

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής  
Αρχιτεκτονική Υπολογιστών  
2018-19

# Αρχιτεκτονικές Συνόλου Εντολών (II)

(Αρχιτεκτονική x86-64)

<http://mixstef.github.io/courses/comparch/>

Μ.Στεφανιδάκης



# Αρχιτεκτονική x86(-64)

- ISA
- Αρχιτεκτονική επεξεργαστών x86(-64)
  - Μία πολύπλοκη αρχιτεκτονική επεξεργαστών
    - 32-bit (IA32, x86) και 64-bit (x86-64, AMD64, Intel 64)
    - Με σύνθετο σετ εντολών
    - Διατήρηση συμβατότητας με παλαιότερους επεξεργαστές της Intel (legacy compatibility)
  - Με πολύ μεγάλη εγκατεστημένη βάση
    - Συστήματα PCs
  - Όχι η βέλτιστη αρχιτεκτονική αλλά:
    - Το μεγάλο μερίδιο που κατέχει στην αγορά επιτρέπει σε Intel και AMD έρευνα και ανάπτυξη για διαρκείς βελτιώσεις απόδοσης
    - “Τελικά η αρχιτεκτονική αυτή είναι βέλτιστη μέσα στην τεράστια πολυπλοκότητά της”

# Αρχιτεκτονική x86(-64)

- ISA

- Οικογένειες επεξεργαστών (Intel)
  - Pentium
    - Pentium Pro, II, III
  - Netburst
    - Pentium 4, D (IA-32 – σε νεώτερες εκδόσεις και 64)
  - Pentium M
    - Intel Core, Core Duo
  - Intel Core (IA-32 -64)
    - Intel Core 2 Duo
  - ...νεώτερες αρχιτεκτονικές

Intel® 64 and IA-32 Architectures Software Developer's Manuals

<https://software.intel.com/en-us/articles/intel-sdm>

# 32-bit και 64-bit λειτουργία

- ISA

- Οι επεξεργαστές x86 των 64 bits λειτουργούν σε
  - 64-bit mode
    - 64-bit OS, 64-bit εφαρμογές
    - 64-bit OS, 32/16-bit εφαρμογές (legacy mode)
  - 32-bit mode
    - 32/16-bit OS, 32/16-bit εφαρμογές
- Ανάλογα με τον τρόπο λειτουργίας αλλάζει η αρχιτεκτονική συνόλου εντολών (ISA)
  - Προσπελάσιμη μνήμη
  - Αριθμός καταχωρητών
  - Μέγεθος δεδομένων/διεθύνσεων μνήμης

# Αρχιτεκτονική Συνόλου Εντολών

- ISA

- Τι περιγράφει;
  - Διαθέσιμες πράξεις/λειτουργίες
    - Κωδικοποίηση εντολών μηχανής
  - Μορφή των δεδομένων
    - Operands
  - Χώροι προσωρινής αποθήκευσης
    - Καταχωρητές
  - Μέθοδοι προσπέλασης μνήμης
    - Προέλευση των δεδομένων
  - Διακοπές και καταστάσεις σφάλματος
    - Ποια η “αντίδραση” του επεξεργαστή

# Κωδικοποίηση Εντολών Μηχανής

- ISA

- Σειρά δυαδικών ψηφίων



Περιγράφει το είδος της πράξης που θα εκτελεστεί

Περιγράφουν την **προέλευση** των δεδομένων εισόδου (αριθμό καταχωρητή, διεύθυνση μνήμης κλπ) και τον **προορισμό** των δεδομένων εξόδου (αποτέλεσμα πράξης)

*Το είδος της πράξης προσδιορίζει τον τύπο, την προέλευση και τον αριθμό των δεδομένων που συμμετέχουν στην πράξη !*

# Εντολές Μηχανής και Assembly

- ISA

- Εντολές μηχανής
  - Σειρές από bits (ομάδες bytes ανά εντολή)
  - Τα bits αυτά κωδικοποιούν **όλα τα χαρακτηριστικά** κάθε εντολής
- Assembly
  - **Μνημονικός** τρόπος αναπαράστασης εντολών
  - Ευκολότερη κατανόηση από τον άνθρωπο



Κάθε διαφορετική  
αρχιτεκτονική  
επεξεργαστών έχει  
διαφορετική  
γλώσσα assembly!

