).




**Σχήμα 2.4.3 - Η βασική δομή του επεξεργαστή του DLX με πίνακα αποτελεσμάτων (Scoreboard).** Η λειτουργία του πίνακα αποτελεσμάτων είναι να ελέγχει την εκτέλεση των εντολών (κάθετες γραμμές ελέγχου). Κατά τη διάρκεια εκτέλεσης του προγράμματος, κάθε εντολή περνά από τον πίνακα αποτελεσμάτων, όπου αποφασίζεται αν η εντολή είναι σε θέση να εκτελέσει ανάγνωση στους τελεστέους ή εγγραφή αποτελεσμάτων. Όλα τα δεδομένα ρέουν μεταξύ του αρχείου καταχωρητών και των λειτουργικών μονάδων μέσω των διαύλων (οριζόντιες γραμμές, ονομάζονται trunks στον CDC 6600). Υπάρχουν δυο λειτουργικές μονάδες πολλαπλασιασμού, μια διαίρεσης, μια πρόσθεσης και μια μονάδα ακεραίων.

## Ανακεφαλαιώνοντας λοιπόν…

**Στο δυναμικό προγραμματισμό το υλικό ρυθμίζει εκ νέου την εκτέλεση των εντολών, με σκοπό να μειώσει τις καθυστερήσεις στη σωλήνωση.**

**Ένας σημαντικός περιορισμός στις μέχρι τώρα τεχνικές σωλήνωσης είναι ότι όλες χρησιμοποιούν την κατά σειρά διευθέτηση των εντολών, η οποία όμως μπορεί να περιοριστεί εφαρμόζοντας την τεχνική της εκτέλεσης εντολών εκτός σειράς.**

**Για να εκτελέσουμε μία ανεξάρτητη εντολή εκτός σειράς, πρέπει να διασπάσουμε το ID στάδιο σε δύο μέρη:**
**1) Διευθέτηση-Αποκωδικοποίηση εντολών και έλεγχος κατασκευαστικών κινδύνων.**
**2) Διάβασμα τελεστέων – Αναμονή μέχρι να μειωθούν οι κίνδυνοι δεδομένων, διάβασμα τελεστέων.**

**Κάθε εντολή κατά την εκτέλεση της ακολουθεί τέσσερα βήματα, που αντικαθιστούν τα ID, EX και WB στάδια στη βασική DLX σωλήνωση: Διευθέτηση, Διάβασμα τελεστέων, Εκτέλεση και Εγγραφή αποτελέσματος.**

**Εντολές προς ολοκλήρωση ονομάζονται οι εντολές που είναι έτοιμες να γράψουν το αποτέλεσμά τους κατά το στάδιο εγγραφής αποτελέσματος.**

**Η καταγραφή αποτελεσμάτων σε πίνακα, είναι μια τεχνική που επιτρέπει στις εντολές να εκτελούνται εκτός σειράς όταν υπάρχουν επαρκείς πόροι και δεν υπάρχουν εξαρτήσεις δεδομένων. Ο πίνακας αποτελεσμάτων αποτελείται από τρία τμήματα :**
**1. Τμήμα κατάστασης εντολής.**
**2. Τμήμα κατάστασης λειτουργικής μονάδας.**
**3. Τμήμα κατάστασης καταχωρητή αποτελέσματος.**