

Εντολές γλώσσας μηχανής

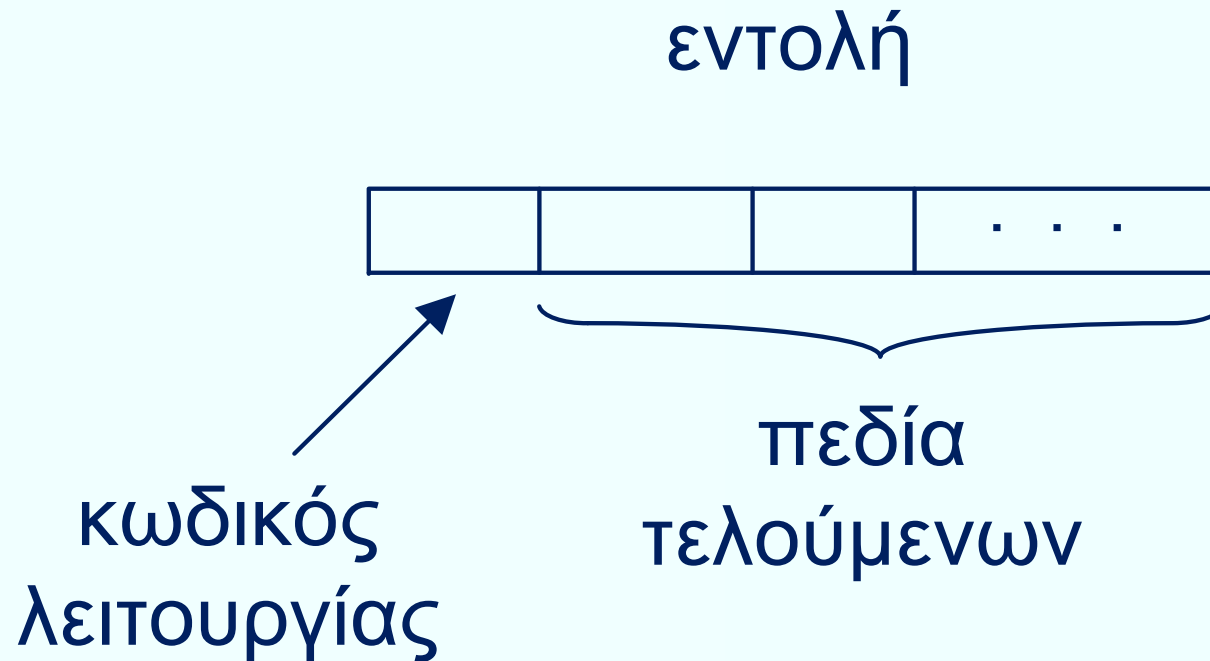
Στον υπολογιστή MIPS η εντολή

“πρόσθεσε τα περιεχόμενα των καταχωρητών 17 και 20 και τοποθέτησε το αποτέλεσμα στον καταχωρητή 9”

έχει την μορφή:

00000010001101000100100000100000

Πεδία εντολής γλώσσας μηχανής



Είδη εντολών γλώσσας μηχανής

- Εντολές μεταφοράς δεδομένων
- Αριθμητικές εντολές
- Εντολές λογικών πράξεων
- Εντολές ελέγχου της ροής του προγράμματος
- Εντολές εισόδου/εξόδου

Είδη και μέγεθος δεδομένων

Οι γλώσσες προγραμματισμού υψηλού επιπέδου υποστηρίζουν μια μεγάλη ποικιλία από είδη δεδομένων

Τα δεδομένα αποθηκεύονται στην μνήμη του υπολογιστή με ένα ενιαίο τρόπο

Έννοια της λέξης

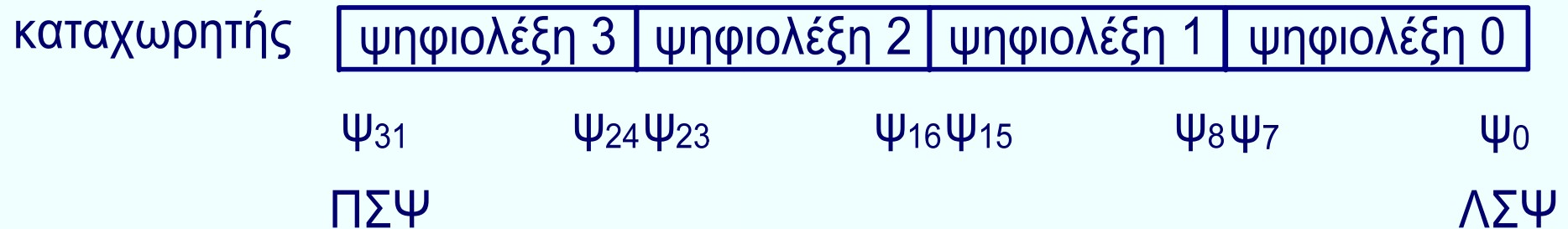
Είδη και μέγεθος δεδομένων

- χαρακτήρες (8 δυαδικά ψηφία)
- ακέραιοι 8 δυαδικών ψηφίων (μια ψηφιολέξη)
- ακέραιοι 16 δυαδικών ψηφίων (μισή λέξη)
- ακέραιοι 32 δυαδικών ψηφίων (μια λέξη)
- ακέραιοι 64 δυαδικών ψηφίων (διπλή λέξη)
- αριθμοί 32 δυαδικών ψηφίων κινητής υποδιαστολής απλής ακρίβειας
- αριθμοί 64 δυαδικών ψηφίων κινητής υποδιαστολής διπλής ακρίβειας

Οργάνωση της κύριας μνήμης

Μια ψηφιολέξη ανά θέση μνήμης

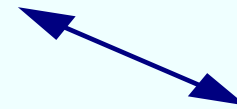
Οργάνωση της πληροφορίας στον καταχωρητή



Απεικόνιση ανάλογη του μεγέθους

ψηφιολέξη 3 | ψηφιολέξη 2 | ψηφιολέξη 1 | ψηφιολέξη 0

καταχωρητής



κύρια
μνήμη

διεύθυνση

100

ψηφιολέξη 0

101

ψηφιολέξη 1

102

ψηφιολέξη 2

103

ψηφιολέξη 3

104

ψηφιολέξη 0

105

ψηφιολέξη 1

⋮

λέξη

μισή
λέξη

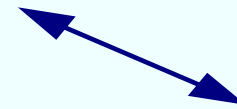
Intel x86

απεικόνιση ανάλογη του μεγέθους
(little endian)

Απεικόνιση αντιστρόφως ανάλογη του μεγέθους

ψηφιολέξη 3 | ψηφιολέξη 2 | ψηφιολέξη 1 | ψηφιολέξη 0

καταχωρητής



κύρια
μνήμη

διεύθυνση

100

ψηφιολέξη 3

101

ψηφιολέξη 2

102

ψηφιολέξη 1

103

ψηφιολέξη 0

104

ψηφιολέξη 1

105

ψηφιολέξη 0

⋮

λέξη

μισή
λέξη

Motorola 680x0

απεικόνιση αντιστρόφως ανάλογη του μεγέθους (big endian)

Πλεονεκτήματα και μειονεκτήματα κάθε τρόπου απεικόνισης

Μεταφορά μεταξύ κύριας μνήμης και καταχωρητή

Μεταφραστής

Αποθήκευση χειρονακτικά

Ανάγνωση δεδομένων από αρχείο

Πλεονεκτήματα και μειονεκτήματα κάθε τρόπου απεικόνισης

Ανάλογη του μεγέθους

Βασική διεύθυνση	+0	ψηφιολέξη	0
Βασική διεύθυνση	+1	ψηφιολέξη	1
Βασική διεύθυνση	+2	ψηφιολέξη	2
Βασική διεύθυνση	+3	ψηφιολέξη	3

Αντιστρόφως ανάλογη του μεγέθους

Βασική διεύθυνση	+0	ψηφιολέξη	3
Βασική διεύθυνση	+1	ψηφιολέξη	2
Βασική διεύθυνση	+2	ψηφιολέξη	1
Βασική διεύθυνση	+3	ψηφιολέξη	0

Ευθυγραμμισμένες διευθύνσεις (data alignment)

Όταν η φυσική διεύθυνση της κύριας μνήμης στην οποία είναι αποθηκευμένο κάθε δεδομένο είναι ακέραιο πολλαπλάσιο του μήκους του δεδομένου (το μήκος του δεδομένου το μετράμε σε πολλαπλάσια του μήκους της θέσης μνήμης), τότε λέμε ότι οι διευθύνσεις των δεδομένων στην κύρια μνήμη είναι ευθυγραμμισμένες (**data alignment**)

Ευθυγραμμισμένες διευθύνσεις (data alignment)

Τελούμενο	Μήκος τελούμενου σε ψηφιολέξεις	Τιμή των 4 λιγότερο σημαντικών δυναδικών ψηφίων της διεύθυνσης
ψηφιολέξη	1	xxxx
Μισή λέξη	2	xxx0
λέξη	4	xx00
Διπλή λέξη	8	x000
Τετραπλή λέξη	16	0000

Ευθυγραμμισμένες διευθύνσεις (data alignment)

```

struct S1 {
  double a;      /* διπλή λέξη*/
  char b;       /* ψηφιολέξη*/
  int c;        /* λέξη*/
  short d;     /* μισή λέξη */
} example1;
  
```

διεύθυνση	μεταβλητή
X00000	a
X00001	
X00010	
X00011	
X00100	
X00101	
X00110	
X00111	
X01000	b
X01001	
X01010	
X01011	
X01100	c
X01101	
X01110	
X01111	
X10000	d
X10001	
X10010	
X10011	

Ευθυγραμμισμένες διευθύνσεις (data alignment)

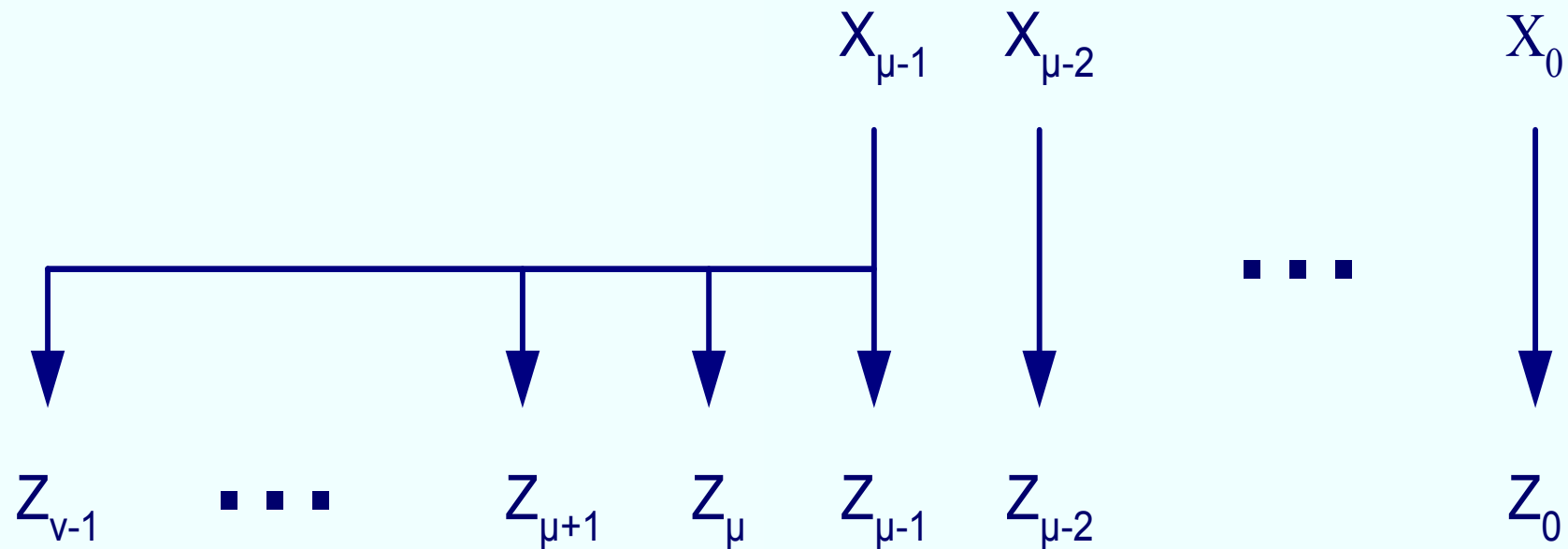
```
struct S2 {  
  double a;      /* διπλή λέξη*/  
  char b;        /* ψηφιολέξη*/  
  short d;       /* μισή λέξη */  
  int c;         /* λέξη*/  
} example2;
```

