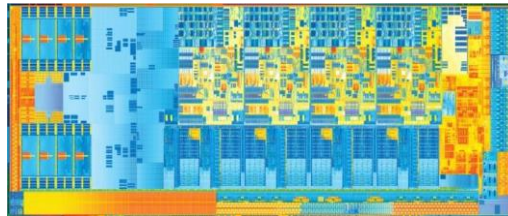


Αρχιτεκτονική Συστημάτων Υπολογιστών

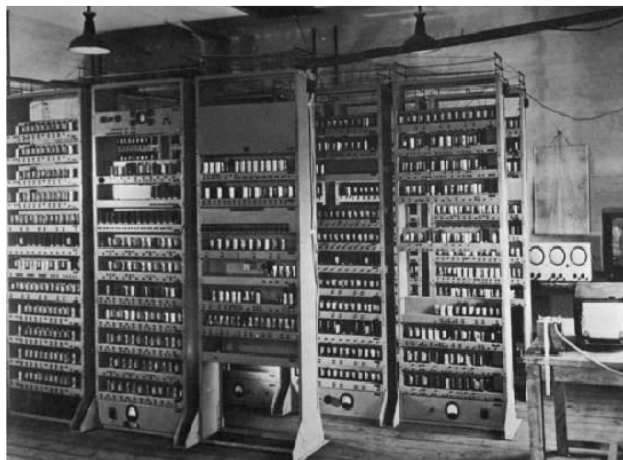
Γιώργος Κορνάρος, Άνοιξη 2023



Ο υπολογιστής
που θα
κατανοεί ο
φοιτητής τον
Ιούνιο

1

Οι Υπολογιστές Τότε...



2

Οι Υπολογιστές Σήμερα...



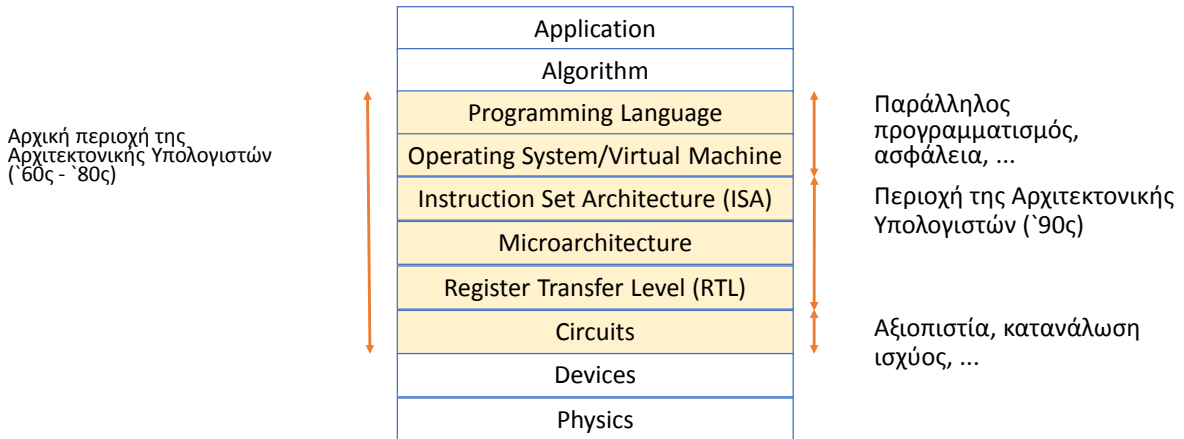
3

Ταξιδεύοντας στο Χρόνο

- Τι κάνουν στην πραγματικότητα οι αρχιτέκτονες υπολογιστών ?
- Παραδείγματα από την ιστορία
 - Οι πρώτες μέρες: ENIAC, EDVAC, and EDSAC
 - Άφιξη των IBM 650 και IBM 360
 - Seymour Cray – CDC 6600, Cray 1
 - Μικροεπεξεργαστές και PCs
 - Πολυπύρηντοι επεξεργαστές (Multicores/Manycores)
 - Κινητά τηλέφωνα
- Επικέντρωση σε ιδέες, μηχανισμούς και αρχές, ειδικά αυτές που έχουν διαχρονικά επηρεάσει την αρχιτεκτονική υπολογιστών

