Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Εισαγωγή στην Αρχιτεκτονική Η/Υ

(θεμελιώδεις αρχές λειτουργίας των υπολογιστών)

http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Τι είναι ένα "υπολογιστικό σύστημα";

- Υπολογιστικά Συστήματα (;)

Ενσωματωμένα συστήματα

- Υπολογιστικά Συστήματα

;

Ποια είναι η αρχιτεκτονική του μικροεπεξεργαστή που παράγεται σε μεγαλύτερο αριθμό ανά έτος;

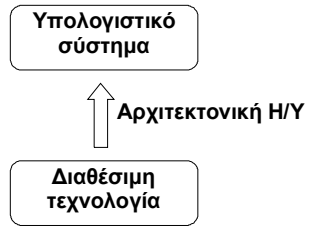
- Υπολογιστικά συστήματα ελέγχου
  - Συστήματα ειδικών απαιτήσεων
    - Ιατρικά, συγκοινωνίες, εργοστάσια, συλλογή πληροφοριών..
  - Καταναλωτικά προϊόντα
- Μια ανερχόμενη αγορά
  - 95% των μικροεπεξεργαστών που πωλούνται ανά έτος καταλήγει σε ένα ενσωματωμένο σύστημα!
- Συστήματα μέσα σε ένα ολοκληρωμένο κύκλωμα
  - Processor cores

Αρχιτεκτονική Η/Υ

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

;

Ανήκει το **λογισμικό** στο διπλανό σχήμα;



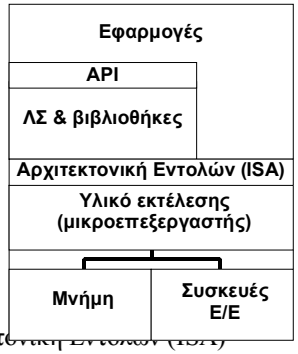
- Η αρχιτεκτονική συγκροτεί υπολογιστικά συστήματα χρησιμοποιώντας την υπάρχουσα τεχνολογία
  - και προδιαγράφει τη μελλοντική

Ο υπολογιστής ως ιεραρχία επιπέδων

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

;

Τι ακριβώς είναι μια **διεπαφή** (interface); Γιατί είναι τόσο σημαντική έννοια;



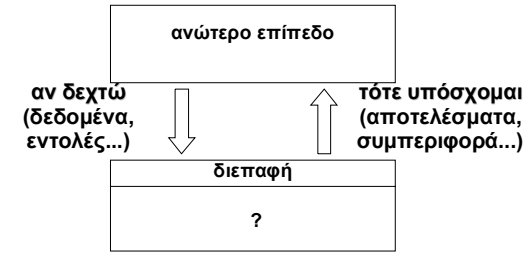
- Αρχιτεκτονική Εντολών (ISA)
  - Η διεπαφή υλικού-λογισμικού

Η σπουδαιότητα των διεπαφών

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

;

Αντιστοιχία με γλώσσες προγραμματισμού;



- Αντιμετώπιση πολυπλοκότητας σχεδιασμού
- Επαναχρησιμοποίηση τμημάτων
- Αξιόπιστος σχεδιασμός

Αρχιτεκτονική: υλικό μόνο ή και λογισμικό;

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

;

Σε ποια συστήματα έχει ιδιαίτερη σημασία η αρχιτεκτονική του λογισμικού;

- Το λογισμικό είναι σημαντικό μέρος ενός υπολογιστικού συστήματος
  - Ορίζει τον τρόπο χρήσης του υλικού
  - Συνεπώς (συν-)διαμορφώνει
    - την απόδοση
    - την κατανάλωση ενέργειας
    - την αξιοπιστία
- "Η αρχιτεκτονική ασχολείται με το υλικό"
  - Η παραδοσιακή αντίληψη
- "Πρέπει να λαμβάνεται υπ'όψη και το λογισμικό"
  - Η σύγχρονη αντίληψη