διεύθυνση	μεταβλητή
X00000	a
X00001	
X00010	
X00011	
X00100	
X00101	
X00110	
X00111	
X01000	b
X01001	
X01010	d
X01011	
X01100	c
X01101	
X01110	
X01111	

Υποστήριξη δεδομένων διαφόρων μεγεθών

Αριθμητικές μονάδες;

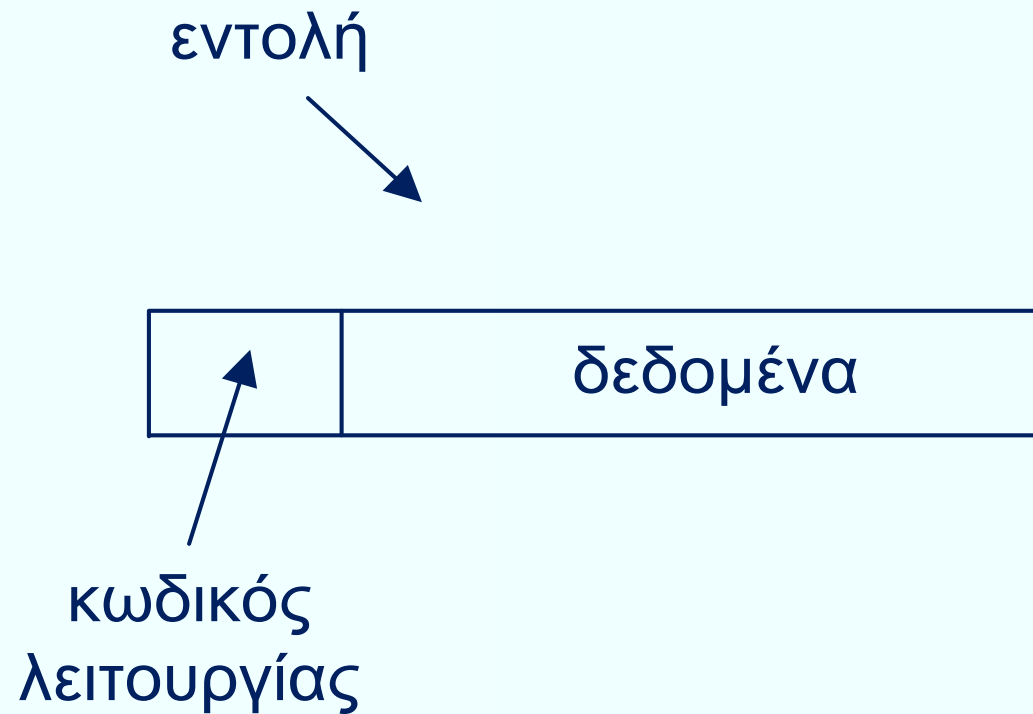
Επέκταση ορίων τελούμενου σε παράσταση συμπληρώματος ως προς 2



Τρόποι διευθυνσιοδότησης της κύριας μνήμης (Addressing Modes)

- Άμεσος τρόπος διευθυνσιοδότησης
(Immediate Addressing Mode)
- Κατ' ευθείαν τρόπος διευθυνσιοδότησης
(Direct Addressing Mode)
 - » Μνήμης
 - » Καταχωρητή
- Έμμεσος τρόπος διευθυνσιοδότησης
(Indirect Addressing Mode)
 - » Με χρήση καταχωρητή
 - » Με χρήση της κύριας μνήμης
- Σχετική διευθυνσιοδότηση
(relative addressing mode)

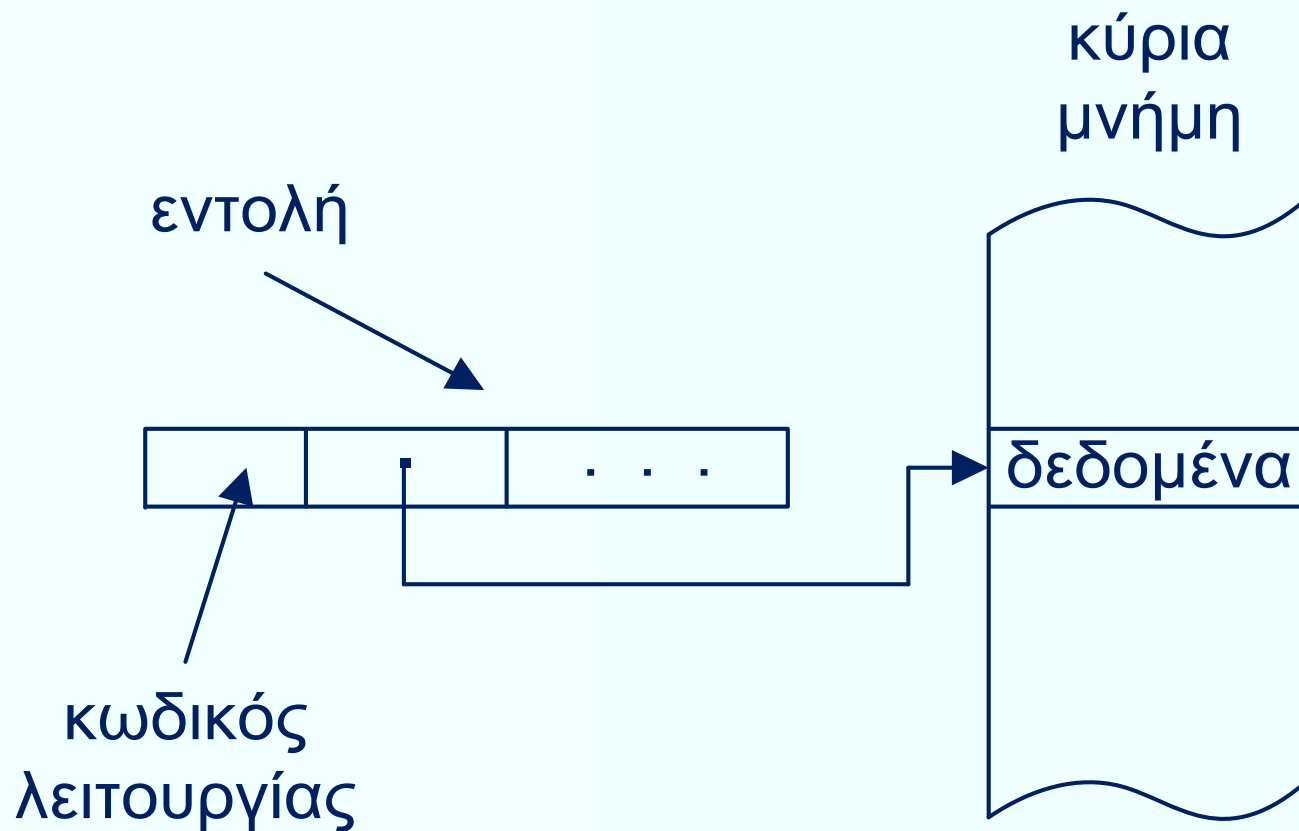
Άμεσος τρόπος διευθυνσιοδότησης



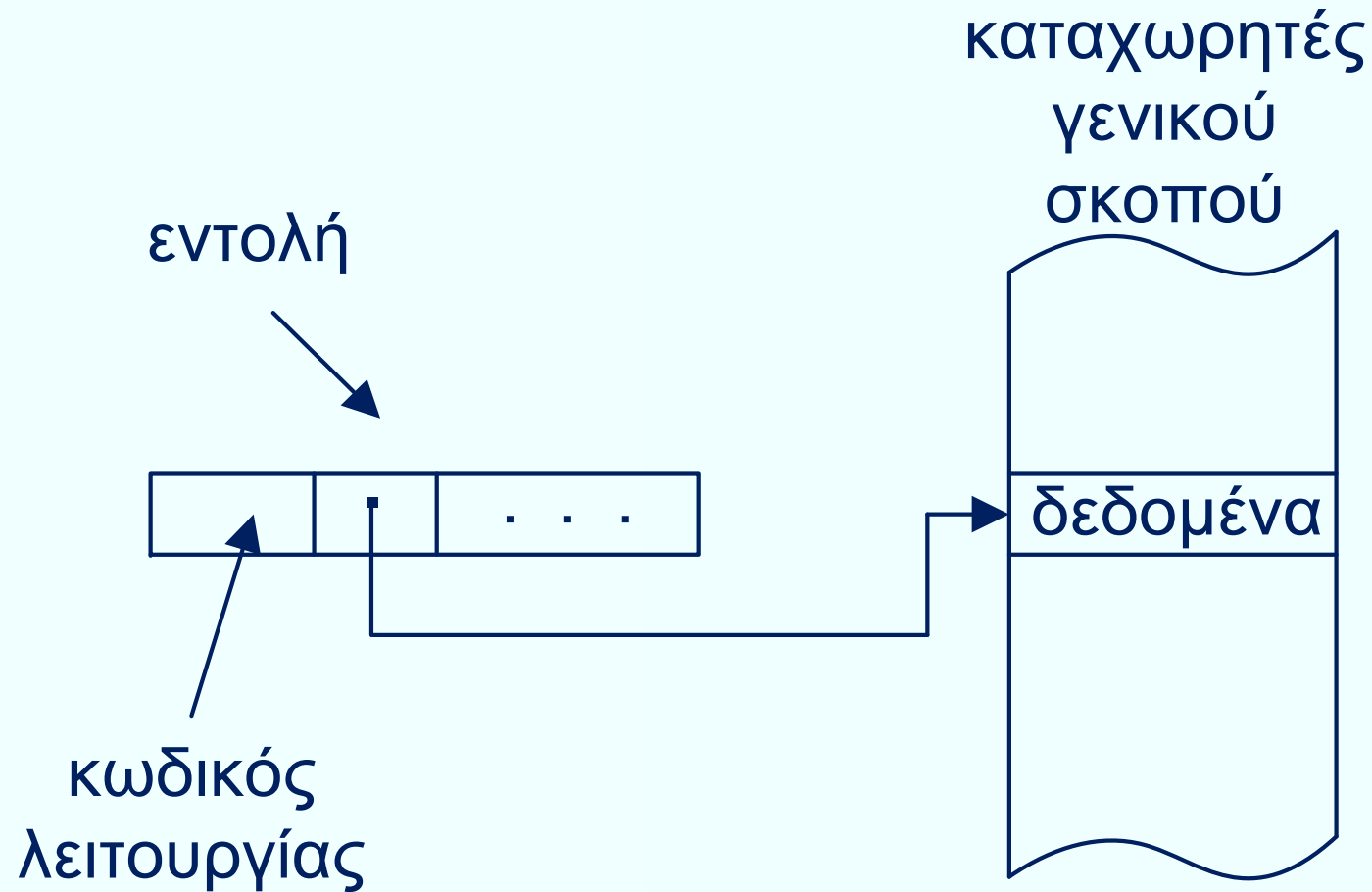
Κατ' ευθείαν τρόπος διευθυνσιοδότησης (Direct Addressing Mode)

