



# Αρχιτεκτονική Υπολογιστών

**Ενότητα 5:** Εντολές αλλαγής ροής. Διακλάδωση χωρίς συνθήκη.  
Διακλάδωση με συνθήκη.

Δρ. Μηνάς Δασυγένης

[mdasyg@ieee.org](mailto: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  
*εινένωση 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



# Η εντολή JMP

---

- Εντολή διακλάδωσης χωρίς συνθήκη. Η εκτέλεση συνεχίζεται από το σημείο που ορίζει η JMP.
- Σύνταξη:  
**JMP (παράμετρος1)**
- Όπου (παράμετρος1) είναι μια ετικέτα μνήμης, δηλαδή ένα αλφαριθμητικό ακολουθούμενο από :

**Για παράδειγμα:**

**JMP ARXH**

σε κάποιο σημείο του κώδικα υπάρχει

**ARXH:**



# Ειδικοί Καταχωρητές

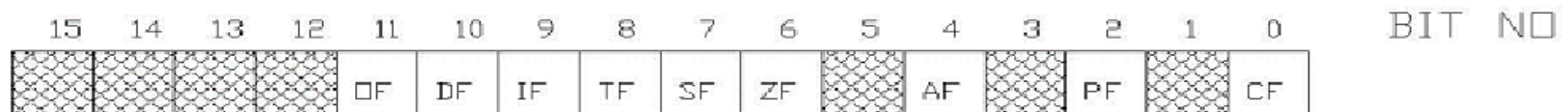
---

- Ο 8086 εκτός από τους καταχωρητές γενικής χρήσης **AX, BX, CX, DX** έχει και τους καταχωρητές ειδικής χρήσης **BP, SP, SI, DI** που χρησιμοποιούνται για τη διευθυνσιοδότηση μνήμης (δες σχετική διάλεξη θεωρίας).
- Υπάρχει επίσης ένας ειδικός καταχωρητής που ονομάζεται Καταχωρητής Κατάστασης ή σημαίων και είναι ο **SR**.



# Καταχωρητής κατάστασης ή σημαιών (Status Register)

- Είναι ο μοναδικός καταχωρητής στον οποίο επιτρέπεται πρόσβαση ανά bit (δηλαδή μπορούμε να διαβάσουμε π.χ. την τιμή του bit 0).
- Κάθε θέση bit αντιστοιχεί σε μια συγκεκριμένη ανίχνευση κατάστασης. Π.χ. Αν το bit0 είναι 1, τότε εμφανίζεται η κατάσταση CF (κρατούμενο) που σημαίνει ότι έχει δημιουργηθεί κρατούμενο από την προηγούμενη αριθμητική πράξη.
- Ο SR ενημερώνεται ΜΟΝΟ από τις αριθμητικές και λογικές πράξεις.



# Σημαντικοί Δείκτες Κατάστασης του SR

---

- **CF** (Carry Flag) Χρησιμοποιείται σαν ένα επιπλέον δυαδικό ψηφίο σε αριθμητικές εντολές (ADD, SUB, ADC, SBC). Εάν προκύψει ένα κρατούμενο (πρόσθεση) ή απαιτηθεί δανεισμός (αφαίρεση), παίρνει τιμή 1 αλλιώς 0.
- **SF** (Sign Flag) Χρησιμοποιείται από αριθμητικές ή λογικές πράξεις.. Εάν το αποτέλεσμα είναι θετικό γίνεται 0, εάν είναι αρνητικό γίνεται 1.
- **ZF** (Zero Flag) Γίνεται 1 αν το αποτέλεσμα μιας αριθμητικής ή λογικής πράξης είναι 0, αλλιώς μένει 0.
- **OF** (Overflow Flag) Γίνεται 1 όταν το προσημασμένο αποτέλεσμα μιας πράξης (σε συμπλήρωμα του δύο) είναι πολύ μεγάλο ή πολύ μικρό για να χωρέσει στον τελεστή του αποδέκτη.



# Διακλαδώσεις με συνθήκη – Συγκρίσεις

---

- Ο 8086 διαθέτει 18 εντολές διακλαδώσεων με τις οποίες υπάρχει η δυνατότητα διακλάδωσης ανάλογα με την τιμή μιας σημαίας ή συνδυασμού σημαιών.
- Ειδικά αναφέρουμε ότι αν ελέγχουμε το αποτέλεσμα πράξεων με μη προσημασμένους αριθμούς τότε λαμβάνεται υπόψη μεταξύ των άλλων η **CF ( Carry Flag)**, ενώ σε προσημασμένους αριθμούς ελέγχεται η **OF (Overflow Flag)** και η **SF (Sign Flag)**.





