

### Ενότητα 3<sup>η</sup> : Καλωδιωμένη Μονάδα Ελέγχου.

**Σκοπός** Ο σκοπός της ενότητας αυτής είναι να παρουσιάσει τον τρόπο με τον οποίο σχεδιάζεται και υλοποιείται η καλωδιωμένη μονάδα ελέγχου του υπολογιστή DLX. Επίσης θα δούμε πώς μετατρέπουμε το αλγοριθμικό διάγραμμα καταστάσεων σε υλικό υπολογιστή, τους εναλλακτικούς τρόπους υλοποίησής του, καθώς και πώς μπορούμε να βελτιώσουμε την επίδοση της καλωδιωμένης μονάδας ελέγχου.

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



να αναφέρετε τα κύρια χαρακτηριστικά της καλωδιωμένης μονάδας ελέγχου,



να σχεδιάζετε το αλγοριθμικό διάγραμμα καταστάσεων για την εκτέλεση των εντολών,



να περιγράφετε και να υλοποιείτε την καλωδιωμένη μονάδα ελέγχου του DLX,



να παρουσιάζετε την επίδοση της καλωδιωμένης μονάδας ελέγχου.



Καλωδιωμένη μονάδα ελέγχου, αλγοριθμικό διάγραμμα καταστάσεων



## Εισαγωγή στην Καλωδιωμένη Μονάδα Ελέγχου

Είδαμε στην προηγούμενη ενότητα ποια είναι τα κύρια τμήματα από τα οποία αποτελείται ένας σύγχρονος υπολογιστής και ότι ο επεξεργαστής είναι το σημαντικότερο κομμάτι του. Η βέλτιστη ορθή λειτουργία του υπολογιστή σχετίζεται άμεσα με το σχεδιασμό της μονάδας ελέγχου, το πιο πολύπλοκο και πιο κρίσιμο τμήμα του επεξεργαστή. Στη μονάδα αυτή βρίσκονται και τα περισσότερα λάθη, όταν ένας νέος υπολογιστής αποσφαλματοποιείται.

Ένας τύπος μονάδας ελέγχου είναι η καλωδιωμένη μονάδα ελέγχου. Η μονάδα ελέγχου ελέγχει το τι κάνει η δίοδος δεδομένων σε κάθε κύκλο ρολογιού κατά τη διάρκεια εκτέλεσης εντολών. Αυτό καθορίζεται από το αλγοριθμικό διάγραμμα καταστάσεων (ASM). Κάθε κατάσταση αντιστοιχεί σε έναν κύκλο ρολογιού, και οι λειτουργίες που πρέπει να εκτελεστούν κατά τη διάρκεια του κύκλου ρολογιού είναι γραμμένες μέσα στην κατάσταση. Κάθε εντολή χρειάζεται αρκετούς κύκλους για να ολοκληρωθεί.



### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Θυμάστε τι είναι το αλγοριθμικό διάγραμμα καταστάσεων (ASM); Για περισσότερες λεπτομέρειες, θα ήταν καλό να ανατρέξετε στο βιβλίο του M. Morris Mano: “Ψηφιακή σχεδίαση” και συγκεκριμένα στο 8<sup>ο</sup> κεφάλαιο.



### ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1

Το διάγραμμα ASM είναι ένας ειδικός τύπος διαγράμματος ροής, κατάλληλο για την περιγραφή των ακολουθιακών λειτουργιών σε ένα ψηφιακό σύστημα. Συγκεκριμένα, περιγράφει τόσο τη σειρά των γεγονότων όσο και τη χρονική σχέση μεταξύ των καταστάσεων ενός ακολουθιακού ελεγκτή, καθώς και τα γεγονότα που συμβαίνουν κατά τη μετάβαση από μία κατάσταση στην επόμενη.



