



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

# ΕΡΓΑΣΤΗΡΙΟ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ Η/Υ

4<sup>ο</sup> Εξάμηνο

Μαδεμλής Ιωάννης

M.Sc Ηλεκτρονικός Φυσικός



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΠΡΟΦΙΛ ΜΑΘΗΜΑΤΟΣ

Αντικείμενο:

Προγραμματισμός σε γλώσσα *Assembly*

Σκοπός:

- Γνώση της *assembly* από τους απόφοιτους του τμήματος
- Χρήση της *assembly* σε αυτοματισμούς

Μέσο:

- Αναπτυξιακό σύστημα με  $\text{i} / \text{ε}$  Intel 8088 (BGC-8088)
- Επικουρικά: Emulator 8088 για MS Windows



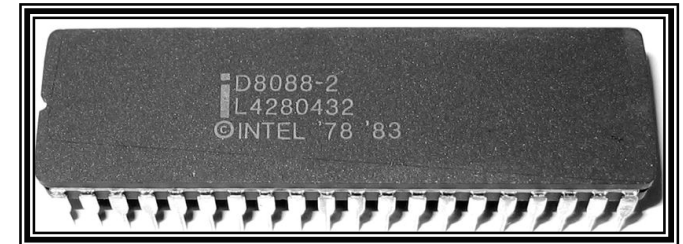
ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ 8088

- Κατασκευαστής: Intel
- Χρονιά παρουσίασης: 1979
- Συχνότητα: 4.77 MHz & 8 MHz
- Αρχιτεκτονική: 16 bit
- Τάση/Ισχύς: 5V / 1.75 Watt
- Data Bus: 16/8 bit
- Address Bus: 20 bit
- Μέγιστη μνήμη: 1 MB (RAM+ROM)
- Bit/θέση μνήμης: 8
- Συσκευασία: 40 pin
- Εντολές: 90





ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ 8088

Καταχωρητές: Μικρές σε μέγεθος, μεγάλης ταχύτητας αποθηκευτικές μονάδες μέσα στον μικροεπεξεργαστή

Καταχωρητές 8088: 14 των 16 bit, γενικού και ειδικού σκοπού

Καταχωρητές ειδικού σκοπού (16 bit) :

- **CS** (Code Segment)
- **DS** (Data Segment)
- **ES** (Extra Segment)
- **SS** (Stack Segment)
- **IP** (Instruction Pointer)
- **FG** (Flag Register)
- **SP** (Stack Pointer)
- **BP** (Base Pointer)



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ 8088

Καταχωρητές γενικού σκοπού (16 bit):

- **AX** (μπορεί να χρησιμοποιηθεί και σαν 2 καταχωρητές 8 bit, **AH, AL**)
- **BX** (μπορεί να χρησιμοποιηθεί και σαν 2 καταχωρητές 8 bit, **BH, BL**)
- **CX** (μπορεί να χρησιμοποιηθεί και σαν 2 καταχωρητές 8 bit, **CH, CL**)
- **DX** (μπορεί να χρησιμοποιηθεί και σαν 2 καταχωρητές 8 bit, **DH, DL**)
- **SI** (Source Index)
- **DI** (Destination Index)