# Υλοποίηση διακλάδωσης

---

Η διακλάδωση υλοποιείται με 5 βήματα:

- Εντολή σύγκρισης (**CMP**).
- Εντολή αλλαγής ροής εκτέλεσης υπό συνθήκη (**J\_\_**).
- Ομάδα εντολών που θα εκτελεστούν αν δεν υλοποιηθεί η αλλαγή ροής εκτέλεσης.
- Ομάδα εντολών που θα εκτελεστούν αν υλοποιηθεί η αλλαγή ροής εκτέλεσης.



# Εντολή σύγκρισης CMP (1/2)

---

- Πολύ συχνά μιας εντολής διακλάδωσης με συνθήκη προηγείται μια εντολή σύγκρισης CMP όποτε και η διακλάδωση πραγματοποιείται ή όχι, ανάλογα με το αποτέλεσμα της σύγκρισης. Η εντολή σύγκρισης συντάσσεται όπως η εντολή αφαίρεσης SUB:
- CMP (παράμετρος1) , (παράμετρος2).
- Όπου (παράμετρος1,2) είναι:
  - Καταχωρητές.
  - Διευθύνσεις μνήμης.
  - Αριθμοί.
- Πρέπει να έχουν ίδιο μέγεθος!



# Εντολή σύγκρισης CMP (2/2)

---

- Παραδείγματα:

**cmp AL,5** ; σύγκριση του AL με την τιμή 5.

**cmp BX,"5"** ; σύγκριση του BX με την ASCII τιμή 5.

**cmp DX,DI** ; σύγκριση του DX με το DI.

- Η CMP έχει ως αποτέλεσμα να ενημερώσει τον καταχωρητή σημαιών SR.
- Δεν επιτελείται καμία άλλη αλλαγή. Δεν αλλάζει κάποιος καταχωρητής.
- Το επόμενο βήμα είναι να ελέγξουμε το κατάλληλο bit του SR.



# Η CMP υλοποιεί μια πράξη αφαίρεσης χωρίς να αποθηκεύεται το αποτέλεσμα

---

- Η CMP συγκρίνει τον προορισμό με την προέλευση, δηλαδή όπως και η SUB αφαιρεί από τον προορισμό την προέλευση, ενημερώνει τους δείκτες κατάστασης (Flags) ανάλογα με το αποτέλεσμα της πράξης, αλλά το αποτέλεσμα της πράξης δεν το αποθηκεύει πουθενά, δηλαδή οι τελεστές της σύγκρισης δεν αλλάζουν τιμή.
- Το επόμενο είναι να χρησιμοποιήσουμε κάποια εντολή που ξεκινάει με J\_\_ η οποία ελέγχει την κατάλληλη σημαία.



# Εντολή αλλαγής ροής εκτέλεσης υπό συνθήκη (J\_\_)

- Οι εντολές J\_\_ περιέχουν μετά το J, ένα ή δύο ή τρία γράμματα από τα παρακάτω. Τα γράμματα αυτά υλοποιούν τη συνθήκη κάτω από την οποία θα πραγματοποιηθεί η αλλαγή ροής εκτέλεσης (δηλαδή, το jump).
  - **A (above)**: μεγαλύτερο για μη προσημασμένους αριθμούς.
  - **B (bellow)**: μικρότερο για μη προσημασμένους αριθμούς.
  - **E (equal)**: ίσο.
  - **N (not)**: όχι, σε συνδυασμό πάντα με ένα από τα προηγούμενα γράμματα.
  - **G (greater)**: μεγαλύτερο για προσημασμένους αριθμούς.
  - **L (lower)**: μικρότερο για προσημασμένους αριθμούς.
  - **P (parity)**: ισοτιμία, δηλαδή  $P=1$
  - **Z (zero)**: μηδενικό αποτέλεσμα, δηλαδή  $Z=1$ .
  - **C (carry)**: κρατούμενο, δηλαδή  $C=1$
  - **O (overflow)**: υπερχείλιση, δηλαδή  $OV=1$
  - **S (sign)**: αρνητικό πρόσημο, δηλαδή  $S=1$



# Σημασία εντολών J\_\_

- Αρχικά επιλέγεται αν θα χρησιμοποιήσουμε εντολές για προσημασμένους αριθμούς ή μη προσημασμένους (στις περισσότερες φορές εμείς θα χρησιμοποιήσουμε μη προσημασμένους αριθμούς).
- Για μη προσημασμένους αριθμούς, δημιουργούμε την εντολή J\_\_ επιλέγοντας ένα ή περισσότερα από τα γράμματα (E B A N). π.χ. αν επιλεχθούν τα NAE, τότε δημιουργείται το J**NAE**, δηλαδή η αλλαγή ροής εκτέλεσης αν ΔEN (**N**) ισχύει να είναι μεγαλύτερο (**A**) ή ίσο (**E**).
- Ομοίως για προσημασμένους, δημιουργούμε την εντολή J\_\_ επιλέγοντας ένα ή περισσότερα από τα γράμματα (E G L N).
- Σε ειδικές περιπτώσεις, δημιουργούμε την εντολή J\_\_ επιλέγοντας ένα ή περισσότερα από τα γράμματα (**N** C O S Z).