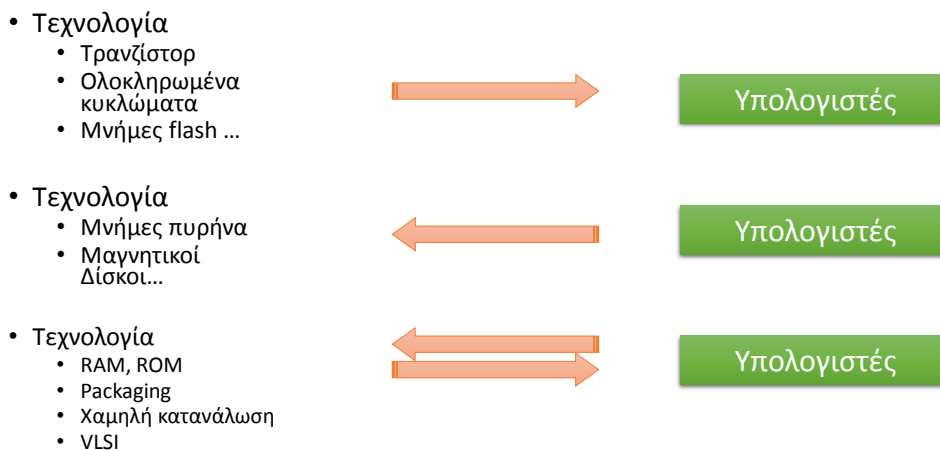
4

Επίπεδα Αφαιρετικότητας



5

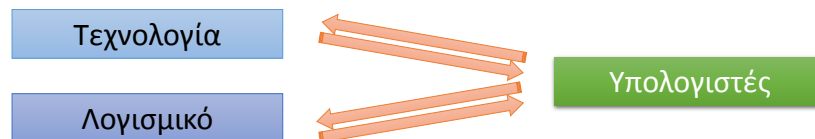
Τεχνολογία στην Εξέλιξη Υπολογιστών



6

Που Είναι το Λογισμικό ?

- Καθώς οι άνθρωποι γράφουν προγράμματα και χρησιμοποιούν υπολογιστές βελτιώνεται η κατανόηση που έχουμε για τον προγραμματισμό και τη συμπεριφορά του λογισμικού
- Το λογισμικό έχει τεράστια επίδραση στην αρχιτεκτονική υπολογιστών – αλλά με αργούς ρυθμούς...
- Οι σύγχρονοι αρχιτέκτονες πρέπει να γνωρίζουν και να κατανοούν τα προβλήματα του λογισμικού και των συμβολομεταφραστών



7

Η Αρχιτεκτονική είναι Σχεδίαση λαμβάνοντας υπόψιν Περιορισμούς

- Σημαντικοί παράγοντες:
 - Απόδοση του συνολικού συστήματος για συγκεκριμένες εφαρμογές
 - Μέση περίπτωση και χειρότερη περίπτωση
 - Κόστος κατασκευής chip και συστημάτων υποστήριξης
 - Απαιτούμενη ισχύς για την λειτουργία του συστήματος
 - Μέγιστη ισχύς και ισχύς ανά (μικρο-)λειτουργία
 - Αξιοπιστία του συστήματος
 - Κόστος σχεδίασης (μηχανικοί, υπολογιστές, εργαλεία CAD)
 - Κόστος ανάπτυξης εφαρμογών και λογισμικού συστήματος
- Οι επιλογές αρχιτεκτονικής διαφοροποιούνται σημαντικά σε συνάρτηση με το ισοζύγιο αυτών των παραγόντων στο πέρασμα του χρόνου

8

Οι Υπολογιστές το '50...

- Ακριβό hardware
- Ο χώρος αποθήκευσης ήταν μικρός (1000 λέξεις)
- Ο χρόνος προσπέλασης της μνήμης ήταν 10 έως 50 φορές μικρότερος από τον κύκλο ρολογιού του επεξεργαστή
- Το βάρος στη σχεδίαση ήταν πολύπλοκα κυκλώματα για την εκτέλεση των εντολών, όχι η ταχύτητα αποκωδικοποίησης ή άλλης λειτουργίας
- Η οπτική του προγραμματιστή για τον υπολογιστή ήταν στενά συνδεδεμένη με το hardware

Τα Πρώιμα Σετ Εντολών – Accumulator-based

| | | |
|-------------|---|--|
| LOAD | x | $AC \leftarrow M[x]$ |
| STORE | x | $M[x] \leftarrow (AC)$ |
| ADD | x | $AC \leftarrow (AC) + M[x]$ |
| SUB | x | |
| MUL | x | Involves a quotient register |
| DIV | x | |
| SHIFT LEFT | | $AC \leftarrow 2 \leftarrow (AC)$ |
| SHIFT RIGHT | | |
| JUMP | x | $PC \leftarrow x$ |
| JGE | x | if $(AC) \geq 0$ then $PC \leftarrow x$ |
| LOAD ADR | x | $AC \leftarrow \text{Extract address field}(M[x])$ |
| STORE ADR | x | |

- Προγράμματα
~ 20-30
εντολών

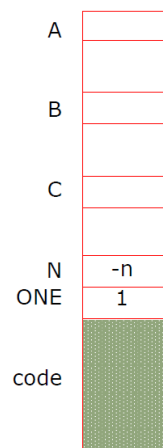
Οι καταχωρητές ήταν ακριβοί !!

