

## Ενότητα 4<sup>η</sup>: Μικροπρογραμματιζόμενη Μονάδα Ελέγχου.

**Σκοπός** Ο σκοπός της ενότητας αυτής είναι να παρουσιάσει τη δομή της μικροπρογραμματιζόμενης μονάδας ελέγχου και τον τρόπο με τον οποίο ομαδοποιούνται τα σήματα ελέγχου. Στη συνέχεια παρουσιάζεται η μορφή της μικροπρογραμματιζόμενης μονάδας ελέγχου στους υπολογιστές DLX και MIPS. Επίσης, γίνεται αντιστοιχία της καλωδιωμένης και μικροπρογραμματιζόμενης μονάδας ελέγχου του DLX. Στη συνέχεια υπολογίζεται η επίδοση ενός υπολογιστή DLX, καθώς και ο τρόπος για να αυξήσουμε την επίδοση αυτή και να ελαττώσουμε το κόστος της μικροπρογραμματιζόμενης μονάδας ελέγχου.

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



αναφέρετε τα κύρια χαρακτηριστικά της μικροπρογραμματιζόμενης μονάδας ελέγχου,



ομαδοποιείτε τα σήματα ελέγχου,



εξηγείτε τον τρόπο με τον οποίο δημιουργείται ένα μικροπρόγραμμα,



περιγράφετε την μικροπρογραμματιζόμενη μονάδα ελέγχου του υπολογιστή DLX και του υπολογιστή MIPS,



συσχετίζετε την καλωδιωμένη και τη μικροπρογραμματιζόμενη μονάδα ελέγχου του DLX,



υπολογίζετε την επίδοση της μικροπρογραμματιζόμενης μονάδας ελέγχου του DLX και να προτείνετε τρόπους για τη βελτίωση της επίδοσης αυτής.



Μικροπρογραμματιζόμενη μονάδα ελέγχου, ομαδοποίηση  
σημάτων ελέγχου, μικροπρόγραμμα, μικροεντολή



## Εισαγωγή στη μικροπρογραμματιζόμενη μονάδα ελέγχου

Η βασική ιδέα για την υλοποίηση της μικροπρογραμματιζόμενης μονάδας ελέγχου είναι η μετατροπή της μονάδας ελέγχου σε έναν υπολογιστή - μικρογραφία. Οι μικροεντολές καθορίζουν τα σήματα ελέγχου που στέλνονται στη δίοδο δεδομένων και αποφασίζουν την επόμενη προς εκτέλεση εντολή. Όπως δηλώνει το όνομα "μικροπρογραμματισμός", από τη στιγμή που έχουν σχεδιαστεί η δίοδος δεδομένων του επεξεργαστή και η μνήμη για τις μικροεντολές, ο έλεγχος γίνεται ουσιαστικά ένα έργο προγραμματισμού.

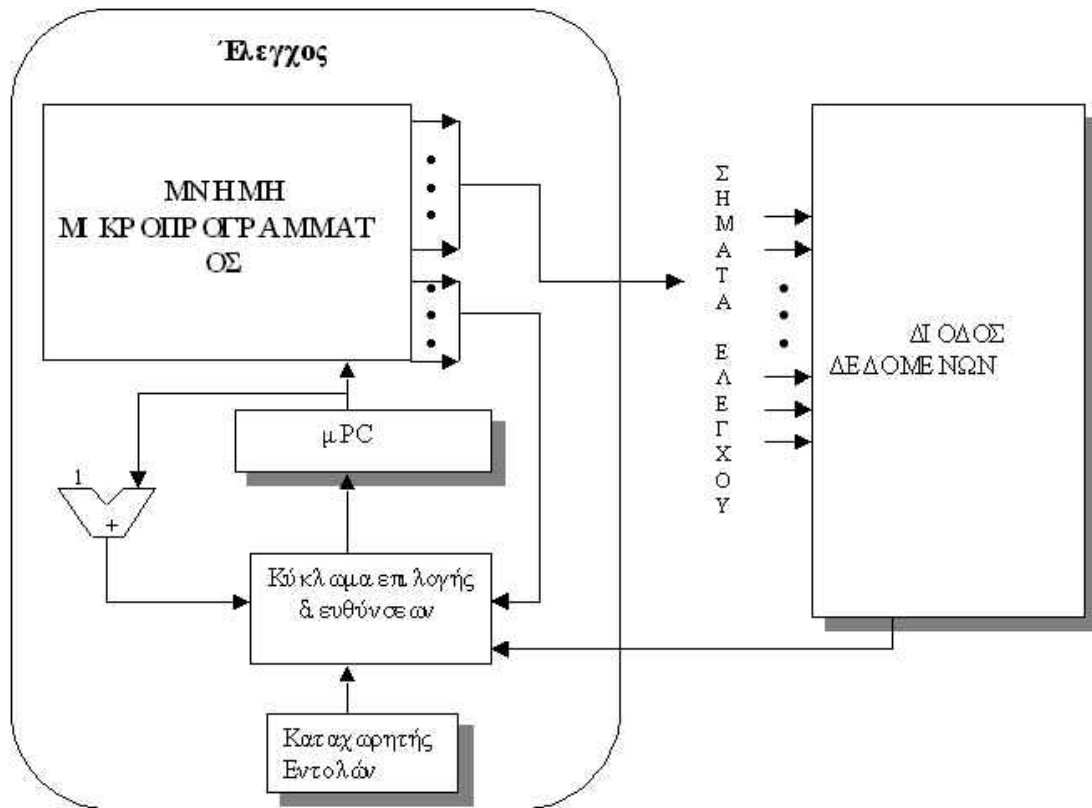
Το μικροπρογράμματα αποτελούνται από μικροεντολές. Υποθέτοντας ότι οι μικροεντολές θα εκτελούνται σειριακά από διαδοχικές διευθύνσεις της μνήμης μικροπρογραμματισμού (καλείται επίσης και μνήμη ελέγχου), είναι φυσικό να χρησιμοποιηθεί ένας απαριθμητής μικροπρογράμματος (μPC) που θα παρακολουθεί τις διευθύνσεις αυτές. Ο μPC θα αυξάνεται κατά ένα, ώστε να ανακληθεί η επόμενη μικροεντολή. Επειδή κάθε εντολή υπολογιστή εκτελείται από ένα αντίστοιχο μικροπρόγραμμα, πρέπει να προσδιοριστεί μία αρχική διεύθυνση για την υπορουτίνα, σε συνάρτηση με το περιεχόμενο του καταχωρητή εντολών. Η δυνατότητα για διακλάδωση μέσα στο μικροπρόγραμμα παρέχεται από μία γεννήτρια διευθύνσεων διακλάδωσης, η οποία παρέχει την ακολουθία αυξήσεων που εκτελεί ο μPC.



### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

**Να προσδιορίσετε το ρόλο των μικροεντολών σε ένα μικροπρόγραμμα, καθώς και τον τρόπο με τον οποίο επιλέγεται η επόμενη εντολή που πρόκειται να εκτελεστεί. Να συγκρίνετε την απάντησή σας με την προηγούμενη παράγραφο.**

Στο παρακάτω σχήμα εμφανίζεται η δομή της μικροπρογραμματιζόμενης μονάδας ελέγχου.



**Σχήμα 3.4.1 - Δομή μονάδας μικροπρογραμματιζόμενου ελέγχου.** Σε αντίθεση με το σχήμα 3.3.2, στο σχήμα αυτό υπάρχει ένα κύκλωμα επιλογής διευθύνσεων και ένα κύκλωμα για την αύξηση του μPC, τα οποία υπολογίζουν τη διεύθυνση της επόμενης μικροεντολής. Υπάρχουν δύο τρόποι για την επιλογή της επόμενης μικροεντολής: χρήση του μPC, όπως φαίνεται στο σχήμα, ή η κάθε εντολής να περιέχει τη διεύθυνση της επόμενης μικροεντολής. Η μικροπρογραμματιζόμενη μνήμη μερικές φορές την ονομάζουμε ROM, επειδή οι περισσότεροι σύγχρονοι υπολογιστές χρησιμοποιούν τη ROM για τις καταστάσεις της μονάδας ελέγχου.



## Ομαδοποίηση σημάτων ελέγχου

Από τη στιγμή που το υλικό του υπολογιστή δεν επηρεάζεται από το πώς είναι ομαδοποιημένες οι γραμμές ελέγχου σε μία μικροεντολή, οι γραμμές ελέγχου που επιτελούν παραπλήσιες λειτουργίες τοποθετούνται δίπλα για ευκολία στην κατανόηση. Οι ομάδες σχετιζόμενων γραμμών ελέγχου καλούνται *πεδία*.

Παρακάτω παρουσιάζεται ένα παράδειγμα μικροεντολής με 8 πεδία:

Destination	ALU operation	Source1	Source2	Constant	Misc	Cond	Jump Address
-------------	---------------	---------	---------	----------	------	------	--------------

**Σχήμα 3.4.2 - Ομαδοποιημένα σήματα ελέγχου.** Τα Destination, ALU operation, Source1, Source2, Constant, Misc, Cond, Jump Address αποτελούν τα πεδία της μικροεντολής.



