

Αρχιτεκτονική Υπολογιστών



Γιώργος Δημητρίου

Ενότητα 2^η: Σύνολα Εντολών

Ιστορική Αναδρομή

- Από τις πρώτες εντολές υπολογιστών...

A n Add the number in storage location n into the accumulator.

E n If the number in the accumulator is greater than or equal to zero execute next the order which stands in storage location n ; otherwise proceed serially.

Z Stop the machine and ring the warning bell.

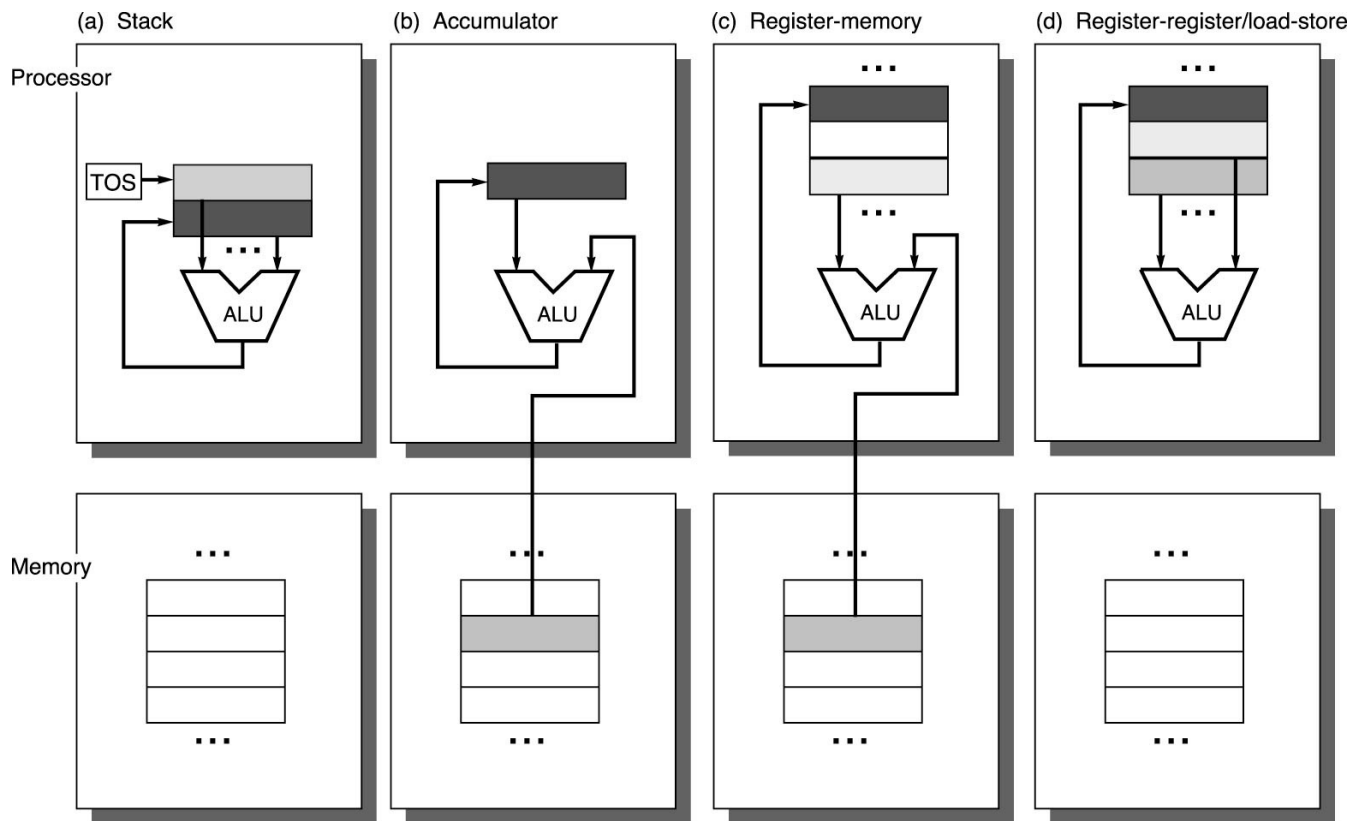
Wilkes and Renwick

*Selection from the List of 18 Machine
Instructions for the EDSAC (1949)*

Αρχιτεκτονική Συνόλου Εντολών

- Στοίβας
- Συσσωρευτή
 - Επέκταση συσσωρευτή
- Καταχωρητών γενικού σκοπού
 - Καταχωρητή-Μνήμης
 - Καταχωρητή-Καταχωρητή
 - (Μνήμης-Μνήμης)