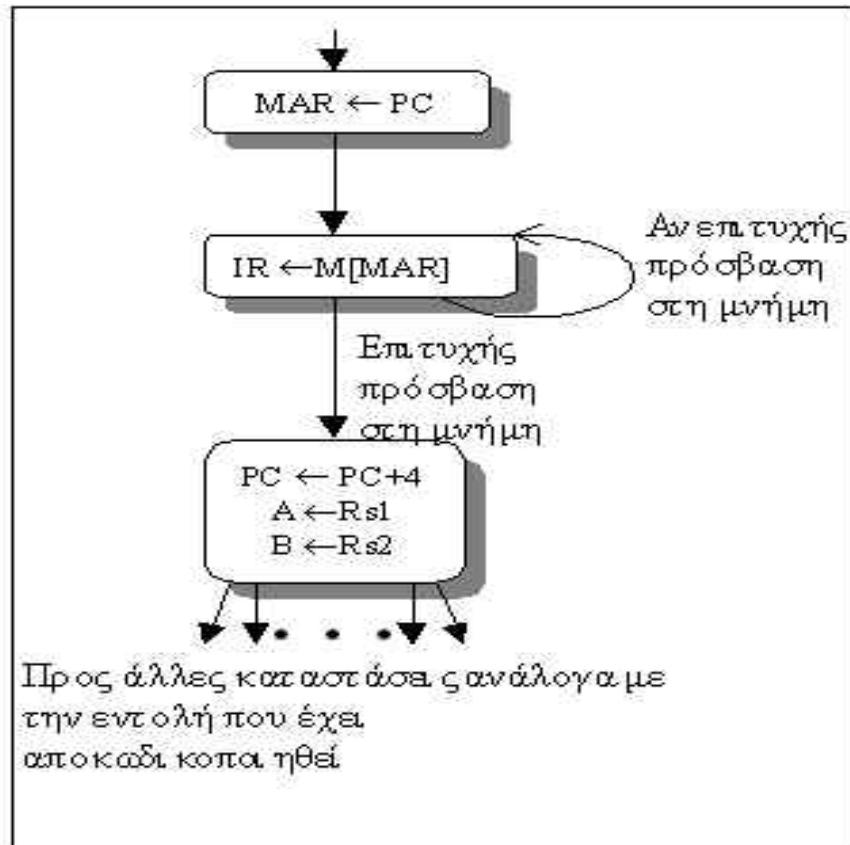
### Παράδειγμα

Τα δύο πρώτα βήματα της εκτέλεσης εντολής στον DLX αντιστοιχούνται σε ένα τμήμα ενός αλγοριθμικού διαγράμματος καταστάσεων. Τα δύο πρώτα βήματα αντιστοιχούν σε τρεις καταστάσεις του αλγοριθμικού διαγράμματος καταστάσεων:

- Κατά τη διάρκεια της πρώτης κατάστασης, το περιεχόμενο του απεριθμητή προγράμματος, δηλαδή η διεύθυνση της εντολής που πρόκειται να εκτελεστεί, μεταφέρεται στον καταχωρητή διευθύνσεων μνήμης (MAR), ο οποίος βρίσκεται σε επικοινωνία με τη μνήμη.

- Στη δεύτερη κατάσταση, η εντολή που πρόκειται να εκτελεστεί, ανακαλείται από τη μνήμη και αποθηκεύεται στον καταχωρητή εντολών.
- Κατά τη διάρκεια της τρίτης κατάστασης, γίνεται αποκωδικοποίηση της εντολής, και ταυτόχρονα γίνεται πρόσβαση στο αρχείο καταχωρητών για να διαβαστούν οι πηγαίοι καταχωρητές Rs1 και Rs2, οι οποίοι φορτώνονται στους καταχωρητές A και B. Ο απριθμητής προγράμματος αυξάνεται έτσι ώστε να δείχνει τη διεύθυνση της επόμενης εντολής προς εκτέλεση.

Το παρακάτω σχήμα δείχνει τις τρεις αυτές καταστάσεις.



**Σχήμα 3.3.1** - Ένα τμήμα του αλγοριθμικού διαγράμματος καταστάσεων, το οποίο περιλαμβάνει τα δύο πρώτα βήματα εκτέλεσης μιας εντολής. 🚩



## Μετατροπή αλγοριθμικού διαγράμματος καταστάσεων σε υλικό υπολογιστή.

Η μετατροπή ενός αλγοριθμικού διαγράμματος καταστάσεων σε υλικό υπολογιστή είναι το επόμενο βήμα. Οι εναλλακτικοί τρόποι για να γίνει αυτό εξαρτώνται από την τεχνολογία υλοποίησης. Ένας τρόπος για να εκτιμήσουμε την πολυπλοκότητα της υλοποίησης αυτής είναι με το γινόμενο:

$$\text{Καταστάσεις} * \text{Είσοδοι Ελέγχου} * \text{Έξοδοι Ελέγχου}$$

Όπου:

**Καταστάσεις** = Ο αριθμός καταστάσεων που είναι απαραίτητες για την εκτέλεση των εντολών του υπολογιστή

**Είσοδοι Ελέγχου** = Ο αριθμός των σημάτων που είναι είσοδοι στη μονάδα ελέγχου

**Έξοδοι Ελέγχου** = Ο αριθμός των σημάτων ελέγχου που δημιουργούνται από τη μονάδα ελέγχου



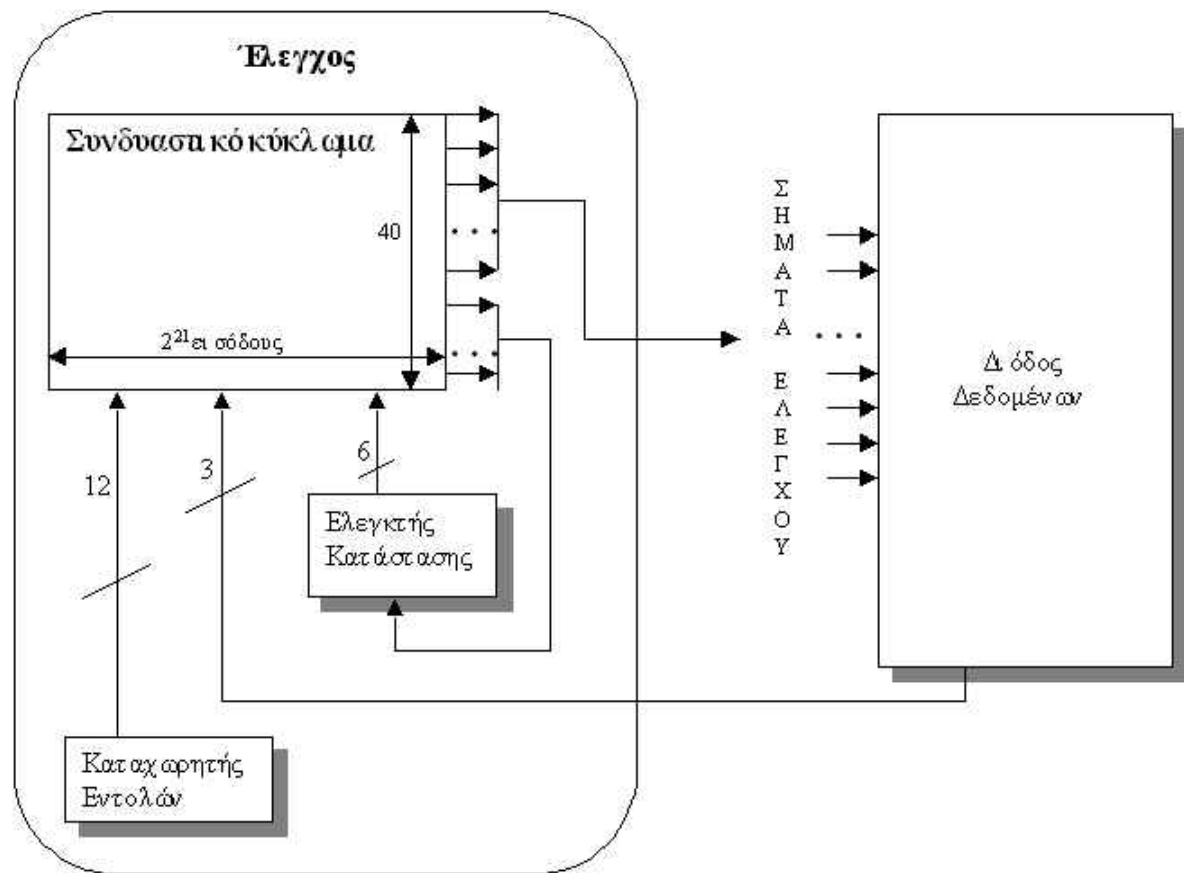
Αν υποθέσουμε ότι το αλγοριθμικό διάγραμμα καταστάσεων για τον DLX περιλαμβάνει 50 καταστάσεις και συνεπώς απαιτούνται 6 bits για την αναπαράσταση της κάθε κατάστασης. Οι είσοδοι ελέγχου πρέπει να περιέχουν αυτά τα 6 bits, έναν αριθμό από bits (για παράδειγμα 3) για την επιλογή των συνθηκών από τη δίοδο δεδομένων και τη μονάδα μνήμης, και bits που προέρχονται από τον καταχωρητή εντολών (IR). Οι πηγαίοι καταχωρητές, καθώς και οι καταχωρητές προορισμού, στέλνονται κατευθείαν στο υλικό, έτσι, δεν υπάρχει ανάγκη να σταλούν και τα 32 bits των εντολών του DLX σαν είσοδοι ελέγχου. Ο κωδικός λειτουργίας της εντολής είναι 6 bits, επίσης χρησιμοποιούνται 6 bits από το πεδίο "func" της εντολής, σχηματίζοντας ένα ολικό άθροισμα 21 bits για τις εισόδους ελέγχου. Έτσι το συνδυαστικό κύκλωμα της καλωδιωμένης μονάδας ελέγχου, δέχεται  $2^{21}$  εισόδους και παράγει ένα αριθμό σημάτων ελέγχου απαραίτητων για τη σωστή λειτουργία του υπολογιστή (ο αριθμός των σημάτων ελέγχου μπορεί να είναι π.χ. 40). Από τα 40 αυτά σήματα ελέγχου ένα μέρος τροφοδοτείται στον ελεγκτή κατάστασης, που είναι το κύκλωμα που καθορίζει την επόμενη κατάσταση.



### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Να σχεδιάσετε τη δομή της καλωδιωμένης μονάδας ελέγχου του υπολογιστή DLX. Να συγκρίνετε το σχήμα που σχεδιάσατε με το σχήμα που ακολουθεί (σχήμα 3.3.2).

Στο παρακάτω σχήμα παρουσιάζεται η δομή της καλωδιωμένης μονάδας ελέγχου:



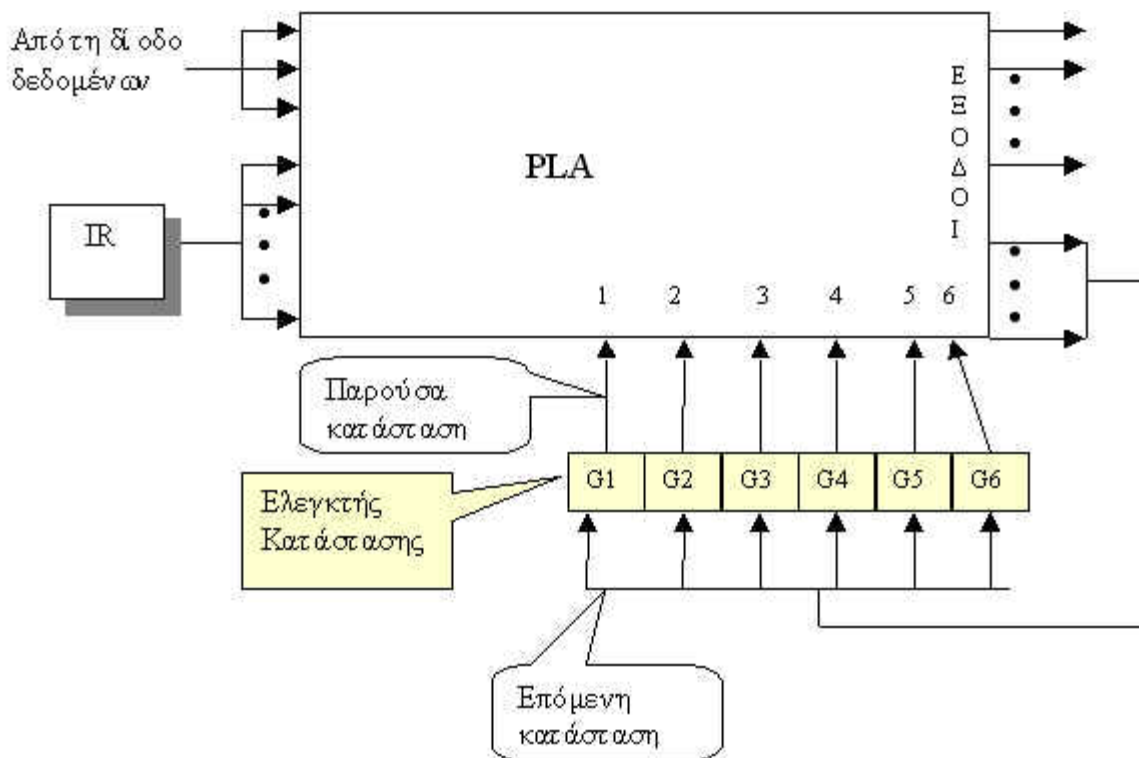
Σχήμα 3.3.2 - Δομή καλωδιωμένης μονάδας ελέγχου.