Αρχιτεκτονική: ο τελικός στόχος

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

;

Ποια η διαφορά μεταξύ **γρήγορης εκτέλεσης** και **αξιόπιστης εκτέλεσης**;

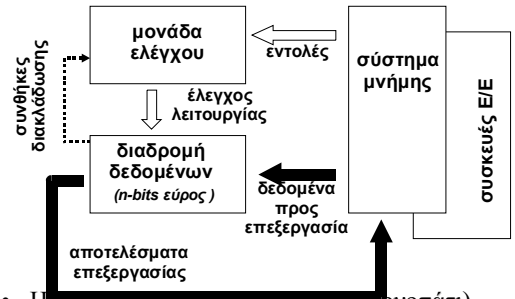
- Η αξιοποίηση με τον καλύτερο δυνατό τρόπο του υλικού (και λογισμικού) για την ανάπτυξη
  - Αποδοτικών συστημάτων
    - Γρήγορη εκτέλεση προγραμμάτων
    - Υπερυπολογιστές, προσωπικοί υπολογιστές
  - Αξιόπιστων συστημάτων
    - Ασφαλής εκτέλεση προγραμμάτων
    - Συστήματα πραγματικού χρόνου
  - Προσιτών συστημάτων
    - Ικανοποιητική εκτέλεση με μικρό κόστος-ενέργεια
    - Καταναλωτικές συσκευές

Οι βασικές μονάδες κάθε υπολογιστή

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες

!

Η διπλανή εικόνα είναι απλοποιημένη!



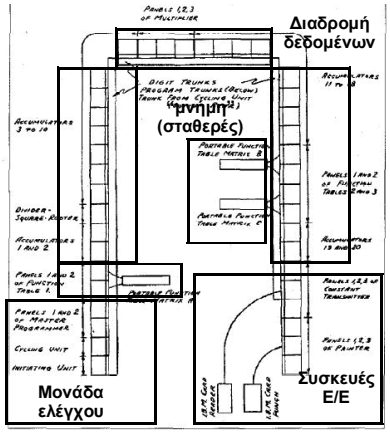
- Η μονάδα ελέγχου, η διαδρομή (μονοπάτι) δεδομένων και μέρος του συστήματος μνήμης βρίσκονται σήμερα μέσα στον μικροεπεξεργαστή (ΚΜΕ)

The Electronic Numerical Integrator and Computer (ENIAC) 1946

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες

i

Ο ENIAC, ο πρώτος ηλεκτρονικός υπολογιστής, διέθετε 18.000 λυχνίες για τα λογικά του κυκλώματα. Δεν υπήρχε μνήμη προγράμματος!

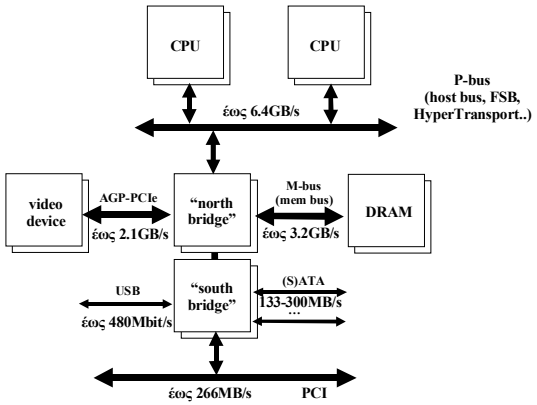


Ένα τυπικό υπολογιστικό σύστημα σήμερα

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες

!

Ο ρυθμός μεταφοράς δεδομένων ανάμεσα στα διάφορα μέρη του υπολογιστή επηρεάζει καθοριστικά τη συνολική απόδοση του συστήματος!