Παράδειγμα: add



Προσπέλαση Μνήμης

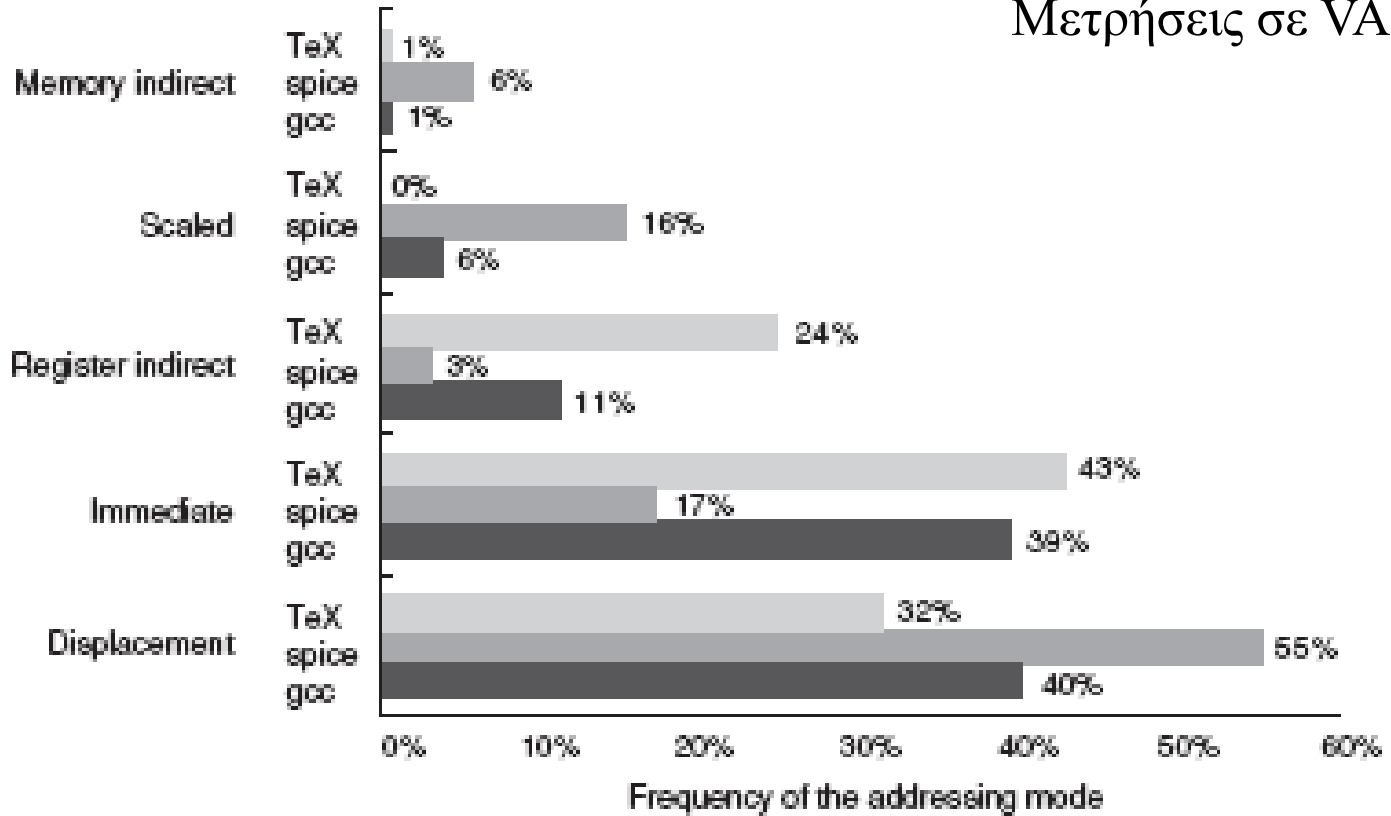
- Σειρά αποθήκευσης ψηφιολέξεων:
 - Σειρά μικρού άκρου (little endian)
 - Σειρά μεγάλου άκρου (big endian)
- Ευθυγράμμιση στη μνήμη:
 - Σύνδεση με ΜΔΜ
 - Προσπέλαση μη ευθυγραμμισμένων δεδομένων
 - Προσπέλαση μικρών δεδομένων

