



Αρχιτεκτονική Υπολογιστών

Ενότητα 3: Καταχωρητές, Τμήματα, Διευθυνσιοδότηση
Μνήμης, SEGMENT, MOV, ADD, SUB, INT, TITLE, LEA

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ece.uowm.gr/mdasyg>



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Πρόσβαση στη μνήμη

- Το εύρος του διαύλου διευθύνσεων είναι 20 bit.
 - Επομένως ο 8086 έχει δυνατότητα προσπέλασης **1 Megabyte** θέσεων μνήμης.
 - Το σύνολο των δ/νσεων είναι από 00000h έως 0FFFFFFh.
- Το περιεχόμενο κάθε θέσης μνήμης είναι **1 byte**.
- Στην μνήμη του μπορούμε να θεωρήσουμε τμήματα (segments) μνήμης καθένα από τα οποία είναι μεγέθους 64 Kbytes.



Τμήματα μνήμης

- Η αρχή κάθε τμήματος από το επόμενο απέχει 16 bytes, δηλαδή τα τμήματα αρχίζουν από δ/νσεις μνήμης που διαιρούνται με το 16 και ονομάζονται παράγραφοι.
- Επομένως έχουμε το μέγιστο 65536 (1MB/16) παραγράφους μνήμης.
- Ο χωρισμός της μνήμης σε τμήματα (segmentation) προήλθε κύρια από το γεγονός ότι είναι αδύνατη η διευθυνσιοδότηση της διαθέσιμης μνήμης εσωτερικά με μόνον ένα από τους διαθέσιμους καταχωρητές των 16 bits του 8086.



Η διευθυνσιοδότηση της μνήμης γίνεται με 2 καταχωρητές (1/2)

- Για την προσπέλαση σε μια θέση μνήμης πρέπει να τεθεί η φυσική διεύθυνση (physical address) των 20 bits στο δίαυλο των διευθύνσεων.
- Για το σχηματισμό της δ/νσης αυτής εσωτερικά στον 8086 χρησιμοποιούνται δύο καταχωρητές.
- Σε έναν από τους καταχωρητές τμημάτων τοποθετείται ο αύξων αριθμός της παραγράφου αρχής ενός τμήματος μνήμης.
- Ο δεύτερος καταχωρητής ο οποίος ονομάζεται κατά περίπτωση pointer ή index (δείκτης) περιέχει την απόσταση από την αρχή του τμήματος, η οποία ονομάζεται λογική διεύθυνση ή μετατόπιση ή ενεργός διεύθυνση (logical address, offset, effective address).



Η διευθυνοδοδότηση της μνήμης γίνεται με 2 καταχωρητές (2/2)

- Ο τελικός προσδιορισμός της φυσικής δ/νσης προκύπτει από την πρόσθεση της λογικής δ/νσης στην δ/νση αρχής του τμήματος σύμφωνα με την απλή σχέση:

$$\langle \text{φυσική διεύθυνση} \rangle = \langle \text{τμήμα} \rangle * 16 + \langle \text{μετατόπιση} \rangle$$

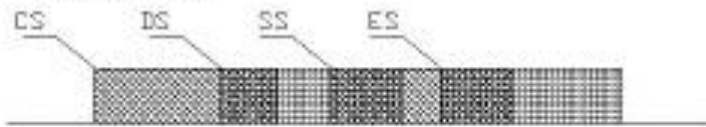
ή

$$\langle \text{φυσική διεύθυνση} \rangle = \langle \text{τμήμα} \rangle * 10h + \langle \text{μετατόπιση} \rangle$$