# x86-64: η γενική μορφή των εντολών

- Μεταβλητός αριθμός bytes ανά εντολή
  - 1 έως 15 bytes
- Συνδυασμός από
  - Προθέματα (prefixes)
    - Αλλάζουν τον τρόπο λειτουργίας, το είδος και το μέγεθος των δεδομένων, τον αριθμό των χρησιμοποιούμενων καταχωρητών ή την κωδικοποίηση της εντολής
  - Opcode (1 byte)
    - Ποια η βασική λειτουργία της εντολής
  - ModRM και SIB (Scale-Index-Base) bytes
    - Επιλογή καταχωρητή/μνήμης και τρόπου σχηματισμού διεύθυνσης
  - Displacement ή/και Immediate bytes (1, 2, 4 ή 8)
    - Offset και άμεσες τιμές μέσα στην εντολή



# x86-64: CISC ή RISC;

- Οι εντολές που περιγράφει η αρχιτεκτονική συνόλου εντολών είναι σύνθετες
  - Πολλές εντολές πέραν των load/store προσπελούν τη μνήμη
  - Ένα τυπικό παράδειγμα αρχιτεκτονικής CISC
  - Αναγκαίο για την τήρηση της προς τα πίσω συμβατότητας
- Αλλά
  - Εσωτερικά, κάθε σύνθετη εντολή μετατρέπεται σε μια σειρά μικρότερες
    - Micro-operations (μops)
    - Αυτόματα, κατά την αποκωδικοποίηση
    - Ουσιαστικά η εκτέλεση είναι σε στυλ RISC!

# x86-64: Είδη εντολών

- Εντολές γενικού σκοπού
  - Αριθμητικές και λογικές πράξεις
  - Μεταφορά δεδομένων
    - Από-πρός Καταχωρητές και Μνήμη (PUSH/POP για στοίβα)
  - Έλεγχος ροής εκτέλεσης
    - Διακλαδώσεις και κλήσεις ρουτινών
- Άλλες κατηγορίες
  - Ειδικές εντολές συστήματος
    - ΛΣ, εικονική μνήμη
  - Επεξεργασία πολλαπλών δεδομένων
    - Χρήσιμο για γραφικά, σειρές χαρακτήρων, multimedia

# Εντολές μεταφοράς δεδομένων

**Table 7-1. Move Instruction Operations**

Type of Data Movement	Source →Destination
From memory to a register	Memory location →General-purpose register Memory location →Segment register
From a register to memory	General-purpose register →Memory location Segment register →Memory location
Between registers	General-purpose register →General-purpose register General-purpose register →Segment register Segment register →General-purpose register General-purpose register →Control register Control register →General-purpose register General-purpose register →Debug register Debug register →General-purpose register
Immediate data to a register	Immediate →General-purpose register
Immediate data to memory	Immediate →Memory location