Μερικοί υπολογιστές αφιερώνουν ένα πεδίο σε κάθε μικροεντολή για τη διεύθυνση της επόμενης μικροεντολής, αντί να χρησιμοποιούν ένα μετρητή μικροπρογράμματος για να παρέχει την επόμενη μικροεντολή.



### Παράδειγμα

Στα δύο παρακάτω σχήματα παρουσιάζεται η διαφορά μεταξύ των μικροεντολών που χρησιμοποιούν μη ομαδοποιημένα σήματα ελέγχου και των μικροεντολών με ομαδοποιημένα σήματα ελέγχου που προορίζονται για την ALU της διόδου δεδομένων.

### Μη ομαδοποιημένα σήματα ελέγχου

Destination	A D D	S U B	A D D	O R	X O R	S L L	S R L	S L L	S R A	R S U B	Pass S1	Pass S2	Source1	Source2	Constant	Misc	Cond	Jump Address
...	1	0	0	0	0	0	0	0	0	0	0	0	...	...	...	...	...	...

**Σχήμα 3.4.3 - Μη ομαδοποιημένα σήματα ελέγχου.** Τα σήματα αυτά καθορίζουν τι πράξη θα εκτελέσει η ALU, που στο συγκεκριμένο παράδειγμα είναι μια πράξη πρόσθεσης. Οι αριθμητικές και λογικές εντολές παρουσιάζονται στο Σχήμα 3.4.7 της επόμενης υποενότητας.


Όπως παρατηρούμε, το bit που αντιστοιχεί στην εντολή ADD είναι 1, ενώ τα υπόλοιπα είναι 0 (αυτό είναι φυσικό αφού η ALU μπορεί να εκτελέσει μόνο μια απ' αυτές τις εντολές). Το σήμα ελέγχου στέλνεται στη δίοδο δεδομένων και ενεργοποιεί την πράξη της πρόσθεσης στην ALU.

#### Ομαδοποιημένα σήματα ελέγχου

Οι αριθμητικές και λογικές εντολές είναι 12, έτσι το  $2^0$  πεδίο της μικροεντολής του σχήματος 3.4.2 θα αποτελείται από 4 bits (αφού  $2^4=16$ , επομένως μπορούν να κωδικοποιηθούν μέχρι 16 εντολές). Έτσι μια εντολή πρόσθεσης (ADD) μπορεί να κωδικοποιηθεί με τα bits: 0000, ενώ μια εντολή αφαίρεσης (SUB) μπορεί να κωδικοποιηθεί με τα bits: 0001 κ.ο.κ.

Έτσι αν η μικροεντολή έχει την παρακάτω μορφή, το σήμα ελέγχου που στέλνεται στη δίοδο δεδομένων ενεργοποιεί την πράξη της αφαίρεσης στην ALU.

Destination	0001	Source1	Source2	Constant	Misc	Cond	Jump Address
-------------	------	---------	---------	----------	------	------	--------------

**Σχήμα 3.4.4 - Ομαδοποιημένα σήματα ελέγχου.** Τα σήματα αυτά καθορίζουν τι πράξη θα εκτελέσει η ALU, που στο συγκεκριμένο παράδειγμα είναι μια πράξη αφαίρεσης. Οι αριθμητικές και λογικές εντολές παρουσιάζονται στο Σχήμα 3.4.7 της επόμενης υποενότητας. 



## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Μπορείτε να εξηγήσετε το λόγο για τον οποίο ομαδοποιούνται τα σήματα ελέγχου; Τι κόστος υπάρχει από την ομαδοποίηση αυτή;



## ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 2

Με την ομαδοποίηση των σημάτων ελέγχου πετυχαίνουμε μείωση του μεγέθους της μικροεντολής αυξάνοντας όμως την πολυπλοκότητα της αποκωδικοποίησης των πεδίων της.



## Μικροπρογραμματιζόμενη μονάδα ελέγχου του DLX.

Πριν ξεκινήσουμε με τον μικροπρογραμματισμό θα πρέπει να καθορίσουμε από τι μπορεί να αποτελούνται τα πεδία μιας μικροεντολής. Το πρώτο βήμα είναι να δημιουργηθεί μια λίστα με τις πιθανές εισόδους. Στα παρακάτω σχήματα παρουσιάζεται η λίστα αυτή, η οποία περιλαμβάνει τον τελεστέο προορισμού (Destination), τον πηγαίο τελεστέο1 (Source1), τον πηγαίο τελεστέο2 (Source2), το πεδίο που καθορίζει τι πράξη κάνει η ALU, το πεδίο misc και το πεδίο cond.

Destination
(None)
C
Temp
PC
IAR
MAR
MDR

**Σχήμα 3.4.5** - Στον πίνακα αυτό παρουσιάζονται οι πιθανοί τελεστέοι προορισμού μιας μικροεντολής (το πεδίο Destination της μικροεντολής του σχήματος 3.4.2).

Source1/Source2
A/B
Temp
PC
IAR
MAR
MDR
IR(16-bit imm)
IR(26-bit imm)
Constant

**Σχήμα 3.4.6** - Στον πίνακα αυτό παρουσιάζονται οι πιθανοί πηγαίοι τελεστές μιας μικροεντολής (τα πεδία Source1 και Source2 της μικροεντολής του σχήματος 3.4.2). Το IR(16-bit imm) και το IR(26-bit imm) σημαίνουν ότι ο πηγαίος τελεστής είναι μια 16-bit ή μια 26-bit αντίστοιχα απευθείας τιμή. Το Constant σημαίνει ότι ο πηγαίος τελεστής είναι μια σταθερά, στην περίπτωση αυτή στο πεδίο C της μικροεντολής υπάρχει η τιμή της σταθεράς.

ALU
ADD
SUB
RSUB (reverse subtract)
AND
OR
XOR
SLL
SRL
SRA
Pass S1
Pass S2

**Σχήμα 3.4.7** - Στον πίνακα αυτό παρουσιάζεται το πεδίο ALU της μικροεντολής του σχήματος 3.4.2. Σε αυτό το πεδίο φαίνεται τι πράξη θα κάνει η ALU. Η πράξη RSUB σημαίνει αντίστροφη αφαίρεση, δηλαδή αντί για την αφαίρεση A-B κάνει την αφαίρεση B-A. Το Pass S1 και Pass S2 δηλώνουν ότι η ALU θα αφήσει να περάσουν τα δεδομένα που έρχονται από την αρτηρία S1 και S2 αντίστοιχα χωρίς να εκτελέσει καμία πράξη σ' αυτά.

Misc	
Instr Read	$IR \leftarrow M[PC]$
Data Read	$MDR \leftarrow M[MAR]$
Write	$M[MAR] \leftarrow MDR$

AB← RF	<i>Load A&amp;B from Reg. File</i>
Rd← C	<i>Write Rd</i>
R31← C	<i>Write R31</i>

**Σχήμα 3.4.8** - Στον πίνακα αυτό παρουσιάζεται το πεδίο Misc της μικροεντολής του σχήματος 3.4.2. Στα αριστερά υπάρχει ο συμβολισμός που χρησιμοποιείται στην μικροεντολή και στα δεξιά υπάρχει η εξήγηση του συμβολισμού. Το Instr Read σημαίνει ότι η εντολή διαβάζεται από τη μνήμη. Το Data Read σημαίνει ότι καταχωρούνται τα δεδομένα που έχουν διαβαστεί από τη μνήμη στον MDR (Load). Το Write σημαίνει ότι αποθηκεύεται στη μνήμη το δεδομένο που ήταν καταχωρημένο στον MDR (Store). Το AB←RF σημαίνει ότι διαβάζονται οι καταχωρητές από το αρχείο καταχωρητών.

<b>Cond</b>	
---	<i>Πήγαινε στην επόμενη μικροεντολή</i>
Uncond	<i>Πάντα μεταπήδηση</i>
Int?	<i>Έλεγχος για διακοπή</i>
Zero?	<i>Είναι η έξοδος της ALU μηδέν;</i>
Mem?	<i>Έχει ολοκληρωθεί η πρόσβαση στη μνήμη;</i>
Negative?	<i>Είναι η έξοδος της ALU αρνητική;</i>
Load?	<i>Είναι μικροεντολή φόρτωσης;</i>
Decode1	<i>Ο πίνακας διευθύνσεων 1 καθορίζει την επόμενη μικροεντολή</i>
Decode2	<i>Ο πίνακας διευθύνσεων 2 καθορίζει την επόμενη μικροεντολή</i>
Decode3	<i>Ο πίνακας διευθύνσεων 3 καθορίζει την επόμενη μικροεντολή</i>

**Σχήμα 3.4.9** - Στον πίνακα αυτό παρουσιάζεται το πεδίο cond της μικροεντολής του σχήματος 3.4.2. Στα αριστερά υπάρχει ο συμβολισμός που χρησιμοποιείται στην μικροεντολή και στα δεξιά υπάρχει η εξήγηση του συμβολισμού. Αν δεν υπάρχει τίποτα στο πεδίο cond τότε εκτελείται η επόμενη κατά σειρά μικροεντολή. Το Uncond σημαίνει ότι η επόμενη μικροεντολή θα είναι εκείνη που καθορίζεται στο πεδίο Jump label. Τα Decode1, Decode2, Decode3 δηλώνουν ότι η επόμενη μικροεντολή καθορίζεται από τους πίνακες διευθύνσεων 1,2 και 3 αντίστοιχα. Οι πίνακες αυτοί παρουσιάζονται παρακάτω.