- Κατ' ευθείαν τρόπος διευθυνσιοδότησης
 - » Μνήμης
 - » Καταχωρητή

Κατ' ευθείαν τρόπος διευθυνσιοδότησης θέσης μνήμης



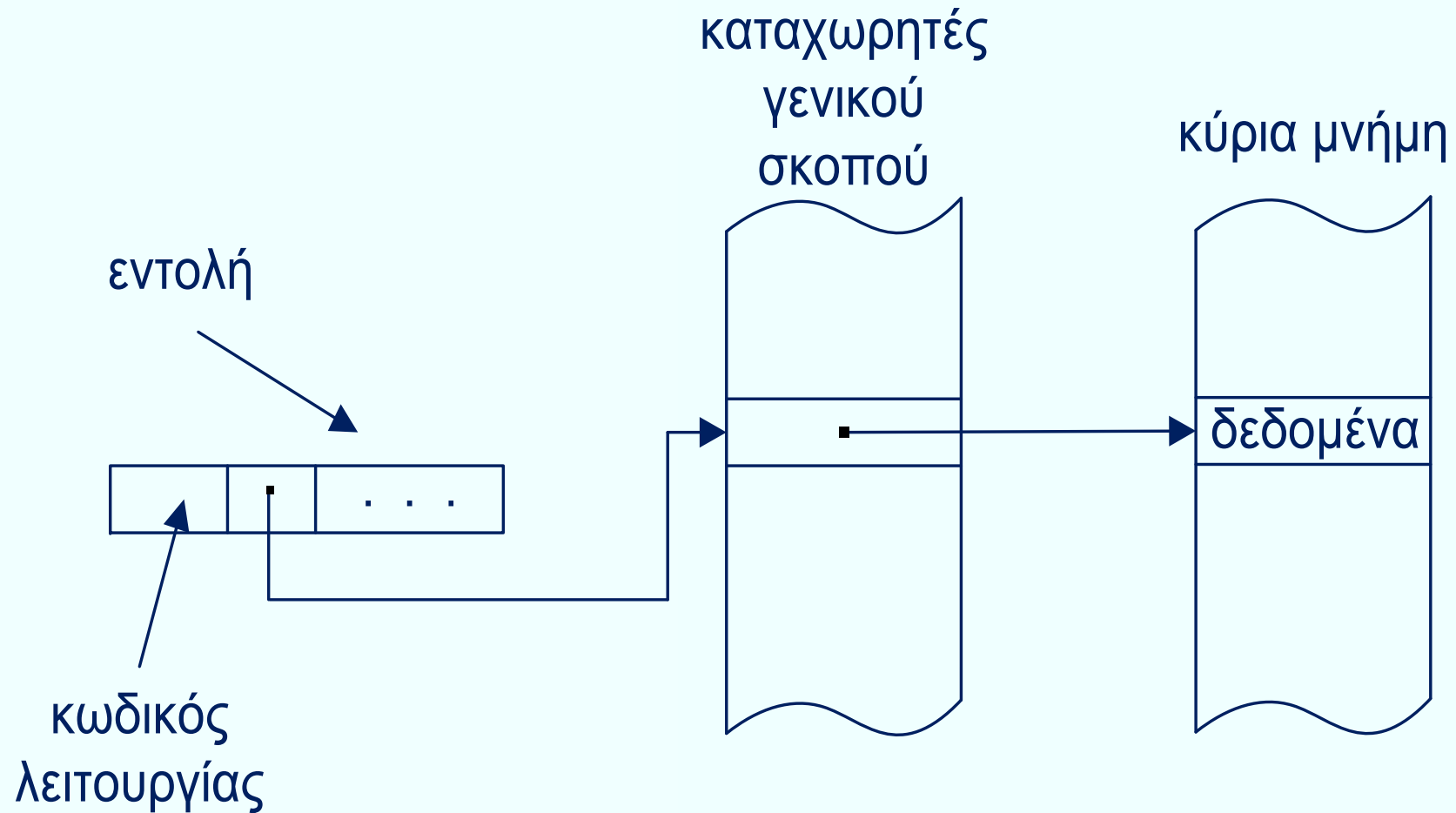
Κατ' ευθείαν τρόπος διευθυνσιοδότησης καταχωρητή



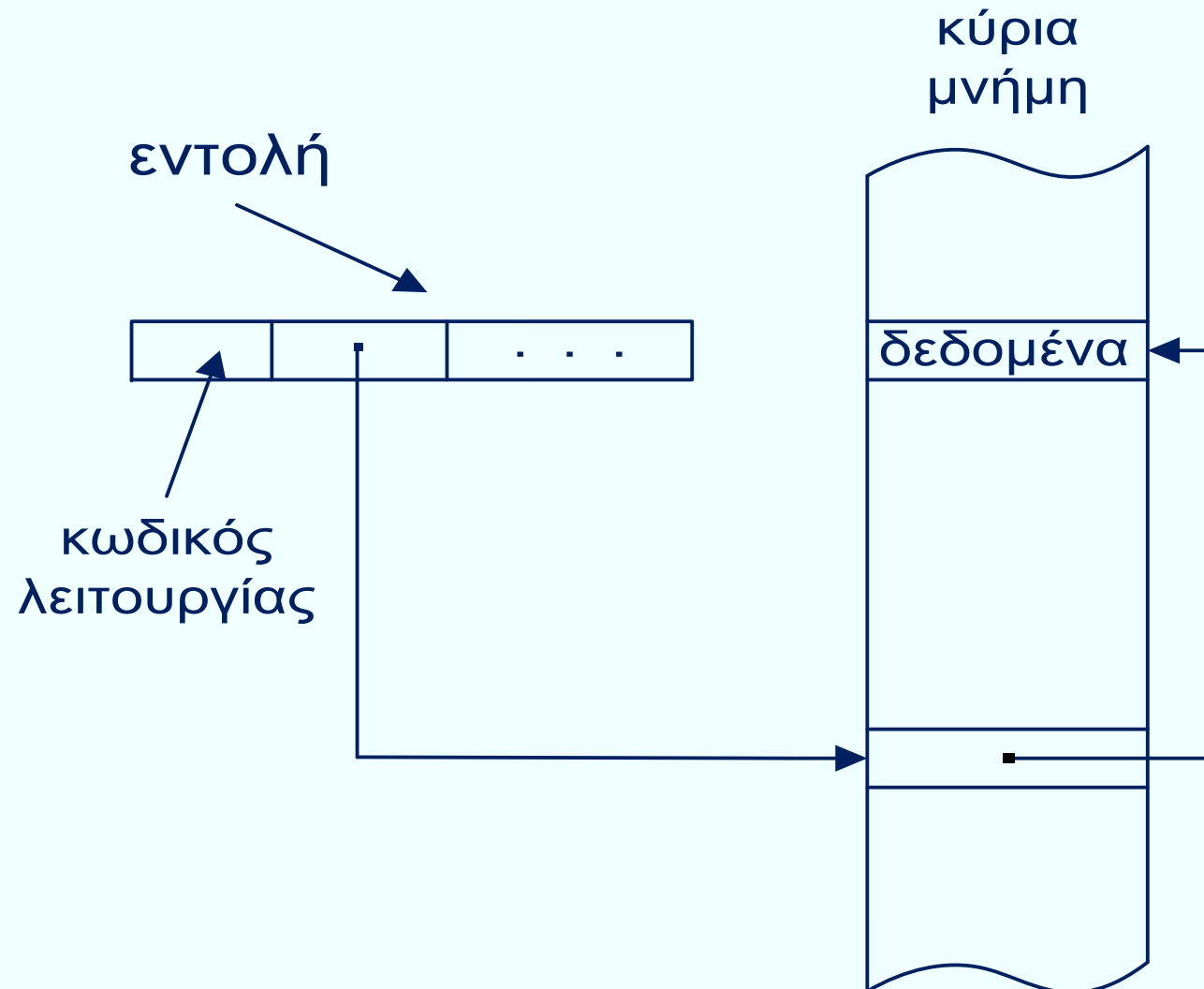
Έμμεσος τρόπος διευθυνσιοδότησης (Indirect Addressing Mode)

- Έμμεσος τρόπος διευθυνσιοδότησης
 - » με χρήση καταχωρητή
 - » με χρήση της κύριας μνήμης

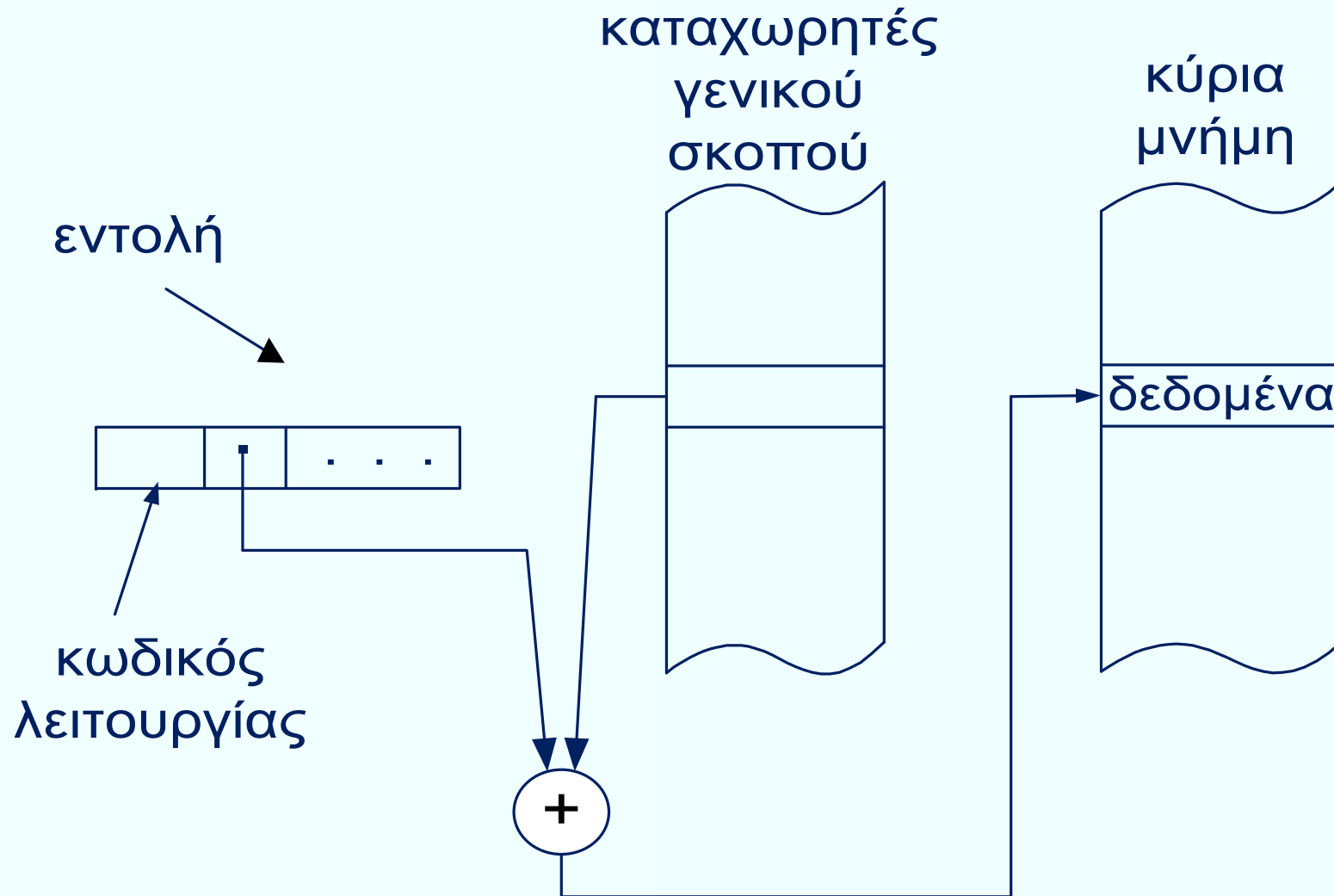
Έμμεσος τρόπος διευθυνσιοδότησης με χρήση καταχωρητή



