

Εισαγωγή στην επιστήμη των υπολογιστών

Οργάνωση υπολογιστών

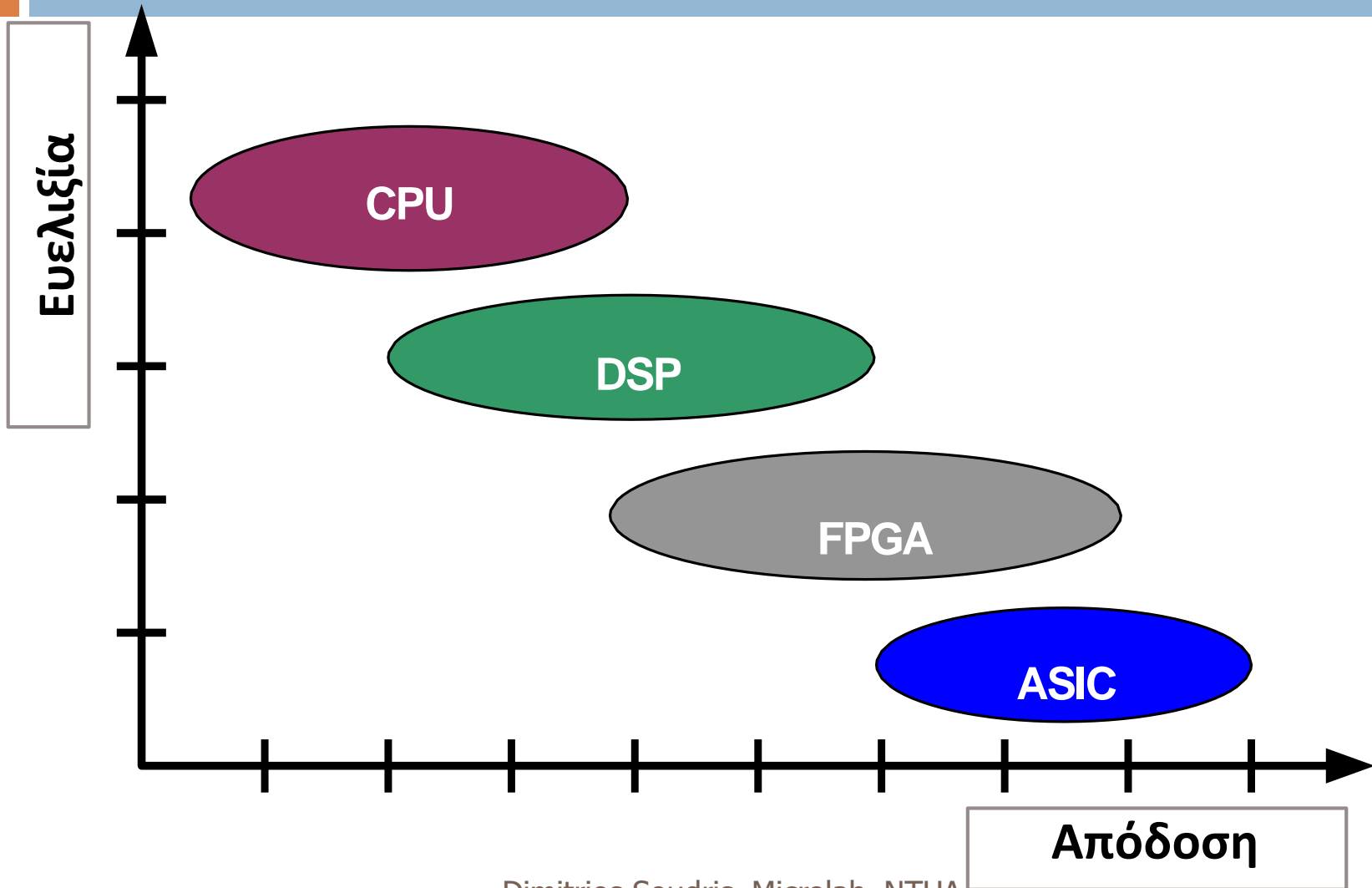
dsoudris@microlab.ntua.gr

Υπολογιστικές Πλατφόρμες

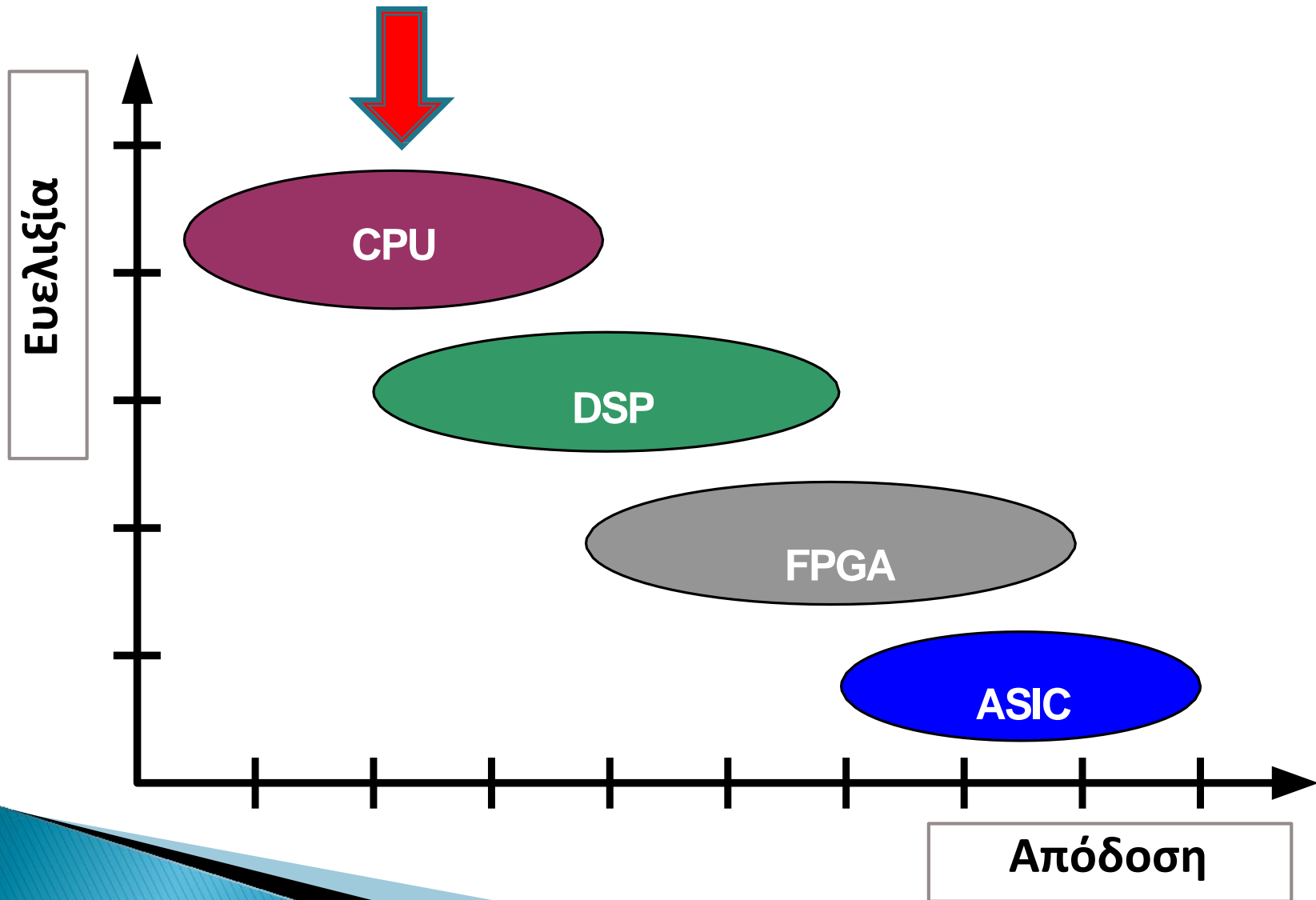
- Επεξεργαστής (processor)
- Επεξεργαστής Ψηφιακού Σήματος (Digital Signal Processor)
- Field programmable Gate Array (FPGA)
- Ολοκληρωμένο Κύκλωμα Ειδικού Σκοπού (Application Specific Integrated Circuit: ASIC)