Ηλεκτρονικά κυκλώματα

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

- Σε κάθε υπολογιστή απαιτούνται
  - Λογικά κυκλώματα
    - Για την εκτέλεση πράξεων
    - Για τη σύγκριση και λήψη αποφάσεων
  - Κυκλώματα μνήμης
    - Για την αποθήκευση δεδομένων
    - Για την αποθήκευση εντολών
  - Κυκλώματα διασύνδεσης
    - Για τη μεταφορά δεδομένων μεταξύ τμημάτων

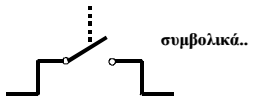
Πριν τα σύγχρονα ηλεκτρονικά

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

- Παλαιότερα χρησιμοποιήθηκαν...
  - Λογικά κυκλώματα
    - Μηχανικά συστήματα (γρανάζια)
    - Ηλεκτρομηχανικά (ρελέ)
    - Λυχνίες κενού
  - Κυκλώματα μνήμης
    - Τα ίδια, αλλά και...
    - Γραμμές υδραργύρου (!) – καθοδική οθόνη (!)
  - Κυκλώματα διασύνδεσης
    - Καλώδια!

Το τρανζίστορ MOS(FET)

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα



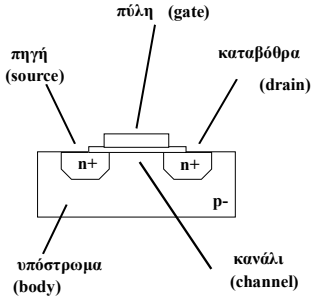
- Ο μικροσκοπικός διακόπτης των σύγχρονων κυκλωμάτων
  - Η θεωρία λειτουργίας του είναι γνωστή από το 1925...
  - ...αλλά τα πρώτα λειτουργικά τρανζίστορ MOS κατασκευάστηκαν στη δεκαετία του 60

Το τρανζίστορ MOS(FET)

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

i

Το τρανζίστορ αυτό είναι τύπου NMOS. Υπάρχει και το συμπληρωματικό PMOS.



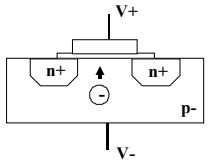
ιδανικό σχήμα..

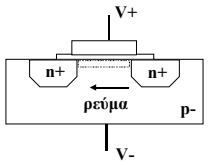
Λειτουργία του τρανζίστορ MOS(FET)

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

;

Τι συμβαίνει στο τρανζίστορ PMOS;





Κατασκευή ολοκληρωμένων κυκλωμάτων

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

i

Λόγω της απαιτούμενης ακρίβειας, μια γραμμή παραγωγής κοστίζει δισ. $

- Γραμμές παραγωγής
  - Φωτολιθογραφία με μάσκες
  - Διεργασίες στα εκτεθειμένα μέρη
    - Οξείδωση, απόξεση, απόθεση μετάλλου, εμφύτευση ιόντων...
  - Επανάληψη βημάτων
- Ένα σύγχρονο ολοκληρωμένο κύκλωμα μικροεπεξεργαστή
  - Έχει επιφάνεια περίπου 280mm^2
  - Και περιέχει από 100 εκ. έως >1 δισ. τρανζίστορ!

Η συρρίκνωση του τρανζίστορ

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

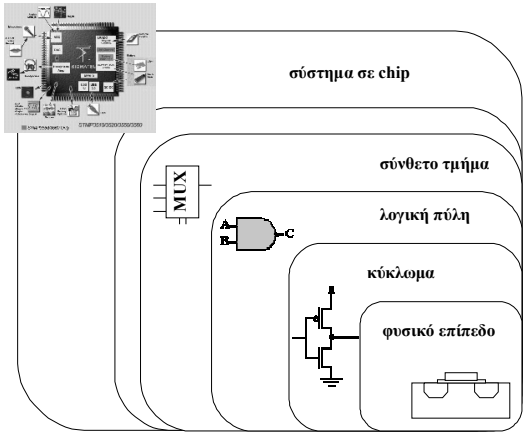
i

Νόμος του Moore: ο αριθμός των τρανζίστορ ανά ολοκληρωμένο διπλασιάζεται κάθε 1,5-2 χρόνια