Έμμεσος τρόπος διευθυνσιοδότησης με χρήση θέσης μνήμης



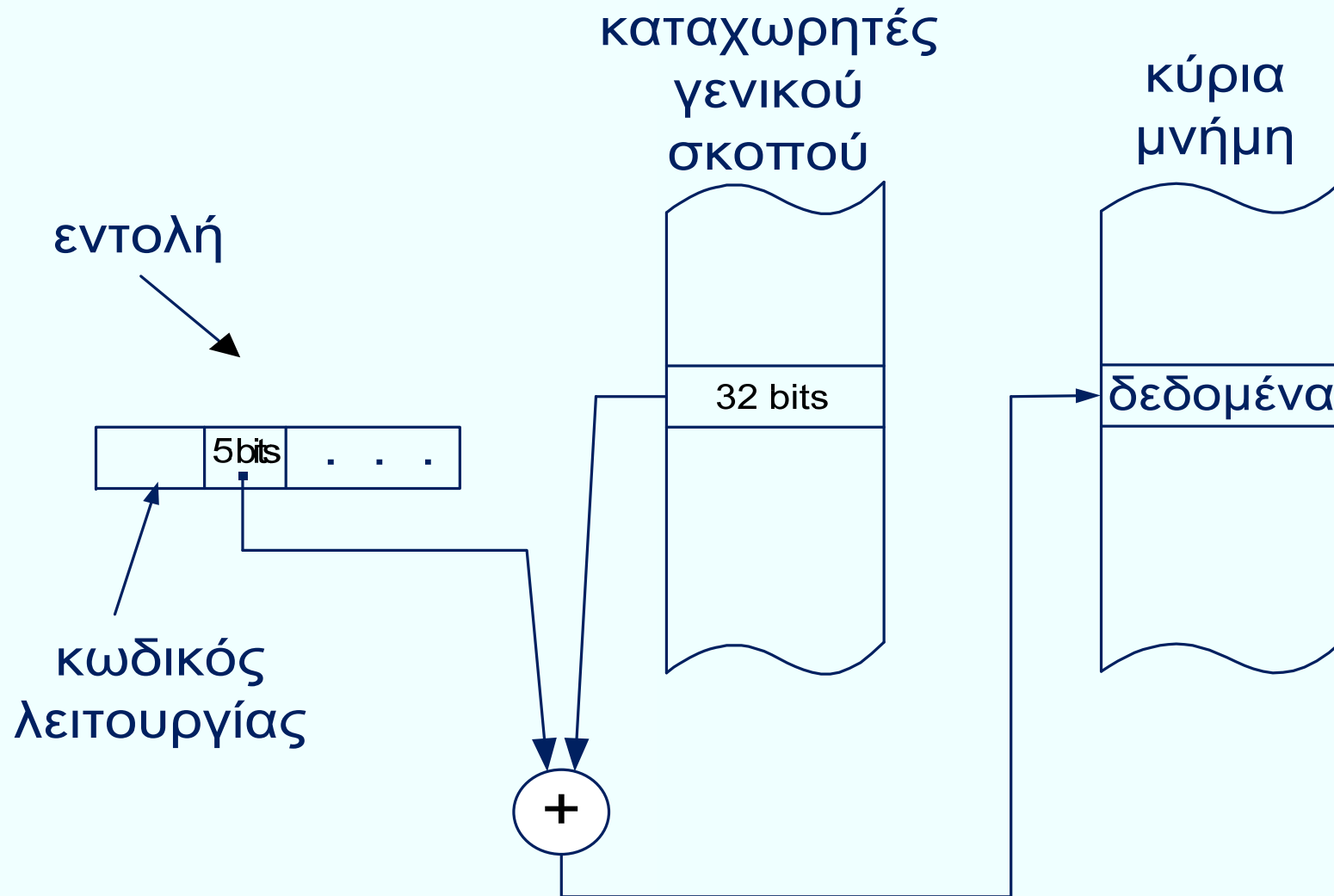
Σχετική διευθυνσιοδότηση



Σχετική διευθυνσιοδότηση

- Επειδή ολόκληρη η διεύθυνση δεν χρειάζεται να υπάρχει στο πεδίο του τελούμενου της εντολής, το μήκος της εντολής μπορεί να ελαττωθεί

Σχετική διευθύνσιοδότηση



Σχετική διευθυνσιοδότηση

- Επειδή ολόκληρη η διεύθυνση δεν χρειάζεται να υπάρχει στο πεδίο του τελούμενου της εντολής, το μήκος της εντολής μπορεί να ελαττωθεί
- Είναι δυνατόν να έχουμε προγράμματα σε εκτελέσιμη μορφή που μπορούν να εκτελεστούν σωστά σε οποιαδήποτε περιοχή της κύριας μνήμης και αν φορτωθούν (statically position independent programs)

Σχετική διευθυνσιοδότηση

...

ADD r1, r2, r3

MULT r1, r4, r5

BRE r3, r5, d

SUB r6, r5, r8

...

Σχετική διευθυνσιοδότηση

- Επειδή ολόκληρη η διεύθυνση δεν χρειάζεται να υπάρχει στο πεδίο του τελούμενου της εντολής, το μήκος της εντολής μπορεί να ελαττωθεί
- Είναι δυνατόν να έχουμε προγράμματα σε εκτελέσιμη μορφή που μπορούν να εκτελεστούν σωστά σε οποιαδήποτε περιοχή της κύριας μνήμης και αν φορτωθούν (statically position independent programs)
- Αλλάζοντας το περιεχόμενο του καταχωρητή A ο επεξεργαστής μπορεί να αλλάξει τις απόλυτες διευθύνσεις που αναφέρονται από μία ομάδα εντολών B (relocatability of programs and data segments)

Σχετική διευθυνσιοδότηση

r10

200

r10

500

...

102 MULT r1, r4, r5

104 JME r3, r5, 52, r10

106 STORE c, r5

...

252 DIV r5, r6, r8

...

...

102 MULT r1, r4, r5

104 JME r3, r5, 52, r10

106 STORE c, r5

...

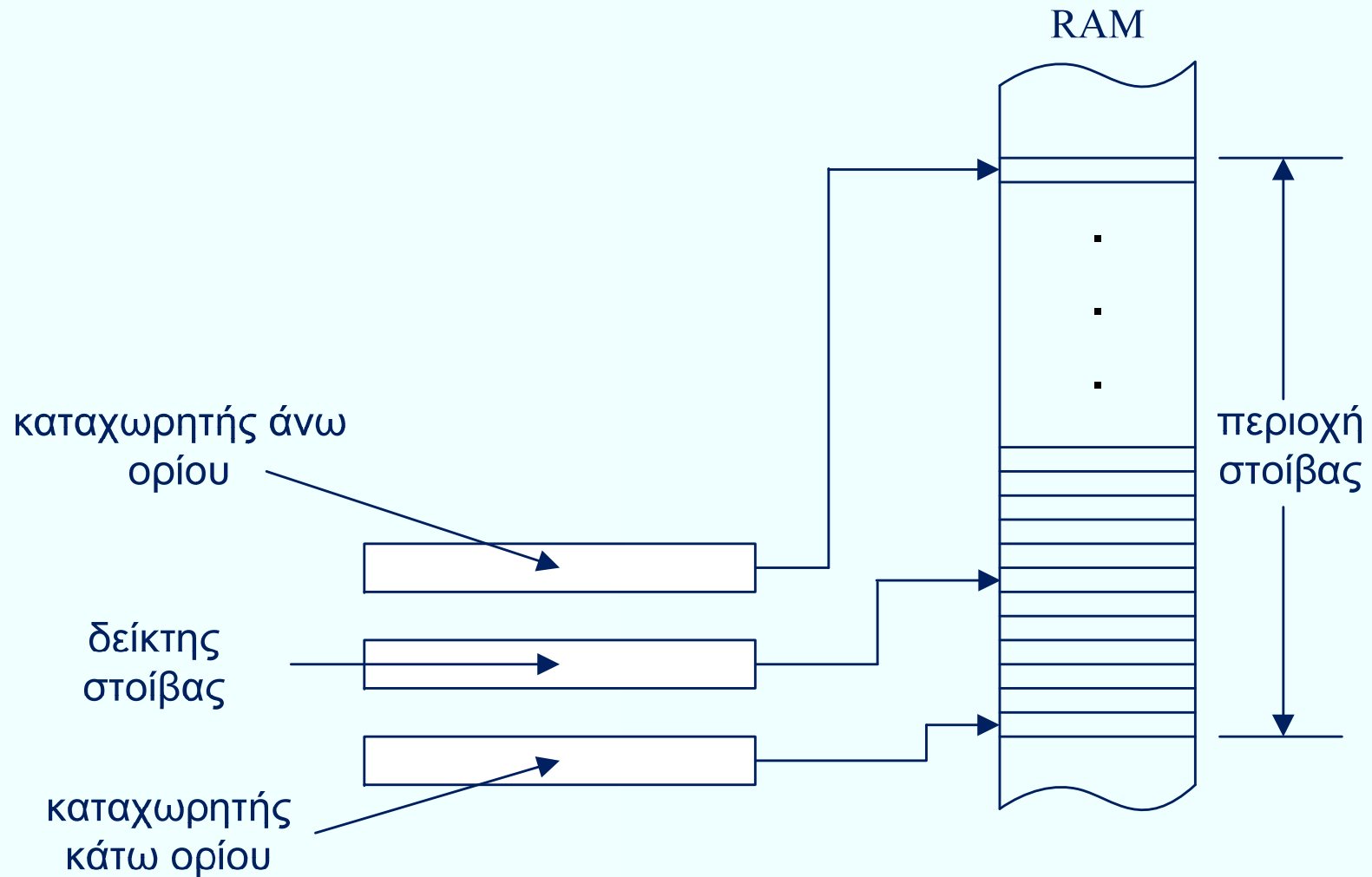
552 DIV r5, r6, r8

...

Σχετική διευθυνσιοδότηση

- Επειδή ολόκληρη η διεύθυνση δεν χρειάζεται να υπάρχει στο πεδίο του τελούμενου της εντολής, το μήκος της εντολής μπορεί να ελαττωθεί
- Είναι δυνατόν να έχουμε προγράμματα σε εκτελέσιμη μορφή που μπορούν να εκτελεστούν σωστά σε οποιαδήποτε περιοχή της κύριας μνήμης και αν φορτωθούν (statically position independent programs)
- Αλλάζοντας το περιεχόμενο του καταχωρητή A ο επεξεργαστής μπορεί να αλλάξει τις απόλυτες διευθύνσεις που αναφέρονται από μία ομάδα εντολών B (relocatability of programs and data segments)
- Ο καταχωρητής A μπορεί να χρησιμοποιηθεί για την αποθήκευση δεικτών ώστε να διευκολυνθεί η επεξεργασία δεδομένων με δείκτες

Μηχανισμός στοίβας (stack)



Ταξινόμηση βάσει του συνόλου εντολών

Αρχιτεκτονικές που βασίζονται στη χρήση:

- του μηχανισμού στοίβας (stack architectures)
- του συσσωρευτή (accumulator architectures)
- καταχωρητών γενικού σκοπού

(general-purpose register architectures)