Υπολογιστικές Πλατφόρμες

3

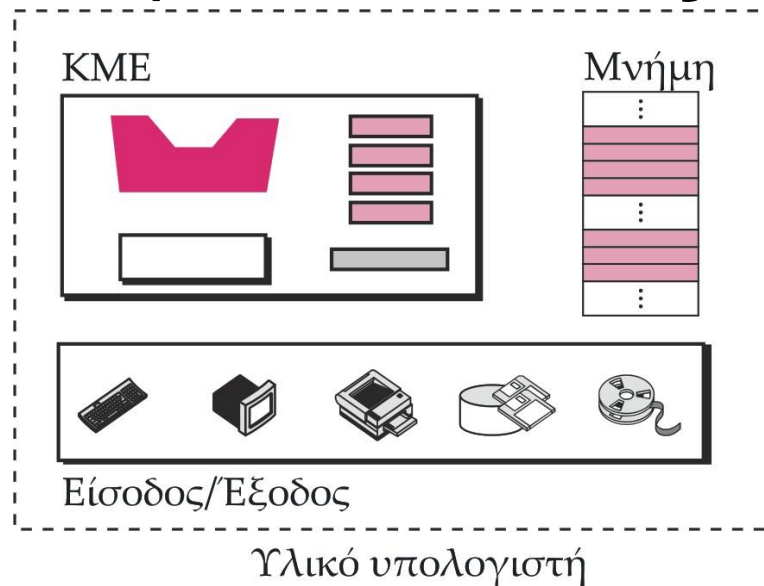


Υπολογιστικές Πλατφόρμες

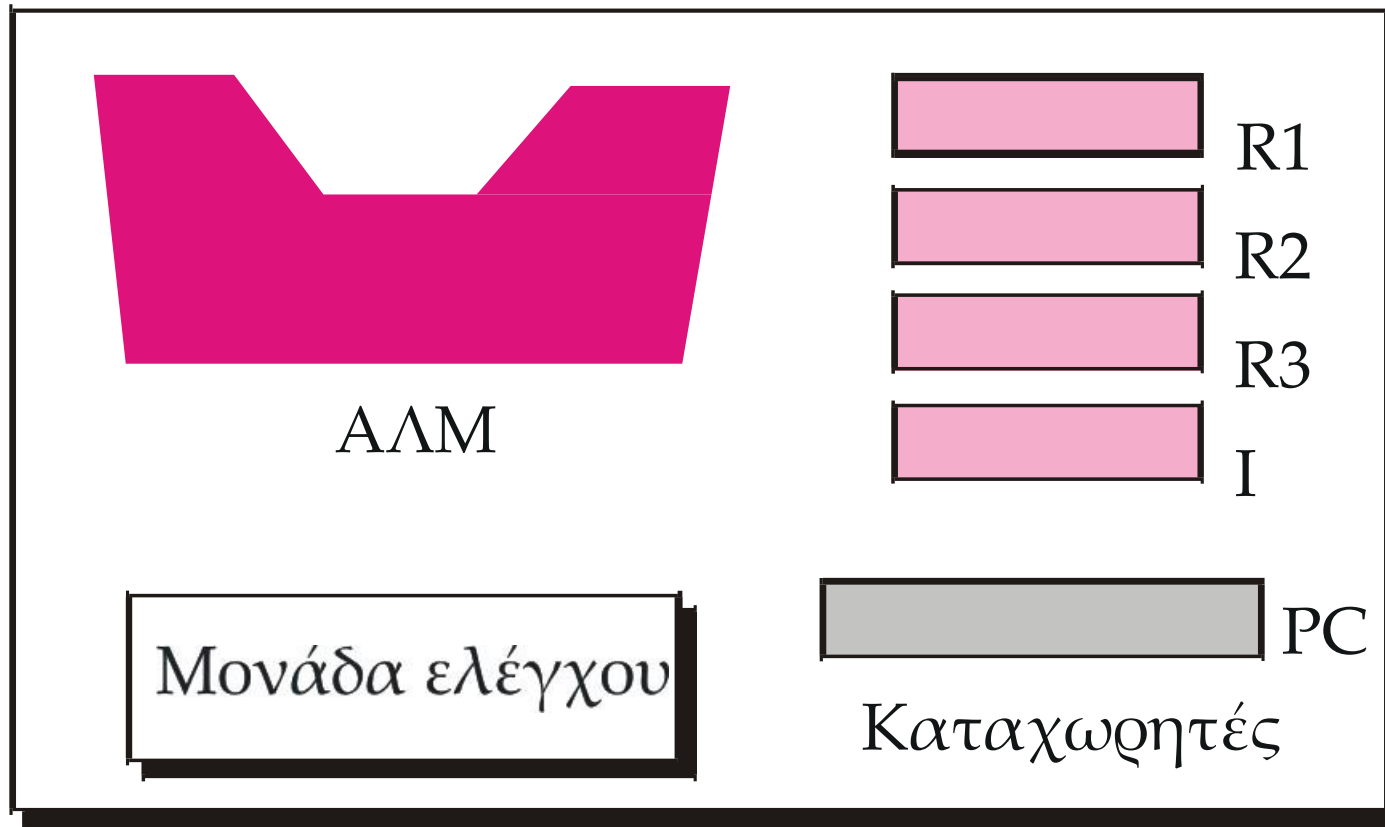


Οργάνωση υπολογιστών

- ▶ ΚΜΕ
- ▶ Κύρια Μνήμη
- ▶ Υποσύστημα εισόδου/εξόδου



Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)



Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)

- ▶ **αριθμητική και λογική μονάδα ή ΑΛΜ** (arithmetic logic unit, ALU)
 - εκτελεί αριθμητικές και λογικές πράξεις
- ▶ **καταχωρητές (registers)**
 - γρήγορες αυτόνομες θέσεις αποθήκευσης για την προσωρινή αποθήκευση δεδομένων
 - Δεδομένων, Εντολών, Μετρητής προγράμματος
- ▶ **μονάδα ελέγχου (control unit)**
 - μοιάζει με το τμήμα του ανθρώπινου εγκεφάλου που ελέγχει τη λειτουργία κάθε οργάνου του σώματος και ο έλεγχος αυτός επιτυγχάνεται μέσω διαύλων που είναι είτε ανοιχτοί είτε κλειστοί