Κωδικός λειτουργίας εντολής (συμβολική αναπαράσταση)	Διεύθυνση	Label
Memory	5	Mem:
Move to special	20	MovI2S:
Move from special	21	MovS2I:
S2=B	23	Reg:
S2=Immediate	24	Imm:
Branch equal zero	50	Beq:
Branch not equal zero	52	Bne:
Jump	54	Jump:
Jump register	55	JReg:
Jump and link	56	JAL:
Jump and link register	58	JALR:
Trap	60	Trap:

**Πίνακας Διευθύνσεων 1** - Ο πίνακας αυτός καθορίζει ποια θα είναι η επόμενη μικροεντολή ανάλογα με τον κωδικό λειτουργίας της εντολής. Στην αριστερή στήλη παρουσιάζεται ο κωδικός λειτουργίας της εντολής, στη μεσαία στήλη η διεύθυνση της μικροεντολής που θα εκτελεστεί ανάλογα με τον κωδικό λειτουργίας της εντολής και στη δεξιά στήλη βρίσκεται ο τίτλος (label) της μικροεντολής που θα εκτελεστεί. Έτσι αν για παράδειγμα ο κωδικός λειτουργίας της εντολής δηλώνει ότι η εντολή είναι μια εντολή μεταπήδησης (Jump) τότε η επόμενη μικροεντολή βρίσκεται στη διεύθυνση 54 και έχει τίτλο (label) Jump: .

Κωδικός λειτουργίας εντολής (συμβολική αναπαράσταση)	Διεύθυνση	Label
Load byte	11	LB:
Load byte unsigned	13	LBU:
Load half	15	LH:
Load half unsigned	17	LHU:
Load word	19	LW:
ADD	25	ADD/I:
SUB	26	SUB/I:
AND	27	AND/I:
OR	28	OR/I:
XOR	29	XOR/I:
SLL	30	SLL/I:
SRL	31	SRL/I:
SRA	32	SRA/I:
LHI	33	LHI:
Set equal	35	SEQ/I:
Set not equal	37	SNE/I:
Set less than	39	SLT/I:
Set greater than or equal	41	SGE/I:
Set greater than	43	SGT/I:
Set less than or equal	45	SLE/I:

**Πίνακας Διευθύνσεων 2 και 3** - Ο πίνακας αυτός καθορίζει ποια θα είναι η επόμενη μικροεντολή ανάλογα με τον κωδικό λειτουργίας της εντολής. Στην αριστερή στήλη παρουσιάζεται ο κωδικός λειτουργίας της εντολής, στη μεσαία στήλη η διεύθυνση της μικροεντολής που θα εκτελεστεί ανάλογα με τον κωδικό λειτουργίας της εντολής και στη δεξιά στήλη βρίσκεται ο τίτλος (label) της μικροεντολής που θα εκτελεστεί. Έτσι αν για παράδειγμα ο κωδικός λειτουργίας της εντολής δηλώνει ότι η εντολή είναι μια εντολή πρόσθεσης (ADD) τότε η επόμενη μικροεντολή βρίσκεται στη διεύθυνση 25 και έχει τίτλο (label) ADD/I: .



## Αντιστοιχία καλωδιωμένης και μικροπρογραμματιζόμενης μονάδας ελέγχου.

Ο μικροπρογραμματιζόμενος έλεγχος περιλαμβάνει έναν απεριθμητή μικροπρογράμματος για να καθορίσει τη διεύθυνση της επόμενης μικροεντολής. Στη συνέχεια θα δούμε την αντιστοιχία μεταξύ διαγραμμάτων καταστάσεων του καλωδιωμένου ελέγχου και των μικροεντολών του μικροπρογραμματιζόμενου ελέγχου του DLX. Στο Σχήμα 3.4.7 υπάρχει το πρώτο τμήμα του μικροπρογράμματος που αντιστοιχεί στο διάγραμμα καταστάσεων του Σχήματος 3.3.5 της ενότητας 3 του κεφαλαίου.

Loc	Label	Dest	ALU	S1	S2	C	Misc	Cond	Jump label	Σχόλια
0	Ifetch							Interrupt?	Intrpt	Έλεγχος για διακοπή
1	Iloop						Instr Read	Mem?	Iloop	$IR \leftarrow M[PC]$
2		PC	ADD	PC	Constant	4	$AB \leftarrow RF$	Decode1		
3	Intrpt:	IAR	Pass S1	PC						Διακοπή
4		PC	Pass S2		Constant	0		Uncond	Ifetch	$PC \leftarrow 0$ και πηγαίνει στην επόμενη εντολή

**Σχήμα 3.4.10** - Η πρώτη μικροεντολή (με διεύθυνση 0) μεταφέρει τη ροή του μικροπρογράμματος στην μικροεντολή με διεύθυνση (Loc.) 3, εάν υπάρχει διακοπή, αλλιώς συνεχίζει με την μικροεντολή 1. Με την μικροεντολή 3 αποθηκεύεται το περιεχόμενο του PC (που βρίσκεται στην αρτηρία S1) στον IAR (Dest) μέσω της ALU. Το πεδίο Misc της μικροεντολής 1 δηλώνει ότι η εντολή διαβάζεται από τη μνήμη, το πεδίο Cond δηλώνει ότι αν η πρόσβαση στη μνήμη δεν ολοκληρώθηκε θα επαναληφθεί η μικροεντολή 1 (Iloop), αλλιώς θα εκτελεστεί η επόμενη μικροεντολή. Έτσι η μικροεντολή 1 ανακαλεί μια εντολή από τη μνήμη και επαναλαμβάνεται αν η πρόσβαση στην μνήμη δεν έχει ολοκληρωθεί. Η μικροεντολή 2 αυξάνει τον PC κατά 4 και φορτώνει στους A και B τιμές από το αρχείο καταχωρητών. Η διεύθυνση της επόμενης εντολής εξαρτάται τώρα από την εντολή που βρίσκεται στον IR. Η μικροεντολή 2 αντιστοιχεί στην 2<sup>η</sup> κατάσταση του διαγράμματος του σχήματος 3.2.5. Στο βήμα αυτό οι A και B φορτώνονται με τιμές από το αρχείο καταχωρητών

(Reg. File), όπως δηλώνει το πεδίο Misc της μικροεντολής. Στο ίδιο βήμα ο PC αυξάνεται κατά 4. Αυτό γίνεται ως εξής: ο PC που βρίσκεται στην αρτηρία S1 και η σταθερά (Constant) 4 που βρίσκεται στην αρτηρία S2 προσθέτονται στην ALU (ADD) και το αποτέλεσμα καταχωρείται στον PC (Dest). Τέλος η διεύθυνση της επόμενης μικροεντολής λαμβάνεται από τον πίνακα διευθύνσεων 1 (Decode1), και εξαρτάται από τον κωδικό λειτουργίας της εντολής που βρίσκεται στον IR. Έτσι αν για παράδειγμα έχουμε μια εντολή μεταπήδησης (Jump) τότε σύμφωνα με τον Πίνακα διευθύνσεων 1 η διεύθυνση της επόμενης μικροεντολής είναι 54 και έχει τίτλο (label) "Jump:" .



## **Η μικροπρογραμματιζόμενη μονάδα ελέγχου για αριθμητικές και λογικές εντολές**

Οι αριθμητικές και λογικές εντολές βρίσκονται στο Σχήμα 3.4.11. Οι δύο πρώτες μικροεντολές αντιστοιχούν στις καταστάσεις που βρίσκονται στην κορυφή του Σχήματος 3.3.6 της ενότητας 3.

Όπως φαίνεται στη μικροεντολή 2 του σχήματος 3.4.10 της προηγούμενης υποενότητας, η διεύθυνση της επόμενης μικροεντολής καθορίζεται από τον Πίνακα Διευθύνσεων 1. Έτσι αν η εντολή είναι μια αριθμητική και λογική εντολή πρέπει να καταχωρηθεί στον Temp η τιμή του B ή η απευθείας τιμή (αν ο κωδικός λειτουργίας δηλώνει ότι ο πηγαίος τελεστέος είναι απευθείας δεδομένο). Όπως βλέπουμε από τον Πίνακα διευθύνσεων αν ο κωδικός λειτουργίας της εντολής δηλώνει ότι ο πηγαίος τελεστέος βρίσκεται στον καταχωρητή B τότε η επόμενη μικροεντολή έχει διεύθυνση 23 και τίτλο (label) "Reg:", ενώ αν ο πηγαίος τελεστέος είναι απευθείας δεδομένο τότε η επόμενη μικροεντολή έχει διεύθυνση 24 και τίτλο (label) "Imm:".