» καταχωρητή-μνήμης

» καταχωρητή –καταχωρητή

Αρχιτεκτονική μηχανισμού στοίβας: μορφή εντολών

μορφή εντολής εκτέλεσης
αριθμητικής ή λογικής πράξης



πεδίο

κωδικού λειτουργίας

μορφή εντολής PUSH ή POP



πεδίο

κωδικού λειτουργίας



πεδίο τελούμενου

Εκτέλεση προγράμματος σε μηχανή που βασίζεται στη χρήση του μηχανισμού στοίβας

Αρχική κατάσταση

Πρόγραμμα

```
PUSH A  
PUSH B  
MUL  
PUSH C  
ADD
```

καταχωρητής άνω
ορίου

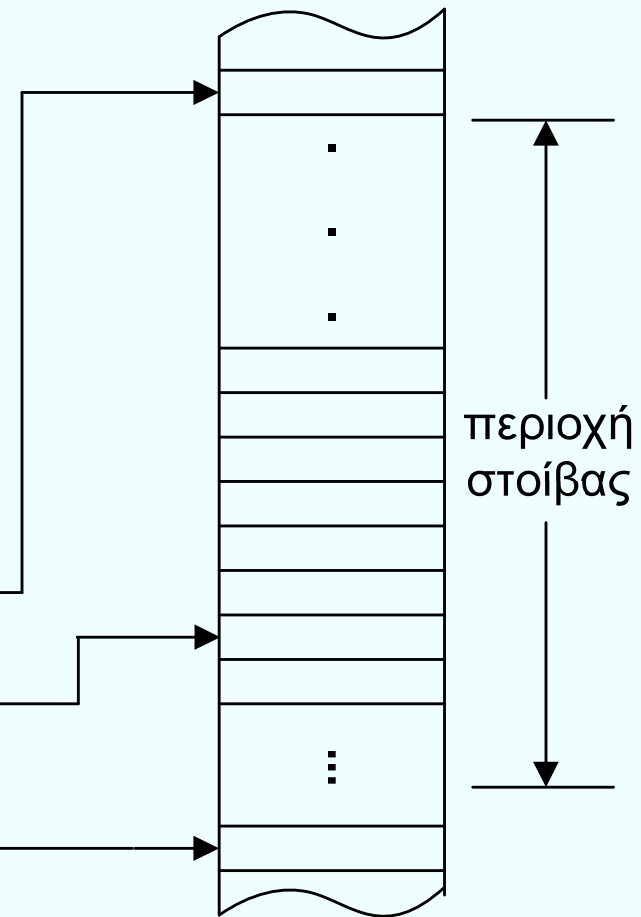
4000

δείκτης στοίβας

1100

καταχωρητής
κάτω ορίου

1000



Μετά την εκτέλεση της εντολής “PUSH A”

Πρόγραμμα

```
PUSH A  
PUSH B  
MUL  
PUSH C  
ADD
```

καταχωρητής άνω
ορίου

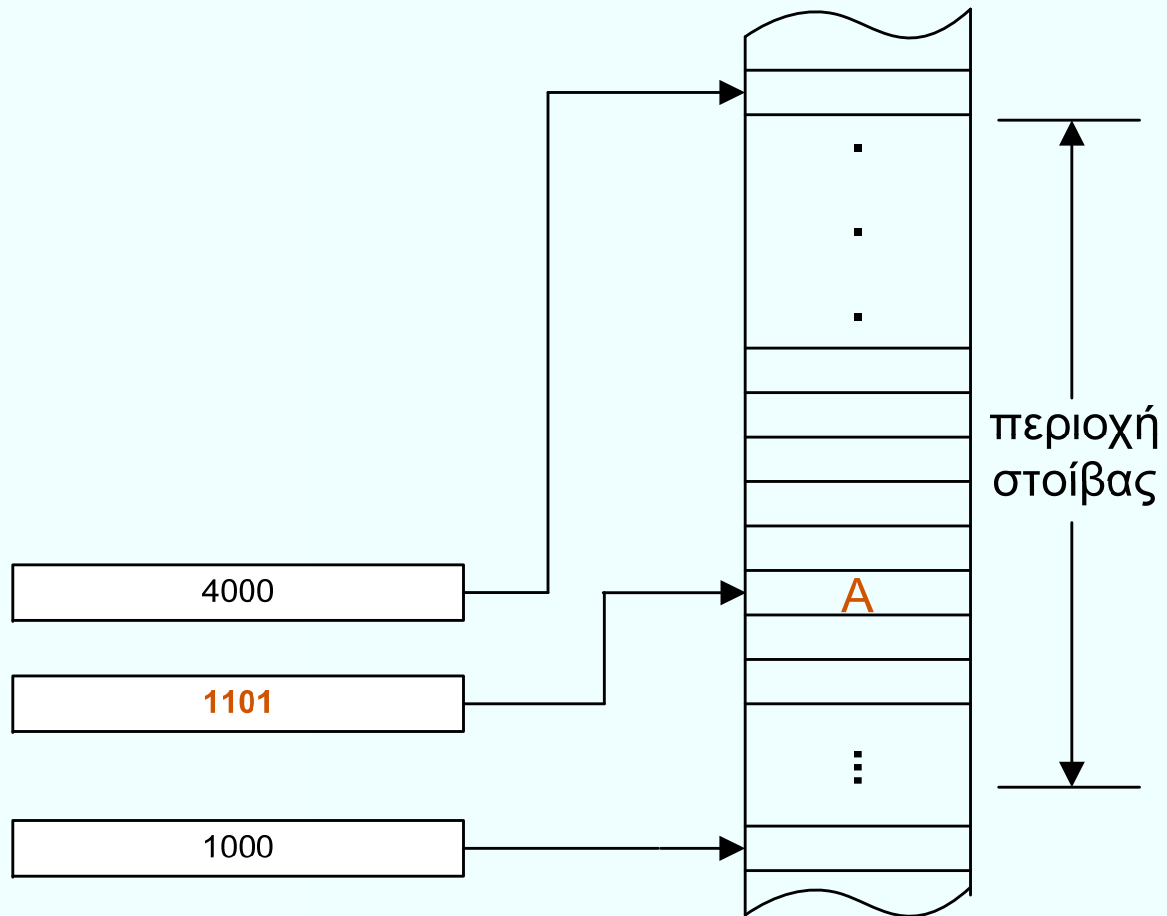
4000

δείκτης στοίβας

1101

καταχωρητής
κάτω ορίου

1000



Μετά την εκτέλεση της εντολής “PUSH B”