Κύρια Μνήμη

- ▶ Η κύρια μνήμη (main memory) είναι μια συλλογή από θέσεις αποθήκευσης, κάθε μία από τις οποίες διαθέτει ένα μοναδικό αναγνωριστικό που ονομάζεται διεύθυνση.
- ▶ Τα δεδομένα μεταφέρονται από και προς τη μνήμη σε ομάδες από bit οι οποίες ονομάζονται λέξεις (words). Μια λέξη μπορεί να είναι μια ομάδα των 8, 16, 32, ή μερικές φορές 64 bit

Ταξινόμηση Αρχιτεκτονικών Υπολογιστών

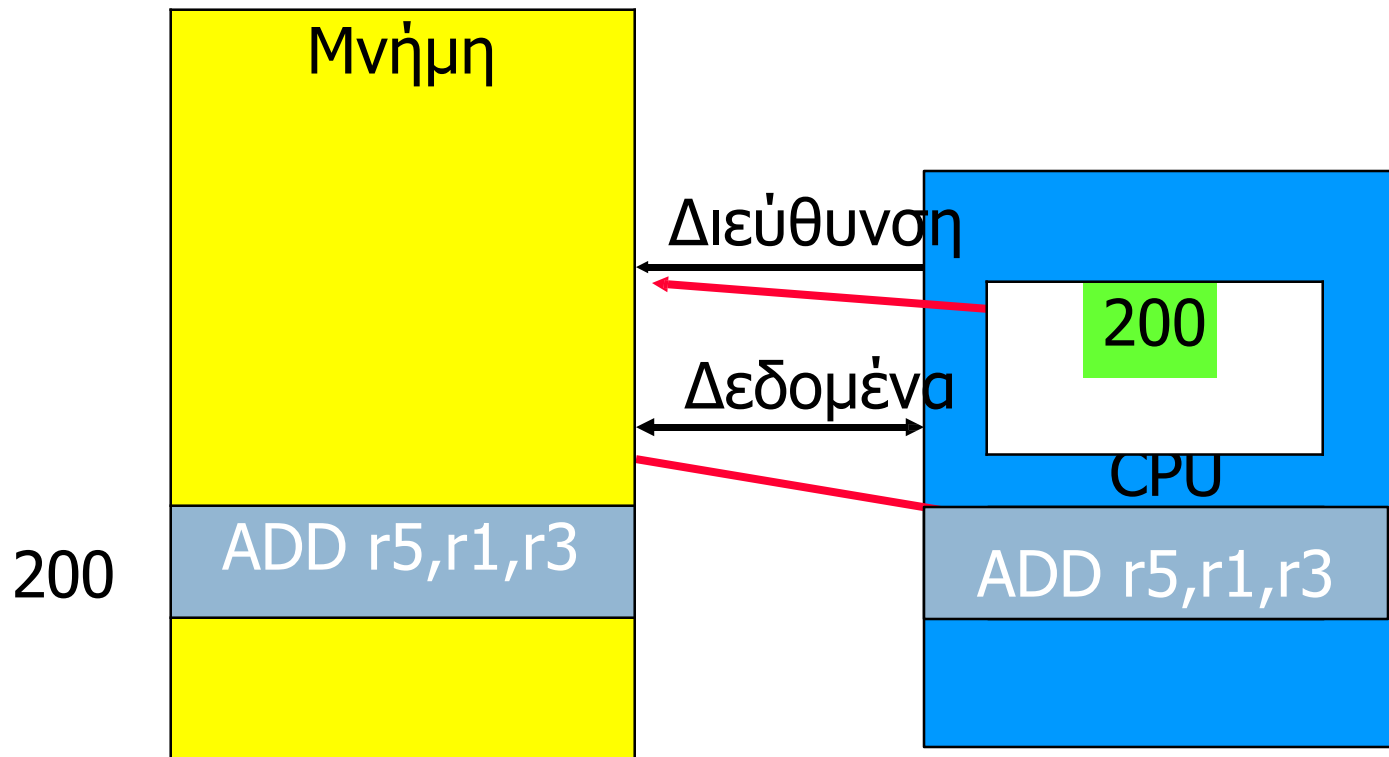
von Neumann architecture

13

- Memory holds data, instructions.
- Central processing unit (CPU) fetches instructions from memory.
 - Separate CPU and memory distinguishes programmable computer.
- CPU registers help out: program counter (PC), instruction register (IR), general-purpose registers, etc.

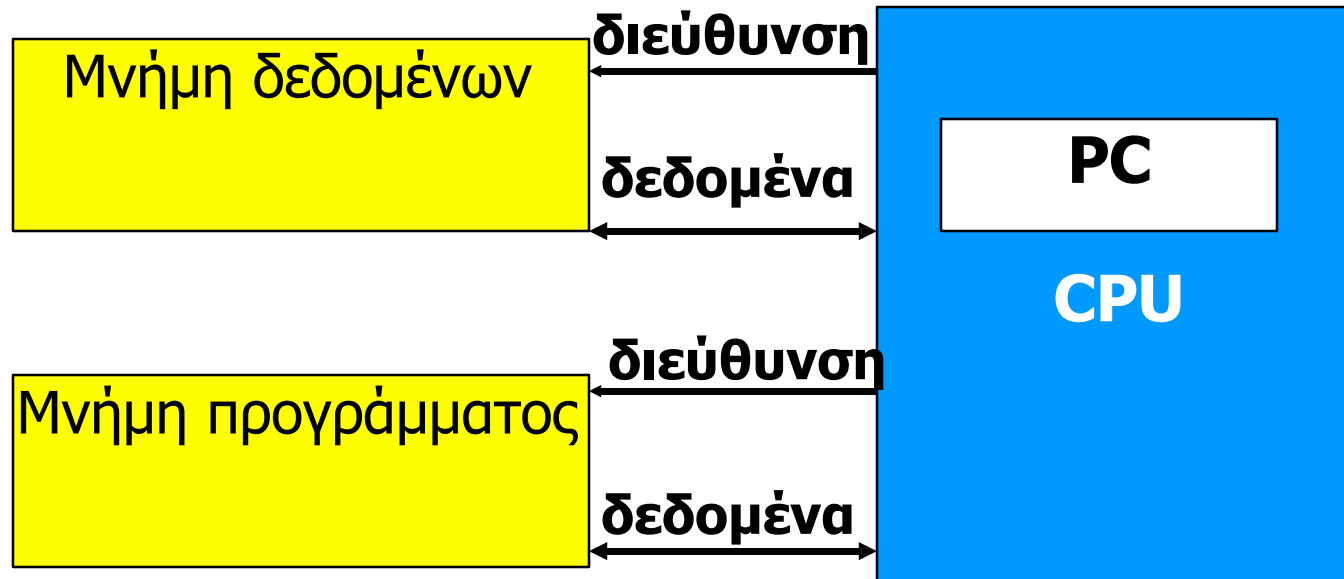
von Neumann: CPU + Μνήμη

14



Αρχιτεκτονική Harvard

15



von Neumann vs. Harvard

16

- Harvard can't use self-modifying code.
- Harvard allows two simultaneous memory fetches.
- Most DSPs use Harvard architecture for streaming data:
 - greater memory bandwidth;
 - more predictable bandwidth.