11

Προγραμματισμός: η Μηχανή του ενός Συσσωρευτή

$$C_i \leftarrow A_i + B_i, 1 \leq i \leq n$$

| | | |
|------|-------|------|
| LOOP | LOAD | N |
| | JGE | DONE |
| | ADD | ONE |
| | STORE | N |
| F1 | LOAD | A |
| F2 | ADD | B |
| F3 | STORE | C |
| | JUMP | LOOP |
| DONE | HLT | |



Πρόβλημα: πως να αλλαχθούν οι διευθύνσεις των A, B, C κατά την εκτέλεση?

12

Αυτο-Τροποποιούμενος Κώδικας

Τροποποίηση του προγράμματος για την επόμενη επανάληψη

| | | |
|------|-----------|------|
| LOOP | LOAD | N |
| | JGE | DONE |
| | ADD | ONE |
| | STORE | N |
| F1 | LOAD | A |
| F2 | ADD | B |
| F3 | STORE | C |
| | LOAD ADR | F1 |
| | ADD | ONE |
| | STORE ADR | F1 |
| | LOAD ADR | F2 |
| | ADD | ONE |
| | STORE ADR | F2 |
| | LOAD ADR | F3 |
| | ADD | ONE |
| | STORE ADR | F3 |
| | JUMP | LOOP |
| DONE | HLT | |

$$C_i \leftarrow A_i + B_i, 1 \leq i \leq n$$

Κάθε επανάληψη περιλαμβάνει:

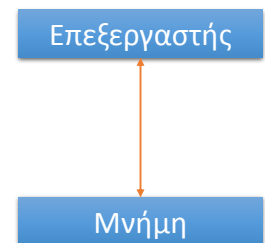
| | Σύνολο | Διαχείριση Μνήμης |
|---------------------|--------|----------------------|
| Προσκόμιση εντολών | 17 | 14 |
| Προσκόμιση τελεστών | 10 | 8 |
| Αποθήκευση | 5 | 4 |

Οι περισσότερες εντολές του προγράμματος είναι για την διαχείριση μνήμης (μεταβλητών)

13

Bottleneck Επεξεργαστή-Μνήμης

- Δυνατότητα indexing
 - Στόχος η μείωση των εντολών διαχείρισης μνήμης
- Γρήγορη τοπική μνήμη μέσα στον επεξεργαστή
 - 8-16 καταχωρητές αντί ενός συσσωρευτή
 - Στόχος η μείωση του αριθμού των load/stores
- Πολύπλοκες εντολές
 - Στόχος η μείωση του αριθμού προσκόμισης εντολών
- Μικρότερες εντολές
 - Bit διεύθυνσης εσωτερικά στην εντολή για τους τελεστές
 - Στόχος η μείωση του κόστους προσκόμισης εντολών



14

Εξέλιξη των Modes Διευθυνσιοδότησης

- Απλός συσσωρευτής, απόλυτη διεύθυνση
 - LOAD X
 - Απλός συσσωρευτής, καταχωρητές index
 - LOAD X, IX
 - Έμμεση διεύθυνση
 - LOAD (X)
 - Πολλαπλοί συσσωρευτές, καταχωρητές index, έμμεση διευθυνσιοδότηση
 - LOAD R, IX, X
 - LOAD R, IX, (X)
 - Έμμεση μέσω καταχωρητών
 - LOAD Ri, (Rj)
- ή $R \leftarrow M[M[x] + (IX)]$
 $R \leftarrow M[M[x + (IX)]]$

17

Ποικιλία Τρόπων Διευθυνσιοδότησης/Formats

- 3 διευθύνσεις ανά εντολή: δύο τελεστές ορίσματα και ένα αποτέλεσμα

(Reg op Reg) to Reg $RI \leftarrow (RJ) \text{ op } (RK)$
 (Reg op Mem) to Reg $RI \leftarrow (RJ) \text{ op } M[x]$

- 2 διευθύνσεις ανά εντολή: ο προορισμός/αποτέλεσμα είναι το ίδιο με ένα από όρισμα

(Reg op Reg) to Reg $RI \leftarrow (RI) \text{ op } (RJ)$
 (Reg op Mem) to Reg $RI \leftarrow (RI) \text{ op } M[x]$

18

Σετ Εντολών στο Παρελθόν....

- Μεγάλη ποικιλία σετ εντολών, εξαρτώμενα από συνιστώσες υλοποίησης
- Η άποψη του προγραμματιστή ήταν άρρηκτα συνδεδεμένη με την υλοποίηση σε hardware