Πρόγραμμα

```
PUSH A  
PUSH B  
MUL  
PUSH C  
ADD
```

καταχωρητής άνω
ορίου

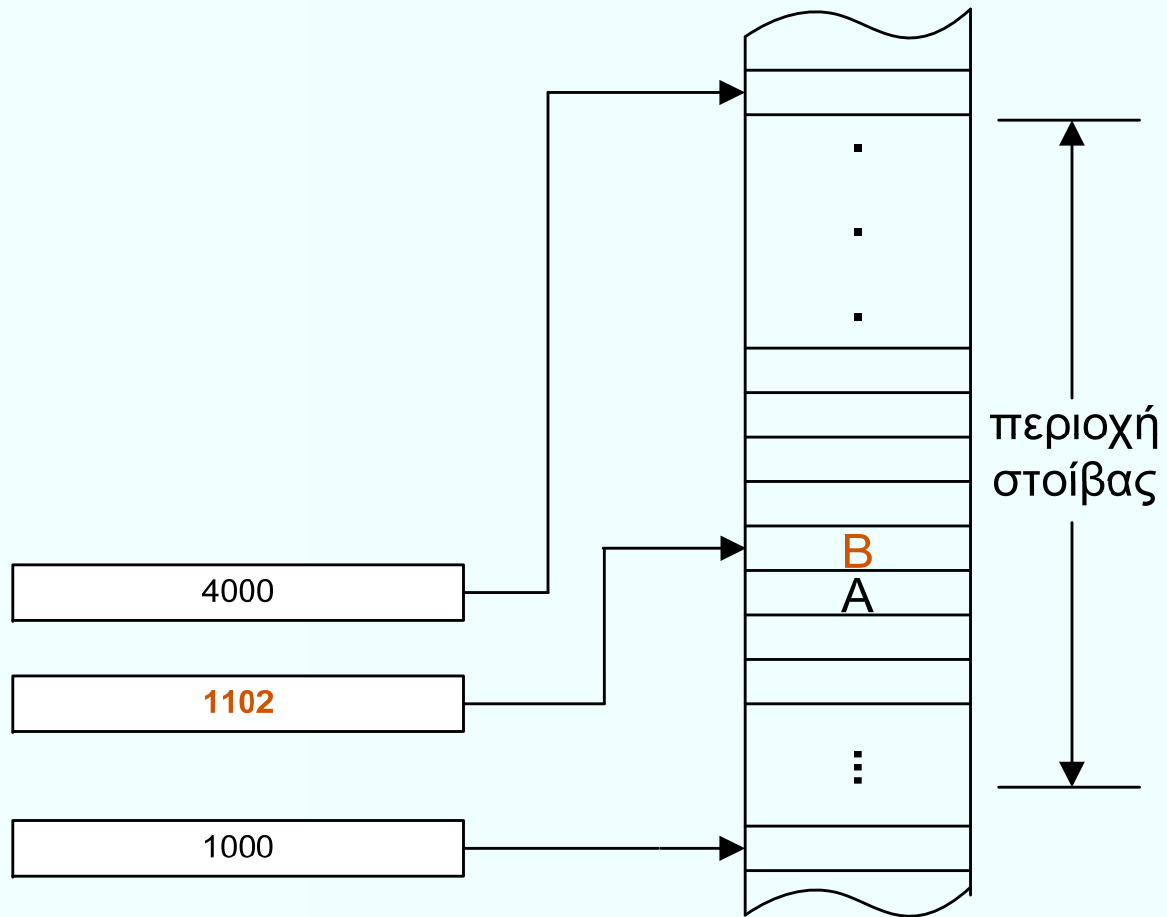
4000

δείκτης στοίβας

1102

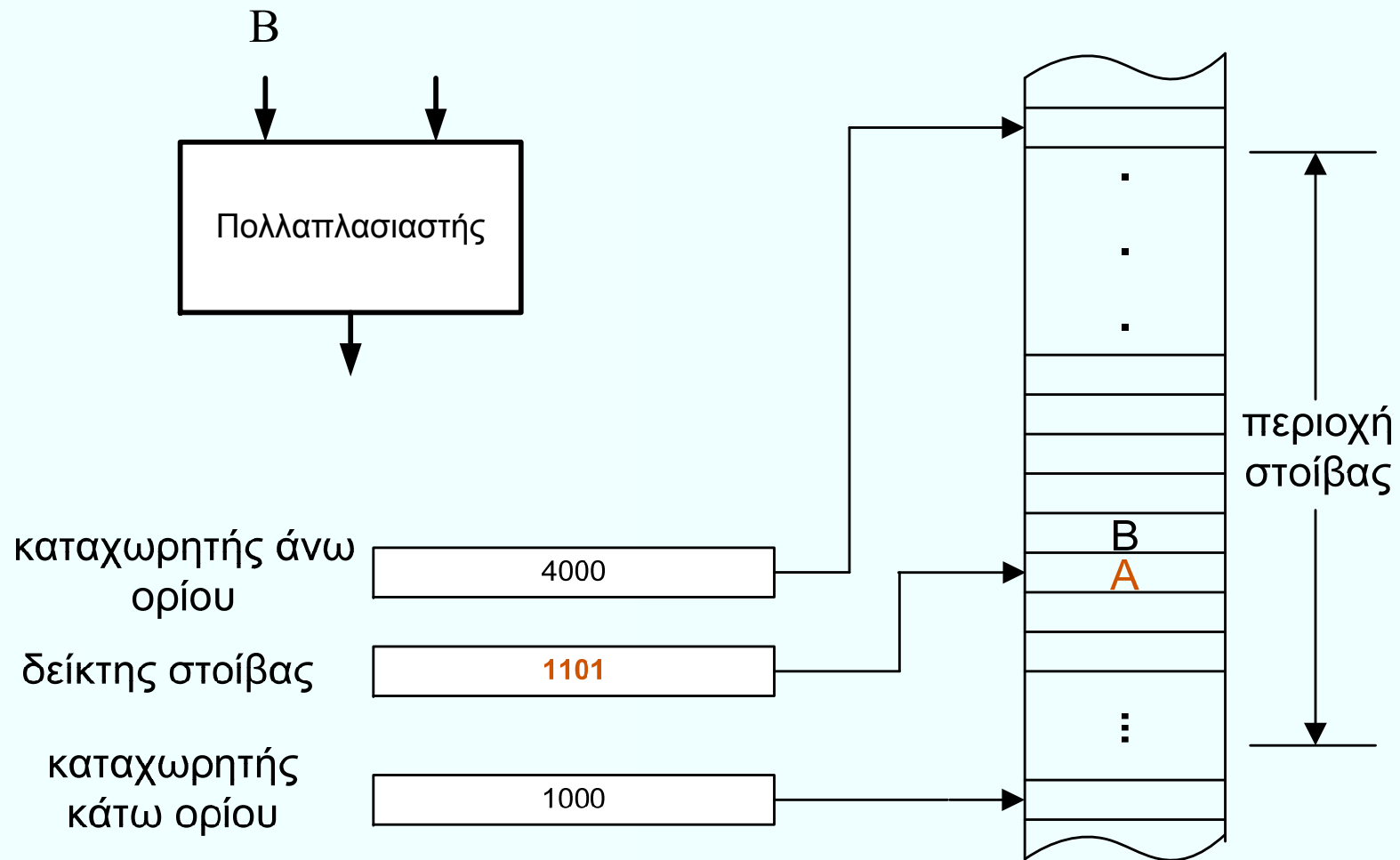
καταχωρητής
κάτω ορίου

1000

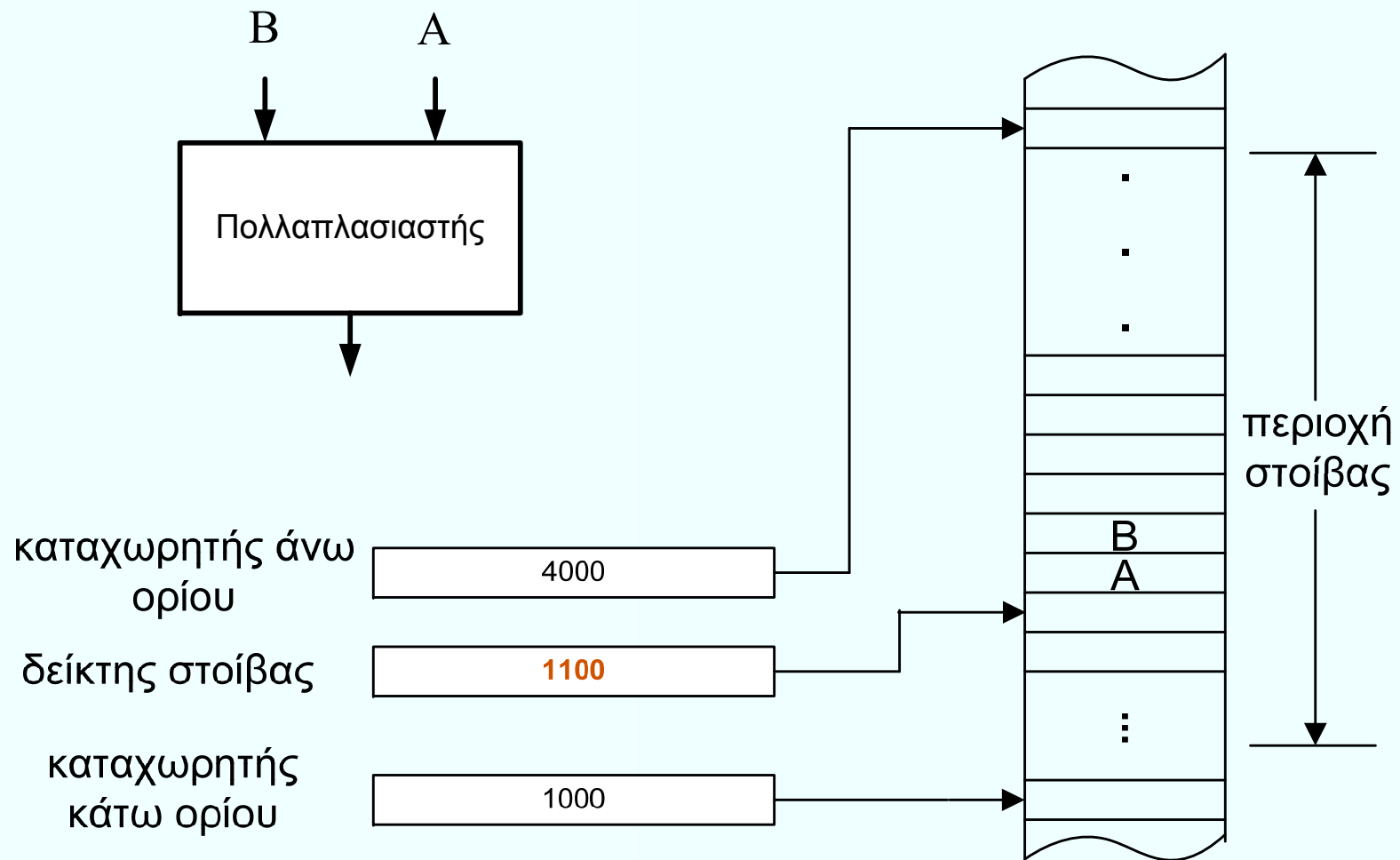


Εκτέλεση της εντολής “MUL”: Πρώτο βήμα

Η εντολή “MUL” εκτελείται σε τρία βήματα



Εκτέλεση της εντολής “MUL”: Δεύτερο βήμα



Μετά την εκτέλεση της εντολής “MUL”

Πρόγραμμα

```
PUSH A  
PUSH B  
MUL  
PUSH C  
ADD
```

καταχωρητής άνω
ορίου

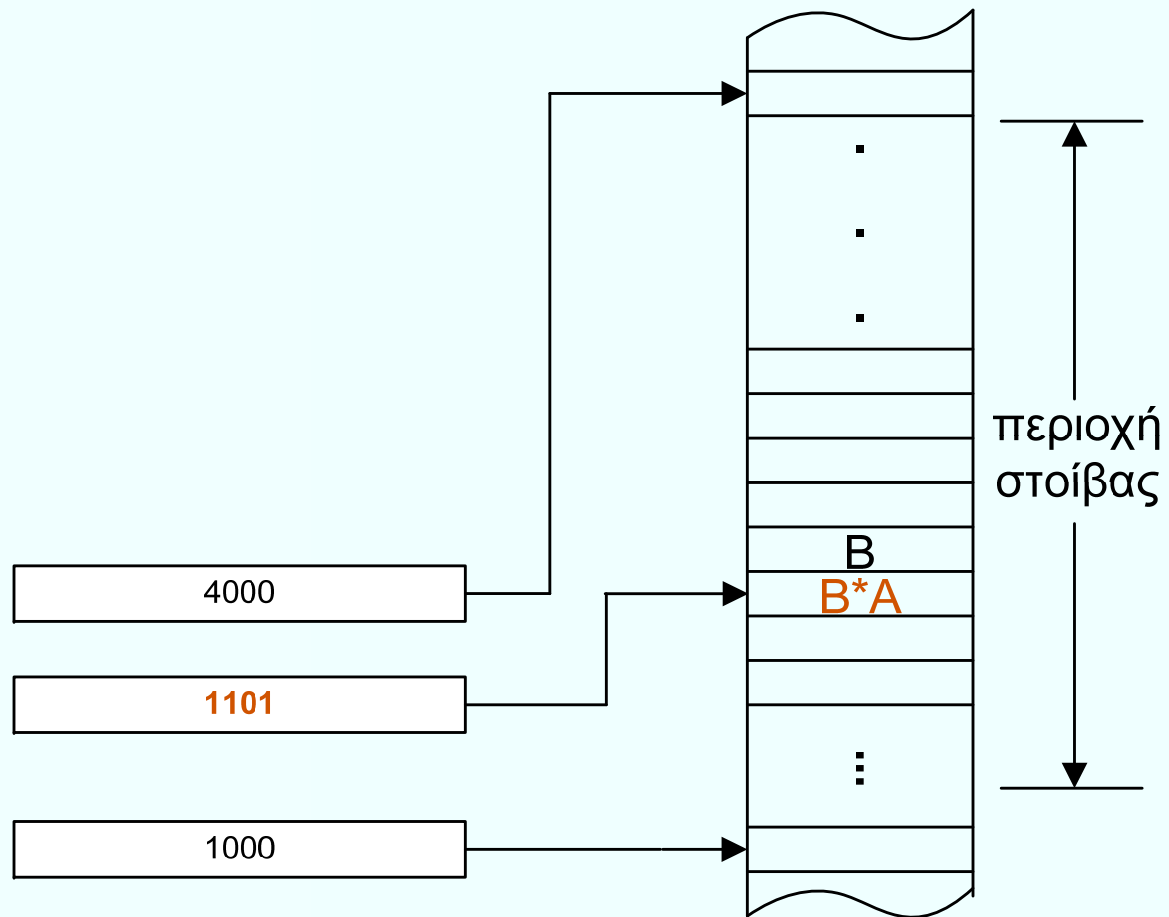
4000

δείκτης στοίβας

1101

καταχωρητής
κάτω ορίου

1000



Μετά την εκτέλεση της εντολής “PUSH C”

