



Ελληνική Δημοκρατία  
Τεχνολογικό Εκπαιδευτικό  
Ίδρυμα Ηπείρου

# Αρχιτεκτονική υπολογιστών

Ενότητα 9 : Ομάδες Εντολών: Ιδιότητες και Λειτουργίες

Φώτης Βαρζιώτης



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Τμήμα Μηχανικών Πληροφορικής Τ.Ε

## Αρχιτεκτονική υπολογιστών

Ενότητα 9 : Ομάδες Εντολών: Ιδιότητες και Λειτουργίες

Φώτης Βαρζιώτης  
Καθηγητής Εφαρμογών  
Άρτα, 2015



# 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 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.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Τι είναι ομάδα εντολών;

- Το σύνολο εντολών που μπορεί να εκτελέσει η CPU
- Κώδικας Μηχανής
- Μια σειρά bit (Binary)
- Αναπαρίστανται με κώδικα assembly (συμβολική παράσταση εντολών μηχανής)



# Στοιχεία εντολής Μηχανής

- Κώδικας Λειτουργίας (Op code)
  - Κάνε..
- Αναφορά σε τελεστή εισόδου
  - Σε...
- Αναφορά σε τελεστή αποτελέσματος
  - Βάλε την απάντηση σε..
- Αναφορά σε επόμενη εντολή
  - Όταν ολοκληρώσεις την εντολή, κάνε...