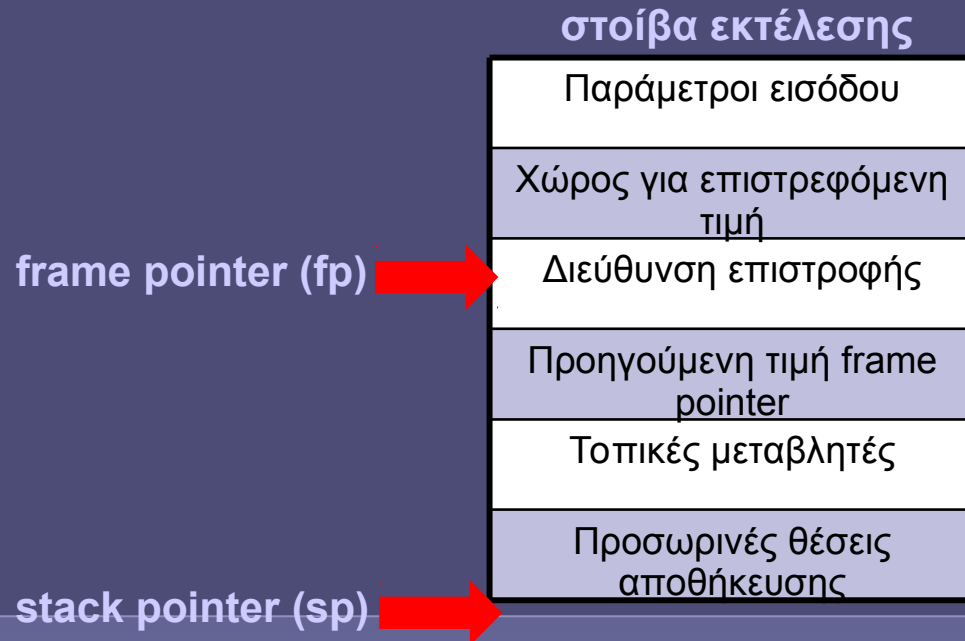
# x86-64: Εντολές διακλάδωσης

- ISA
  - Με ή χωρίς συνθήκη
    - Σε απόλυτη ή σχετική διεύθυνση
      - πιθανώς αποθηκευμένη σε μνήμη ή καταχωρητή
    - Συνθήκη: καταχωρητής κατάστασης EFLAGS
    - Μεταξύ τμημάτων κώδικα
      - της ίδιας ή διαφορετικής διεργασίας
  - Κλήση συναρτήσεων
    - Όπως η διακλάδωση με αποθήκευση πρόσθετης πληροφορίας (program stack)
      - για επιστροφή από συνάρτηση στο σημείο μετά την κλήση

# Εντολές διακλάδωσης

- ISA

- **Κλήση συναρτήσεων και επιστροφή**
  - Αποθήκευση της τρέχουσας διεύθυνσης εκτέλεσης (καταχωρητή PC)
    - Πριν τη μετάβαση στη συνάρτηση
  - Χρήση αποθηκευμένης τιμής
    - Κατά την επιστροφή (return)



# x86-64: Προέλευση δεδομένων

- ISA
- Προέλευση δεδομένων

- Τα δεδομένα που επεξεργάζονται οι εντολές προέρχονται από
  - Καταχωρητές
  - Μνήμη (ή I/O ports)
  - Τιμή μέσα στην εντολή (immediate values)

Και μπορούν να αποθηκευτούν σε

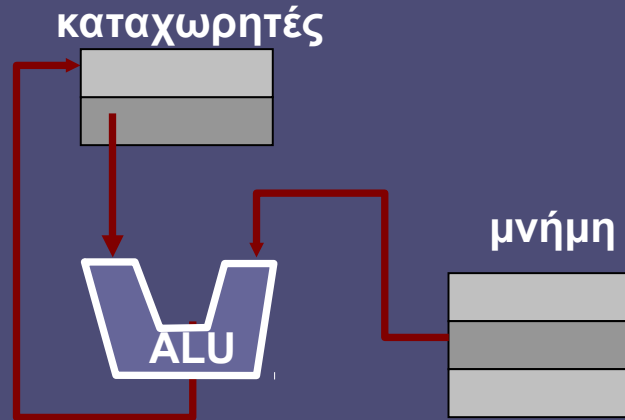
- Καταχωρητές
- Μνήμη (ή I/O ports)

# x86-64: Αρχιτεκτονική με καταχωρητές

- ISA
- Προέλευση δεδομένων



Καταχωρητές:  
προσωρινές θέσεις  
αποθήκευσης  
αποτελεσμάτων.



## ■ Memory-register

- Οποιαδήποτε εντολή γενικού σκοπού μπορεί να προσπελάσει τη μνήμη
- Όχι μόνο οι εντολές load/store

# x86-64: αρχιτεκτονική memory-register

- ISA
- Προέλευση δεδομένων

- Πολλές εντολές γενικού σκοπού
  - πέρα από τις load και store
  - έχουν ως πηγή ή προορισμό τη μνήμη, π.χ.