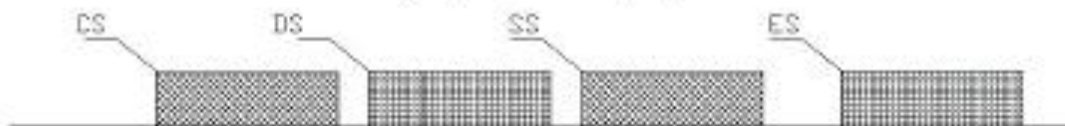


4 είναι τα βασικά τμήματα

Αλληλοεπικαλυπτόμενα
τμήματα



Ανεξάρτητα τμήματα



Ταυτιζόμενα τμήματα

CS=DS=ES=SS



Τμήμα Κώδικα (Code Segment)

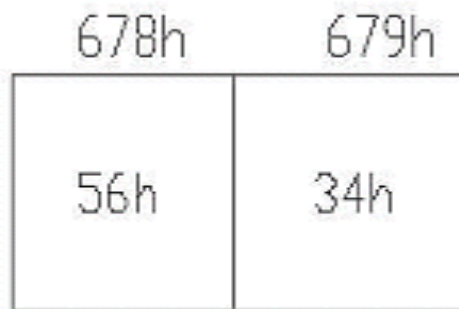
- Ο καταχωρητής **CS** περιέχει την δ/νση αρχής του τμήματος όπου έχουν αποθηκευτεί οι κωδικοί των εντολών του προγράμματος (CODE). Η μετατόπιση στο τμήμα αυτό καθορίζεται αποκλειστικά και μόνο από τον καταχωρητή **IP**.
- Το δίδυμο δηλ. CS:IP αντικαθιστά τον μετρητή προγράμματος (PC) των άλλων επεξεργαστών Π.χ
- αν CS = 345Ah και IP = 712Ch, τότε η φυσική δ/νση της επόμενης εντολής είναι:

$$\mathbf{CS*10h + IP = 345Ah*10h + 712Ch = 345A0h + 712Ch = 3B6CCh}$$



Προσπέλαση των Λέξεων στη Μνήμη

- Το περιεχόμενο κάθε θέσης μνήμης είναι 1 byte. Οι λέξεις (word = 2 bytes) δεδομένα αποθηκεύονται με το περισσότερο σημαντικό byte στην χαμηλότερη δ/νση.
- Έτσι π.χ. ο αριθμός 34 56h θα αποθηκευτεί, στις διαδοχικές δ/νσεις 678h και 679h σαν 56 34h σύμφωνα με το σχήμα:



Τοποθέτηση 4 byte στη μνήμη

Οι διπλές λέξεις δεδομένων όπως αυτές που αφορούν πραγματικές δ/ (segment:offset) αποθηκεύονται σε διαδοχικές θέσεις μνήμης αντίστροφα. Π $DS=1234h$ και $SI=5678h$ η πραγματική δ/νση (pointer) $DS:SI$ αποθηκεύεται διαδοχικές θέσεις μνήμης $0300h$ έως και $0303h$ ως εξής:

Η $DS:SI = 1234:5678h$ τοποθετείται ως εξής:



Η εντολή MOV (1/2)

=> Η εντολή MOV τοποθετεί μια τιμή σε έναν καταχωρητή.

- Δέχεται 2 παραμέτρους. Σύνταξη:

MOV (πaráμετροςA) , (πaráμετροςB) .

- Η τιμή γράφεται στην (πaráμετροςA).
- Η τιμή που θα γραφεί καθορίζεται από την (πaráμετροςB).

Παράδειγμα:

```
MOV AX,02
```

Γράφεται η τιμή 02 στον καταχωρητή AX.



Η εντολή MOV (2/2)

- Επιτρέπονται οι εξής μετακινήσεις:
 - καταχωρητή σε καταχωρητή (π.χ MOV AX,BX).
 - καταχωρητή σε θέση μνήμης (π.χ MOV APOI,AL).
 - θέση μνήμης σε καταχωρητή (π.χ MOV DL,APOI).
 - σταθερά τιμή σε καταχωρητή (π.χ MOV AH,21h).
 - σταθερά τιμή σε θέση μνήμης (π.χ MOV ARII,5).
 - Η τελευταία περίπτωση δεν επιτρέπεται σε καταχωρητές τμημάτων.
MOV DS,DATASEG; δεν επιτρέπεται.
- Και οι δυο τελεστές πρέπει να είναι ίσου μήκους. Δεν επιτρέπεται π.χ. MOV ax,dl.