## Υλοποίηση της καλωδιωμένης μονάδας ελέγχου

Η υλοποίηση του συνδυαστικού κυκλώματος του σχήματος 3.3.2 γίνεται με μία μνήμη ROM (Read Only Memory - Μνήμη ανάγνωσης μόνο), ενώ ο ελεγκτής κατάστασης υλοποιείται με έναν καταχωρητή των 6 bits. Για την παραπάνω υλοποίηση, θα χρειαστούμε  $2^{21}$  λέξεις με εύρος 40 bits η κάθε μία (δηλ. 10 MB ROM). Καταλαβαίνουμε βέβαια ότι αυτό είναι υπερβολικά μεγάλη απαίτηση για υλικό. Επειδή πολλές θέσεις της μνήμης περιέχουν μηδενικά μπορούμε να τις παραλείψουμε κρατώντας μόνο εκείνες που περιέχουν πληροφορία. Αυτό έχει όμως σαν αποτέλεσμα την αύξηση της πολυπλοκότητας αποκωδικοποίησης των σημάτων. Μία τέτοια κατασκευή υλικού καλείται *Προγραμματιζόμενη Λογική Παράταξη* ή *PLA* (*PLA - programmed logic array*). Με τη χρήση του PLA έχουμε ελάττωση της απαιτούμενης μνήμης από  $2^{21}$  λέξεις σε πολύ λιγότερες (για παράδειγμα μια συνηθισμένη τιμή είναι 50), αυξάνοντας όμως την πολυπλοκότητα αποκωδικοποίησης των σημάτων.

Στο παρακάτω σχήμα φαίνεται πως επικοινωνούν το συνδυαστικό κύκλωμα (που στο σχήμα μας είναι μια PLA) και ο ελεγκτής κατάστασης.



**Σχήμα 3.3.3 - Το συνδυαστικό κύκλωμα (PLA) και ο ελεγκτής κατάστασης μιας καλωδιωμένης μονάδας ελέγχου.** Όπως παρατηρούμε ο ελεγκτής κατάστασης αποτελείται από έναν καταχωρητή με 6 flip-flops τα οποία δέχονται σαν είσοδο την επόμενη κατάσταση που έχει παραχθεί από την PLA και τροφοδοτούν την PLA με την παρούσα κατάσταση. Η PLA παράγει ένα πλήθος σημάτων ελέγχου, από τα οποία ένα μέρος στέλνεται στη δίοδο δεδομένων και ένα μέρος αποτελεί την επόμενη κατάσταση.



### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Η πιο εύχρηστη διάταξη προγραμματιζόμενης λογικής (PLD) είναι η προγραμματιζόμενη λογική παράταξη (PLA). Θυμάστε τους άλλους τύπους PLD, καθώς και τα κυκλώματα από τα οποία αποτελείται μια PLA; Για περισσότερες λεπτομέρειες για τις PLD, καθώς και για την PLA, θα ήταν καλό να ανατρέξετε στο βιβλίο του M. Morris Mano: “Ψηφιακή σχεδίαση” και συγκεκριμένα στο 5<sup>ο</sup> κεφάλαιο.



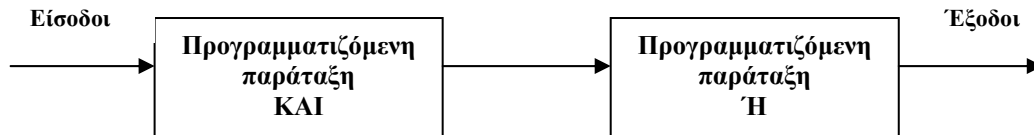
### ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3

Οι τρεις τύποι διάταξης προγραμματιζόμενης λογικής (PLD) είναι:

- 1) Προγραμματιζόμενη μνήμη ανάγνωσης μόνο (PROM)
- 2) Προγραμματιζόμενη παράταξη λογικής (PAL)
- 3) Προγραμματιζόμενη λογική παράταξη (PLA)

Οι τρεις τύποι συνδέσμων διαφέρουν στην τοποθέτηση των συνδέσμων στην παράταξη ΚΑΙ – Ή. Η πιο εύχρηστη (PLD) είναι η προγραμματιζόμενη λογική παράταξη (PLA), όπου και οι ΚΑΙ και οι Ή παρατάξεις μπορούν να προγραμματιστούν.

Η βασική δομή μιας PLA είναι:





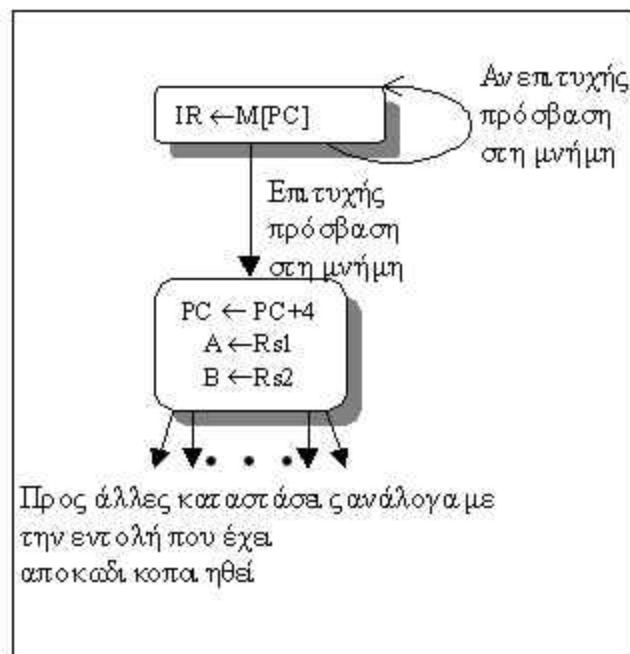
## Επίδοση της καλωδιωμένης μονάδας ελέγχου

Όταν σχεδιάζουμε τη μονάδα ελέγχου ενός υπολογιστή, θέλουμε να ελαχιστοποιήσουμε το μέσο CPI, τον κύκλο ρολογιού, το ποσό του υλικού που προσδιορίζει τον έλεγχο και το χρόνο για την ανάπτυξη ενός σωστού ελεγκτή. Ελαχιστοποίηση του CPI σημαίνει ελάττωση του μέσου αριθμού καταστάσεων κατά την εκτέλεση μίας εντολής, αφού κάθε κύκλος ρολογιού αντιστοιχεί σε μία κατάσταση. Αυτό για παράδειγμα γίνεται κάνοντας αλλαγές στη δίοδο δεδομένων ώστε να συνδυαστούν ή να απομακρυνθούν καταστάσεις.



### Παράδειγμα

Αν αλλάξουμε το υλικό έτσι ώστε ο PC να μπορεί να χρησιμοποιηθεί απευθείας στη διεύθυνση μνήμης χωρίς να πηγαίνει διαμέσου του MAR πρώτα, τότε θα πρέπει να καταργήσουμε την πρώτη κατάσταση στην οποία ο PC αντιγράφεται στο MAR. Εξοικονομείται έτσι ένας κύκλος ρολογιού για κάθε εντολή. Έτσι, αν η μέση τιμή του CPI ήταν αρχικά 7, με την αλλαγή αυτή θα κατέβαινε στο 6. Με την προϋπόθεση ότι δεν υπήρχε επίδραση στο χρόνο κύκλου ρολογιού, η αλλαγή θα έκανε τη μηχανή 17% ταχύτερη. 🚩

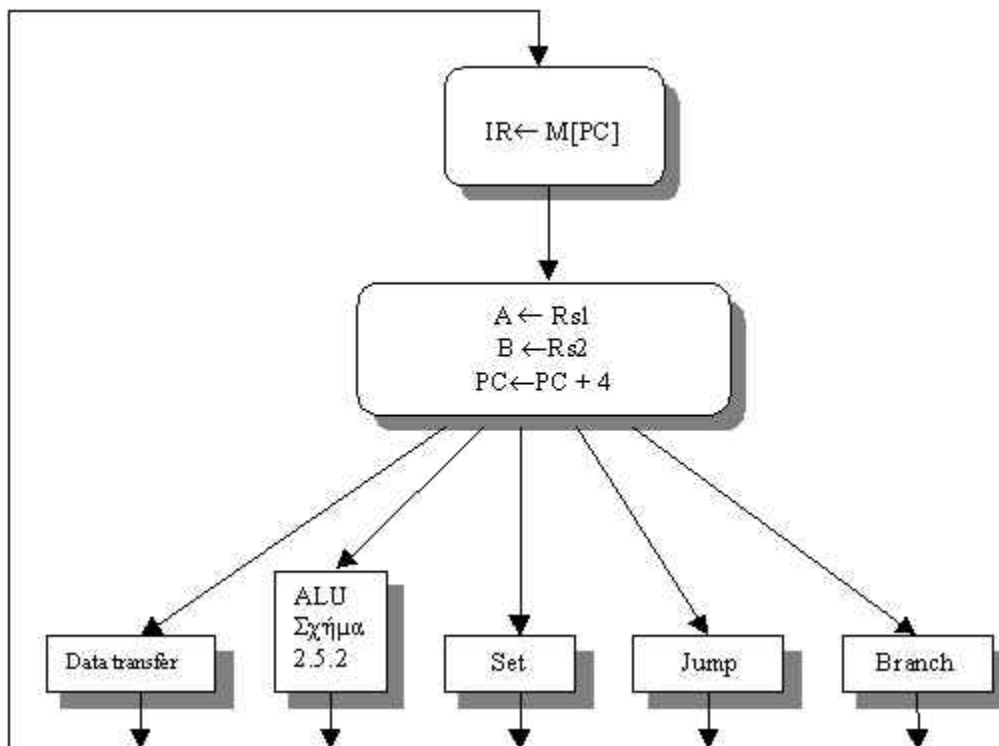


**Σχήμα 3.3.4 - Τροποποίηση των βημάτων IF και ID εκτέλεσης μιας εντολής με δύο καταστάσεις.** Παρατηρούμε ότι το πλήθος των καταστάσεων έχει μειωθεί κατά ένα.