## AND—Logical AND

Opcode	Instruction
...	
81 /4 iw	AND r/m16, imm16
81 /4 id	AND r/m32, imm32
...	
21 /r	AND r/m16, r16
21 /r	AND r/m32, r32
REX.W + 21 /r	AND r/m64, r64
22 /r	AND r8, r/m8
...	
23 /r	AND r32, r/m32
REX.W + 23 /r	AND r64, r/m64
...κλπ..	



# x86-64: καταχωρητές

- ISA
- Προέλευση δεδομένων

- 16 καταχωρητές γενικού σκοπού
  - 64 bits: RAX, RBX, RCX, RDX, RBP, RSI, RDI, RSP, R8–R15
  - 32 bits (το χαμηλό μισό των προηγούμενων): EAX, EBX, ECX, EDX, ESI, EDI, ESP, EBP, R8D-R15D
  - 16 bits (το χαμηλό τέταρτο): AX, BX, CX, DX, SI, DI, SP, BP, R8W-R15W
  - 8 bits (τμήματα των προηγούμενων): AL, BL, CL, DL, AH, BH, CH, DH, SIL, DIL, SPL, BPL, R8L-R15L
- Κάποιοι από τους καταχωρητές αυτούς χρησιμοποιούνται με ειδικό τρόπο!

# x86-64: καταχωρητές (2)

- ISA
- Προέλευση δεδομένων

- RIP (EIP στα 32 bits)
  - Program counter
- RFLAGS (EFLAGS)
  - Σημαίες κατάστασης
- Καταχωρητές πράξεων SIMD
  - Εύρος καταχωρητών έως 512 bits
- Λοιποί καταχωρητές ελέγχου
  - Διαχείριση μνήμης, διεργασιών, προνομίων, τρόπου λειτουργίας κ.ο.κ

# x86-64: Μοντέλο μνήμης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

- Ο χώρος διευθύνσεων που βλέπει κάθε εκτελούμενο πρόγραμμα
  - Στα 64 bit: “flat model”
    - Ενιαίος χώρος διευθύνσεων, από το 0 μέχρι το μέγιστο δυνατό ( $2^{64}-1$ )
    - Στην πράξη οι σύγχρονοι επεξεργαστές διαθέτουν 48-52 γραμμές διεύθυνσης
    - Στα 32 bit το μοντέλο είναι πιο πολύπλοκο (segmentation)
  - Προσοχή: πρόκειται για εικονικές διεύθυνσεις
    - Παράγονται από την ΚΜΕ
    - Θα ακολουθήσει μετάφραση σε φυσική διεύθυνση μέσω του συστήματος εικονικής μνήμης
    - Virtual Memory (σε επόμενο μάθημα)

# Μέθοδοι προσπέλασης μνήμης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

- Πολλές εντολές γενικού σκοπού προσπελάνουν τη μνήμη
  - για ανάγνωση ή εγγραφή δεδομένων
  - Πώς σχηματίζεται η διεύθυνση προσπέλασης;
  - Η γενική ιδέα: υποβοήθηση του λογισμικού
  - Διαφορετικός σχηματισμός διεύθυνσης για
    - Τοπικές μεταβλητές
    - Δείκτες (έμμεση προσπέλαση)
    - Στατικά δεδομένα
    - Διάσχιση πινάκων
    - (Σταθερές τιμές)
- Υποστήριξη ανάλογα με αρχιτεκτονική

# x86-64: Σχηματισμός διευθύνσεων

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

- Στατική τιμή διεύθυνσης
  - Εντός εντολής
- Υπολογιζόμενη τιμή διεύθυνσης
$$address = scale * index + base + offset$$
  - Scale: 2, 4 ή 8
  - Index: η τιμή ενός καταχωρητή γενικού σκοπού
  - Base: η τιμή ενός καταχωρητή γενικού σκοπού
  - Offset (ή displacement): μια 8-, 16- ή 32-bit τιμή (εντός εντολής)