Οι διπλοί καταχωρητές

- Οι καταχωρητές AX, BX, CX, DX είναι 16bit.
- Μπορούμε να χρησιμοποιήσουμε όμως όχι ολόκληρο τον κάθε καταχωρητή, αλλά το “μισό”.
- Ο κάθε καταχωρητής μπορεί να υποστηρίζει 2 ειδών προσπελάσεις:
 - είτε ως ένας καταχωρητής των 16bit,
 - είτε ως δύο καταχωρητές των 8bit.

ΠΡΟΣΟΧΗ:

Και με τους δυο τρόπους προσπέλασης έχουμε προσπέλαση στον **ίδιο χώρο μνήμης**.



Διπλή πρόσβαση

- Οι προηγούμενοι καταχωρητές των 2 Byte υποστηρίζουν πρόσβαση είτε στο υψηλότερης αξίας Byte (High Byte, H), είτε στο χαμηλότερης αξίας Byte (Low Byte, L) ως εξής:

AX (2 Byte) := AH (High Byte), AL (Low Byte)

BX (2 Byte) := BH (High Byte), BL (Low Byte)

CX (2 Byte) := CH (High Byte), CL (Low Byte)

DX (2 Byte) := DH (High Byte), DL (Low Byte)



Παραδείγματα

- Η εντολή `MOV AX, 1234h`
γράφει την τιμή 1234 στον AX.
=>Ο AH έχει τώρα τιμή 12h, ο AL 34h.
- Αν τώρα δώσουμε `MOV AL,45h`
δηλαδή, γράφουμε την τιμή 45h στον AL.
=>Ο AH έχει τιμή 12h, ο AX 1245h.
- Στη συνέχεια με την εντολή `MOV AH,00.`
=>Ο AH έχει τιμή 00, ο AL 45h και ο AX 0045h.



Η εντολή LEA

=> Η εντολή LEA (load effective address) βρίσκει τη διεύθυνση μνήμης μιας ετικέτας και την τοποθετεί στον καταχωρητή στην πρώτη παράμετρο.

- Δέχεται 2 παραμέτρους. Σύνταξη:

LEA (παράμετροςA) , (παράμετροςB)

- Η διεύθυνση μνήμης της παραμέτρουB τοποθετείται στην παράμετρο A.

Παράδειγμα:

LEA DX, message

Η διεύθυνση μνήμης που βρίσκεται το message τοποθετείται στον καταχωρητή DX.



Η λέξη SEGMENT

=> Δεν είναι εντολή, αλλά οδηγία προς τον μεταφραστή. Ορίζει την αρχή ενός τμήματος.

- Δέχεται 1 παραμέτρο. Σύνταξη:

SEGMENT (πaráμετροςA) . . . (πaráμετροςA) **ENDS**

- Ξεκινάει ένα νέο τμήμα με όνομα (πaráμετροςA) το οποίο τελειώνει στη γραμμή που υπάρχει το ENDS (End Segment).
- Θα πρέπει να ορίσουμε στο πρόγραμμα το κάθε τμήμα τι είναι με το να τοποθετήσουμε τη διεύθυνση του σε κάποιον από τους καταχωρητές CS (code), DS (data), SS (stack), ES (extra).



Η λέξη db

=> Δεν είναι εντολή, αλλά οδηγία προς τον μεταφραστή. Ορίζει το μέγεθος μνήμης (ότι είναι byte) των στοιχείων που ακολουθούν.