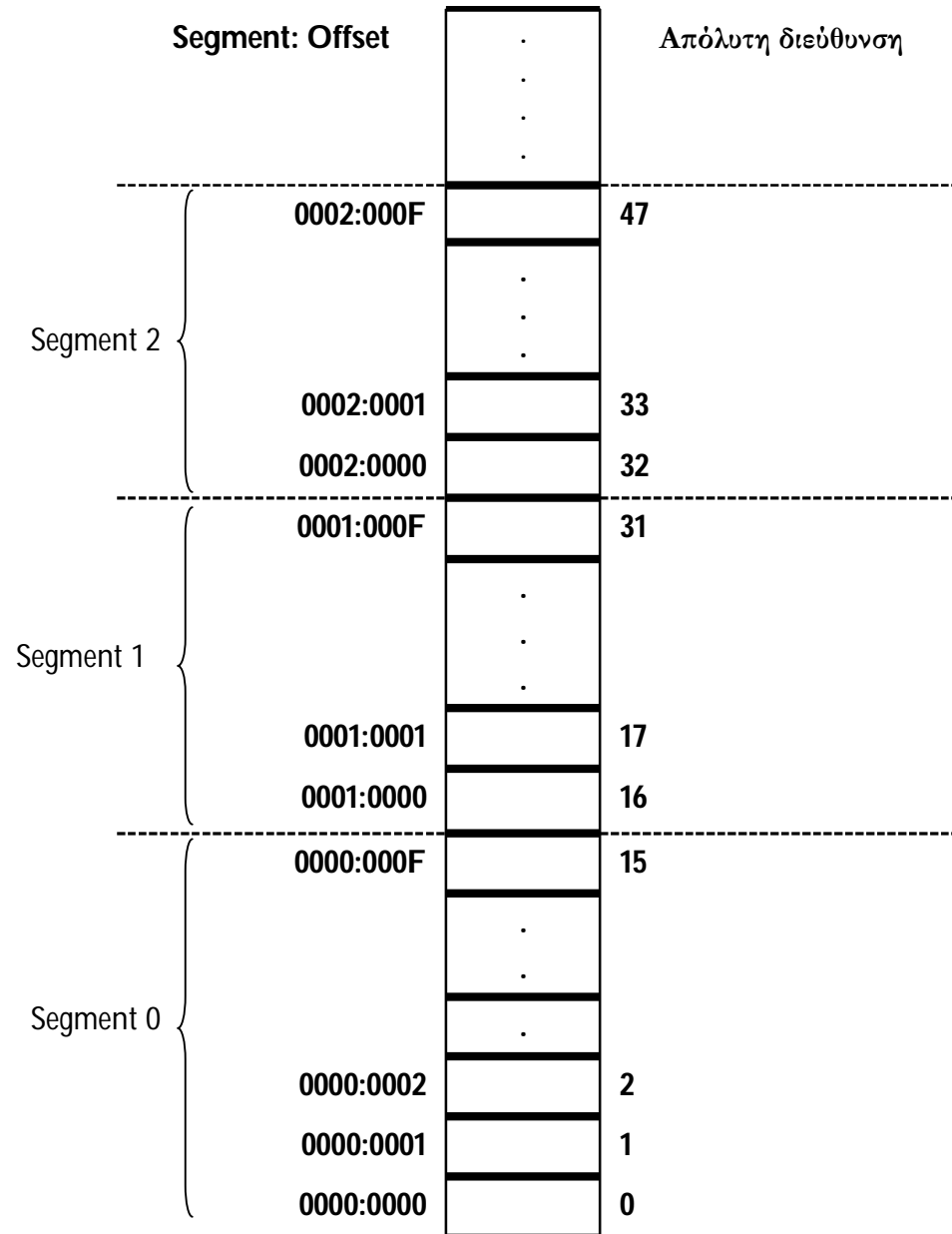


ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

# ΔΙΑΧΕΙΡΙΣΗ ΜΝΗΜΗΣ ΤΟΥ 8088





ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΔΙΑΧΕΙΡΙΣΗ ΜΝΗΜΗΣ ΤΟΥ 8088

- Χρησιμοποιούνται **segments** (16άδες θέσεων μνήμης) και **offset** (θέση μέσα στο segment)
- Κάθε διεύθυνση γράφεται σαν **Segment:Offset** (π.χ. 0FFF:0008)
- Υπάρχουν συνολικά **65.536** segments
- Συνολική μνήμη =  $65.536 \text{ segments} \times 16 \text{ bytes} = 1 \text{ MB}$
- Απόλυτη διεύθυνση =  $\text{segment} \times 16 + \text{offset}$