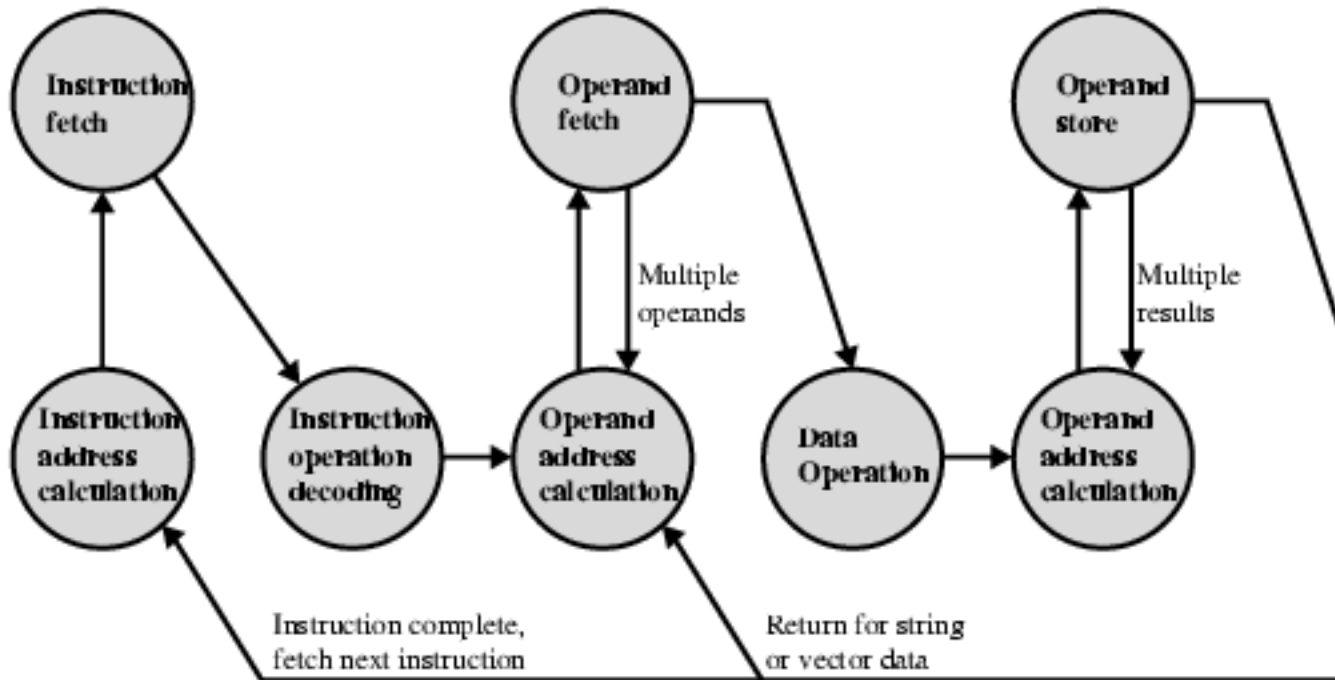


# Που βρίσκονται οι τελεστές;

- Στην κύρια μνήμη (ή εικονική μνήμη ή cache)
- Σε καταχωρητή της CPU
- Σε συσκευή I/O



# Διάγραμμα Κατάστασης Κύκλων Εντολής





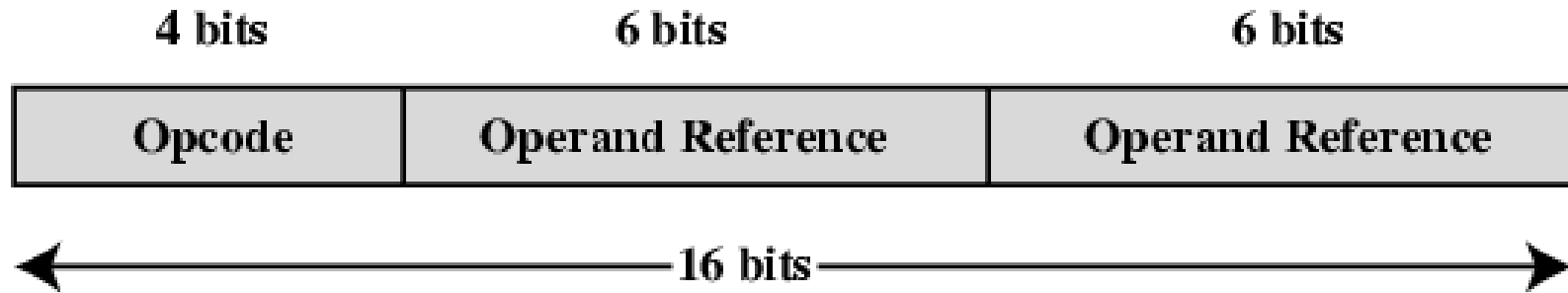


# Παράσταση Εντολών

- Στον κώδικα μηχανής κάθε εντολή αναπαρίσταται με μια μοναδική ακολουθία bit
- Για χρήση από προγραμματιστές διατίθεται μια συμβολική αναπαράσταση εντολής
  - π.χ. ADD, SUB, LOAD
- Οι τελεστές μπορούν να αναπαρασταθούν με παρόμοιο τρόπο
  - ADD A,B



# Μορφοποίηση Εντολής





# Είδη Εντολών

- Επεξεργασία Δεδομένων
- Αποθήκευση Δεδομένων (κύρια μνήμη)
- Μετακίνηση δεδομένων (I/O)
- Έλεγχος (Εντολές δοκιμών και διακλάδωσης)



# Πλήθος Διευθύνσεων (α)

- **3 Διευθύνσεις**
  - Τελεστής 1, Τελεστής 2, Αποτέλεσμα
  - $a = b + c$ ;
  - Μπορεί να απαιτείται και τέταρτος – επόμενη εντολή (συνήθως εσωτερικά)
  - Εξαιρετικά σπάνιο
  - Γενικά δημιουργείται πολύ μεγάλη λέξη



# Πλήθος Διευθύνσεων (β)

- 2 διευθύνσεις
  - Διττός ο ρόλος μιας διεύθυνσης ως τελεστής και αποτέλεσμα
  - $a = a + b$
  - Μειώνεται το μήκος της εντολής
  - Αυξάνεται λίγο η περιπλοκότητα υλοποίησης
    - Απαιτείται προσωρινός χώρος αποθήκευσης για τη διατήρηση κάποιων αποτελεσμάτων