- Τα στοιχεία θα πρέπει να έχουν τιμή από 0 έως 255.
- Δέχεται πολλαπλές παραμέτρους. Σύνταξη:

```
mess db 'Hi there',10,13,'$'
```

- Δηλώνεται ότι κάθε χαρακτήρας που ακολουθεί μετά το db θα καταλαμβάνει 1 byte (εκτός από τα ' "). Στο συγκεκριμένο παράδειγμα 8Byte + 2Byte+1Byte = 11 Byte.
- Όταν ένα στοιχείο βρίσκεται μέσα σε απλά ή διπλά εισαγωγικά, τότε μπαίνει η ASCII τιμή του.



Ετικέτες Διευθύνσεων μεταβλητών

```
mess db 'Hi there',10,13,'$'
```

- Με την ανωτέρω δήλωση στο παράδειγμα αυτό δημιουργείται η ετικέτα μεταβλητής (VAR) με το όνομα mess.
- Οι ετικέτες αντιστοιχούν σε διευθύνσεις μνήμης.
- Η ετικέτα μιας μεταβλητής ΠΑΝΤΑ δείχνει στη διεύθυνση του πρώτου στοιχείου (αν έχει πολλά στοιχεία). Στο παράδειγμα αυτό η ετικέτα mess δείχνει στη διεύθυνση που αποθηκεύεται το 'H'.

Άλλα παραδείγματα:

```
number1 db 0
```

```
array123 db 1,2,3
```



Η λέξη TITLE

=> Δεν είναι εντολή, αλλά οδηγία προς το μεταφραστή.
Ορίζει το όνομα το προγράμματος.

- Είναι προαιρετικό.

Παράδειγμα:

TITLE HELLOWORLD

- Δίνει την οδηγία προς το μεταφραστή ότι το πρόγραμμα που ακολουθεί θα ονομαστεί HELLOWORLD.



Η λέξη END

⇒ Δεν είναι εντολή, αλλά οδηγία προς το μεταφραστή. Ορίζει το τέλος του προγράμματος και το σημείο που θα ξεκινήσει η εκτέλεση.

Παράδειγμα:

END ARXH

- Δίνει την οδηγία προς το μεταφραστή:
 - ότι το πρόγραμμα έχει ολοκληρωθεί,
 - και θα πρέπει να ξεκινήσει η εκτέλεση από το σημείο που βρίσκεται η ετικέτα μνήμης ARXH:
- Ότι υπάρχει ύστερα από την εντολή αυτή θα αγνοηθεί από το μεταφραστή.



Η εντολή INT

=> Η εντολή INT (Interrupt) προκαλεί μια διακοπή στο πρόγραμμα προκειμένου να εκτελεστεί μια συνάρτηση από το λειτουργικό σύστημα.

- Δέχεται 1 παραμέτρο. Σύνταξη:

INT (παράμετροςA)

- Η (παράμετροςA) είναι ο αριθμός διακοπής.
- Ο πιο διαδεδομένος αριθμός διακοπής είναι το 21h (=33).

Παράδειγμα:

INT 21h

- Απαιτείται στις προηγούμενες 1 ή 2 εντολές να έχουν τοποθετηθεί οι κατάλληλες τιμές σε καταχωρητές.
- Απαιτείται τιμή σίγουρα στον καταχωρητή AH (ή AX).



Παραδείγματα int 21h

- Τερματισμός Προγράμματος:

```
MOV AH,4Ch
```

```
INT 21h
```

- Εκτύπωση string με ετικέτα msg στην οθόνη:

```
LEA DX,msg
```

```
MOV AH,09
```

```
INT 21h
```

*** ΠΡΟΣΟΧΗ: Το msg για να εκτυπωθεί στην οθόνη πρέπει να έχει ως τελευταίο χαρακτήρα το \$.



Η εντολή ADD

=> Η εντολή ADD (addition) προσθέτει τις 2 παραμέτρους και τοποθετεί το αποτέλεσμα στην πρώτη παράμετρο.