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

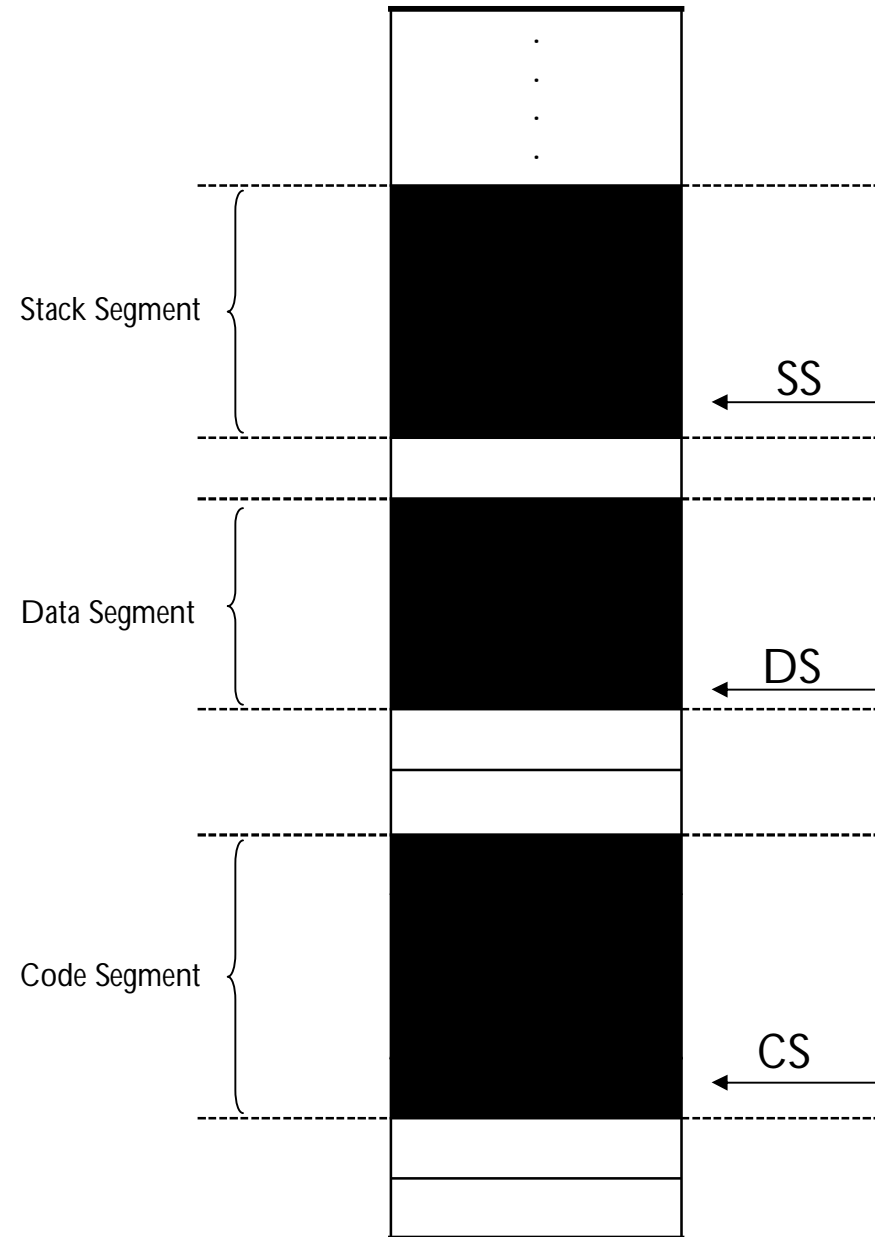
# ΔΙΑΧΕΙΡΙΣΗ ΜΝΗΜΗΣ ΤΟΥ 8088

Κάθε πρόγραμμα στον 8088 καταλαμβάνει 3 περιοχές στη μνήμη:

- **Code Segment** (Κώδικας)
- **Data Segment** (Δεδομένα)
- **Stack Segment** (Σωρός)

Η διεύθυνση της επόμενης προς εκτέλεση εντολής είναι:

**CS:IP**





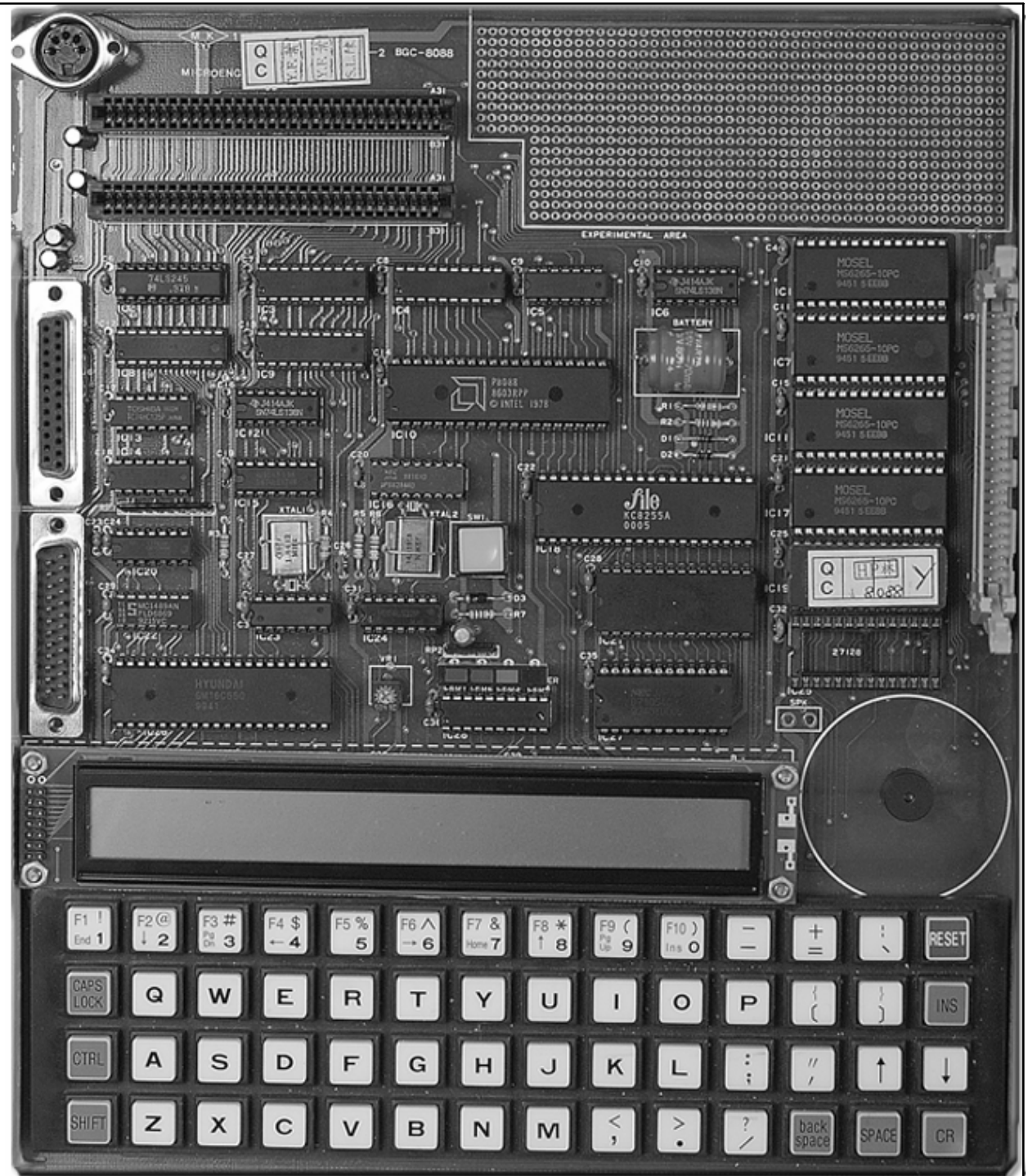


ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

BGC-8088





ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 1

## ΤΟ MONITOR ΤΟΥ BGG-8088

- Είναι ένα απλοποιημένο Λειτουργικό Σύστημα που διευκολύνει τη χρήση του BGC από τους χρήστες.
- Εκτελείται αυτόματα με το άνοιγμα του BGC
- Δέχεται από τον χρήστη εντολές σε μορφή κειμένου
- Οι εντολές αυτές ισχύουν μόνο για το συγκεκριμένο αναπτυξιακό σύστημα



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 1

## ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ ΤΟΥ MONITOR

- Προβολή περιεχομένων καταχωρητών: Εντολή **R**
- Αλλαγή περιεχομένων καταχωρητών: Εντολή **Rxx** όπου **xx** το όνομα του καταχωρητή
  - Π.χ. **RAX** για αλλαγή του **AX**
- Προβολή/αλλαγή περιεχομένων του **FLAG REGISTER**:  
Εντολή **RF**
  - Π.χ **RF**  
**FLAG STATUS=NV UP EI PL ZR AC PE NC**  
**NEW STATUS=**



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 1

## ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ ΤΟΥ MONITOR

- Προβολή περιεχομένων μνήμης:

Εντολή **D** θέση\_μνήμης

- Π.χ. D200 => D DS:0200

0100:0200 41000000-00010001 A .....

0100:0208 00020000-00010050 ..... P

- Αλλαγή περιεχομένων θέσεων μνήμης:

Εντολή **E** θέση\_μνήμης

- Π.χ. E200 => E DS:200

0100:0200 00 \_\_\_\_