Μέθοδοι Διευθυνσιοδότησης

Addressing mode	Example instruction	Meaning	When used
Register	Add R4,R3	$\text{Regs}[R4] \leftarrow \text{Regs}[R4] + \text{Regs}[R3]$	When a value is in a register.
Immediate	Add R4,#3	$\text{Regs}[R4] \leftarrow \text{Regs}[R4] + 3$	For constants.
Displacement	Add R4,100(R1)	$\text{Regs}[R4] \leftarrow \text{Regs}[R4] + \text{Mem}[100+\text{Regs}[R1]]$	Accessing local variables (+ simulates register indirect, direct addressing modes).
Register indirect	Add R4,(R1)	$\text{Regs}[R4] \leftarrow \text{Regs}[R4] + \text{Mem}[\text{Regs}[R1]]$	Accessing using a pointer or a computed address.
Indexed	Add R3,(R1+R2)	$\text{Regs}[R3] \leftarrow \text{Regs}[R3] + \text{Mem}[\text{Regs}[R1]+\text{Regs}[R2]]$	Sometimes useful in array addressing: R1 = base of array; R2 = index amount.
Direct or absolute	Add R1,(1001)	$\text{Regs}[R1] \leftarrow \text{Regs}[R1] + \text{Mem}[1001]$	Sometimes useful for accessing static data; address constant may need to be large.
Memory indirect	Add R1,@(R3)	$\text{Regs}[R1] \leftarrow \text{Regs}[R1] + \text{Mem}[\text{Mem}[\text{Regs}[R3]]]$	If R3 is the address of a pointer p , then mode yields $*p$.
Autoincrement	Add R1,(R2)+	$\text{Regs}[R1] \leftarrow \text{Regs}[R1] + \text{Mem}[\text{Regs}[R2]]$ $\text{Regs}[R2] \leftarrow \text{Regs}[R2] + d$	Useful for stepping through arrays within a loop. R2 points to start of array; each reference increments R2 by size of an element, d .
Autodecrement	Add R1,-(R2)	$\text{Regs}[R2] \leftarrow \text{Regs}[R2] - d$ $\text{Regs}[R1] \leftarrow \text{Regs}[R1] + \text{Mem}[\text{Regs}[R2]]$	Same use as autoincrement. Autodecrement/increment can also act as push/pop to implement a stack.
Scaled	Add R1,100(R2)[R3]	$\text{Regs}[R1] \leftarrow \text{Regs}[R1] + \text{Mem}[100+\text{Regs}[R2] + \text{Regs}[R3]*d]$	Used to index arrays. May be applied to any indexed addressing mode in some computers.

Συχνότητα Διευθynσιοδοτήσεων

Μετρήσεις σε VAX



Τελούμενα Εντολών

- Κωδικοποίηση τύπου στον κωδικό λειτουργίας
- Διαχωρισμός ακέραιων από πραγματικούς
- Πυκνωμένοι BCD αριθμοί
 - Πρόβλημα διαφορετικής ακρίβειας
- Συμβολοσειρές
- Μέγεθος τελούμενων

Εντολές

- Οι 10 συχνότερες εντολές του 80x86:

1	load	22%
2	conditional branch	20%
3	compare	16%
4	store	12%
5	add	8%
6	and	6%
7	sub	5%
8	move register-register	4%
9	call	1%
10	return	1%
Σύνολο		~96%