# Πλήθος Διευθύνσεων ( $\gamma$ )

- **1 διεύθυνση**
  - Εσωτερική 2<sup>η</sup> διεύθυνση
  - Συνήθως ένας καταχωρητής (Συσσωρευτής)
  - Κοινή αρχιτεκτονική στους πρώτους επεξεργαστές



# Πλήθος Διευθύνσεων (δ)

- 0 (μηδέν) διευθύνσεις
  - Όλες οι διευθύνσεις διαχειρίζονται εσωτερικά
  - Χρησιμοποιείται ένας σωρός
  - π.χ. `push a`
  - `push b`
  - `add`
  - `pop c`
- $c = a + b$



# Επιλογή Πλήθους Διευθύνσεων

- Πολλές διευθύνσεις
  - Περίπλοκες (αλλά ισχυρές;) εντολές
  - Πολλοί καταχωρητές
    - Οι πράξεις με χρήση καταχωρητών γίνονται γρήγορα
  - Λίγες εντολές ανά πρόγραμμα
- Λίγες διευθύνσεις
  - Απλές (αλλά ισχυρές?) εντολές
  - Πολλές εντολές ανά πρόγραμμα
  - Γρήγορη προσκόμιση / εκτέλεση εντολών





# Σχεδιαστικές αποφάσεις<sub>1/2</sub>

- Σετ λειτουργιών
  - Πόσες πράξεις?
  - Τι μπορούν να κάνουν?
  - Πόσο περίπλοκες είναι?
- Τύποι δεδομένων
- Μορφοποίηση Εντολής
  - Μήκος πεδίου op code
  - Πλήθος διευθύνσεων



# Σχεδιαστικές αποφάσεις<sub>2/2</sub>

- Καταχωρητές
  - Πλήθος καταχωρητών στη CPU
  - Ποιες λειτουργίες μπορούν να εκτελεστούν σε ποιους καταχωρητές;
- Διευθυνσιοδότηση (αργότερα...)
- RISC vs CISC



# Τύποι τελεστών

- Διευθύνσεις
- Αριθμοί
  - Ακέραιοι/Κινητής υποδιαστολής
- Χαρακτήρες
  - ASCII etc.
- Λογικά δεδομένα
  - Bits ή flags



# Είδη δεδομένων σε Pentium

- 8 bit Byte
- 16 bit word
- 32 bit double word
- 64 bit quad word
- Η διευθυνσιοδότηση γίνεται σε επίπεδο Byte
- Μια 32 bit double word διαβάζεται σε διευθύνσεις που είναι διαιρετές δια του 4