Δύο διαφορετικές Αρχιτεκτονικές

17

- CISC (complex instruction set computer)
Συμπεριλαμβάνονται σύνθετες εντολές
Εύκολος προγραμματισμός
Επιπλέον φόρτος
Intel-Pentium
- RISC (reduced instruction set computer)
Οι σύνθετες εντολές προσομοιώνονται με τη χρήση υποσυνόλου απλών εντολών
Δύσκολος και χρονοβόρος προγραμματισμός
Apple-PowerPC

- CISC (προφέρεται sisk) αντιστοιχεί σε Complex Instruction Set Computer (CISC). Η στρατηγική πίσω από αρχιτεκτονικές CISC είναι να έχουμε ένα μεγάλο σύνολο εντολών, συμπεριλαμβανομένων και συνθέτων εντολών. Προγραμματισμός σε CISC υπολογιστές είναι ευκολότερος από ό, τι σε άλλους σχεδιασμούς, διότι υπάρχει μια εντολή τόσο για τα απλά και και σύνθετα έργα. Ως εκ τούτου οι προγραμματιστές δεν χρειάζονται να γράψουν μια σύνολο εντολών για την εκτέλεση ενός πολύπλοκου έργου.

RISC

5.19

- RISC (προφέρεται risk) αντιστοιχεί Reduced Instruction Set Computer. Η στρατηγική πίσω από την αρχιτεκτονική RISC είναι να έχουμε ένα μικρό σύνολο από εντολές που εκτελούν έναν ελάχιστο αριθμό από απλές εργασίες. Πολύπλοκες εντολές προσομοιώνονται με ένα υποσύνολο των απλές εντολές. Ο προγραμματισμός σε RISC είναι πιο δύσκολη και χρονοβόρα διαδικασία από ό, τι σε άλλο σχεδιασμό, γιατί οι περισσότερες από τις πολύπλοκες εντολές θα πρέπει να προσομοιωθούν χρησιμοποιώντας απλές εντολές.

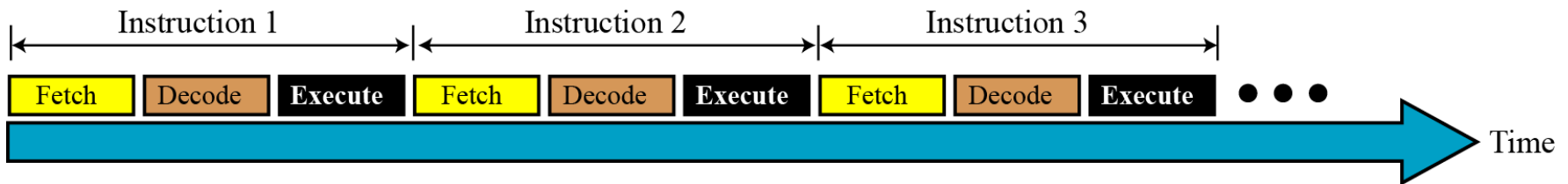
Pipelining (Διοχεύτηση) [1/2]

5.20

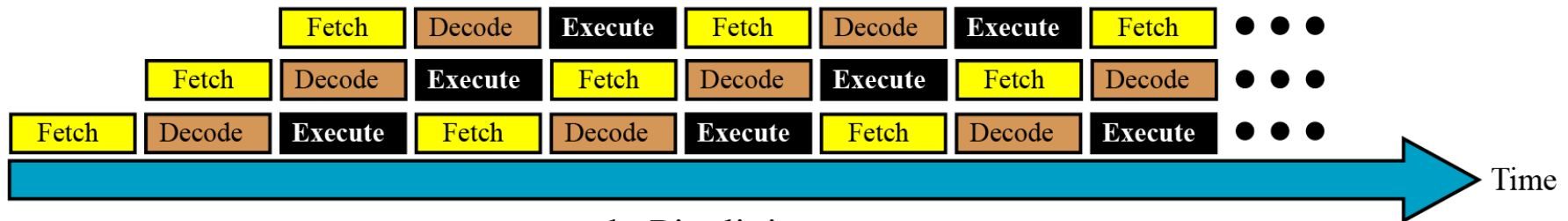
- Έχουμε μάθει ότι ένας υπολογιστής χρησιμοποιεί τρεις φάσεις: (i) την κλήση, (ii) αποκωδικοποίηση και (iii) εκτέλεση, για κάθε εντολή. Στους πρώτους υπολογιστές, οι τρεις αυτές φάσεις έπρεπε να γίνει σε σειρά για κάθε εντολή. Με άλλα λόγια, μια εντολή, n , χρειάζεται να ολοκληρώσει όλες τις φάσεις πριν ξεκινήσει η εντολή $n+1$. Οι σύγχρονοι υπολογιστές χρησιμοποιούν μια τεχνική που ονομάζεται **διοχεύση (pipelining)** για τη βελτίωση της απόδοσης (ο συνολικός αριθμός των εντολών που εκτελούνται σε κάθε περίοδο του χρόνου). Η ιδέα είναι ότι εάν η μονάδα ελέγχου μπορεί να κάνει δύο ή τρεις από αυτές τις φάσεις ταυτόχρονα, η επόμενη εντολή μπορεί να αρχίσει πριν από το προηγούμενο, έχει τελειώσει.

Pipelining (Διοχεύτωση) [2/2]

5.21



a. No pipelining

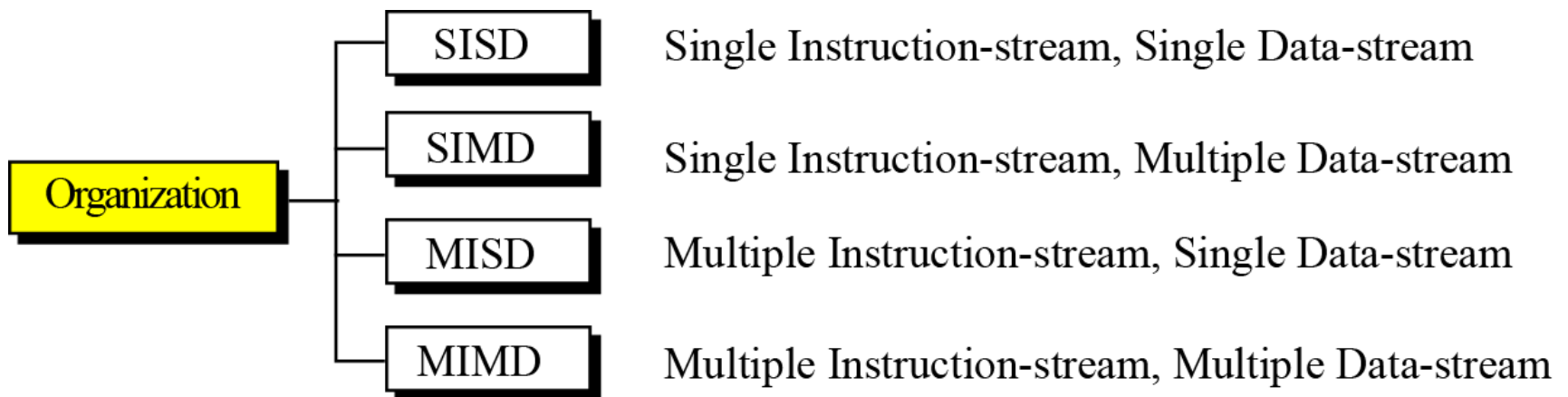


b. Pipelining

Παράλληλη επεξεργασία (Parallel processing)