Πρόγραμμα

```
PUSH A  
PUSH B  
MUL  
PUSH C  
ADD
```

καταχωρητής άνω
ορίου

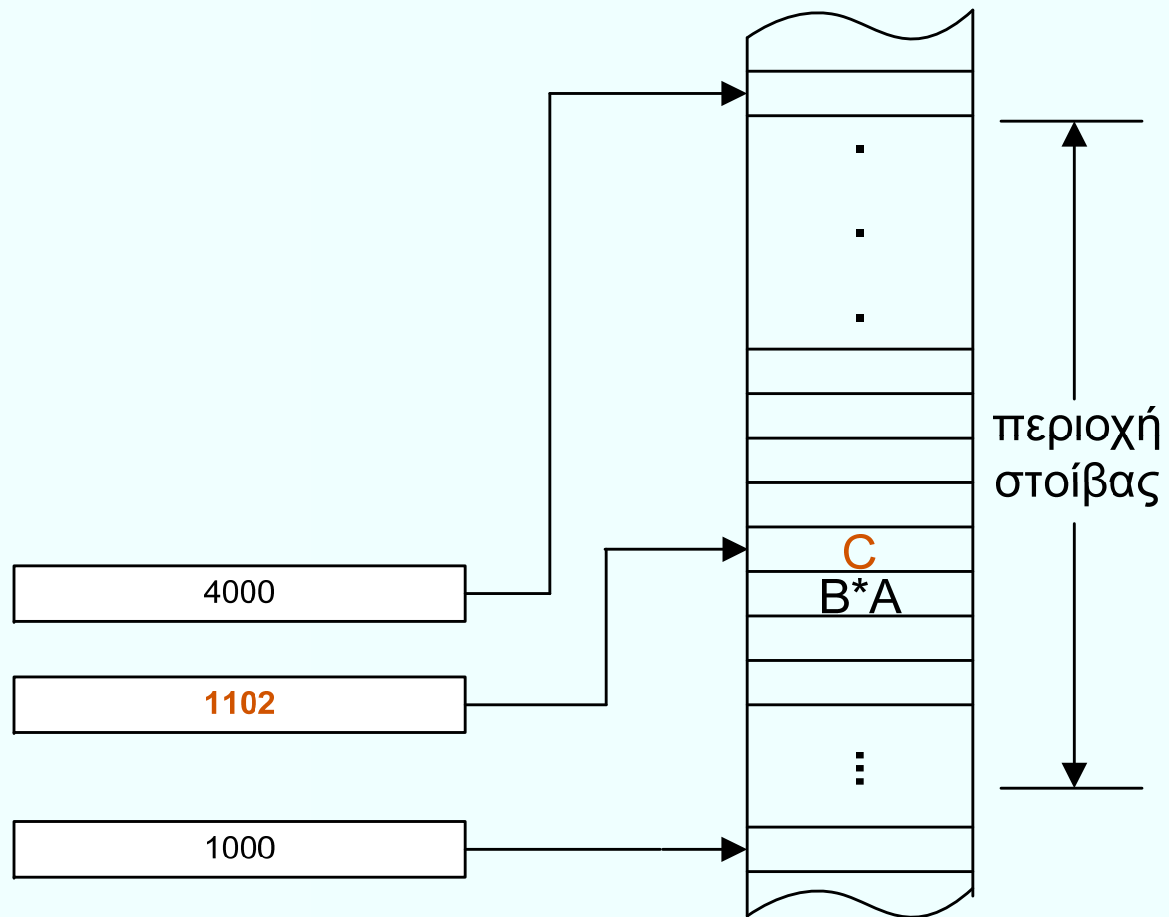
4000

δείκτης στοίβας

1102

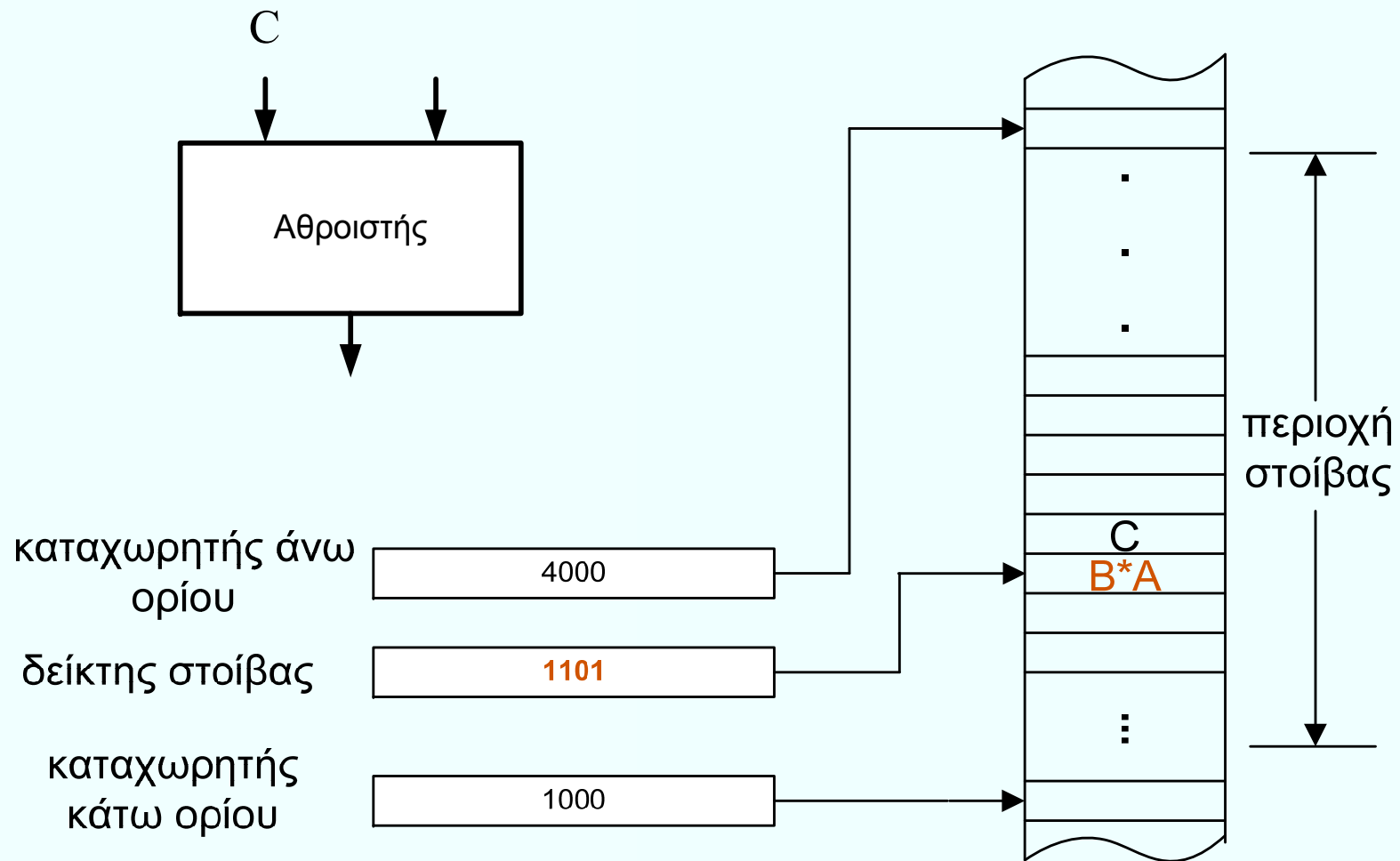
καταχωρητής
κάτω ορίου

1000

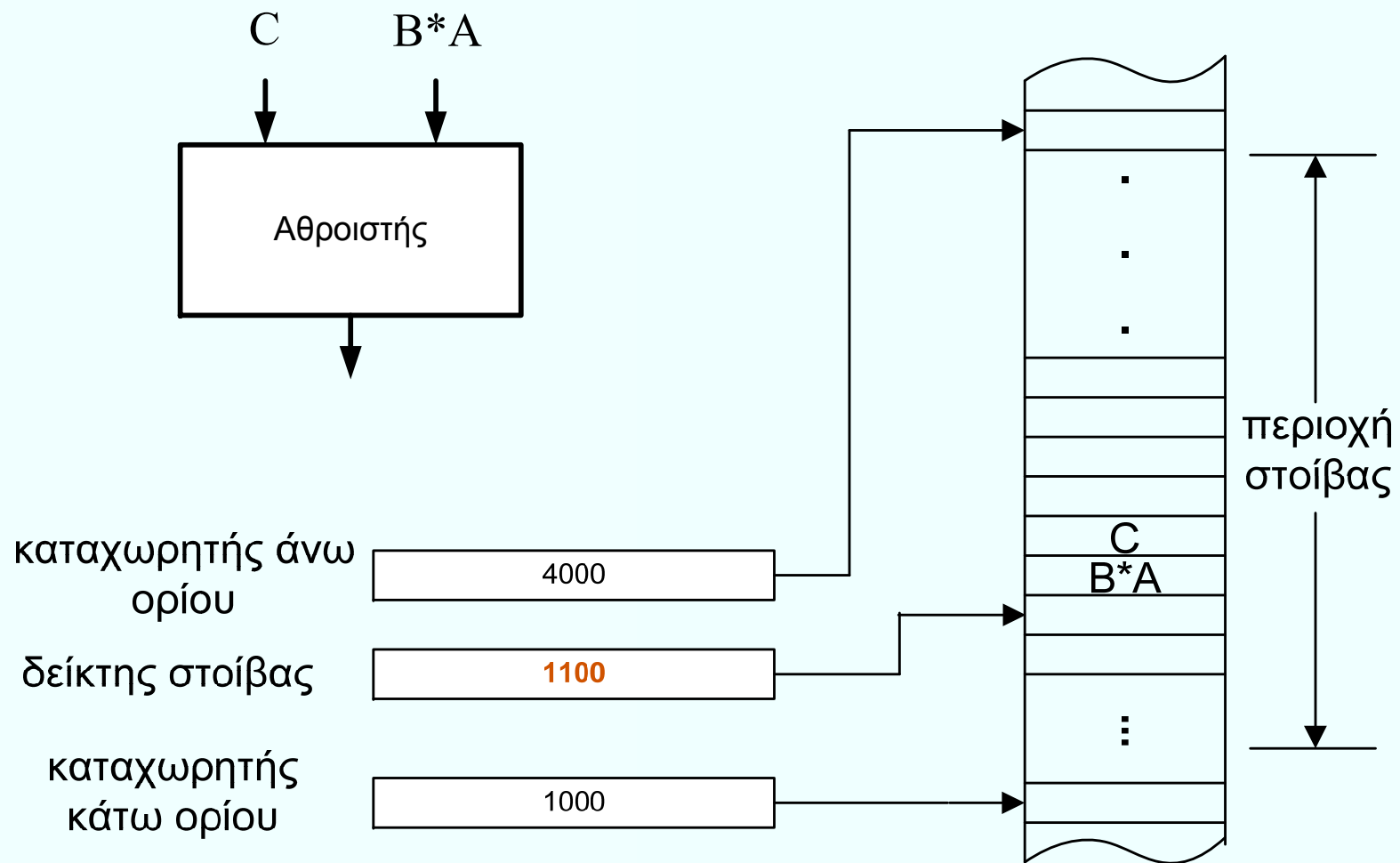


Εκτέλεση της εντολής “ADD”: Πρώτο βήμα

Η εντολή “ADD” εκτελείται σε τρία βήματα



Εκτέλεση της εντολής “ADD”: Δεύτερο βήμα



Μετά την εκτέλεση της εντολής “ADD”

Πρόγραμμα

```
PUSH A  
PUSH B  
MUL  
PUSH C  
ADD
```

καταχωρητής άνω
ορίου

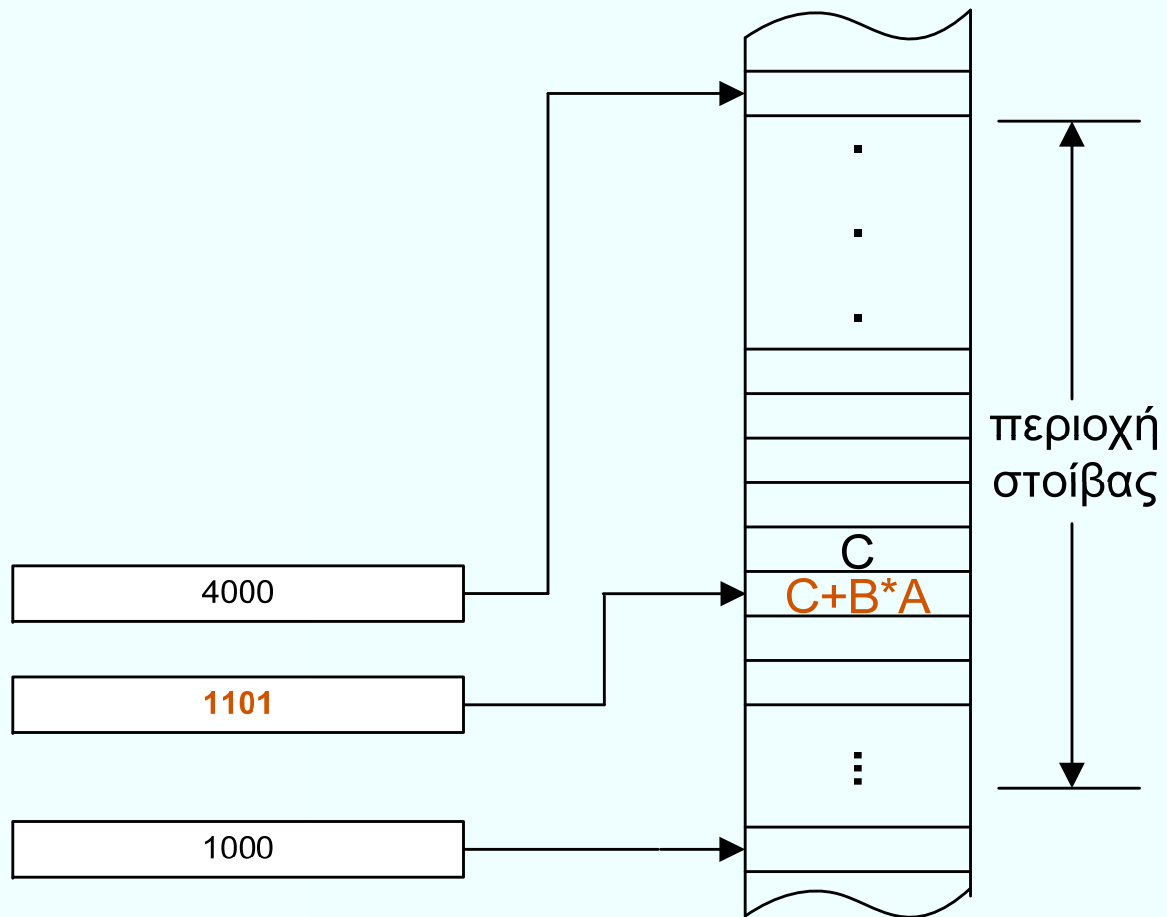
4000

δείκτης στοίβας

1101

καταχωρητής
κάτω ορίου

1000



Αρχιτεκτονική μηχανισμού στοίβας: infix συμβολισμός → postfix συμβολισμό

$$E = A * (B + D) - C * F$$

$$\begin{aligned} E &= A * (B + D) - C * F = \\ &= A * \underline{BD+} - \underline{CF*} = \\ &= \underline{ABD+*} - \underline{CF*} = \\ &= \underline{CF*ABD+*} - \end{aligned}$$