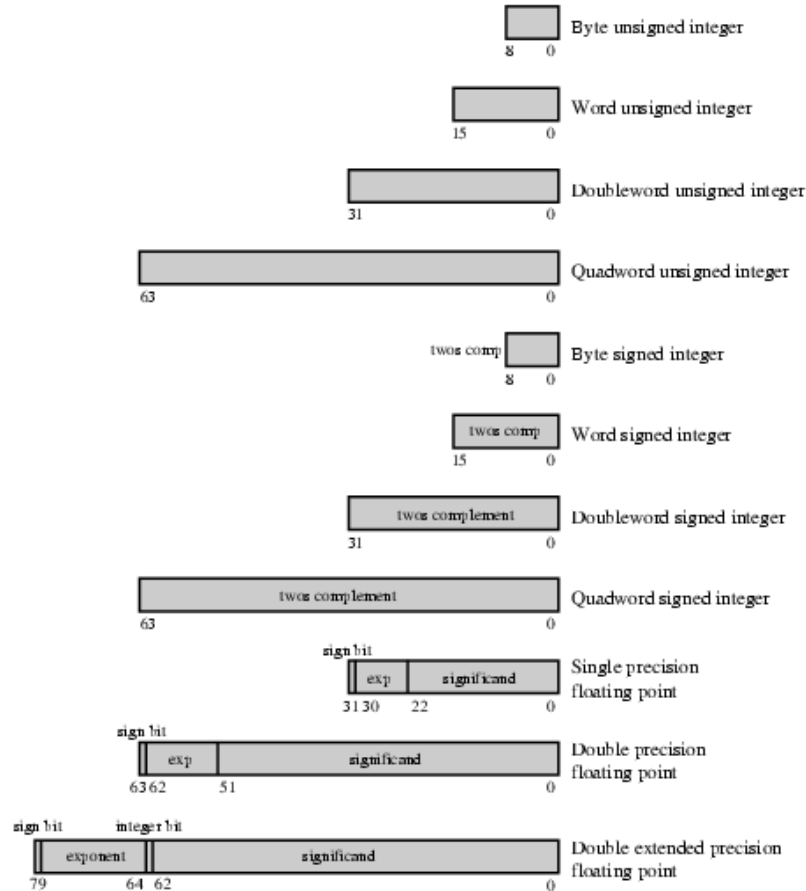


# Συγκεκριμένοι τύποι δεδομένων

- Γενικός – Μια αυθαίρετη ακολουθία bit
- Ακέραιος- Δυαδική τιμή με πρόσημο που περιέχεται σε byte, λέξη, ή διπλή λέξη με χρήση συμπληρώματος του 2
- Τακτικός – Ακέραιος χωρίς πρόσημο
- Unpacked BCD – Ένα ψηφίο ανά Byte
- Packed BCD - 2 BCD ψηφία ανά byte
- Near Pointer - 32 bit μετατόπιση εντός τμήματος
- Bit field
- Byte String
- Floating Point



# Τύποι δεδομένων κινητής υποδιαστολής σε Pentium





# Είδη πράξεων

- Μεταφορά δεδομένων
- Αριθμητική
- Λογική
- Μετατροπή
- I/O
- Έλεγχος συστήματος
- Μεταφορά ελέγχου



# Μεταφορά δεδομένων

- Πρέπει να ορίζονται
  - Η πηγή
  - Ο προορισμός
  - Ο πλήθος δεδομένων
- Μπορεί να υπάρχουν διαφορετικές εντολές για διαφορετικές μετακινήσεις
  - π.χ. IBM 370
- Η μια εντολή με χρήση ξεχωριστών διευθύνσεων
  - π.χ. VAX



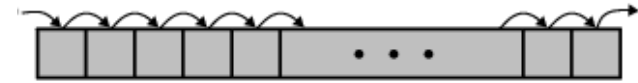


# Αριθμητική

- Add, Subtract, Multiply, Divide
- Signed Integer
- Floating point ?
- Μπορεί να περιλαμβάνει
  - Increment ( $a++$ )
  - Decrement ( $a--$ )
  - Negate ( $-a$ )



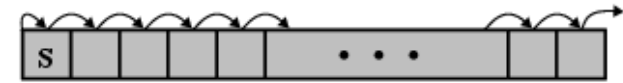
# Λειτουργίες Ολίσθησης και Μετατόπισης



(a) Logical right shift



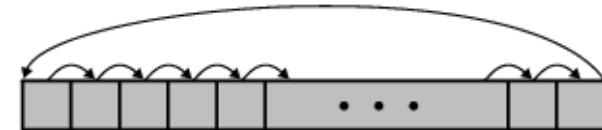
(b) Logical left shift



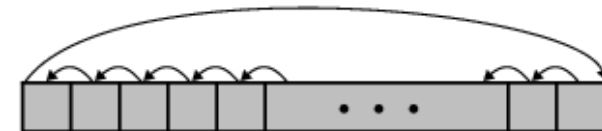
(c) Arithmetic right shift



(d) Arithmetic left shift



(e) Right rotate



(f) Left rotate



# Λογική

- Bitwise operations
- AND, OR, NOT



# Conversion

- π.χ. Δυαδικός σε δεκαδικό



# Input/Output

- Μπορεί να είναι εξειδικευμένες εντολές
- Μπορεί να πραγματοποιείται μέσω εντολών μεταφοράς δεδομένων (με χαρτογράφηση μνήμης)
- Μπορεί να υλοποιείται με ξεχωριστό ελεγκτή (DMA)