- Πλεονεκτήματα
  - Ταχύτερη λειτουργία
    - Πιο γρήγοροι χρόνοι ON-OFF
  - Μικρότερη κατανάλωση ενέργειας
    - Για τον ίδιο αριθμό τρανζίστορ!
  - Μεγαλύτερη ολοκλήρωση
    - Μείωση κόστους παραγωγής και αύξηση λειτουργικότητας
- Τρέχουσα εμπορική τεχνολογία:
  - Μέγεθος (καναλιού) τρανζίστορ = 65nm
- Το μέλλον; 45 και 25 nm

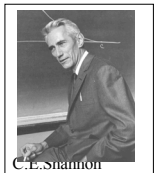
Ψηφιακά Ηλεκτρονικά: Ιεραρχία σχεδίασης

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα



Ψηφιακά Ηλεκτρονικά και Δυαδική λογική

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα



C.E.Shannon

- Η δυαδική λογική ταιριάζει με την τεχνολογία του τρανζίστορ
  - 2 καταστάσεις: ON-OFF, 1-0
  - Ψηφιακά ηλεκτρονικά (2 στάθμες)
- Δυαδική άλγεβρα Boole
  - Λογική άλγεβρα
  - Συσχέτιση με διακοπτικά κυκλώματα
    - Η εργασία του Shannon (1938)

Άλγεβρα Boole: επανάληψη

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

- A + B (A OR B)
- A • B (ή απλά AB, A AND B)
- $\overline{A}$ (NOT A)
  - A + 0 = A και A • 1 = A
  - A + 1 = 1 και A • 0 = 0
  - A + $\overline{A}$ = 1 και A • $\overline{A}$ = 0
  - A + B = B + A και A • B = B • A
  - A+(B+C)=(A+B)+C και
  - A(BC)=(AB)C

Άλγεβρα Boole: επανάληψη

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

- A(B+C) = (AB)+(AC) και
- A+(BC) = (A+B)(A+C)

- $\overline{(A+B)} = \overline{A} \cdot \overline{B}$ και
- $\overline{(A \cdot B)} = \overline{A} + \overline{B}$ (DeMorgan)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Ψηφιακή Λογική και Σχεδίαση

(σχεδίαση συνδυαστικών κυκλωμάτων)

http://www.ionio.gr/~mistral/tp/comparch/

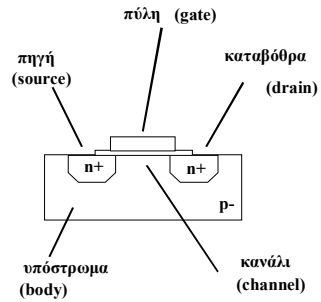
Μ.Στεφανιδάκης

Το τρανζίστορ MOS(FET)

• Ηλεκτρονικά κυκλώματα

i

Το τρανζίστορ αυτό είναι τύπου NMOS. Υπάρχει και το συμπληρωματικό PMOS.

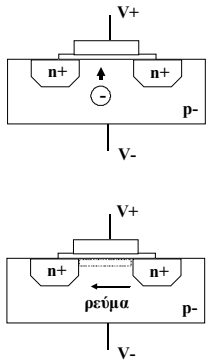


Λειτουργία του τρανζίστορ MOS(FET)

• Ηλεκτρονικά κυκλώματα

;

Τι συμβαίνει στο τρανζίστορ PMOS;



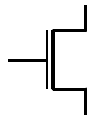
Τρανζίστορ NMOS και PMOS

• Ηλεκτρονικά κυκλώματα

i

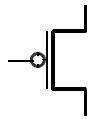
Η πλειοψηφία των σημερινών κυκλωμάτων χρησιμοποιεί και τα δύο είδη τρανζίστορ (τεχνολογία CMOS)