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 1

## ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ ΤΟΥ MONITOR

- Εισαγωγή προγράμματος Assembly:  
Εντολή **A** θέση\_μνήμης

- Π.χ. A100 => A CS:0100  
0100:0100 MOV AX,05  
0100:0102 —

- Προβολή προγράμματος Assembly από τη μνήμη:  
Εντολή **U** θέση\_μνήμης

- Π.χ. U100 => U CS:0100  
0100:0100 BB0000 MOV BX,0000  
0100:0003 B80000 MOV AX,0000

.....



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 1

## ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ ΤΟΥ MONITOR

- Εκτέλεση προγράμματος **Assembly**:

Εντολή **G=θέση\_μνήμης**

- Π.χ.  $G=100 \Rightarrow G=CS:0100$

- Βηματική Εκτέλεση προγράμματος **Assembly** από τη μνήμη:

Εντολή **T=θέση\_μνήμης**

- Π.χ.  $T=100 \Rightarrow T=CS:0100$

..... ; 1<sup>η</sup> εντολή

T←

..... ; 2<sup>η</sup> εντολή

T←



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 1

## ΑΣΚΗΣΗ 1.1

- Εισάγετε στην διεύθυνση 0100:0000 το παρακάτω πρόγραμμα :

```
MOV AX,1234  
MOV BX,AX  
MOV [200],BX  
MOV CX,[200]  
MOV BY[202],FF  
INT 3
```

- Δείτε το πρόγραμμα με την εντολή **U** και διορθώστε τυχόν σφάλματα.
- Μέσω της εντολής **RXX** βάλτε στους καταχωρητές **AX,BX,CX** τις τιμές 1111, 2222 και 3333 αντίστοιχα
- Μέσω της εντολής **E** Βάλτε στις θέσεις μνήμης 0200 έως και 0202 τις τιμές 55, 66, 77
- Εκτελέστε το πρόγραμμα βήμα-βήμα με την εντολή **T**.
- Δείτε ενδιάμεσα και την μνήμη για επιβεβαίωση της σωστής εκτέλεσης των εντολών.