Στο Σχήμα 3.4.11 παρουσιάζεται η σειρά των μικροεντολών για την εκτέλεση αριθμητικών και λογικών εντολών μετά την αποκωδικοποίηση (έχουν προηγηθεί οι μικροεντολές του σχήματος 3.4.10):

# Αρχιτεκτονική Υπολογιστών I

Loc.	Label	Dest	ALU	S1	S2	C	Misc	Cond	Jump label	Σχόλια
23	Reg	Temp	Pass S2		B			Decode2		Source2=reg
24	Imm	Temp	Pass S2		Imm			Decode3		Source2=imm
25	ADD/I	C	ADD	A	Temp			Uncond	Write2	ADD
26	SUB/I	C	SUB	A	Temp			Uncond	Write2	SUB
27	AND/I	C	AND	A	Temp			Uncond	Write2	AND
28	OR/I	C	OR	A	Temp			Uncond	Write2	OR
29	XOR/I	C	XOR	A	Temp			Uncond	Write2	XOR
30	SLL/I	C	SLL	A	Temp			Uncond	Write2	SLL
31	SRL/I	C	SRL	A	Temp			Uncond	Write2	SRL
32	SRA/I	C	SRA	A	Temp			Uncond	Write2	SRA
33	LHI	C	SLL	Temp	Constant	16		Uncond	Write2	LHI
34	Write2						Rd←C	Uncond	Ifetch	Επανεγγραφή και πήγαινε στην επόμενη εντολή

**Σχήμα 3.4.11** - Όπως στις δύο πρώτες καταστάσεις του σχήματος 3.2.6 της υποενότητας 2, οι μικροεντολές 23 και 24 φορτώνουν στον Temp έναν πηγαίο τελεστέο και στη συνέχεια δείχνουν στην επόμενη μικροεντολή, που εξαρτάται από τον κωδικό λειτουργίας της εντολής. Η μικροεντολή 23 φορτώνει στον Temp το περιεχόμενο του καταχωρητή B, ενώ η εντολή 24 φορτώνει στον Temp την απευθείας τιμή. Αυτό γίνεται ως εξής: ο πηγαίος τελεστέος (B ή Imm) που βρίσκεται στην αρτηρία S2 περνά στον Temp μέσω της ALU. Η ALU δεν εκτελεί καμία πράξη με τον τελεστέο αυτό απλά τον αφήνει να περάσει (αυτό σημαίνει ο συμβολισμός Pass S2). Για να εξοικονομηθεί χώρος χρησιμοποιείται η ίδια μικροεντολή για τελεστέο που είναι καταχωρητής και για τελεστέο που είναι απευθείας τιμή. Η επόμενη μικροεντολή καθορίζεται από τον Πίνακα Διευθύνσεων 2 και 3 και εξαρτάται από τον κωδικό λειτουργίας της εντολής. Έτσι αν η εντολή είναι μια εντολή πρόσθεσης (ADD) τότε σύμφωνα με τον Πίνακα Διευθύνσεων 2 η επόμενη μικροεντολή έχει διεύθυνση 25 και τίτλο (label) "ADD/I". Ο Temp που βρίσκεται στην αρτηρία S2 και ο A που βρίσκεται στην αρτηρία S1 περνούν στην ALU και γίνεται πρόσθεση (ADD). Το αποτέλεσμα καταχωρείται στον C όπως φαίνεται στο πεδίο Dest της μικροεντολής. Η επόμενη μικροεντολή (χωρίς κανένα έλεγχο-Uncond) έχει τίτλο (label) "Write2" και διεύθυνση 34. Η μικροεντολή αυτή καταχωρεί το περιεχόμενο του C στον Rd. Η επόμενη μικροεντολή έχει τίτλο "Ifetch", δηλαδή ανακαλείται η επόμενη εντολή του προγράμματος.



### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Να εξηγήσετε τον τρόπο με τον οποίο εκτελούνται οι εντολές στο παρακάτω τμήμα μικροκώδικα.

Loc.	Label	Dest	ALU	S1	S2	C	Misc	Cond	Jump label
50	Beq:		SUB	A	Constant	0		0?	Branch
51								Uncond	Ifetch
52	Bne:		SUB	A	Const	0		0?	Branch
53	Branch:	PC	ADD	PC	Imm16			Uncond	Ifetch
54	Jump:	PC	ADD	PC	Imm26			Uncond	Ifetch
55	JReg:	PC	Pass S1	A				Uncond	Ifetch
56	JAL:	C	Pass S1	PC					
57		PC	ADD	PC	Imm26		R31← C	Uncond	Ifetch
58	JALR:	C	Pass S1	PC					
59		PC	Pass S1	A			R31← C	Uncond	Ifetch
60	Trap:	IAR	Pass S1	PC					
61		PC	Pass S2		Imm26				



### ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 3

Το τμήμα του μικροκώδικα αντιστοιχεί στις εντολές διακλάδωσης (Branch) και μεταπήδησης (Jump) του DLX:

Η μικροεντολή 50 αντιστοιχεί στην εντολή Branch equal zero (Διακλάδωση εάν ίσο με μηδέν). Εάν ο A είναι ίσος με μηδέν τότε η επόμενη μικροεντολή είναι η μικροεντολή 53 με τίτλο "Branch:", αλλιώς η επόμενη μικροεντολή είναι η μικροεντολή με τίτλο "Ifetch:". Ο έλεγχος για το αν ο A είναι 0 γίνεται ως εξής: από τον A που βρίσκεται στην αρτηρία S1 αφαιρείται το 0 που βρίσκεται στην αρτηρία S2. Η πράξη της αφαίρεσης γίνεται στην ALU (SUB). Σύμφωνα με το πεδίο Cond της μικροεντολής αν το αποτέλεσμα της αφαίρεσης είναι 0 τότε η επόμενη μικροεντολή είναι η μικροεντολή 53 με τίτλο "Branch:", αλλιώς η επόμενη μικροεντολή είναι η 51. Η μικροεντολή 51 δηλώνει ότι η επόμενη μικροεντολή είναι η μικροεντολή με τίτλο "Ifetch:". Σύμφωνα με τη μικροεντολή 53 προστίθεται στον PC (που βρίσκεται στην αρτηρία S1) η απευθείας τιμή imm16 (που βρίσκεται στην αρτηρία S2) ώστε να υπολογιστεί ο στόχος της διακλάδωσης. Το αποτέλεσμα της πρόσθεσης που γίνεται στην ALU (ADD) καταχωρείται στον PC (Dest) και στη συνέχεια γίνεται ανάκληση της εντολής που δείχνει ο PC, αφού το πεδίο Jump label δηλώνει ότι η επόμενη μικροεντολή είναι η μικροεντολή με τίτλο "Ifetch:" που κάνει ανάκληση εντολής. Παρόμοια μπορούμε να εξηγήσουμε και την εντολή Branch if Not Equal (BNE).

Η μικροεντολή 56 είναι μια εντολή Jump And Link. Αρχικά το περιεχόμενο του PC που βρίσκεται στην αρτηρία S1 περνά στον καταχωρητή C (Dest) μέσω της ALU. Η ALU δεν εκτελεί καμία πράξη στο περιεχόμενο του PC (Pass S1). Η επόμενη μικροεντολή είναι η μικροεντολή 57. Τώρα υπολογίζεται ο στόχος της μεταπήδησης. Στον PC (που βρίσκεται στην αρτηρία S1) προστίθεται η απευθείας τιμή Imm26 (που βρίσκεται στην αρτηρία S2) ώστε να υπολογιστεί ο στόχος. Η πρόσθεση γίνεται στην ALU (ADD) και το αποτέλεσμα καταχωρείται στον PC (Dest). Το πεδίο Misc της μικροεντολής δηλώνει ότι στον καταχωρητή R31 καταχωρείται η τιμή του C. Η επόμενη μικροεντολή έχει τίτλο "Ifetch:", δηλαδή ανακαλείται η επόμενη εντολή με διεύθυνση το στόχο της μεταπήδησης. Με ίδιο τρόπο μπορούν να εξηγηθούν και οι υπόλοιπες μικροεντολές.



## Επίδοση της μικροπρογραμματιζόμενης μονάδας ελέγχου του DLX.

Αρχικά πρέπει να υπολογίσουμε πόσους κύκλους ρολογιού απαιτεί κάθε εντολή. Όλες οι εντολές εκτελούν τις μικροεντολές 0,1 και 2 του σχήματος 3.4.10, οι οποίες απαιτούν 3 κύκλους ρολογιού για κάθε εντολή συν τους κύκλους καθυστέρησης (λόγω μη επιτυχών προσβάσεων στη μνήμη). Οι κύκλοι ρολογιού που απαιτούνται για τα υπόλοιπα βήματα εκτέλεσης μιας αριθμητικής ή λογικής εντολής είναι όσες και οι καταστάσεις του Σχήματος 3.2.6, δηλαδή τρεις. Οι κύκλοι ρολογιού που απαιτούνται για τα υπόλοιπα βήματα εκτέλεσης των εντολών ανάλογα με την κατηγορία εντολής είναι:

Εντολές	Κύκλοι ρολογιού
Stores	4 + καθυστέρηση
Load word	5 + καθυστέρηση
Load byte ή Load half	6 + καθυστέρηση
ALU	3
Set	4
Branch equal zero	2
Branch not equal (taken)	2
Branch not equal (not taken)	1
Jumps	1
Jump and link	2

Οι καθυστερήσεις στις εντολές Store, Load word, Load byte ή half οφείλονται σε ανεπιτυχείς προσβάσεις στη μνήμη. Αν κατά μέσο όρο χρειάζεται ένας κύκλος ρολογιού για τις καθυστερήσεις που οφείλονται σε ανεπιτυχείς προσβάσεις στη μνήμη τότε το CPI που υπολογίζεται από τον τύπο:

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

είναι 7.68. Το πηλίκο  $I_i / Instruction\_count$  είναι το ποσοστό εμφάνισης της  $i$  εντολής σε ένα πρόγραμμα. Παρατηρούμε ότι είναι μεγαλύτερο από αυτό του καλωδιωμένου ελέγχου, γιατί ο έλεγχος για διακοπές απαιτεί στο ξεκίνημα έναν κύκλο ρολογιού, οι φορτώσεις και οι αποθηκεύσεις είναι πιο αργές και οι διακλαδώσεις που απαιτούν

σύγκριση με το 0 είναι πιο αργές στην περίπτωση που είναι διακλαδώσεις που δεν ακολουθούνται (untaken).



#### ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Να προσδιορίσετε τους λόγους για τους οποίους το CPI της μικροπρογραμματιζόμενης μονάδας ελέγχου είναι μεγαλύτερο από αυτό της καλωδιωμένης.



#### ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

Το CPI της μικροπρογραμματιζόμενης μονάδας ελέγχου είναι μεγαλύτερο από αυτό της καλωδιωμένης. Αυτό συμβαίνει επειδή ο έλεγχος για διακοπές απαιτεί ξεκινώντας έναν κύκλο ρολογιού. Οι εντολές φόρτωσης και οι εντολές αποθήκευσης είναι πιο αργές. Επίσης, οι εντολές διακλάδωσης απαιτούν σύγκριση με το 0 είναι πιο αργές στην περίπτωση που είναι εντολές διακλάδωσης που δεν ακολουθούνται (untaken).



#### Μέγεθος μικροεντολών του DLX

Το μέγεθος μιας ολοκληρωμένης μη κωδικοποιημένης έκδοσης μικροεντολής του DLX υπολογίζεται από τον αριθμό των εισόδων των Σχημάτων 3.4.5, 3.4.6, 3.4.7, 3.4.8, 3.4.9 συν το μέγεθος του πεδίου σταθεράς (Constant) και του πεδίου της διεύθυνσης μεταπήδησης (Jump address). Η μεγαλύτερη σταθερά είναι 24 και απαιτεί 5 bits για την κωδικοποίηση (χρησιμοποιείται για πράξεις σε byte), και η μεγαλύτερη διεύθυνση είναι 61 και απαιτεί 16 (προσδιορίζεται από το σύνολο των μικροεντολών)bits. Το παρακάτω σχήμα δείχνει τα πεδία των μικροεντολών και το πλήθος των bits που απαιτούνται για τις δύο μορφές των μικροεντολών: κωδικοποιημένη (Encoded), μη κωδικοποιημένη (Unencoded). Με την κωδικοποιημένη μορφή απαιτείται ο μισός χώρος αποθήκευσης.

	<b>Dest</b>	<b>ALU</b>	<b>S1</b>	<b>S2</b>	<b>Constant</b>	<b>Misc</b>	<b>Cond</b>	<b>Jump address</b>	<b>Total</b>
<b>Μη κωδικοποιημένη</b>	7	11	9	9	5	6	10	6	=63 bits
<b>Κωδικοποιημένη</b>	3	4	4	4	5	3	4	6	=33 bits



## Μικροπρογραμματιζόμενη μονάδα ελέγχου του MIPS

Για να υλοποιήσουμε τη μικροπρογραμματιζόμενη μονάδα ελέγχου του υπολογιστή MIPS, σχεδιάζουμε ένα μικροπρόγραμμα το οποίο υλοποιεί τις εντολές του MIPS. Το μικροπρόγραμμα αυτό είναι μία συμβολική αναπαράσταση της μονάδας ελέγχου και μεταφράζεται από ένα άλλο πρόγραμμα από τη μονάδα ελέγχου.

Η κάθε μικροεντολή αποτελείται από 8 πεδία. Στο σχήμα 3.4.12 φαίνονται τα πεδία των εντολών και η λειτουργία του κάθε πεδίου.

Όνομα πεδίου	Λειτουργία πεδίου
Μονάδα ελέγχου της ALU	Καθορίζει τη λειτουργία της ALU κατά τη διάρκεια του κύκλου ρολογιού.
SRC1	Καθορίζει τον πρώτο πηγαίο τελεστέο της ALU.
SRC2	Καθορίζει το δεύτερο πηγαίο τελεστέο της ALU.
Έξοδος της ALU	Καθορίζει τον καταχωρητή στον οποίο πρόκειται να εγγραφεί το αποτέλεσμα από την ALU.
Μνήμη	Καθορίζει την εγγραφή ή την ανάγνωση και τη διεύθυνση της μνήμης.
Καταχωρητής μνήμης	Καθορίζει τον καταχωρητή προορισμού (για ανάγνωση στη μνήμη) ή τη διεύθυνση των τιμών (για εγγραφή στη μνήμη).
PCWrite control	Καθορίζει την εγγραφή στο απαριθμητή προγράμματος (PC).
Sequencing	Καθορίζει την επόμενη εντολή που πρόκειται να εκτελεστεί.

**Σχήμα 3.4.12 – Τα 8 πεδία της μικροεντολής.** Τα επτά πρώτα πεδία ελέγχουν τη δίοδο δεδομένων, ενώ το τελευταίο πεδίο (Sequencing), καθορίζει τον τρόπο με τον οποίο επιλέγεται η επόμενη εντολή που πρόκειται να εκτελεστεί.



Οι μικροεντολές συνήθως τοποθετούνται σε ROM ή PLA, οπότε μπορούμε να βρούμε τη διεύθυνση των μικροεντολών. Συνήθως, η διεύθυνση της επόμενης εντολής είναι αυτή που ακολουθεί στη σειρά, με τον ίδιο ακριβώς τρόπο που γίνεται η επιλογή του αριθμού της επόμενης κατάστασης στις μηχανές πεπερασμένων καταστάσεων. Υπάρχουν τρεις μέθοδοι για την επιλογή της επόμενης εντολής που πρόκειται να εκτελεστεί:

- ▶ Αύξηση της διεύθυνσης της εκτελούμενης εντολής, έτσι ώστε να καθοριστεί η διεύθυνση της επόμενης εντολής. Αυτό καθορίζεται στο μικροπρόγραμμα επιλέγοντας την τιμή Seq στο πεδίο Sequencing.
- ▶ Διακλάδωση της μικροεντολής έτσι ώστε να εκτελεστεί η επόμενη εντολή του υπολογιστή MIPS. Η αρχική μικροεντολή (η οποία βρίσκεται στην κατάσταση 0), είναι εντολή Ανάκλησης και γίνεται επιλογή της τιμής Ανάκληση στο πεδίο Sequencing έτσι ώστε να καθοριστεί η λειτουργία της ανάκλησης.
- ▶ Η επιλογή της επόμενης μικροεντολής βασίζεται στην είσοδο της μονάδας ελέγχου. Η διαδικασία σύμφωνα με την οποία η επιλογή της επόμενης εντολής

βασίζεται στην είσοδο μίας μονάδας ονομάζεται *αποστολή*. Οι λειτουργίες της αποστολής υλοποιούνται με τη δημιουργία ενός πίνακα ο οποίος περιέχει τις διευθύνσεις των εντολών που πρόκειται να εκτελεστούν. Η είσοδος της μονάδας ελέγχου λειτουργεί ως ευρετήριο για τον πίνακα αυτό, ο οποίος μπορεί να υλοποιηθεί σε ROM ή PLA. Για την υλοποίηση ενός σύνθετου πίνακα αποστολής, χρειάζονται δύο πίνακες αποστολής, ένας για την αποστολή από την κατάσταση 1 και ένα για την αποστολή από την κατάσταση 2. Η επόμενη μικροεντολή καθορίζεται από μία λειτουργία αποστολής τοποθετώντας την τιμή Dispatch i στο επίπεδο Sequencing.

Το σχήμα που ακολουθεί περιγράφει τις τιμές που μπορεί να έχει κάθε πεδίο της μικροεντολής, καθώς και την επίδραση των διαφορετικών αυτών τιμών.