# Εντολές J\_\_ για αλλαγή ροής εκτέλεσης υπό συνθήκη (1/2)

---

Όνομα	Περιγραφή	Δείκτες που ελέγχονται
JB/JNAE	Διακλάδωση εάν είναι μικρότερο μη προσημασμένο	CF=1
JAE/JNB	Διακλάδωση εάν δεν είναι μικρότερο μη προσημασμένο	CF=0
JBE/JNA	Διακλάδωση εάν δεν είναι μεγαλύτερο μη προσημασμένο	CF=1 ή ZF=1
JA/JNBE	Διακλάδωση εάν είναι μεγαλύτερο μη προσημασμένο	CF=0 ΚΑΙ ZF=0
JE/JZ	Διακλάδωση αν είναι ίσο	ZF=1
JNE/JNZ	Διακλάδωση αν είναι άνισα	ZF=0
JL/JNGE	Διακλάδωση εάν είναι μικρότερο προσημασμένο	SF<>OF
JGE/JNL	Διακλάδωση εάν δεν είναι μικρότερο προσημασμένο	SF=OF
JLE/JNG	Διακλάδωση εάν δεν είναι μεγαλύτερο προσημασμένο	ZF=1 ή SF<>OF



# Εντολές J\_\_\_ για αλλαγή ροής εκτέλεσης υπό συνθήκη (2/2)

---

JG/JNLE	Διακλάδωση εάν είναι μεγαλύτερο προσημασμένο	ZF=0 και SF = OF
JP/JPE	Διακλάδωση εάν υπάρχει ισοτιμία	PF=1
JNP/JPO	Διακλάδωση εάν δεν υπάρχει ισοτιμία	PF=0
JS	Διακλάδωση εάν αρνητικό αποτέλεσμα	SF=1
JNS	Διακλάδωση εάν θετικό αποτέλεσμα	SF=0
JC	Διακλάδωση εάν υπάρχει κρατούμενο	CF=1
JNC	Διακλάδωση εάν δεν υπάρχει κρατούμενο	CF=0
JO	Διακλάδωση εάν υπάρχει υπέρβαση κρατουμένου	OF=1
JNO	Διακλάδωση εάν δεν υπάρχει υπέρβαση κρατουμένου	OF=0





# Παραδείγματα (1)

- Αν ο AL είναι μικρότερος από 10 να εκτελείται το τμήμα κώδικα [AAA] ενώ διαφορετικά να εκτελείται το τμήμα κώδικα [BBB].

`cmp AL, 5` ;σύγκρινε AL με 5

`JB partAAA` ;αν  $AL < 5$  (jump if bellow) εκτέλεσε partAAA

**PartBBB** ;εδώ εντολές partBBB. Δεν απαιτείται jump

`jmp telos` ;ΠΡΟΣΟΧΗ: `jmp telos` για να μην εκτελεστεί και το `partAAA`

`PartAAA:` ;ετικέτα μνήμης PartAAA

**partAAA** ;εδώ εντολές partAAA

`telos:` ;ετικέτα μνήμης telos



# Παραδείγματα (2)

- Αν ο AL είναι μεγαλύτερος ή ίσος από 10 να πάμε μερικές γραμμές πίσω.

**againinput:**

;ετικέτα μνήμης againinput.

[..π.χ. Είσοδος ξανά του AL]

**cmp AL,10**

;σύγκρινέ το AL με το 10.

**jae againinput**

;αν μεγαλύτερος ή ίσος

(jump above-equal) πήγαινε πίσω στο againinput.



# Βρόχος επανάληψης για 5 φορές

Να επαναληφθεί το τμήμα κώδικα AAAA 5 φορές  
Θεωρούμε έναν καταχωρητή ως μετρητή. Έστω CX

```
mov cx, 5           ;αρχική τιμή 5 στον CX
startloop:         ;ετικέτα που αρχίζει το loop
AAAA              ;κώδικας που θα επαναληφθεί 5 φορές.
sub cx, 1          ;μείωσε του CX κατά 1.
cmp cx, 0          ;σύγκριση του cx με το 0.
jne startloop      ;αν δεν είναι cx=0 (jump not equal) τότε πήγαινε startloop
                  ;σε διαφορετική περίπτωση έχουμε κάνει 5 επαναλήψεις.
```



---

# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο