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

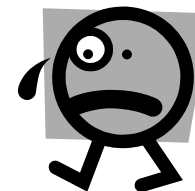
Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΕΡΓΑΣΤΗΡΙΟ 1

### ■ Η εντολή **MOV** και οι τρόποι σύνταξής της

- |  |                   |
|--|-------------------|
| ■ MOV καταχωρητής1, καταχωρητής2       | MOV AX,BX         |
| ■ MOV καταχωρητής, [θέση μνήμης]       | MOV AX,[200]      |
| ■ MOV [θέση μνήμης], καταχωρητής       | MOV [200],CX      |
| ■ MOV καταχωρητής, τιμή                | MOV DX, 1AF4      |
| ■ MOV BY[διεύθυνση μνήμης], τιμή 8bit  | MOV BY[200], 7A   |
| ■ MOV WO[διεύθυνση μνήμης], τιμή 16bit | MOV WO[200], 10F5 |

~~MOV [θέση μνήμης], [θέση μνήμης]~~







ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

ΕΡΓΑΣΤΗΡΙΟ 2

## ΑΡΙΘΜΗΤΙΚΕΣ ΕΝΤΟΛΕΣ ΤΟΥ MONITOR

- Μετατροπή από δεκαδικό σε δυαδικό  
Εντολή **B**δεκαδικός\_αριθμός
  - Π.χ. B10 → 1010B
- Μετατροπή από δυαδικό σε δεκαδικό  
Εντολή **V**δυαδικός\_αριθμός
  - Π.χ. V10100 → 20D
- Μετατροπή από δεκαδικό σε δεκαεξαδικό  
Εντολή **J**δεκαδικός\_αριθμός
  - Π.χ. J200 → C8H
- Μετατροπή από δεκαεξαδικό σε δεκαδικό  
Εντολή **S**δεκαεξαδικός\_αριθμός
  - Π.χ. S5F → 95D





ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΕΡΓΑΣΤΗΡΙΟ 2

### ΑΝΑΠΑΡΑΣΤΑΣΗ ΠΡΟΣΗΜΑΣΜΕΝΩΝ ΑΡΙΘΜΩΝ

- Μετατροπή θετικού δεκαδικού αριθμού σε δυαδικό στα 8 bit,

π.χ.  $(+20)_{10}$

$$(20)_{10} \rightarrow (10100)_2 \text{ και } (+20)_{10} \rightarrow (\underline{000}10100)_2$$

- Μετατροπή αρνητικού δεκαδικού αριθμού σε δυαδικό στα 8 bit,

π.χ.  $(-20)_{10}$

Θετική τιμή αριθμού στο δυαδικό στα 8 bit:  $(20)_{10} \rightarrow (00010100)_2$

Αντιστρέφω όλα τα bit (συμπλήρωμα ως προς 1)  $\rightarrow (11101011)_2$

Προσθέτω τη μονάδα (συμπλήρωμα ως προς 2)

$$\begin{array}{r} \phantom{11101011} + \phantom{11101011} 1 \\ \hline 11101100 \end{array}$$

Άρα  $(-20)_{10} \rightarrow (11101100)_2 \rightarrow (EC)_{16}$



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΕΡΓΑΣΤΗΡΙΟ 2

# ΑΠΟΘΗΚΕΥΣΗ ΑΡΙΘΜΩΝ ΣΤΗ ΜΝΗΜΗ

- Αριθμός 16 bit, π.χ.  $(12AB)_{16}$

AB	200
12	201

- Αριθμός 32 bit,  
π.χ.  $(12ABFF40)_{16}$

40	200
FF	201
AB	202
12	203



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΕΡΓΑΣΤΗΡΙΟ 2

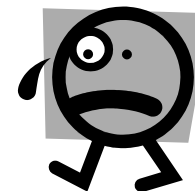
# ΕΝΤΟΛΕΣ ΠΡΟΣΘΕΣΗΣ

## ■ Η εντολή **ADD** και οι τρόποι σύνταξής της

Πρόσθεση χωρίς κρατούμενο,  $ADD\ a,b \rightarrow a=a+b$

- $ADD$  καταχωρητής1, καταχωρητής2
  - $ADD$  καταχωρητής, [θέση μνήμης]
  - $ADD$  [θέση μνήμης], καταχωρητής
  - $ADD$  καταχωρητής, τιμή
  - $ADD\ BY$ [διεύθυνση μνήμης], τιμή 8bit
  - $ADD\ WO$ [διεύθυνση μνήμης], τιμή 16bit
- |                     |
|---------------------|
| $ADD\ AX,BX$        |
| $ADD\ AX,[200]$     |
| $ADD\ [200],CX$     |
| $ADD\ DX,1AF4$      |
| $ADD\ BY[200],7A$   |
| $ADD\ WO[200],10F5$ |

~~$ADD$  [θέση μνήμης], [θέση μνήμης]~~





ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΕΡΓΑΣΤΗΡΙΟ 2

# ΕΝΤΟΛΕΣ ΠΡΟΣΘΕΣΗΣ

### ■ Η εντολή **ADC** και οι τρόποι σύνταξής της

Πρόσθεση με κρατούμενο,  $ADC\ a,b \rightarrow a=a+b+\text{κρατούμενο}$

- |  |                     |
|--|---------------------|
| ■ $ADC$ καταχωρητής1, καταχωρητής2         | $ADC\ AX,BX$        |
| ■ $ADC$ καταχωρητής, [θέση μνήμης]         | $ADC\ AX,[200]$     |
| ■ $ADC$ [θέση μνήμης], καταχωρητής         | $ADC\ [200],CX$     |
| ■ $ADC$ καταχωρητής, τιμή                  | $ADC\ DX,1AF4$      |
| ■ $ADC\ BY$ [διεύθυνση μνήμης], τιμή 8bit  | $ADC\ BY[200],7A$   |
| ■ $ADC\ WO$ [διεύθυνση μνήμης], τιμή 16bit | $ADC\ WO[200],10F5$ |

- Οι εντολές **CLC** (μηδένισε κρατούμενο) & **STC** (κάνε το κρατούμενο 1)



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΕΡΓΑΣΤΗΡΙΟ 2

### ΑΣΚΗΣΗ 2.1

Γράψτε ένα πρόγραμμα που να προσθέτει δύο μη προσημασμένους αριθμούς των 8 bits που είναι περιεχόμενα των θέσεων μνήμης 0200 και 0201 και να αποθηκεύει το άθροισμα στη θέση μνήμης 0202.

- Εκτελέστε το πρόγραμμά σας για τις τρεις περιπτώσεις που υπάρχουν στον παρακάτω πίνακα και συμπληρώστε τον.
- Στις περιπτώσεις που υπάρχει κρατούμενο η θέση 0202 κρατάει το σωστό αποτέλεσμα;

0200		0201		0202				
16-δικός	10-δικός	16-δικός	10-δικός	Θεωρ. 10-δικός	Εργ. 16-δικός	Εργ. 10-δικός	Σωστό? (Ναι/Όχι)	Κρατούμενο (Ναι/Όχι)
8F	143	83	131	274	12	18	OXI	NAI
2E	46	41	65	111	6F	111	NAI	OXI
3B	59	C5	197	256	00	00	OXI	NAI



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΕΡΓΑΣΤΗΡΙΟ 2

### ΑΣΚΗΣΗ 2.2

Γράψτε ένα πρόγραμμα που να προσθέτει δύο προσημασμένους αριθμούς των 8 bits που είναι περιεχόμενα των θέσεων μνήμης 0200 και 0201 και να αποθηκεύει το άθροισμα στη θέση μνήμης 0202.