# Ελέγχου συστήματος

- Προνομιακές εντολές
- Η CPU έχει τεθεί σε συγκεκριμένη κατάσταση
  - Ring 0 on 80386+
  - Kernel mode
- Για χρήση από το λειτουργικό σύστημα

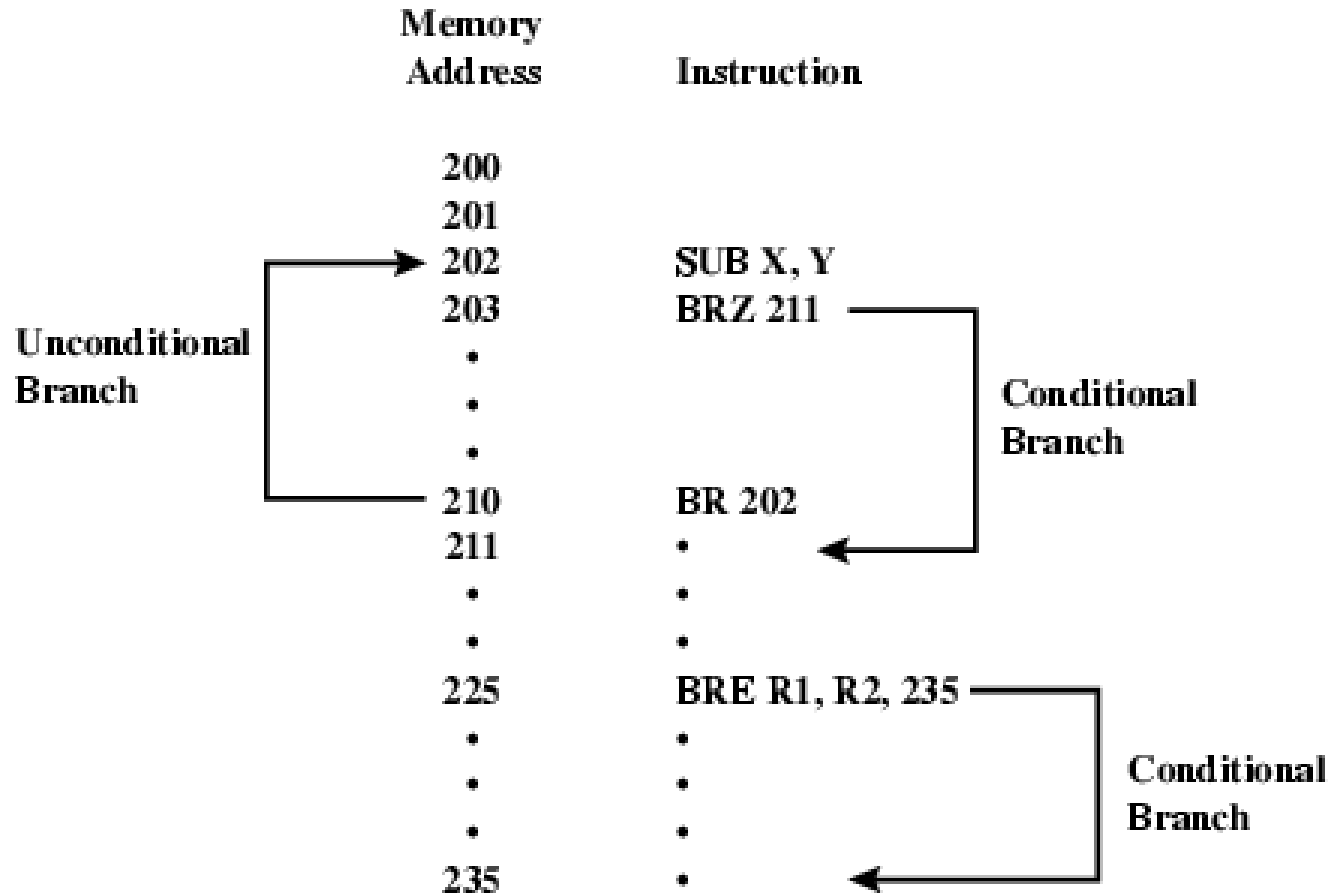


# Μεταφορά Ελέγχου

- Διακλάδωση
  - π.χ. Διακλάδωση σε  $x$  αν το αποτέλεσμα είναι 0
- Υπερπήδηση
  - π.χ. increment and skip if zero
  - ISZ Register1
  - Branch xxxx
  - ADD A
- Κλήση υπορουτίνας
  - π.χ. κλήση διακοπής