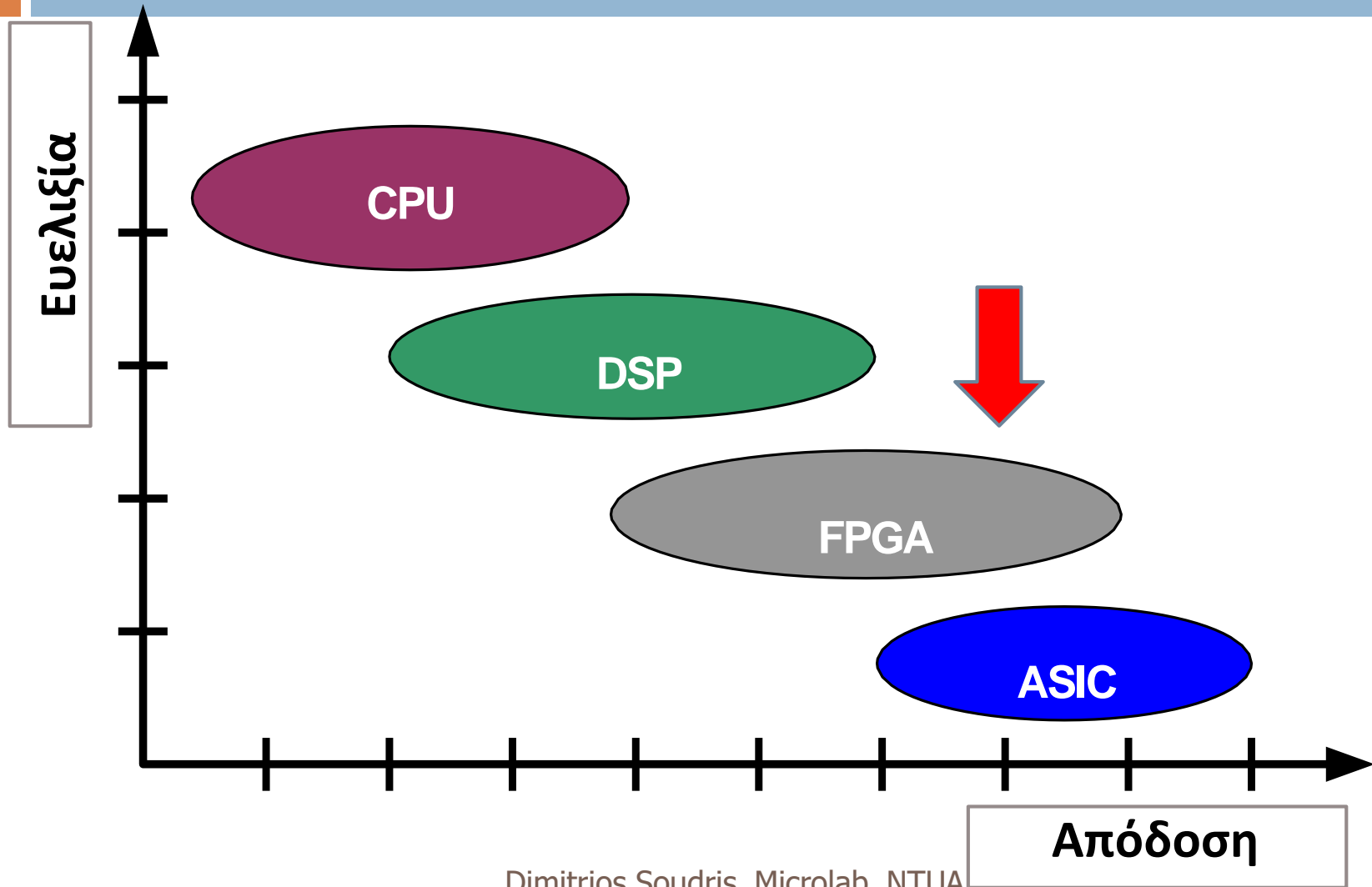
5.22

- Παραδοσιακά, ένας υπολογιστής είχε μια ενιαία μονάδα ελέγχου, μία αριθμητική λογική μονάδα και μία μονάδα μνήμης. Με την εξέλιξη της τεχνολογίας και τη μείωση του κόστους του υλικού του υπολογιστή, σήμερα μπορούμε να έχουμε έναν μόνο υπολογιστή με πολλαπλές μονάδες ελέγχου, πολλαπλές μονάδες αριθμητικής λογικής και πολλαπλές μονάδες μνήμης. Η ιδέα αυτή αναφέρεται ως παράλληλη επεξεργασία. Όπως διοχέτευση, έτσι και η παράλληλη επεξεργασία μπορεί να βελτιώσει απόδοση.



Υπολογιστικές Πλατφόρμες

27



FPGA:

28



The chip that flip-flops

Dimitrios Soudris, Microlab, NTUA

Γιατί FPGAs ?

29

- **Ιδανικά για εξεικεδευμένους σχεδιασμούς**
Διαφοροποίηση προϊόντων σε μια αγορά που αλλάζει γρήγορα
- **Προσφέρει πλεονεκτήματα για υψηλή ολοκλήρωση**
Υψηλή πολυπλοκότητα, πυκνότητα, αξιοπιστία
Χαμηλό κόστος, κατανάλωση ισχύος, μικρό φυσικό μέγεθος
- **Αποφεύγονται τα προβλήματα των ASICs**
Υψηλά μη-επαναλαμβανόμενα κόστη, μεγάλη καθυστέρηση στο σχεδιασμό και δοκιμή για ηλεκτρικά χαρακτηριστικά
- **Γρήγορο Time-to-Market,**
- **Γρήγορη απόκτηση στις αλλαγές της αγοράς**