Όνομα πεδίου	Τιμή πεδίου	Λειτουργία του πεδίου
Μονάδα ελέγχου της ALU	Add	Γίνεται πρόσθεση στην ALU.
	Func code	Χρησιμοποιεί τον κωδικό λειτουργίας της εντολής για να καθορίσει τη μονάδα ελέγχου της ALU.
	Subt	Γίνεται αφαίρεση στην ALU.
SRC1	PC	Η πρώτη είσοδος της ALU είναι ο PC.
	rs	Η πρώτη είσοδος της ALU είναι ο καταχωρητής rs.
SRC2	4	Η δεύτερη είσοδος της ALU είναι το 4.
	Extend	Η δεύτερη είσοδος της ALU είναι η έξοδος της μονάδας επέκτασης προσήμου.
	Extshft	Η δεύτερη είσοδος της ALU είναι η έξοδος της μονάδας ολίσθησης κατά 2.
	rt	Η δεύτερη είσοδος της ALU είναι ο καταχωρητής rt.
Έξοδος της ALU	Target	Γίνεται εγγραφή της εξόδου της ALU στον καταχωρητή Στόχο.
	rd	Γίνεται εγγραφή της εξόδου της ALU στον καταχωρητή rd.
Μνήμη	Ανάγνωση του PC	Η διεύθυνση για την ανάγνωση της μνήμης είναι το περιεχόμενο του PC.
	Ανάγνωση της ALU	Η διεύθυνση για την ανάγνωση της μνήμης είναι η έξοδος της ALU.
	Εγγραφή της ALU	Η διεύθυνση για την εγγραφή της μνήμης είναι η έξοδος της μνήμης.
Καταχωρητής μνήμης	IR	Γίνεται εγγραφή των δεδομένων από τη μνήμη, στον καταχωρητή εντολών.
	Εγγραφή rt	Γίνεται εγγραφή των δεδομένων από τη μνήμη, στον καταχωρητή rt.
	Ανάγνωση rt	Τα δεδομένα για την εγγραφή της μνήμης, προέρχονται από τον καταχωρητή rt.
PCWrite control	ALU	Γίνεται εγγραφή της εξόδου της ALU στον PC.
	Target – cond	Αν η έξοδος Zero της ALU είναι ενεργοποιημένη, τότε γίνεται εγγραφή των περιεχομένων του καταχωρητή Στόχος, στον PC.
	Διεύθυνση μεταπήδησης	Γίνεται εγγραφή της διεύθυνσης μεταπήδησης της εντολής, στον PC.

Sequencing	Seq	Η επόμενη μικροεντολή είναι αυτή που ακολουθεί διαδοχικά.
	Fetch	Πήγαινε στην πρώτη μικροεντολή, έτσι ώστε να ξεκινήσει η επόμενη εντολή.
	Dispatch i	Η αποστολή χρησιμοποιεί τη ROM, η οποία καθορίζεται από το i (1 ή 2).

**Σχήμα 3.4.13 - Οι τιμές που μπορούν να πάρουν τα πεδία μιας μικροεντολής και η λειτουργία των πεδίων, ανάλογα με την τιμή που έχουν.**



## Δημιουργία του μικροπρογράμματος

Για τη δημιουργία του μικροπρογράμματος της μονάδας ελέγχου, χωρίζουμε τις εντολές του μικροπρογράμματος σε επίπεδα. Τα επίπεδα αυτά μπορούν να χρησιμοποιηθούν για να καθορίσουν τα περιεχόμενα στους πίνακες αποστολής. Στο μικροπρόγραμμα υπάρχουν δύο περιπτώσεις στις οποίες το πεδίο μίας εντολής μπορεί να είναι κενό. Όταν το πεδίο το οποίο ελέγχει τη μονάδα λειτουργίας, ή το πεδίο το οποίο προκαλεί την εγγραφή της κατάστασης (όπως το πεδίο Μνήμης ή το πεδίο ALU destination), είναι κενό, τότε τα αντίστοιχα σήματα ελέγχου δεν είναι ενεργοποιημένα. Όταν το πεδίο το οποίο ελέγχει μόνο τον πολυπλέκτη που καθορίζει την είσοδο της μονάδας λειτουργίας, (όπως το πεδίο SRC1), μένει κενό, σημαίνει πως η είσοδος της λειτουργικής μονάδας είναι αδιάφορη (ή η έξοδος του πολυπλέκτη είναι αδιάφορη).



Για την καλύτερη κατανόηση του μικροπρογράμματος, θα το χωρίσουμε σε τμήματα το καθένα από τα οποία σχετίζεται με ένα βήμα εκτέλεσης της εντολής.

Το πρώτο βήμα της εκτέλεσης των εντολών είναι η ανάκληση της εντολής, η αποκωδικοποίηση και ο υπολογισμός του επόμενου περιεχομένου του απαριθμητή προγράμματος, ή ο στόχος διακλάδωσης του απαριθμητή προγράμματος. Το παραπάνω βήμα αντιστοιχεί στα δύο πρώτα βήματα εκτέλεσης της εντολής. Οι δύο μικροεντολές που χρειάζονται για τα δύο αυτά βήματα απεικονίζονται στο παρακάτω σχήμα.

### Μικροεντολές για τα δύο πρώτα βήματα εκτέλεσης των εντολών:

Επίπεδο	Μονάδα ελέγχου της ALU	SRC1	SRC2	ALU destination	Μνήμη	Καταχωρητής μνήμης	PCWrite control	Sequencing
Ανάκληση	Add	PC	4		Ανάγνωση του PC	IR	ALU	Seq
	Add	PC	Extshft	Στόχος				Dispatch 1

Για την **πρώτη μικροεντολή**, η **ενεργοποίηση των πεδίων**, καθώς και τα αποτελέσματά τους φαίνονται στο παρακάτω σχήμα.

Πεδίο	Αποτέλεσμα
Μονάδα ελέγχου ALU, SRC1, SRC2	Υπολογισμός του PC + 4
Μνήμης και καταχωρητής μνήμης	Ανάκληση της εντολής στον IR
PCWrite control	Η έξοδος της ALU εγγράφεται στον PC
Sequencing	Πήγαινε στην επόμενη μικροεντολή

Για τη **δεύτερη μικροεντολή**, η ανάγνωση των καταχωρητών θα γίνει χρησιμοποιώντας τα πεδία του καταχωρητή εντολών. Οι υπόλοιπες λειτουργίες που ελέγχονται από τη μικροεντολή απεικονίζονται στο παρακάτω σχήμα.

Πεδίο	Αποτέλεσμα
Μονάδα ελέγχου ALU, SRC1, SRC2, ALU destination	Αποθήκευση του περιεχομένου του PC + sign-extension (IR[15-0]) << 2 στον καταχωρητή Στόχο
Sequencing	Χρήση του πίνακα αποστολής 1 για την επιλογή της διεύθυνσης της επόμενης μικροεντολής

Μπορούμε να θεωρήσουμε τη λειτουργία της αποστολής σαν μία εντολή case με το πεδίο opcode και τον πίνακα αποστολής 1, για την επιλογή ενός από τα τέσσερα είδη εντολών (αναφορά στη μνήμη, εντολές τύπου R, διακλάδωση με συνθήκη και μεταπήδηση). Το μικροπρόγραμμα για τις εντολές αναφοράς στη μνήμη, αποτελείται από τέσσερις μικροεντολές και απεικονίζεται στο σχήμα που ακολουθεί. Η πρώτη εντολή υπολογίζει τη διεύθυνση μνήμης. Οι δύο επόμενες εντολές την εντολή φόρτωσης (ανάγνωση της μνήμης και εγγραφή του καταχωρητή). Τέλος, για την εντολή αποθήκευσης χρειάζεται μόνο μία εντολή μετά τον υπολογισμό της διεύθυνσης μνήμης.

#### Μικροεντολές για εντολές αναφοράς στη μνήμη:

Επίπεδο	Μονάδα ελέγχου της ALU	SRC1	SRC2	ALU destination	Μνήμη	Καταχωρητής μνήμης	PCWrite control	Sequencing
LWSW1	Add	rs	Επέκταση					Dispatch 2
LW2	Add	rs	Επέκταση		Ανάγνωση της ALU			Seq
	Add	rs	Επέκταση		Ανάγνωση της ALU	Εγγραφή του rt		Ανάκληση
SW2	Add	rs	Επέκταση		Εγγραφή της ALU	Ανάγνωση του rt		Ανάκληση

**Τα πεδία για την πρώτη μικροεντολή (LWSW1) και το αποτέλεσμα αυτών όταν είναι ενεργοποιημένα είναι:**

Πεδίο	Αποτέλεσμα
Μονάδα ελέγχου ALU, SRC1, SRC2	Υπολογισμός της διεύθυνσης μνήμης: Καταχωρητής (rs) + sign-extension (IR[15-0])
Sequencing	Ο δεύτερος πίνακας αποστολής καθορίζει αν η επόμενη μικροεντολή θα είναι η LW2 ή η SW2

**Η επόμενη μικροεντολή στη σειρά (LW2) έχει τα ακόλουθα πεδία:**

Πεδίο	Αποτέλεσμα
Μονάδα ελέγχου ALU, SRC1, SRC2	Η έξοδος της ALU καθορίζει τη διεύθυνση μνήμης
Μνήμη	Γίνεται ανάγνωση της μνήμης (η διεύθυνση στην οποία θα γίνει ανάγνωση είναι η έξοδος της ALU)
Sequencing	Πήγαινε στην επόμενη μικροεντολή

**Η επόμενη εντολή ολοκληρώνει την εκτέλεση της εντολής και αποτελείται από τα εξής πεδία:**

Πεδίο	Αποτέλεσμα
Μονάδα ελέγχου ALU, SRC1, SRC2	Η έξοδος της ALU καθορίζει τη διεύθυνση μνήμης
Μνήμη	Γίνεται ανάγνωση της μνήμης (η διεύθυνση στην οποία θα γίνει ανάγνωση καθορίζεται από την έξοδο της ALU) και εγγραφή του αποτελέσματος στον καταχωρητή rt
Sequencing	Πήγαινε στην μικροεντολή Ανάκληση

Παρατηρούμε πως τα πεδία των δύο μικροεντολών για την υλοποίηση μιας εντολής φόρτωσης δε συγχέονται. Επομένως μπορούμε να τα ενώσουμε σε μία μικροεντολή η οποία θα έχει την εξής μορφή:

Επίπεδο	Μονάδα ελέγχου της ALU	SRC1	SRC2	ALU destination	Μνήμη	Καταχωρητής μνήμης	PCWrite control	Sequencing
LW2	Add	rs	Επέκταση		Ανάγνωση της ALU	Εγγραφή του rt		Ανάκληση



Πολλές φορές χρησιμοποιούμε βελτιστοποίηση του μικροκώδικα έτσι ώστε να μειώσουμε τον αριθμό των μικροεντολών. Ωστόσο, με αυτή την αλλαγή υπάρχει περίπτωση να αυξηθεί το μήκος του κύκλου ρολογιού, αφού η προσπέλαση της μνήμης και η εγγραφή του καταχωρητή πρέπει να γίνονται σε μία μόνο μικροεντολή. Επίσης, κάθε μικροεντολή πρέπει να αντιστοιχεί σε ένα κύκλο ρολογιού. Έτσι, όταν προσπαθούμε να βελτιστοποιήσουμε το μικροκώδικα πρέπει να γνωρίζουμε ποιες από τις λειτουργίες της διόδου δεδομένων μπορούν να εκτελεστούν κατά τη διάρκεια ενός κύκλου ρολογιού.