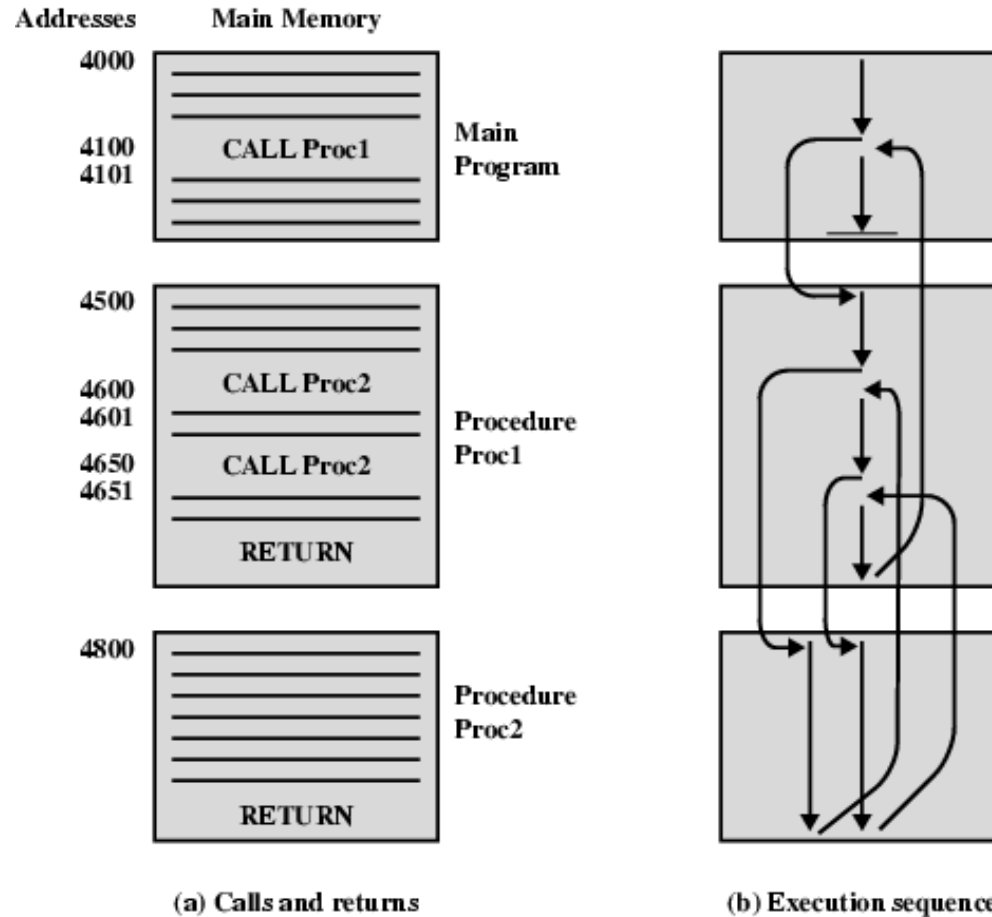


# Εντολή Διακλάδωσης



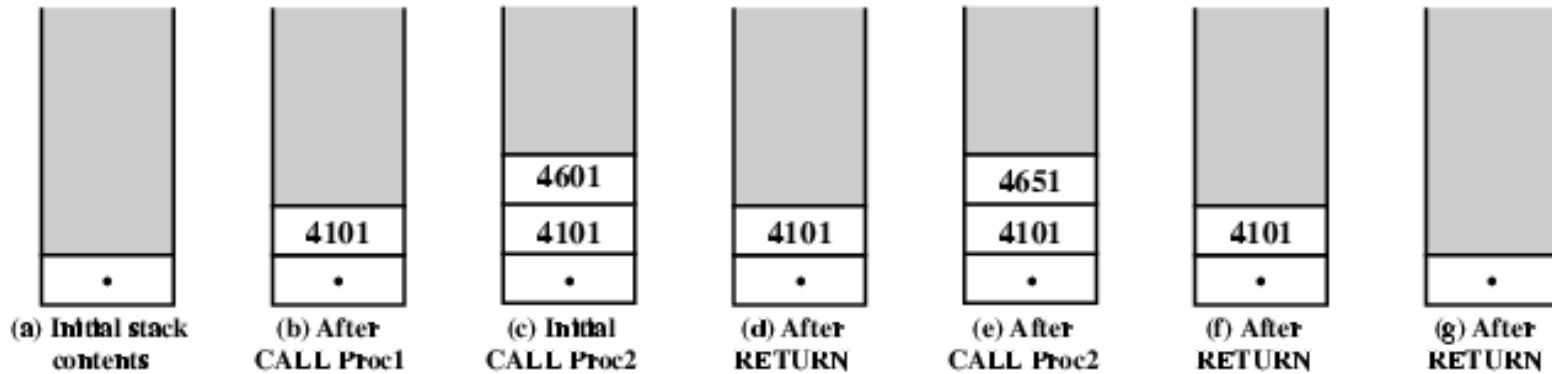


# Εμφωλευμένες Διαδικασίες





# Χρήση Σωρών





# Εξάσκηση

- Σετ εντολών για Pentium και PowerPC
- Αρχικά από βιβλίο
- Στη συνέχεια από web sites



# Η διάταξη των Byte

- Με ποια σειρά διαβάζουμε αριθμούς που περιγράφονται από περισσότερα του ενός bytes;
- π.χ. (Οι αριθμοί δίνονται σε hex για ευκολότερη ανάγνωση)
- 12345678 can be stored in 4x8bit locations as follows



# Η διάταξη των Byte(παράδειγμα)

- Address                      Value (1)                      Value(2)  
    184                              12                              78  
    185                              34                              56  
    186                              56                              34  
    186                              78                              12
- Προς ποια κατεύθυνση διαβάζουμε?



# Ονόματα για την διάταξη των bytes στη μνήμη

- Το πρόβλημα καλείται “Endian”
- Το πρώτο σύστημα έχει το πιο σημαντικό ψηφίο στην μικρότερη διεύθυνση
- Καλείται “big-endian”
- Το δεύτερο σύστημα έχει το πιο σημαντικό ψηφίο στην μεγαλύτερη διεύθυνση
- Καλείται “little-endian”



# Παράδειγμα για C Data Structure

```

struct {
    int    a;        //0x1112_1314           word
    int    pad;     //
    double b;       //0x2122_2324_2526_2728       doubleword
    char*  c;       //0x3132_3334           word
    char   d[7];   //'A','B','C','D','E','F','G'  byte array
    short  e;      //0x5152               halfword
    int    f;      //0x6161_6364         word
} s;
    
```