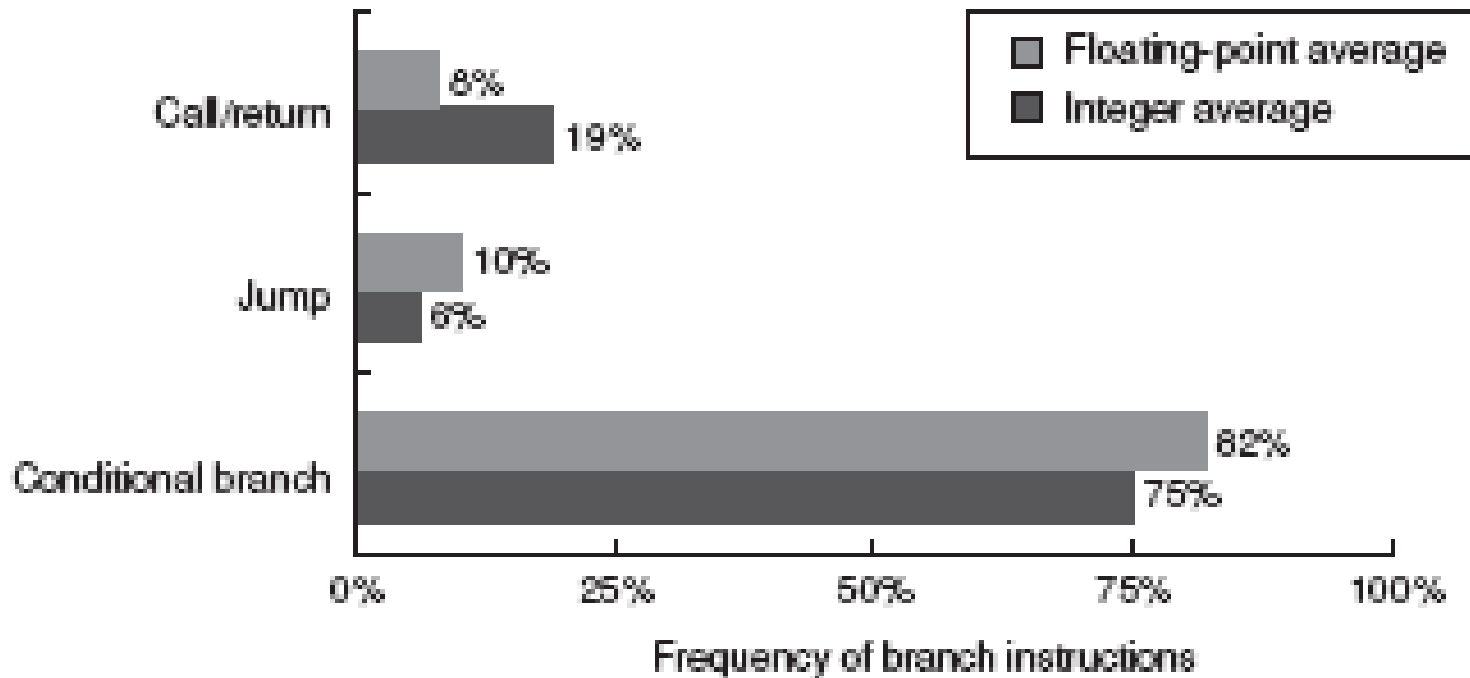
Εντολές για Πολυμέσα & Επεξεργασία Σημάτων

- Εντολές SIMD:
 - Εκμεταλλεύονται τη δυνατότητα διάσπασης λέξεων σε μικρότερα δεδομένα
 - Ανεξαρτησία πράξεων μεταξύ ανεξάρτητων τμημάτων
- Εντολές χωρίς υπερχείλιση
 - Αποφυγή υπερχείλισης με φραγμό
- Ειδικές μέθοδοι στρογγυλοποίησης
- Εντολές πρόσθεσης-γινομένου
- Διευθυνσιοδότηση για FFT

Εντολές Ελέγχου Ροής

- Διακλαδώσεις (με συνθήκη)
- Άλματα (χωρίς συνθήκη)
- Κλήσεις υποπρογραμμάτων
- Επιστροφές από υποπρογράμματα
- Υπολογισμός διευθύνσεων προορισμού
 - Διευθυνσιοδότηση σχετική με τον PC
- Άγνωστες διευθύνσεις
 - Έμμεσα άλματα