Αρχιτεκτονική μηχανισμού στοίβας: Παράδειγμα εκτέλεσης προγράμματος

$$E = A * (B + D) - C * F = \underline{CF * ABD + * } -$$

ΕΝΤΟΛΗ

ΣΤΟΙΒΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ ΕΝΤΟΛΗΣ
(ΚΟΡΥΦΗ ΣΤΟΙΒΑΣ ΣΤΑ ΑΡΙΣΤΕΡΑ)

<i>PUSH C</i>	<i>C</i>
<i>PUSH F</i>	<i>F, C</i>
<i>MUL</i>	<i>F * C</i>
<i>PUSH A</i>	<i>A, F * C</i>
<i>PUSH B</i>	<i>B, A, F * C</i>
<i>PUSH D</i>	<i>D, B, A, F * C</i>
<i>ADD</i>	<i>D + B, A, F * C</i>
<i>MUL</i>	<i>(D + B) * A, F * C</i>
<i>SUB</i>	<i>(D + B) * A - F * C</i>
<i>POP E</i>	<i>KENO</i>

Αρχιτεκτονική συσσωρευτή: μορφή εντολών

μορφή εντολής



Αρχιτεκτονική συσσωρευτή: Παράδειγμα εκτέλεσης προγράμματος

$$\underline{E = A * (B + D) - C * F}$$

ΕΝΤΟΛΗ

ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ

Σ = ΣΥΣΣΩΡΕΥΤΗΣ)

LOAD C

Σ = C

MUL F

Σ = C * F

STORE E

E = C * F

LOAD B

Σ = B

ADD D

Σ = B + D

MUL A

Σ = (B + D) * A

SUB E

Σ = (B + D) * A - C * F

STORE E

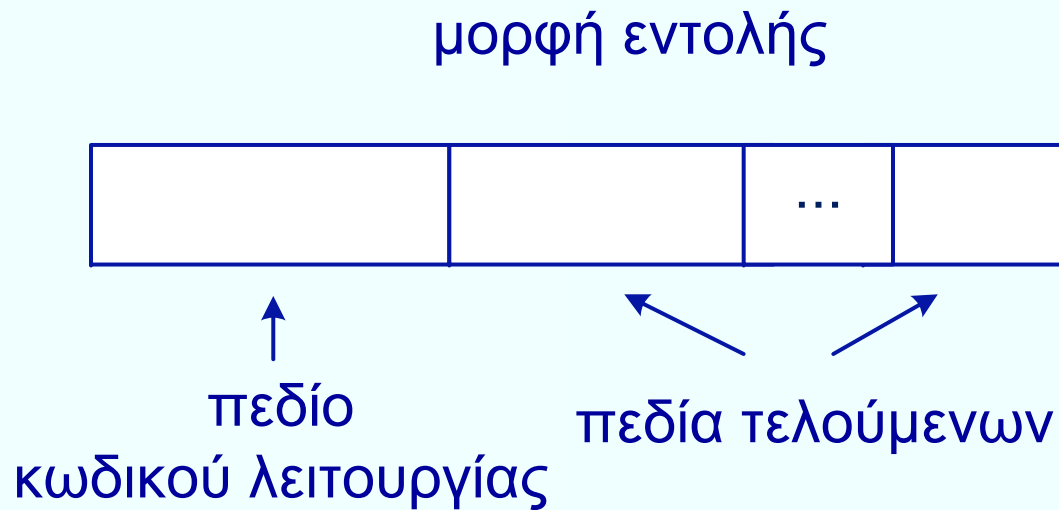
E = (B + D) * A - C * F

Αρχιτεκτονικές καταχωρητών γενικού σκοπού

» καταχωρητή-μνήμης

» καταχωρητή –καταχωρητή

Αρχιτεκτονική καταχωρητή-μνήμης: μορφή εντολών



Παράδειγμα αρχιτεκτονικής καταχωρητή- μνήμης: εκτέλεση προγράμματος

$$\underline{E = A * (B + D) - C * F}$$

ΕΝΤΟΛΗ

ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ

LOAD R1, C

$$R1 = C$$

MUL R1, F

$$R1 = R1 * F = C * F$$

STORE E, R1

$$E = R1 = C * F$$

LOAD R2, B

$$R2 = B$$

ADD R2, D

$$R2 = R2 + D = B + D$$

MUL R2, A

$$R2 = R2 * A = (B + D) * A$$

SUB R2, E

$$R2 = R2 - E = (B + D) * A - C * F$$

STORE E, R2

$$E = R2 = (B + D) * A - C * F$$

Αρχιτεκτονική καταχωρητή- καταχωρητή: μορφή εντολών

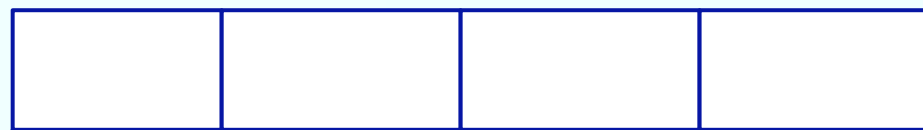
μορφή εντολής LOAD ή STORE



↑
πεδίο
κωδικού
λειτουργίας

↙ ↘
πεδία τελούμενων

μορφή εντολής εκτέλεσης αριθμητικής ή
λογικής πράξης



↑
πεδίο
κωδικού
λειτουργίας

↙ ↑ ↘
πεδία τελούμενων

Αρχιτεκτονική καταχωρητή-καταχωρητή: Παράδειγμα εκτέλεσης προγράμματος

$$\underline{E = A * (B + D) - C * F}$$

ΕΝΤΟΛΗ

ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ

LOAD R1, A

R1 = A

LOAD R2, B

R2 = B

LOAD R3, D

R3 = D

ADD R4, R2, R3

R4 = R2 + R3 = B + D

MUL R5, R1, R4

*R5 = R1 * R4 = A * (B + D)*

LOAD R6, C

R6 = C

LOAD R7, F

R7 = F

MUL R8, R6, R7

*R8 = R6 * R7 = C * F*

SUB R9, R5, R8

*R9 = A * (B + D) - C * F*

STORE E, R9

*E = R9 = A * (B + D) - C * F*

Αρχιτεκτονική με εντολές καταχωρητή-μνήμης και καταχωρητή -καταχωρητή: Παράδειγμα εκτέλεσης προγράμματος

$$\underline{E = A * (B + D) - C * F}$$

ΕΝΤΟΛΗ

LOAD R1, C

MUL R1, F

LOAD R2, B

ADD R2, D

MUL R2, A

SUB R2, R1

STORE E, R2

ΑΠΟΤΕΛΕΣΜΑ ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΗΣ

$$R1 = C$$

$$R1 = R1 * F = C * F$$

$$R2 = B$$

$$R2 = R2 + D = B + D$$

$$R2 = R2 * A = (B + D) * A$$

$$R2 = R2 - R1 = (B+D) * A - C * F$$

$$E = R2 = (B+D) * A - C * F$$

Σύνολο εντολών γλώσσας μηχανής

- Πρέπει να υποστηρίζει τις λειτουργίες που είναι χρήσιμες
- Πρέπει να επιτρέπει την αποδοτική υλοποίηση όσον αφορά
 - Ταχύτητα εκτέλεσης
 - Κόστος υλοποίησης
 - Κατανάλωση ισχύος

Σύνολο εντολών γλώσσας μηχανής

**η εννοιολογική διαφορά μεταξύ
γλωσσών προγραμματισμού υψηλού
επιπέδου
και
γλώσσας μηχανής
γεφυρώνεται από τον μεταγλωττιστή**

**Άρα ποια είναι τα χαρακτηριστικά που
πρέπει να έχει το σύνολο εντολών;**

Υπολογιστές πολύπλοκου συνόλου εντολών, CISC (Complex Instruction Set Computers)

Το σύνολο των εντολών σε επίπεδο γλώσσας μηχανής περιλαμβάνει:

- Πολύπλοκες και πανίσχυρες εντολές που εννοιολογικά βρίσκονται κοντά στις εντολές των γλωσσών προγραμματισμού υψηλού επιπέδου
- Μεγάλο αριθμό από κωδικούς λειτουργίας, τρόπους διευθυνσιοδότησης και είδη δεδομένων

Υπολογιστές πολύπλοκου συνόλου εντολών

Παράδειγμα εντολής

$\text{mem}(r1) = \text{mem}(r2+r3) * \text{mem}(r4+\text{disp})$

Υπολογιστές πολύπλοκου συνόλου εντολών

Μειονεκτήματα:

- α. οι μεταγλωττιστές δεν μπορούσαν να εκμεταλλευτούν πάντα με τον καλύτερο τρόπο τις πολύπλοκες εντολές γλώσσας μηχανής
- β. πολύπλοκη μονάδα ελέγχου με αποτέλεσμα να επιβραδύνεται η εκτέλεση και των πιο απλών εντολών
- γ. η μονάδα ελέγχου καταλάμβανε μεγάλο μέρος του επεξεργαστή

Υπολογιστές απλού συνόλου εντολών, RISC (Reduced Instruction Set Computers)

Το σύνολο των εντολών σε επίπεδο γλώσσας μηχανής περιλαμβάνει:

- απλές εντολές που εκτελούν στοιχειώδεις λειτουργίες
- εντολές με μεγάλο βαθμό κανονικότητας
- μικρός αριθμός από απλούς τρόπους διευθυνσιοδότησης

Υπολογιστές πολύπλοκου συνόλου εντολών και Υπολογιστές απλού συνόλου εντολών

Εντολή σε CISC

mem(r1) = mem(r2+r3)*mem(r4+disp)

Πρόγραμμα σε RISC

r5 = r2+r3

/* υπολογισμός της διεύθυνσης του
πρώτου αριθμού */

loadf fp1, (r5)

/* μεταφορά σε καταχωρητή κινητής
υποδιαστολής του πρώτου αριθμού */

loadf fp2, (r4+disp)

/* μεταφορά σε καταχωρητή κινητής
υποδιαστολής του δεύτερου αριθμού*/

multf fp3, fp1 fp2

/* πολλαπλασιασμός αριθμών κινητής
υποδιαστολής */

store (r1), fp3

/* αποθήκευση αποτελέσματος στη
μνήμη */

Υπολογιστές απλού συνόλου εντολών

Πλεονεκτήματα

- Η μετάφραση από μια γλώσσα προγραμματισμού σε γλώσσα μηχανής είναι απλή
- Απαιτείται λιγότερη επιφάνεια πυριτίου για την υλοποίηση του επεξεργαστή.
- Το απλούστερο σύνολο εντολών οδηγεί επίσης σε μικρότερους σχεδιαστικούς χρόνους, ευκολότερη επιβεβαίωση σχεδιασμού και ευκολότερο έλεγχο ορθής λειτουργίας

Μειονεκτήματα

- Για συγγραφή του ίδιου προγράμματος απαιτείται μεγαλύτερο πλήθος εντολών γλώσσας μηχανής

Κωδικοποίηση Συνόλου Εντολών

Ο τρόπος αναπαράστασης των εντολών μέσω του δυαδικού επηρεάζει:

- α. Το μέγεθος του προγράμματος σε γλώσσα μηχανής
- β. Την υλοποίηση της μονάδας ελέγχου

Κωδικοποίηση Συνόλου Εντολών

- α. Κωδικοποίηση του τρόπου διευθυνσιοδότησης
- β. Έχουν όλες οι εντολές το ίδιο μήκος;
 - » εντολές με διαφορετικά μήκη: *VAX*
 - » κάθε είδος εντολής έχει σταθερό μήκος: *Intel 80x86*
 - » όλες οι εντολές το ίδιο μήκος: *MIPS, PowerPC, SPARC*

Υποστήριξη γλωσσών προγραμματισμού υψηλού επιπέδου

α. Εντολές

β. Υποστήριξη χρόνου ζωής μεταβλητών

γ. Υποστήριξη συναρτήσεων και διαδικασιών

Χρήση μνήμης