NMOS: άγει όταν στην πύλη εφαρμόζεται '1'
Περνά ισχυρό '0'



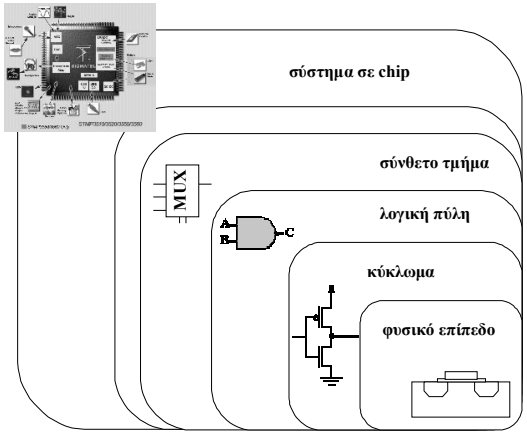
σύμβολα

PMOS: άγει όταν στην πύλη εφαρμόζεται '0'
Περνά ισχυρό '1'



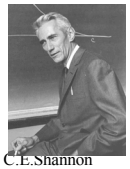
Ψηφιακά Ηλεκτρονικά: Ιεραρχία σχεδίασης

- Ηλεκτρονικά κυκλώματα



Ψηφιακά Ηλεκτρονικά και Δυαδική λογική

- Ηλεκτρονικά κυκλώματα


C.E.Shannon

- Η δυαδική λογική ταιριάζει με την τεχνολογία του τρανζίστορ
  - 2 καταστάσεις: ON-OFF, 1-0
  - Ψηφιακά ηλεκτρονικά (2 στάθμες)
- Δυαδική άλγεβρα Boole
  - Λογική άλγεβρα
  - Συσχέτιση με διακοπτικά κυκλώματα
    - Η εργασία του Shannon (1938)

Άλγεβρα Boole: επανάληψη

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

- $A + B$ (A OR B)
- $A \bullet B$ (ή απλά AB, A AND B)
- $\overline{A}$ (NOT A)
  - $A + 0 = A$ και $A \bullet 1 = A$
  - $A + 1 = 1$ και $A \bullet 0 = 0$
  - $A + \overline{A} = 1$ και $A \bullet \overline{A} = 0$
  - $A + B = B + A$ και $A \bullet B = B \bullet A$
  - $A+(B+C)=(A+B)+C$ και
  - $A(BC)=(AB)C$

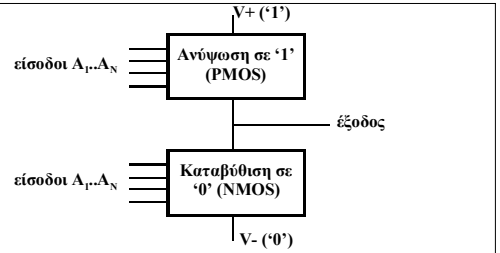
Άλγεβρα Boole: επανάληψη

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

- $A(B+C) = (AB)+(AC)$ και
- $A+(BC) = (A+B)(A+C)$
- $\overline{(A+B)} = \overline{A} \bullet \overline{B}$ και
- $\overline{(A \bullet B)} = \overline{A} + \overline{B}$ (DeMorgan)

Απλές λογικές πύλες

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες

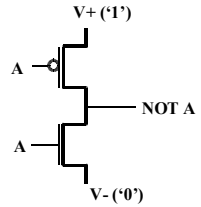


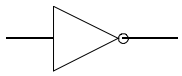
- Στατική τεχνολογία CMOS
  - Αναστρέφουσες συναρτήσεις
    - NOT, NAND, NOR …
  - Αξιόπιστη λειτουργία, εύκολη σχεδίαση
  - Όχι πάντα η αποδοτικότερη λύση