**Η μικροεντολή αποθήκευσης (SW2), λειτουργεί με παρόμοιο τρόπο με τη μικροεντολή φόρτωσης (LW2). Τα πεδία που αποτελείται η μικροεντολή αποθήκευσης είναι:**

Πεδίο	Αποτέλεσμα
Μονάδα ελέγχου ALU, SRC1, SRC2	Η έξοδος της ALU καθορίζει τη διεύθυνση μνήμης
Μνήμη και καταχωρητής μνήμης	Γίνεται ανάγνωση της μνήμης (η διεύθυνση στην οποία θα γίνει ανάγνωση καθορίζεται από την έξοδο της ALU) και το περιεχόμενο του καταχωρητή που καθορίζεται από το πεδίο rt, είναι η τιμή η οποία πρόκειται να εγγραφεί
Sequencing	Πήγαινε στην μικροεντολή Ανάκληση

Το μικροπρόγραμμα για τις εντολές τύπου R αποτελείται από δύο μικροεντολές: στην πρώτη εκτελούνται οι λειτουργίες της ALU και στη δεύτερη γίνεται εγγραφή του αποτελέσματος στο αρχείο καταχωρητών.

#### Μικροεντολές για τις εντολές τύπου R:

Επίπεδο	Μονάδα ελέγχου της ALU	SRC1	SRC2	ALU destination	Μνήμη	Καταχωρητής μνήμης	PCWrite control	Sequencing
Rformat1	Κωδικός Func	rs	rt					Seq
	Κωδικός Func	rs	rt	rd				Ανάκληση

Οι δύο αυτές μικροεντολές μπορούν να ενωθούν έτσι ώστε να έχουμε μόνο μία μικροεντολή, όπως ακριβώς και στην εντολή φόρτωσης. Έτσι, οι λειτουργίες της ALU και η επανεγγραφή του καταχωρητή θα εκτελούνται σε ένα κύκλο ρολογιού, ο οποίος προφανώς θα είναι μεγαλύτερος. Για να ενώσουμε τις δύο μικροεντολές, αρχικά θα δούμε τα πεδία που αποτελούνται και το αποτέλεσμά τους όταν αυτά είναι ενεργοποιημένα. **Η πρώτη μικροεντολή** σχετίζεται με τις λειτουργίες που εκτελούνται στην ALU και αποτελείται από τα παρακάτω πεδία:

Πεδίο	Αποτέλεσμα
Μονάδα ελέγχου ALU, SRC1, SRC2	Η ALU εκτελεί την πράξη ανάμεσα στα περιεχόμενα των καταχωρητών rs και rt, χρησιμοποιώντας το πεδίο func το οποίο καθορίζει την πράξη που πρόκειται να εκτελεστεί
Sequencing	Πήγαινε στην επόμενη μικροεντολή

Στη **δεύτερη μικροεντολή** γίνεται εγγραφή του αποτελέσματος (έξοδος της ALU), στο αρχείο καταχωρητών. Τα πεδία που αποτελούν τη δεύτερη μικροεντολή είναι:

Πεδίο	Αποτέλεσμα
Μονάδα ελέγχου ALU, SRC1, SRC2, ALU destination	Το πεδίο ALU dest καθορίζει ότι το rd χρησιμοποιείται για την επιλογή του καταχωρητή προορισμού
Sequencing	Πήγαινε στην μικροεντολή Ανάκληση

Το **μικροπρόγραμμα** για τις εντολές διακλάδωσης με συνθήκη αποτελείται από μία μικροεντολή:

Επίπεδο	Μονάδα ελέγχου της ALU	SRC1	SRC2	ALU destination	Μνήμη	Καταχωρητής μνήμης	PCWrite control	Sequencing
BEQ1	Subt	rs	rt				Target-cond	Ανάκληση

Τα **πεδία** που ενεργοποιούνται στη **μικροεντολή διακλάδωσης με συνθήκη** είναι:

Πεδίο	Αποτέλεσμα
Μονάδα ελέγχου ALU, SRC1, SRC2	Η ALU εκτελεί αφαίρεση στο περιεχόμενο του καταχωρητή για να δημιουργήσει την έξοδο Zero
PCWrite control	Γίνεται εγγραφή στον απαριθμητή προγράμματος της τιμής του καταχωρητή Στόχος, εάν η έξοδος Zero της ALU είναι αληθής
Sequencing	Πήγαινε στην μικροεντολή Ανάκληση

Το **μικροπρόγραμμα** για τις εντολές μεταπήδησης αποτελείται από μία μικροεντολή:

Επίπεδο	Μονάδα ελέγχου της ALU	SRC1	SRC2	ALU destination	Μνήμη	Καταχωρητής μνήμης	PCWrite control	Sequencing
JUMP1							Διεύθυνση μεταπήδησης	Ανάκληση

Μόνο δύο από τα **πεδία** της **μικροεντολής μεταπήδησης** είναι ενεργοποιημένα. Τα πεδία αυτά είναι:

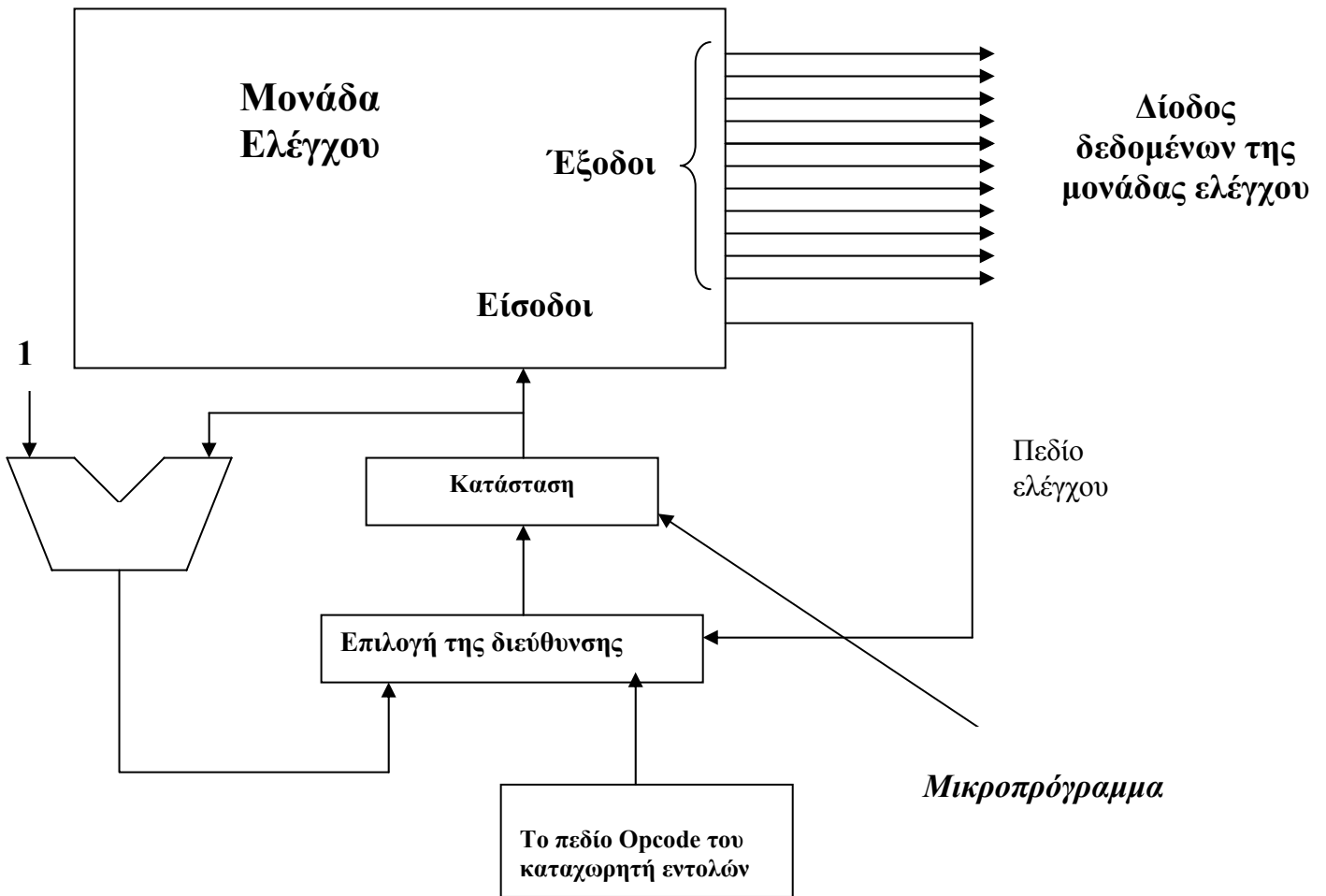
Πεδίο	Αποτέλεσμα
PCWrite control	Γίνεται εγγραφή στον απαριθμητή προγράμματος χρησιμοποιώντας το πεδίο jump
Sequencing	Πήγαινε στην μικροεντολή Ανάκληση