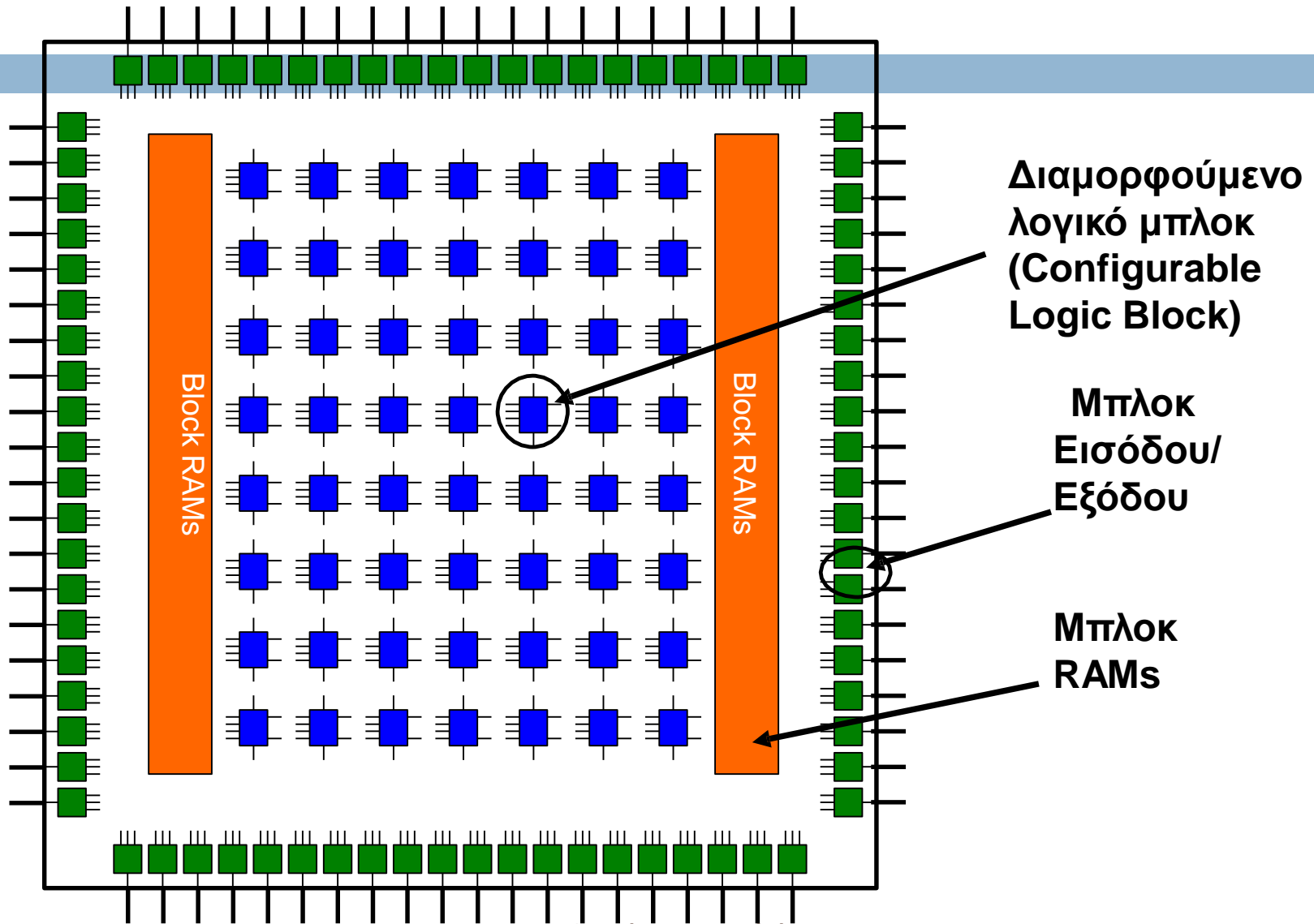
Πλεονεκτήματα **FPGAs**

30

- **Πολυ γρήγορη εξειδικευμένη λογική**
Μαζική παράλληλη λειτουργία
- **Γρηγότερα από μικροελεγκτές και μικροϋπολογιστές**
Γρηγορότερα από μηχανές DSP
- **Πιο ευέλικτα από εξειδικευμένα chipsets**
Επιτρέπει από την συνεχή διαφοροποίηση ενός προϊόντος
- **Πιο οικονομικά και λιγότερο επιρρεπή σε κινδύνους από τα ASICs**
Μη-επαναλαμβανόμενα κόστη, μικρό μέγεθος παραγγελίας
- **Επαναπρογραμματιζόμενο οποιαδήποτε στιγμή**

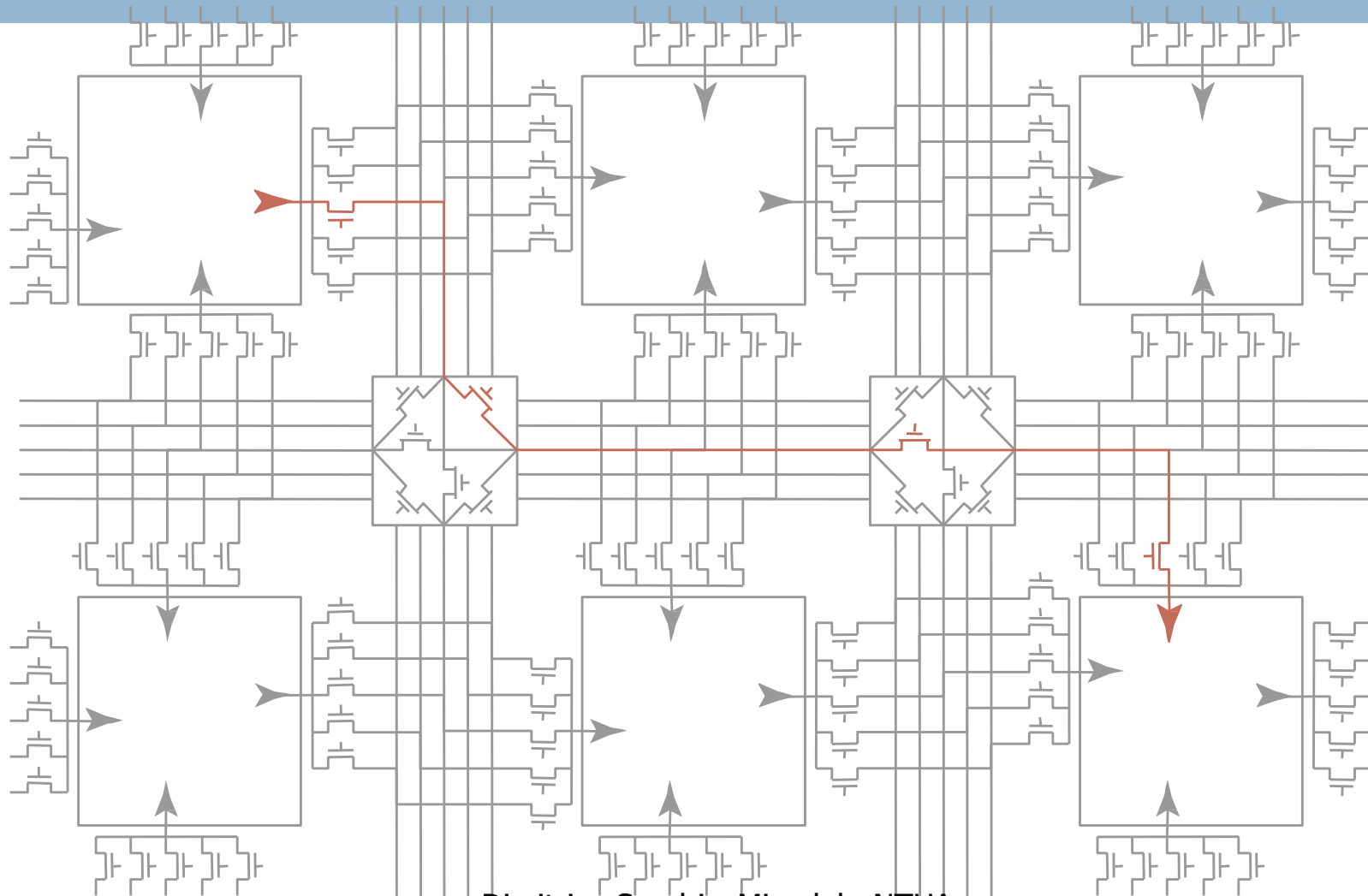
Τι είναι ένα FPGA?