Η πύλη NOT

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες

A	Y
0	1
1	0

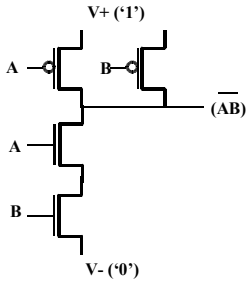


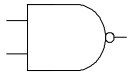


Η πύλη NAND

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

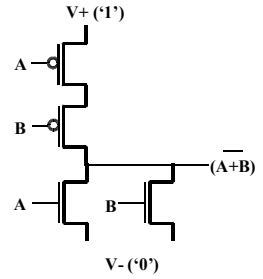


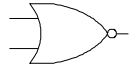


Η πύλη NOR

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες

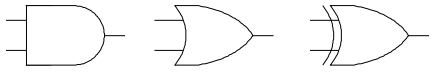
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0





Άλλες λογικές πύλες

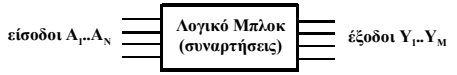
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες



- Ως συνδυασμός των βασικών πυλών
  - NOT, NAND, NOR
- Υπάρχουν και εναλλακτικές μέθοδοι σχεδίασης!

Συνδυαστική Λογική

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική



- Μπλοκ λογικών συναρτήσεων
  - Οι έξοδοι εξαρτώνται αποκλειστικά από την τρέχουσα τιμή των εισόδων
  - Δεν υπάρχει μνήμη προηγούμενων καταστάσεων
  - Αλλαγή των εισόδων θα επηρεάσει τις εξόδους μετά από χρονικό διάστημα (καθυστέρηση διάδοσης)
  - Η συνάρτηση που υλοποιεί το μπλοκ μπορεί να εκφραστεί με έναν πίνακα αλήθειας

Υλοποίηση συναρτήσεων

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική

!

Πάντοτε προσπαθούμε να απλοποιήσουμε τις συναρτήσεις (πίνακες Karnaugh ή άλλες υπολογιστικές μέθοδοι)

A	B	C	Y	Ελαχιστόροι	Μεγιστόροι
0	0	0	1	a'b'c'	a+b+c
0	0	1	0	a'b'c	a+b+c'
0	1	0	0	a'bc'	a+b'+c
0	1	1	1	a'bc	a+b'+c'
1	0	0	0	ab'c'	a'+b+c
1	0	1	0	ab'c	a'+b+c'
1	1	0	1	abc'	a'+b'+c
1	1	1	1	abc	a'+b'+c'

- Y = a'b'c' + a'bc + abc' + abc
- Y = (a+b+c')(a+b'+c)(a'+b+c)(a'+b+c')

Βασικά συνδυαστικά τμήματα

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

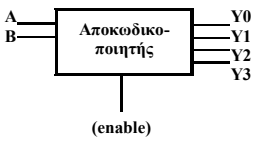
- Αποκωδικοποιητής (decoder)
  - N είσοδοι ενεργοποιούν 1 από 2^N εξόδους
- Πολυπλέκτης (multiplexer)
  - N είσοδοι επιλέγουν 1 από 2^N εισόδους
- Αθροιστής (adder)
  - Πρόσθεση δυαδικών ψηφίων
  - Διάφοροι τρόποι παραγωγής-διάδοσης κρατουμένου

Αποκωδικοποιητής (decoder)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Πώς υλοποιείται ένας αποκωδικο-ποιητής;



A	B	Y_0	Y_1	Y_2	Y_0
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

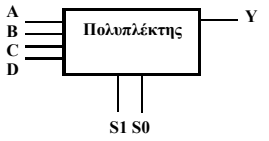
- Αποκωδικοποιητής N-σε-2^N
  - Ν είσοδοι ενεργοποιούν μία από 2^N εξόδους

Πολυπλέκτης (multiplexer)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Πώς υλοποιείται ένας πολυπλέκτης;