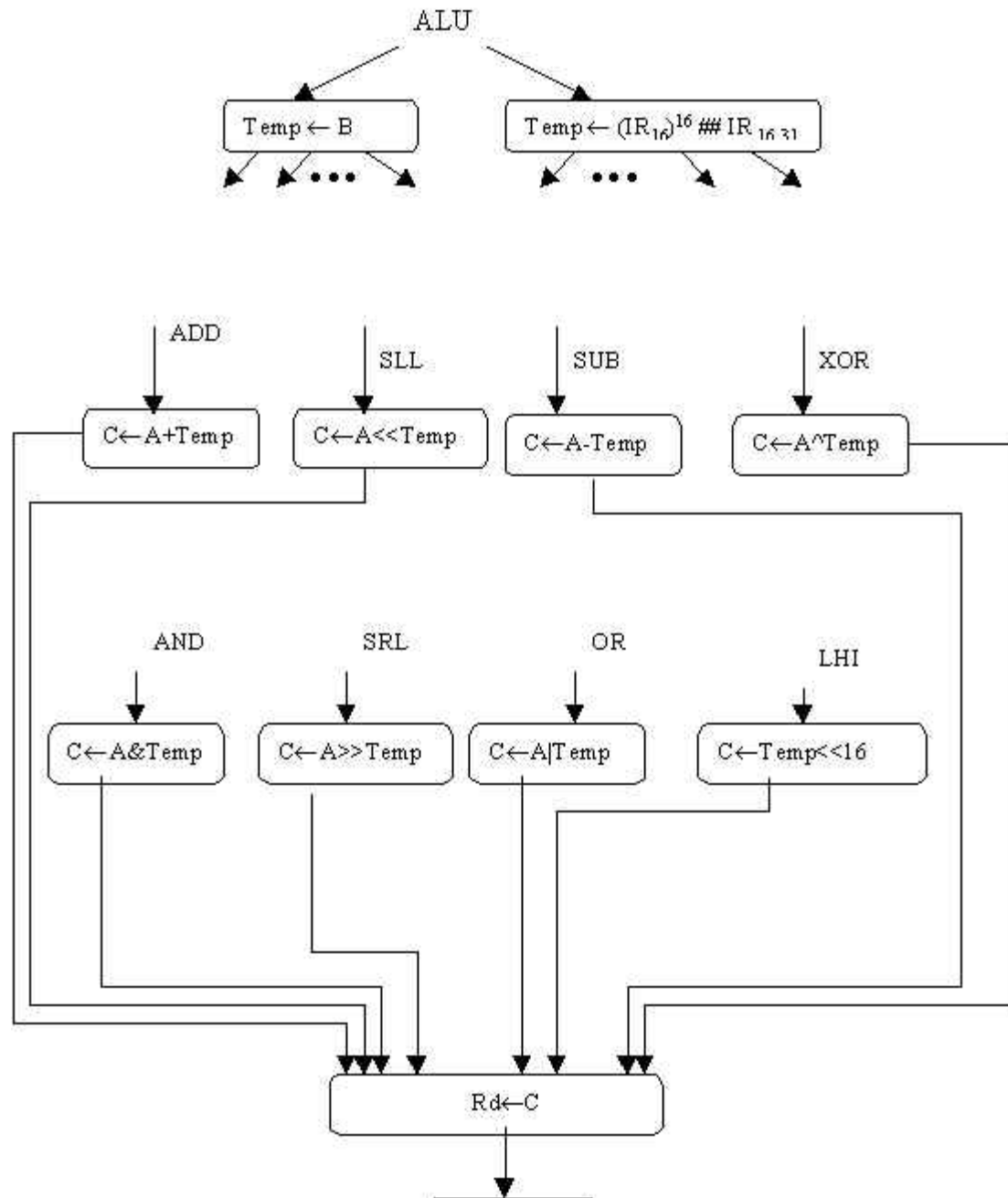


## Καλωδιωμένη μονάδα ελέγχου του DLX

Οι εντολές που εκτελούνται στον DLX είναι: μεταφορές δεδομένων (Data transfer) από και προς τη μνήμη, αριθμητικές και λογικές εντολές (ALU), εντολές που θέτουν τον τελεστέο προορισμού ανάλογα με το αποτέλεσμα μιας λογικής πράξης (Set), μεταπήδησης (Jump), διακλάδωσης (Branch). Το παρακάτω σχήμα (Σχήμα 3.2.5) μας δείχνει το υψηλότερο επίπεδο του αλγοριθμικού διαγράμματος καταστάσεων για τον DLX:



**Σχήμα 3.3.5** - Τα κουτιά με ετικέτες Data transfer, ALU, Set, Jump, Branch αντιστοιχούν στο διάγραμμα καταστάσεων που εκτελεί τις αντίστοιχες εντολές. Έτσι για το κουτί που αναφέρεται σε αριθμητικές και λογικές εντολές (ALU) έχουμε το παρακάτω αλγοριθμικό διάγραμμα καταστάσεων.



Σχήμα 3.3.5

**Σχήμα 3.3.6** - Οι καταστάσεις του βήματος εκτέλεσης και του βήματος επανεγγραφής των αριθμητικών και λογικών εντολών του DLX.