- Το πρόγραμμα αυτό σε τι διαφέρει από το προηγούμενο;
- Εκτελέστε το πρόγραμμά σας για τις περιπτώσεις που υπάρχουν στον παρακάτω πίνακα και συμπληρώστε τον.
- Ποια αποτελέσματα στη 0202 θέση είναι σωστά; Γιατί;

0200		0201		0202				
16-δικός	10-δικός	16-δικός	10-δικός	Θεωρ. 10-δικός	Εργ. 16-δικός	Εργ. 10-δικός	Σωστό? (Ναι/Όχι)	Υπερχείλιση (Ναι/Όχι)
<b>6A</b>	<b>+106</b>	<b>59</b>	<b>+89</b>	<b>+195</b>	<b>C3</b>	<b>-61</b>	<b>OXI</b>	<b>NAI</b>
<b>7F</b>	<b>+127</b>	<b>03</b>	<b>+3</b>	<b>+130</b>	<b>82</b>	<b>-126</b>	<b>OXI</b>	<b>NAI</b>
<b>20</b>	<b>+32</b>	<b>FD</b>	<b>-3</b>	<b>+29</b>	<b>1D</b>	<b>+29</b>	<b>NAI</b>	<b>OXI</b>
<b>A6</b>	<b>-90</b>	<b>C2</b>	<b>-62</b>	<b>-152</b>	<b>68</b>	<b>+104</b>	<b>OXI</b>	<b>NAI</b>





ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΕΡΓΑΣΤΗΡΙΟ 2

### ΑΣΚΗΣΗ 2.3

Γράψτε ένα πρόγραμμα που να προσθέτει δύο 16-bit προσημασμένους αριθμούς που ο ένας είναι αποθηκευμένος στις θέσεις μνήμης 0200 (χαμηλής τάξης BYTE), 0201 (υψηλής τάξης BYTE), και ο άλλος στις 0202 (X.T.B.) και 0203 (Y.T.B.). Το αποτέλεσμα να αποθηκευθεί στις θέσεις μνήμης 0204 (X.T.B.) και 0205 (Y.T.B.).

- Το πρόγραμμα αυτό σε τι διαφέρει από το προηγούμενο;
- Εκτελέστε το προγράμμα σας για τα δεδομένα του παρακάτω πίνακα

0200,0201 Προσθετέος		0202,0203 Προσθετέος		0204,0205 Άθροισμα		Υπερχείλιση (Ναι/Όχι)
25	63	41	AB	66	0E	OXI
83	C7	F1	B4	74	7C	NAI

6325

+AB41

0E66

C783

+ B4F1

7C74



ΤΕΙ ΣΕΡΡΩΝ

Τμήμα Πληροφορικής & Επικοινωνιών

Τομέας Αρχιτεκτονικής Υπολογιστών & Βιομηχανικών Εφαρμογών

## ΕΡΓΑΣΤΗΡΙΟ 2

### ΑΣΚΗΣΗ 2.4

Γράψτε ένα πρόγραμμα που να προσθέτει δύο 32-bit προσημασμένους αριθμούς που ο ένας είναι αποθηκευμένος στις θέσεις μνήμης 0200...0203, και ο άλλος στις 0204...0207. Το αποτέλεσμα να αποθηκευθεί στις θέσεις μνήμης 0208...020B.

- Το πρόγραμμα αυτό σε τι διαφέρει από το προηγούμενο;
- Εκτελέστε το προγράμμα σας για τα δεδομένα του παρακάτω πίνακα

0200...0203 Προσθετέος	0204...0207 Προσθετέος	0208...020B Άθροισμα	Υπερχείλιση (Ναι/Όχι)
12345678	23456789	3579BD01	OXI
7FFFFFFF	00000001	7FFFFFF0	OXI

78563412

+89674523

**01BD7935**

FFFFFFFF7F

+ 01000000

**00FFFFFF7F**