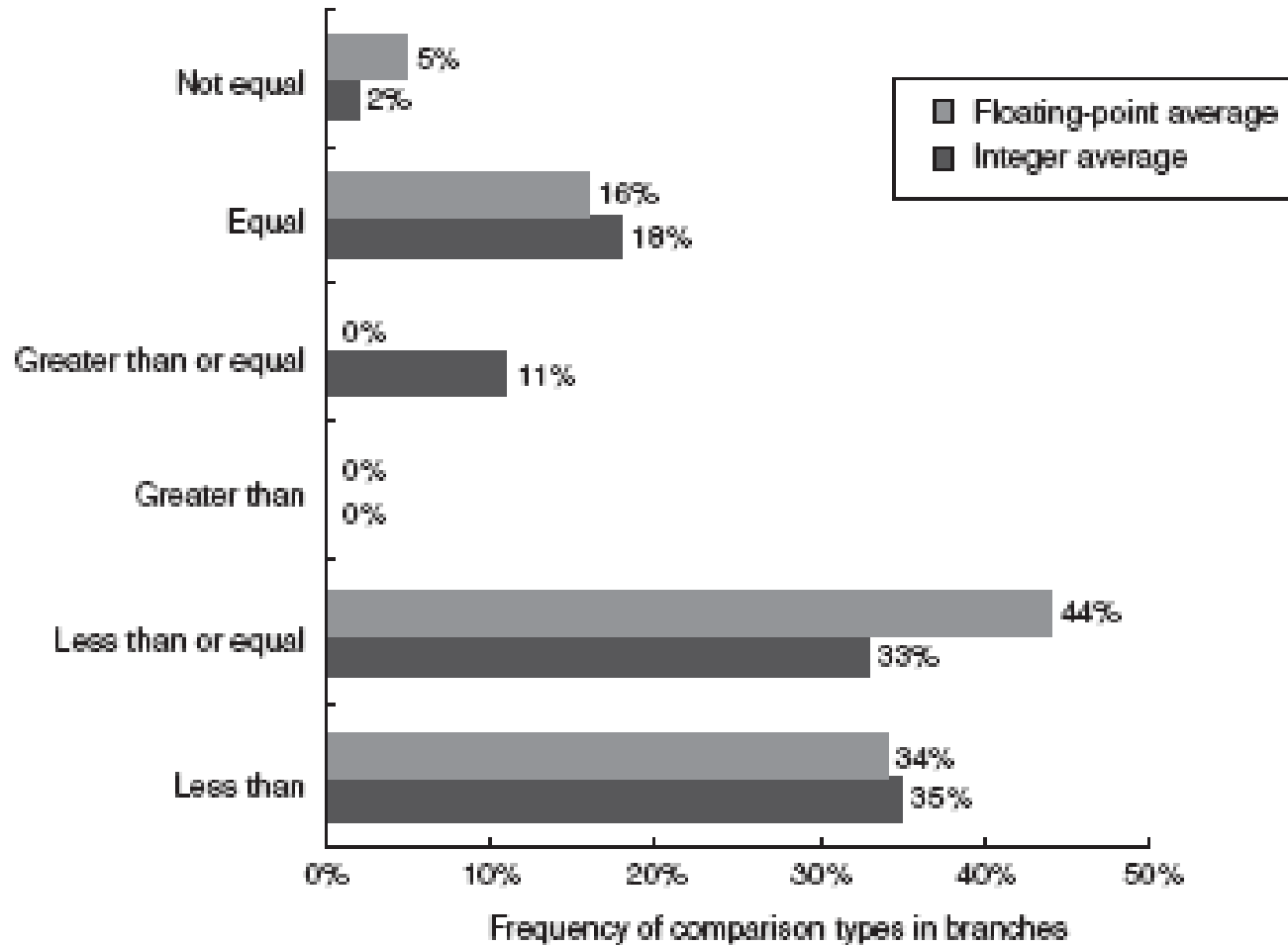
Συχνότητα Εντολών Ελέγχου Ροής



Υλοποίηση Ελέγχου Ροής

Name	Examples	How condition is tested	Advantages	Disadvantages
Condition code (CC)	80x86, ARM, PowerPC, SPARC, SuperH	Tests special bits set by ALU operations, possibly under program control.	Sometimes condition is set for free.	CC is extra state. Condition codes constrain the ordering of instructions since they pass information from one instruction to a branch.
Condition register	Alpha, MIPS	Tests arbitrary register with the result of a comparison.	Simple.	Uses up a register.
Compare and branch	PA-RISC, VAX	Compare is part of the branch. Often compare is limited to subset.	One instruction rather than two for a branch.	May be too much work per instruction for pipelined execution.

Συνθήκες Ελέγχου Ροής



Κωδικοποίηση Εντολών

- CISC
 - Πολλές μέθοδοι διευθυνσιοδότησης απαιτούν ειδική κωδικοποίηση για τη διευθυνσιοδότηση
 - Πολλαπλά τελούμενα με διαφορετικές μεθόδους διευθυνσιοδότησης οδηγούν σε μεταβλητό μέγεθος από εντολή σε εντολή
- RISC
 - Μία ή δύο μέθοδοι διευθυνσιοδότησης οδηγούν σε σταθερό μέγεθος εντολής

Μέθοδοι Κωδικοποίησης



(a) Variable (e.g., Intel 80x86, VAX)



(b) Fixed (e.g., Alpha, ARM, MIPS, PowerPC, SPARC, SuperH)



(c) Hybrid (e.g., IBM 360/370, MIPS16, Thumb, TI TMS320C54x)