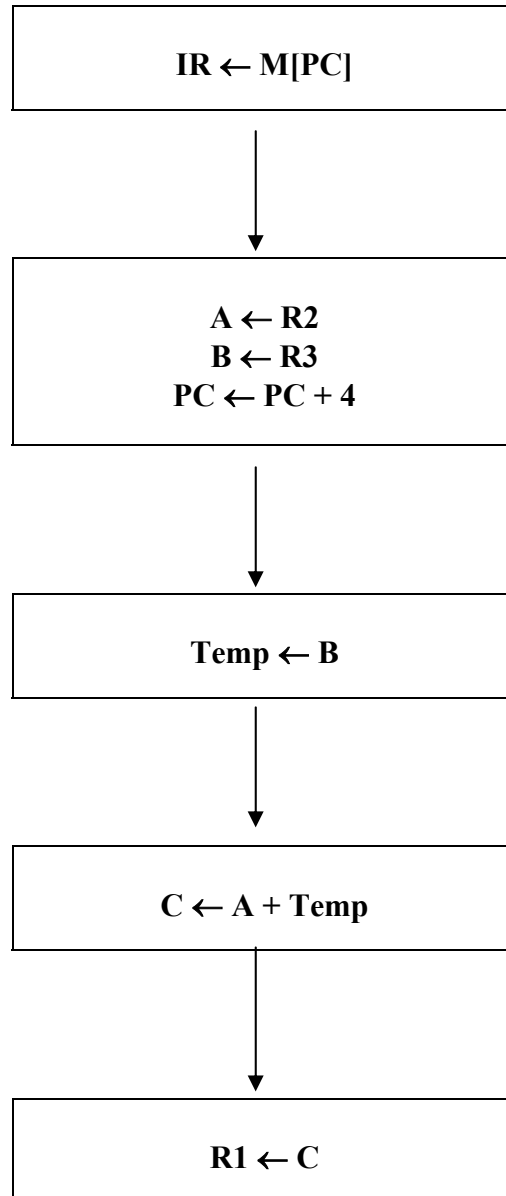


#### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Να σχεδιάσετε το αλγοριθμικό διάγραμμα καταστάσεων για την εκτέλεση της εντολής **ADD R1, R2, R3**.

**ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4**

Το αλγοριθμικό διάγραμμα καταστάσεων για την εκτέλεση της εντολής **ADD R1, R2, R3** είναι:





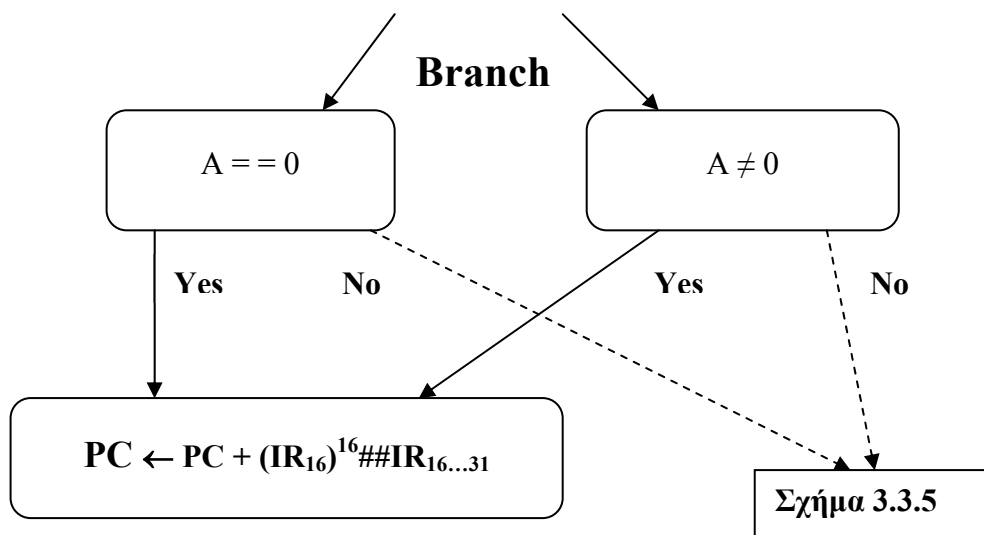
## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Να σχεδιάσετε το αλγοριθμικό διάγραμμα καταστάσεων για τις εντολές διακλάδωσης, μετά το στάδιο της αποκωδικοποίησης.



## ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Το σχήμα που ακολουθεί περιέχει τις καταστάσεις για τα υπόλοιπα βήματα εκτέλεσης των εντολών διακλάδωσης με συνθήκη. Τα δύο πρώτα βήματα, δηλαδή το στάδιο της αποκωδικοποίησης είναι ίδιο για όλους τους τύπους των εντολών. (Στο σχήμα υπάρχουν αρχικά δύο βέλη: το ένα είναι για την BEQZ και το άλλο για την BNEZ.)





## Επίδοση της καλωδιωμένης μονάδας ελέγχου του DLX.

Σκοπός των σχεδιαστών μονάδων ελέγχου είναι η μείωση του CPI, της διάρκειας του κύκλου ρολογιού και των κυκλωμάτων. Το CPI, όπως γνωρίζουμε, είναι ο μέσος αριθμός καταστάσεων που απαιτούνται κατά την εκτέλεση μιας εντολής.



### Παράδειγμα

**Ποιό είναι το CPI ενός υπολογιστή DLX που χρησιμοποιεί καλωδιωμένη μονάδα ελέγχου;**

### Απάντηση:

Ο αριθμός των κύκλων ρολογιού για την εκτέλεση κάθε εντολής του DLX καθορίζεται από το πλήθος των καταστάσεων κάθε εντολής. Ξεκινώντας από την κορυφή του αλγοριθμικού διαγράμματος καταστάσεων, κάθε εντολή σπαταλά τουλάχιστον δύο κύκλους ρολογιού, όπως φαίνεται στο Σχήμα 3.3.5. Ο ακριβής αριθμός εξαρτάται από το μέσο αριθμό προσπελάσεων της μνήμης, αφού πολλές φορές χρειάζεται να επαναληφθεί η προσπέλαση γιατί η μνήμη δεν είναι έτοιμη. Σε υπολογιστές με κρυφή μνήμη ο αριθμός των προσπελάσεων είναι 1. 🚩

Στον παρακάτω πίνακα εμφανίζονται για κάθε εντολή, ο ελάχιστος αριθμός κύκλων ρολογιού, πόσες προσβάσεις στη μνήμη απαιτούνται, ο ολικός αριθμός κύκλων ρολογιού κατά την εκτέλεση μιας εντολής, καθώς και το ποσοστό εμφάνισης κάθε εντολής.

<i>Εντολές DLX</i>	<b>Ελάχιστος αριθμός κύκλων ρολογιού</b>	<b>Προσπελάσεις μνήμης</b>	<b>Ολικός αριθμός κύκλων ρολογιού</b>	<b>Ποσοστό εμφάνισης</b>
Loads	6	2	8	21%
Stores	5	2	7	12%
ALU	5	1	6	37%
Set	6	1	7	6%
Jumps	3	1	4	2%
Jump and link	5	1	6	0%

Branch(taken)	4	1	5	12%
Branch(not taken)	3	1	4	11%

**Πίνακας 1**

Το CPI υπολογίζεται με τον παρακάτω τύπο:

$$CPI = \sum_{i=1}^n (CPI_i \times \frac{I_i}{Instruction\_count})$$

Το πηλίκο  $I_i/Instruction\_count$  είναι το ποσοστό εμφάνισης της εντολής  $i$  στο πρόγραμμα.

Το CPI για κάθε εντολή εμφανίζεται στον παρακάτω πίνακα:

Loads	$8 \times 21\% = 1.68$
Stores	$7 \times 12\% = 0.84$
ALU	$6 \times 37\% = 2.22$
Set	$7 \times 6\% = 0.42$
Jumps	$4 \times 2\% = 0.08$
Jump and link	$6 \times 0\% = 0$
Branch(taken)	$5 \times 12\% = 0.6$
Branch(not taken)	$4 \times 11\% = 0.44$
<b>Ολικό CPI : 6.28</b>	

**Πίνακας 2**

Έτσι το ολικό CPI είναι περίπου 6.3.



### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 6

Με ποιον τρόπο μπορεί να βελτιωθεί η επίδοση της καλωδιωμένης μονάδας ελέγχου του DLX; Να συγκρίνετε της απάντησή σας με την παράγραφο που ακολουθεί.



## Βελτίωση της επίδοσης της καλωδιωμένης μονάδας ελέγχου του DLX

Η επίδοση της καλωδιωμένης μονάδας ελέγχου του DLX βελτιώνεται με τη μείωση του αριθμού των καταστάσεων από τις οποίες πρέπει να περάσει μια εντολή κατά την εκτέλεσή της.

### Ελάττωση αριθμού καταστάσεων:

Αν καταργηθεί η πρώτη κατάσταση στην οποία είχαμε  $MAR \leftarrow PC$  και η δεύτερη κατάσταση τροποποιηθεί έτσι ώστε να εκτελείται η λειτουργία  $IR \leftarrow M[PC]$ , τότε ο αριθμός των καταστάσεων μειώνεται κατά 1.

Αν έχουμε αριθμητικές και λογικές εντολές μπορούμε να μειώσουμε τις καταστάσεις με έναν από τους παρακάτω τρόπους.

### Χρήση πολυπλέκτη:

Χρησιμοποιούμε έναν πολυπλέκτη MUX στην είσοδο της ALU ο οποίος επιλέγει είτε τον καταχωρητή B είτε την 16-bit απευθείας τιμή, ανάλογα με τον κωδικό λειτουργίας της εντολής που βρίσκεται στον IR. Για παράδειγμα στην εντολή ADD επιλέγεται ο καταχωρητής B, ενώ στην εντολή ADDI επιλέγεται η 16-bit απευθείας τιμή. Έτσι παραλείπεται η κατάσταση στην οποία φορτωνόταν στον Temp ο B ή η 16-bit απευθείας τιμή, μειώνοντας έτσι τον αριθμό καταστάσεων κατά 1.

### Ορισμός διαφορετικών καταστάσεων:

Ορίζουμε διαφορετικές καταστάσεις για τις αριθμητικές και λογικές εντολές που χρησιμοποιούν απευθείας τιμές ή καταχωρητές. Όπως και στην προηγούμενη μέθοδο παραλείπεται η κατάσταση στην οποία φορτώνεται στον Temp ο B ή η 16-bit απευθείας τιμή, μειώνοντας έτσι τον αριθμό καταστάσεων κατά 1.

## Ανακεφαλαιώνοντας λοιπόν ...



Η μονάδα ελέγχου είναι το πιο κρίσιμο κομμάτι κατά τη σχεδίαση ενός επεξεργαστή.



Η υλοποίηση της καλωδιωμένης μονάδας ελέγχου γίνεται με ένα συνδυαστικό κύκλωμα που μπορεί να είναι είτε με μια μνήμη ROM είτε με μια PLA και έναν καταχωρητή (ελεγκτή κατάστασης) που συνδέεται με το συνδυαστικό κύκλωμα.



Η επίδοση της μονάδας καλωδιωμένου ελέγχου βελτιώνεται μειώνοντας τον αριθμό των καταστάσεων από τις οποίες πρέπει να περάσει μια εντολή κατά την εκτέλεσή της, με τη χρήση ενός πολυπλέκτη στην είσοδο της ALU ή με τον ορισμό διαφορετικών καταστάσεων.



Η επίδοση της καλωδιωμένης μονάδας ελέγχου του DLX βελτιώνεται με τη μείωση του αριθμού των καταστάσεων από τις οποίες πρέπει να περάσει μια εντολή κατά την εκτέλεσή της.



Αν έχουμε αριθμητικές και λογικές εντολές μπορούμε να μειώσουμε τις καταστάσεις με έναν από τους παρακάτω τρόπους: χρήση πολυπλέκτη, ορισμός διαφορετικών καταστάσεων.