S_1	S_0	Y
0	0	A
0	1	B
1	0	C
1	1	D

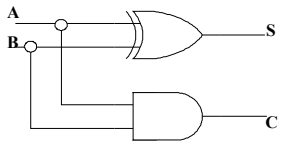
- Πολυπλέκτης 2^N γραμμών σε 1
  - Επιλογή μίας από 2^N εισόδους με τη βοήθεια Ν σημάτων ελέγχου

Ημιαθροιστής (half-adder)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Αν απαιτείται πρόσθεση αριθμών με περισσότερα bits;



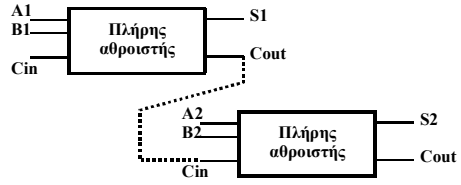
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Πλήρης αθροιστής (full-adder)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Πώς υλοποιείται ένας πλήρης αθροιστής;



- Πολλαπλά τμήματα πλήρη αθροιστή
  - Όμως: πόσο γρήγορα διαδίδεται το κρατούμενο; (ripple carry)
  - Τεχνικές πρόβλεψης κρατουμένου (carry look-ahead)

Πέρα από τη συνδυαστική λογική...

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

- Στο επόμενο μάθημα...
  - Πώς εισάγω την έννοια της κατάστασης ενός λογικού τμήματος;
  - Που φυλάσσεται η κατάσταση;
  - Πότε ενημερώνεται;
  - Πώς συγχρονίζονται τα διάφορα τμήματα λογικής;

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Ψηφιακή Λογική και Σχεδίαση

(στοιχεία μνήμης και μέθοδοι χρονισμού)

http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Πέρα από τη συνδυαστική λογική

- Ακολουθιακή λογική

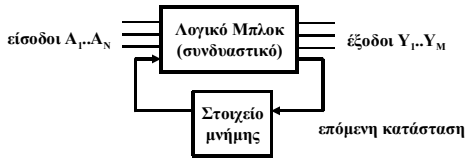
- Βασικά ερωτήματα
  - Πώς εισάγω την έννοια της κατάστασης ενός λογικού τμήματος;
  - Που φυλάσσεται η κατάσταση;
  - Πότε ενημερώνεται;
  - Πώς συγχρονίζονται τα διάφορα τμήματα λογικής;

Ακολουθιακή λογική

- Ακολουθιακή λογική

;

Πότε τελειώνει η τρέχουσα και πότε αρχίζει η επόμενη κατάσταση;



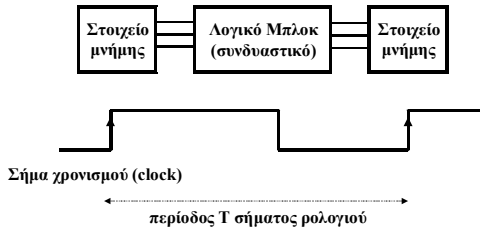
- Η έννοια της κατάστασης: οι έξοδοι εξαρτώνται από τις εισόδους ΚΑΙ από την προηγούμενη κατάσταση

Χρονισμός σε ακολουθιακή λογική

- Ακολουθιακή λογική

!

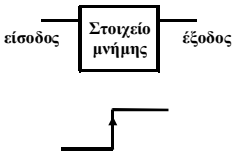
Το clock καθορίζει τη **χρονική στιγμή αποθήκευσης** στα στοιχεία μνήμης



- Η πλειοψηφία των ψηφιακών κυκλωμάτων χρησιμοποιεί ένα περιοδικό σήμα συγχρονισμού (ρολόι - clock)

Χρονισμός στις ακμές του ρολογιού

- Ακολουθιακή λογική



Σήμα χρονισμού (clock)