31



Υλοποίηση ενός βρόχου με τρανζίστορ

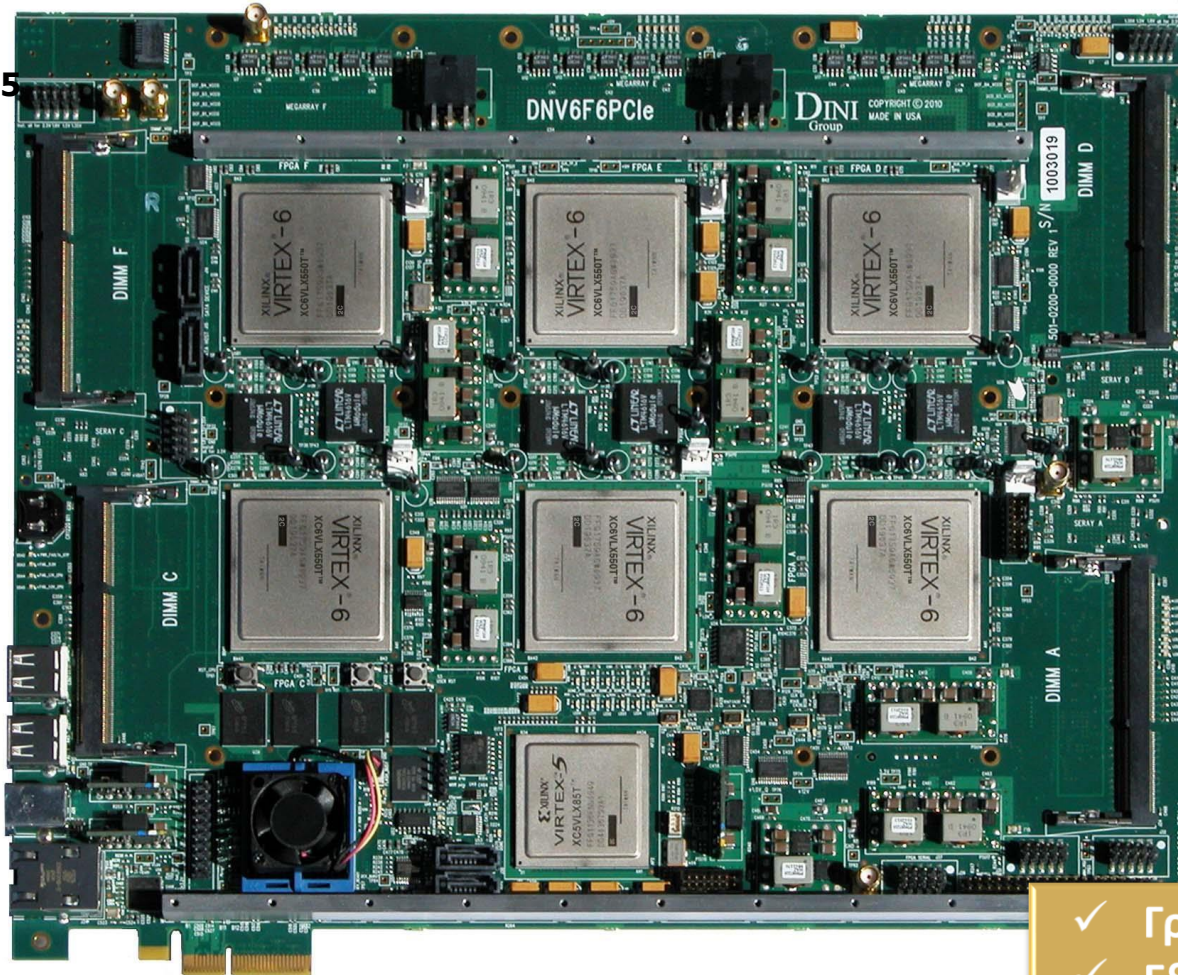
33



Dimitrios Soudris, Microlab, NTUA
Courtesy Dehon and Wawrzyniek

FPGAs και εφαρμογές

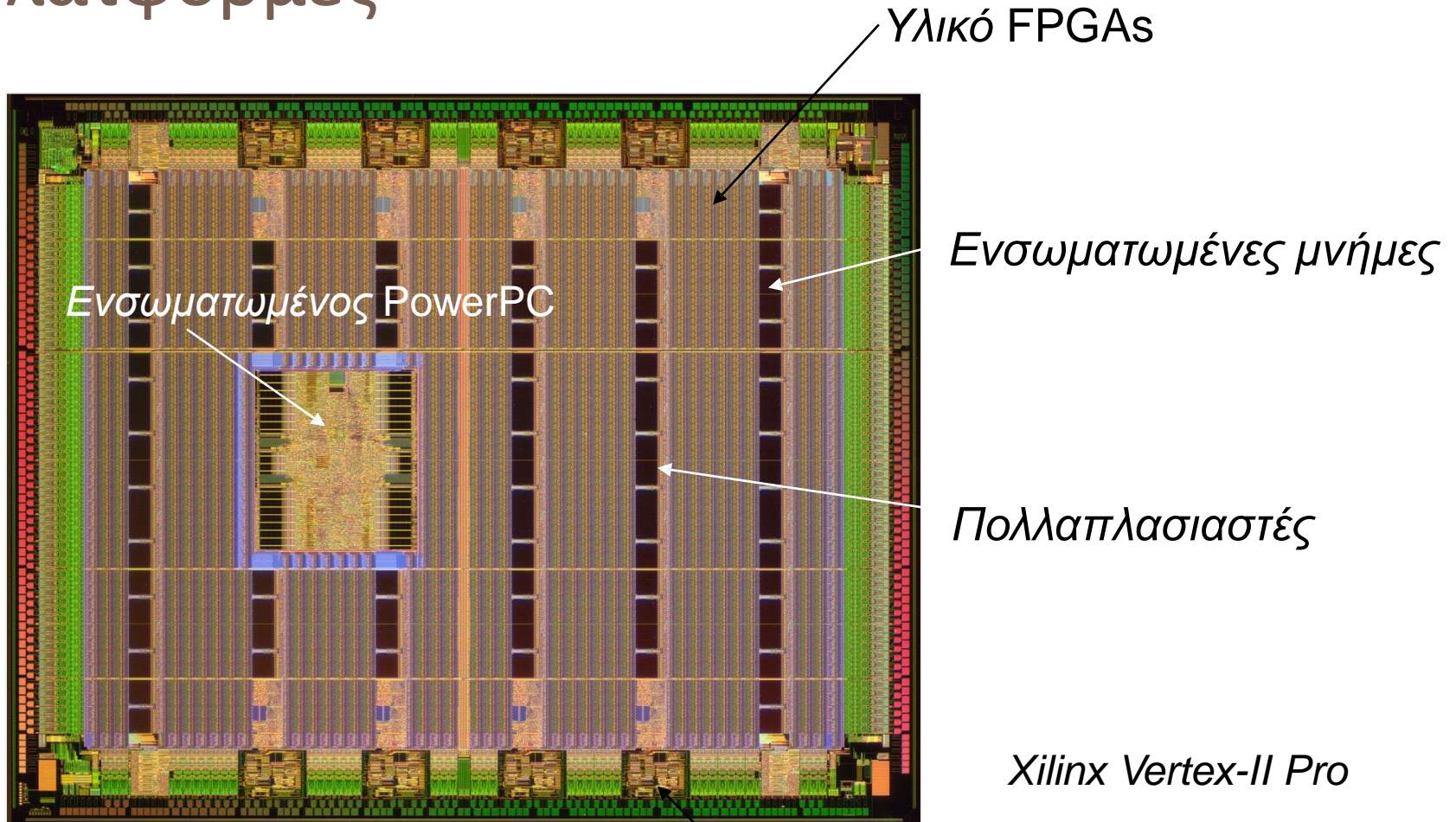
35



- ✓ Γρήγορη Πρωτοτυποποίηση
- ✓ Εξομίωση
- ✓ Προσομίωση
- ✓ Επιστημονικός Υπολογισμός
- ✓ Υπολογισμός Νέφους

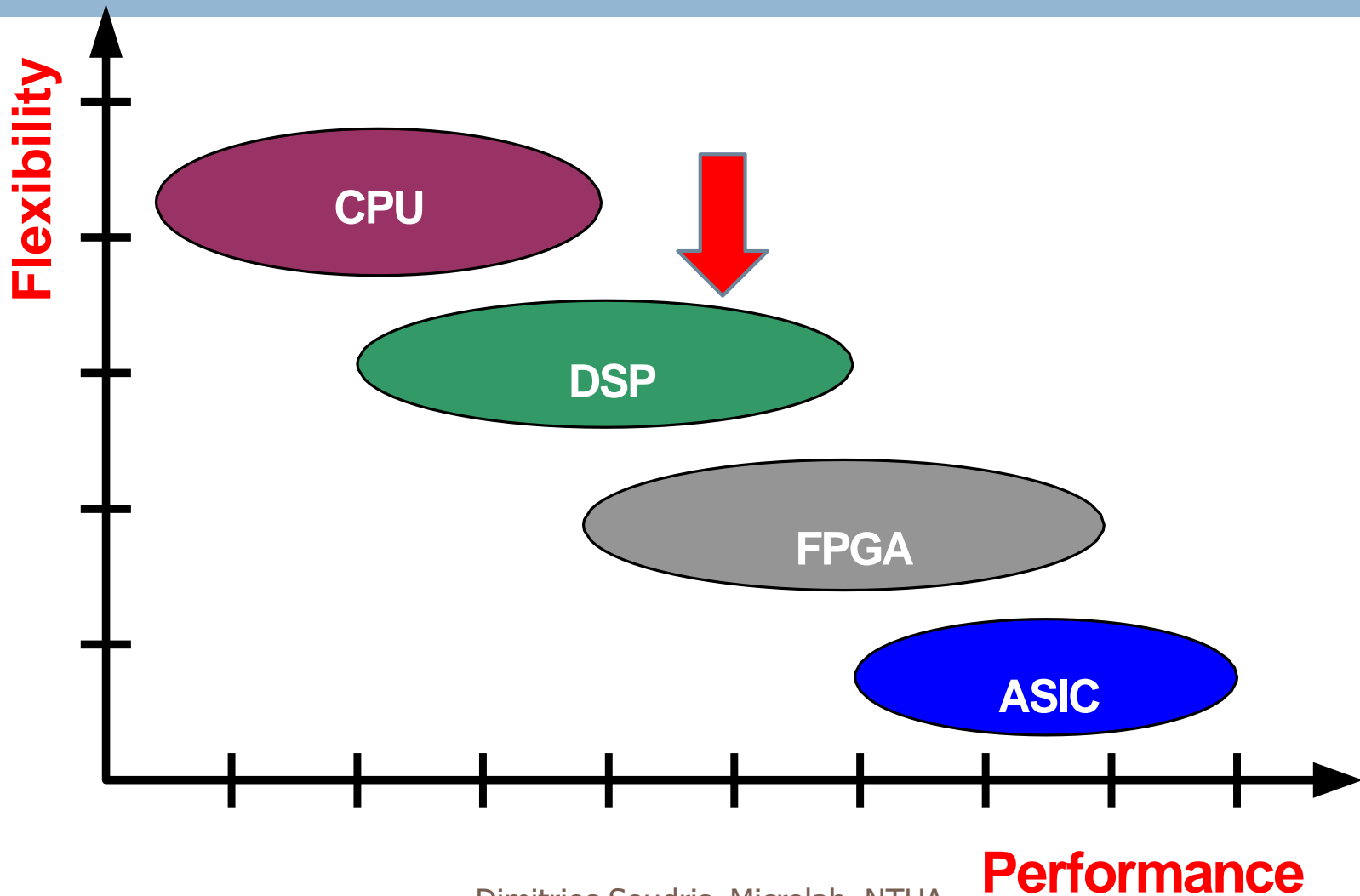
Dimitrios Soudris, Microlab, NTUA

Ετερογενείς Προγραμματιζόμενες Πλατφόρμες



Υπολογιστικές Πλατφόρμες

71



Χαρακτηριστικά εφαρμογών DSPs

73

- Απαιτούν μεγάλη υπολογιστική πολυπλοκότητα
- Απαιτεί προσοχή στην αριθμητική πιστότητα
- Απαιτήσεις στη μνήμη για υψηλό εύρος ζώνης
- Συνεχής ροή δεδομένων (Streaming Data)
- Προβλέψιμα μοτίβα πρόσβασης δεδομένων (access pattern)
- Τοπικότητα στην Εκτέλεση
- Δύσκολες απαιτητικές Μαθηματικές πράξεις
- Απαίτηση για εκτέλεση σε πραγματικό χρόνο (real-time processing)

DSP Algorithms Shape DSPs

74

Computational demands	→	Multiple parallel execution units, hardware acceleration of common DSP functions
Numeric fidelity	→	Accumulator registers, guard bits, saturation hardware
High memory bandwidth	→	Harvard architecture, support for parallel moves
Predictable data access patterns	→	Specialized addressing modes, e.g., modulo, bit-reversed

DSP Algorithms Shape DSPs

75

Execution-time locality	→	Hardware looping, streamlined interrupt handling
Math-centricity	→	Single-cycle multiplier(s) or MAC unit(s), MAC instruction
Streaming data	→	Data memory usually SRAM, not cache; DMA
Real-time constraints	→	Few dynamic features, on-chip SRAM instead of cache
Standards	→	16-bit data types; rounding, saturation modes

Χρήσιμες διεθύνσεις

70

- **Authors@Google: Christos Papadimitriou**
http://www.youtube.com/watch?v=yF-cq06C9Ck&feature=Playlist&p=42B3E03AAE2B3B02&playnext_from=PL&playnext=1&index=32
- **The Birth of the Transistor - 4 of 4**
<http://www.youtube.com/watch?v=ptDYeDXXGOY>
- **MICROPROCESSOR TUTORIAL**
<http://www.eastaughs.fsnet.co.uk/cpu/index.htm>