**Big-endian address mapping**

Byte Address	11	12	13	14				
00	00	01	02	03	04	05	06	07
	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>
08	08	09	0A	0B	0C	0D	0E	0F
	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	'A'	'B'	'C'	'D'
10	10	11	12	13	14	15	16	17
	'E'	'F'	'G'		51	52		
18	18	19	1A	1B	1C	1D	1E	1F
	<b>61</b>	<b>62</b>	<b>63</b>	<b>64</b>				
20	20	21	22	23				

**Little-endian address mapping**

								11	12	13	14	Byte Address
07	06	05	04	03	02	01	00					00
<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>					
0F	0E	0D	0C	0B	0A	09	08					08
'D'	'C'	'B'	'A'	31	32	33	34					
17	16	15	14	13	12	11	10					10
		51	52		'G'	'F'	'E'					
1F	1E	1D	1C	1B	1A	19	18					18
								<b>61</b>	<b>62</b>	<b>63</b>	<b>64</b>	
								23	22	21	20	20



# Άλλη άποψη της μνήμης

00	11
	12
	13
	14
04	
08	21
	22
	23
	24
0C	25
	26
	27
	28
10	31
	32
	33
	34
14	'A'
	'B'
	'C'
	'D'
18	'E'
	'F'
	'G'
1C	51
	52
20	61
	62
	63
	64

(a) Big-endian

00	14
	13
	12
	11
04	
08	28
	27
	26
	25
0C	24
	23
	22
	21
10	34
	33
	32
	31
14	'A'
	'B'
	'C'
	'D'
18	'E'
	'F'
	'G'
1C	52
	51
20	64
	63
	62
	61

(b) Little-endian





# Standard...Ποιο Standard?

- Pentium (80x86), VAX είναι little-endian
- IBM 370, Motorola 680x0 (Mac), και οι περισσότεροι RISC είναι big-endian
- Το Internet είναι big-endian
  - Το να γράφεις προγράμματα για διαδίκτυο είναι λίγο περίεργο!
  - Το WinSock παρέχει τις συναρτήσεις htoi and itoh (Host to Internet & Internet to Host) για μετατροπή



# Βιβλιογραφία

William Stallings. (2011). Αρχιτεκτονική & Οργάνωση Υπολογιστών. Εκδόσεις Τζιόλα.

Δημοσθένης Ε. Μπολανάκης. (2011). Αρχιτεκτονική Μικροϋπολογιστών: αρχές προγραμματισμού χαμηλού επιπέδου και εφαρμογές με το μικροελεγκτή M68HC908GP32, Εκδόσεις Σύγχρονη Παιδεία.

Tanenbaum Andrew S. (1995). Η Αρχιτεκτονική των Υπολογιστών μια δομημένη προσέγγιση Συγγραφέας Tanenbaum Andrew S. Εκδόσεις Κλειδάριθμος.

Luce T. (1991). Αρχιτεκτονική των Υπολογιστών. Εκδόσεις Τζιόλα.

Gilmore. (1999). Μικροεπεξεργαστές θεωρία και εφαρμογές. Εκδόσεις Τζιόλα.

Predko M. (2000). Προγραμματίζοντας τον Μικροελεγκτή PIC, Εκδόσεις Τζιόλα.

Μπεκάκος Μ.Π. (1994). Αρχιτεκτονική υπολογιστών & τεχνολογία παράλληλης επεξεργασίας, Εκδόσεις Σταμούλης.



# Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Φώτης Βαρζιώτης.  
Αρχιτεκτονική υπολογιστών.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή  
διεύθυνση:

<http://eclass.teiep.gr/OpenClass/courses/COMP115/>

# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, Διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



# Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης  
Άρτα, 2015



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Τέλος Ενότητας

Ομάδες Εντολών: Ιδιότητες και Λειτουργίες



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