Στο σχήμα 3.4.14 απεικονίζεται το μικροπρόγραμμα για την εκτέλεση όλων των εντολών, το οποίο αποτελείται από δέκα μικροεντολές. Απ' τη στιγμή που το μικροπρόγραμμα αποτελείται από μικροεντολές και πίνακες αποστολής, μπορεί να υλοποιηθεί από ROM ή PLA. Τα σήματα ελέγχου της διόδου δεδομένων σε κάθε μικροεντολή, μπορούν να υλοποιηθούν χρησιμοποιώντας λογικές πύλες και τα χρησιμοποιούμε για την υλοποίηση μιας μηχανής πεπερασμένων καταστάσεων.

Επίπεδο	Μονάδα ελέγχου της ALU	SRC1	SRC2	ALU destination	Μνήμη	Καταχωρητής μνήμης	PCWrite control	Sequencing
Ανάκληση	Add	PC	4		Ανάγνωση του PC	IR	ALU	Seq
	Add	PC	Extshft	Στόχος				Dispatch i
LWSW1	Add	rs	Extshft					Dispatch i
LW2	Add	rs	Extshft		Ανάγνωση της ALU			Seq
	Add	rs	Extshft		Ανάγνωση της ALU	Εγγραφή του rt		Ανάκληση
SW2	Add	rs	Extshft		Εγγραφή της ALU	Ανάγνωση του rt		Ανάκληση
Rformat1	Κωδικός func	rs	rt					Seq
	Κωδικός func	rs	rt	rd				Ανάκληση
BEQ1	Subt	rs	rt				Target-cond	Ανάκληση
JUMP1							Διεύθυνση μεταπήδησης	Ανάκληση

**Σχήμα 3.4.14 - Το μικροπρόγραμμα για τη μονάδα ελέγχου.** Τα επίπεδα που χρησιμοποιούνται για να καθοριστούν οι λειτουργίες της αποστολής. Η αποστολή 1 (Dispatch 1) εκτελεί μεταπήδηση η οποία βασίζεται στον IR στο επίπεδο 1. Το PCWrite control καθορίζει αν πρόκειται να γίνει εγγραφή υπό συνθήκη στον PC. Υπάρχουν τρεις πιθανές τιμές για την εγγραφή στον PC: η έξοδος της ALU, το περιεχόμενο του καταχωρητή που της διεύθυνσης του στόχου διακλάδωσης (καταχωρητής Στόχος), και η διεύθυνση του στόχου μεταπήδησης, η οποία προέρχεται από το συνδυασμό των 26 λιγότερων σημαντικών ψηφίων της εντολής και του PC.

Το σχήμα 3.4.15 απεικονίζει ένα τρόπο με τον οποίο μπορούμε να υλοποιήσουμε το μικροπρόγραμμα. Σε αυτό το είδος υλοποίησης χρησιμοποιούνται συνδυαστικά κυκλώματα, που καθορίζουν την τιμή των γραμμών ελέγχου της διόδου δεδομένων, έτσι ώστε να γίνεται η επιλογή της επόμενης κατάστασης. Για την επιλογή της διεύθυνσης της επόμενης προς εκτέλεση μικροεντολής, χρησιμοποιούνται οι πίνακες αποστολής



**Σχήμα 3.4.15 - Η μικροπρογραμματιζόμενη μονάδα ελέγχου του MIPS.** Η μονάδα ελέγχου χρησιμοποιεί συνδυαστικά κυκλώματα για να καθορίσει τη δίοδο δεδομένων μίας κατάστασης, καθώς επίσης και να καθορίσει την επιλογή της επόμενης προς εκτέλεση μικροεντολής. Η επιλογή της επόμενης διεύθυνσης περιέχει τους πίνακες αποστολής για την επιλογή των επόμενων καταστάσεων: ο έλεγχος για την επιλογή της επόμενης μικροεντολής γίνεται από τις εξόδους της μονάδας ελέγχου.



## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Τι είναι ο μικροπρογραμματισμός και γιατί είναι καλύτερη μέθοδος για την υλοποίηση της μονάδας ελέγχου, από τη μέθοδο των μηχανών πεπερασμένων καταστάσεων. Να αιτιολογήσετε την απάντησή σας.



## ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 5

Ο μικροπρογραμματισμός είναι μία τεχνική για την υλοποίηση της μονάδας ελέγχου. Ο μικροπρογραμματισμός είναι καλύτερη τεχνική από ότι οι μηχανές πεπερασμένων καταστάσεων, στην περίπτωση που το σχήμα της μονάδας ελέγχου αναπαρασταθεί από μια μηχανή πεπερασμένων καταστάσεων και αποτελείται από πολλές καταστάσεις.



## ΔΡΑΣΤΗΡΙΟΤΗΤΑ 6

Να αναφέρετε τα πλεονεκτήματα και τα μειονεκτήματα του μικροπρογραμματισμού. Να αιτιολογήσετε την απάντησή σας.



## ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 6

Υπάρχουν πολλά πλεονεκτήματα με τη χρήση του μικροπρογραμματισμού:

- 1) μπορούμε να χειριστούμε περισσότερες εντολές από ότι με τις μηχανές πεπερασμένων καταστάσεων,
- 2) αφού ο μικροπρογραμματισμός είναι μια τεχνική λογισμικού, υπάρχουν πολλά εργαλεία λογισμικού τα οποία υποστηρίζουν τη συγκεκριμένη τεχνική,
- 3) οι τεχνικές οι οποίες χρησιμοποιούνται για την υλοποίηση του μικροπρογραμματισμού είναι σχετικά εύκολες, με αποτέλεσμα ο μικροπρογραμματισμός να καθίσταται ο ευκολότερος τρόπος υλοποίησης της μονάδας ελέγχου.

Υπάρχουν όμως και αρκετά μειονεκτήματα όπως:

- 1) οι μικροεντολές πρέπει να είναι αμετάβλητες, δηλαδή κάθε πεδίο δεν μπορεί να έχει παραπάνω από μία τιμή για μία συγκεκριμένη μικροεντολή,
- 2) η ορθότητα των προγραμμάτων δεν είναι πάντα εύκολη υπόθεση, με αποτέλεσμα ο τεχνικός έλεγχος του λογισμικού να εξαρτάται από το υλικό του υπολογιστή, το οποίο μπορεί να έχει προβλήματα,
- 3) χρησιμοποιείται μια ειδική μνήμη μικροπρογραμματισμού, η οποία δεν είναι απαραίτητα ταχύτερη στους υπολογιστές RISC.

## Ανακεφαλαιώνοντας λοιπόν ...



Η βασική ιδέα για την υλοποίηση της μικροπρογραμματιζόμενης μονάδας ελέγχου είναι η μετατροπή της μονάδας ελέγχου σε έναν υπολογιστή-μικρογραφία. Οι μικροεντολές καθορίζουν τα σήματα ελέγχου που στέλνονται στη δίοδο δεδομένων και αποφασίζουν υπό όρους ποια μικροεντολή θα εκτελεστεί μετά.



Με την ομαδοποίηση των σημάτων ελέγχου πετυχαίνουμε μείωση του μεγέθους της μικροεντολής αυξάνοντας όμως την πολυπλοκότητα της αποκωδικοποίησης των σημάτων.



Το CPI της μικροπρογραμματιζόμενης μονάδας ελέγχου του υπολογιστή DLX είναι μεγαλύτερο απ' αυτό της καλωδιωμένης, γιατί ο έλεγχος για διακοπές απαιτεί στο ξεκίνημα έναν κύκλο ρολογιού, οι φορτώσεις και οι αποθηκεύσεις είναι πιο αργές και οι διακλαδώσεις με συνθήκη που απαιτούν σύγκριση με το 0 είναι πιο αργές στην περίπτωση που είναι διακλαδώσεις με συνθήκη που δεν ακολουθούνται (untaken).



Κάθε μικροεντολή στο μικροπρόγραμμα του υπολογιστή MIPS αποτελείται από 8 πεδία. Κάθε πεδίο μπορεί να εκτελεί διαφορετικές λειτουργίες, ανάλογα με τις τιμές που παίρνει κάθε φορά.



Για τη δημιουργία του μικροπρογράμματος της μονάδας ελέγχου του υπολογιστή MIPS, χωρίζουμε τις εντολές του μικροπρογράμματος σε επίπεδα.