- Δέχεται 2 παραμέτρους. Σύνταξη:

ADD (παράμετροςA) , (παράμετροςB)

- Θα γίνει η πράξη (παράμετροςA)+(παράμετροςB).
- Το αποτέλεσμα θα τοποθετηθεί στην (παράμετροςA).
- Θα πρέπει να έχουν ίδιο μέγεθος.

Παραδείγματα:

ADD AH,AL

ADD DX,30h



Η εντολή SUB

=> Η εντολή SUB (subtraction) αφαιρεί από την πρώτη παράμετρο τη 2η παράμετρο και τοποθετεί το αποτέλεσμα στην πρώτη παράμετρο.

- Δέχεται 2 παραμέτρους. Σύνταξη:

SUB (παράμετροςA) , (παράμετροςB)

- Θα γίνει η πράξη (παράμετροςA)-(παράμετροςB).
- Το αποτέλεσμα θα τοποθετηθεί στην (παράμετροςA).
- Θα πρέπει να έχουν ίδιο μέγεθος.

Παραδείγματα:

```
SUB AX,DX
```

```
SUB BH,30h
```



Μετατροπή από προς το σύστημα ASCII

- Η Είσοδος/Εξοδος διαβάζει γράφει στο σύστημα ASCII.
- Ο επεξεργαστής κάνει πράξεις στο κανονικό αριθμητικό σύστημα.
- Απαιτείται μετατροπή:
Κανονικό Αρ. Σύστημα \Leftrightarrow ASCII.
- Για να γίνει μετατροπή μιας αριθμητικής τιμής από 0 έως 9 προς το σύστημα ASCII ακολουθούμε 2 κανόνες:
Κανονικός Αριθμός + 30h \Rightarrow Αριθμός στο ASCII.
Αριθμός στο ASCII - 30h \Rightarrow Κανονικός Αριθμός.



Παραδείγματα μετατροπών

- Παράδειγμα:
 - Αν θέλουμε να εμφανίσουμε τον αριθμό 7 στην οθόνη, τότε θα πρέπει να προσθέσουμε το 30h, οπότε θα έχουμε τελικό αποτέλεσμα 37h, το οποίο αντιστοιχεί στον χαρακτήρα “7” στον πίνακα ASCII.
 - Αν διαβάσουμε από το πληκτρολόγιο έναν χαρακτήρα π.χ. το “8”, τότε θα έχουμε διαβάσει την ASCII τιμή του “8”, δηλαδή το 38h. Θα πρέπει να αφαιρέσουμε το 30h για έχουμε την πραγματική καθαρή τιμή, δηλαδή το 8.



Εκτύπωση χαρακτήρα στην οθόνη

- Η εμφάνιση χαρακτήρα στην οθόνη είναι λειτουργία Εισόδου/Εξόδου (E/E).
- Απαιτεί τη χρήση του `int 21h`.
- Θα πρέπει το `AH` να έχει τιμή `02`.
- Ο χαρακτήρας που θα εμφανιστεί στην οθόνη θα πρέπει να έχει τοποθετηθεί στον καταχωρητή `DL` με την ASCII τιμή του.

Παραδείγματα:

`mov dl,'H'` `mov dl,67`

`mov ah,02` `mov ah,02`

`int 21h` `int 21h`



Template (A)

(Κοινός σκελετός αρχείου ASM)

```
TITLE MYPROGRAM
DEDOMENA SEGMENT
[.....εδώ πέρα τοποθετούνται τα
δεδομένα.....]
DEDOMENA ENDS
KODIKAS SEGMENT
ARXH: MOV AX,DEDOMENA
      MOV DS,AX
[.....εδώ τοποθετείται ο κύριος
κώδικας του προγράμματος...]

      MOV AH,4CH
      INT 21H
KODIKAS ENDS
END ARXH
```

Όλα τα προγράμματα σε ASM 8086 χρησιμοποιούν το διπλανό κώδικα ως σκελετό. Προσθέτετε δεδομένα και εντολές στις αντίστοιχες περιοχές.



Τέλος Ενότητας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