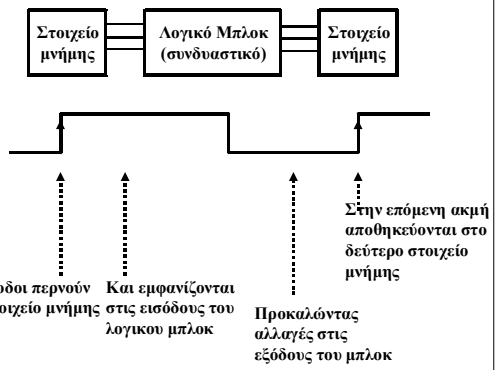
- Η απλούστερη λύση: αποθήκευση στην ανερχόμενη (ή κατερχόμενη) ακμή του ρολογιού
  - Τη στιγμή αυτή η είσοδος αποθηκεύεται στο στοιχείο μνήμης
  - Και εμφανίζεται στην έξοδο μετά από την καθυστέρηση διάδοσης του στοιχείου μνήμης.

!

Η είσοδος πρέπει να είναι σταθερή λίγο **πριν** (setup) και λίγο **μετά** (hold) την ακμή του ρολογιού

Χρονισμός και διάδοση σημάτων

- Ακολουθιακή λογική

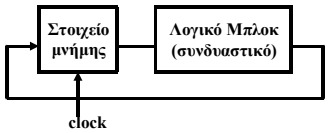


;

Μπορεί η περίοδος του ρολογιού να γίνει όσο μικρή θέλουμε;

Ανάδραση και στοιχεία μνήμης

- Ακολουθιακή λογική



- Απολύτως επιτρεπτή συνδεσμολογία!
- Λόγω του σήματος ρολογιού δεν δημιουργείται πρόβλημα με την ανατροφοδότηση του σήματος εξόδου στην είσοδο

Συγχρονισμός τμημάτων

- Ακολουθιακή λογική

- Σύγχρονο σύστημα
  - Κοινή πολιτική χρονισμού
  - Κοινό ρολόι
  - Σχεδίαση τήρησης χρόνων διάδοσης
- Τμήματα με διαφορετικά ρολόγια
  - Ή ασύγχρονες είσοδοι
  - Αδυναμία τήρησης χρόνων setup – hold
  - Τα στοιχεία μνήμης μπορούν να περιέλθουν σε απροσδιόριστη κατάσταση (metastability)
  - Ειδικά κυκλώματα συγχρονισμού

Στοιχεία μνήμης

- Ακολουθιακή λογική
- Στοιχεία μνήμης

!

Η έξοδος ενός στοιχείου μνήμης εξαρτάται από τις εισόδους του και την τρέχουσα κατάστασή του!

- Μαζί με λογικά κυκλώματα
  - Για αποθήκευση ενδιάμεσης κατάστασης μεταξύ συνδυαστικών συναρτήσεων
  - Μικρή χωρητικότητα
  - Παράδειγμα: οι καταχωρητές της ΚΜΕ
- Σε μεγάλες συστοιχίες μνήμης
  - Κύρια μνήμη συστήματος
  - Μεγάλη χωρητικότητα
  - Μεγαλύτερη ολοκλήρωση κυκλωμάτων

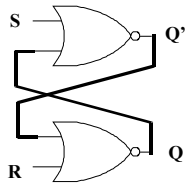
Απλά στοιχεία μνήμης χωρίς ρολόι

- Ακολουθιακή λογική
- Στοιχεία μνήμης

!

Τα S και R δεν επιτρέπεται να είναι ταυτόχρονα '1'

S	R	Q	Q'
0	0	Q	Q'
0	1	0	1
1	0	1	0
~~1~~	~~1~~	~~0~~	~~0~~

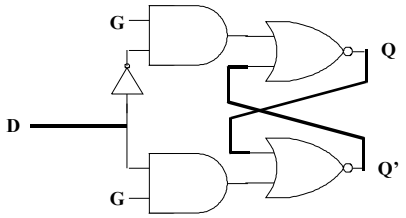


Ο "μανδαλωτής" (latch)

- Ακολουθιακή λογική
- Στοιχεία μνήμης

!

Το latch περνά την είσοδο στην έξοδο ανάλογα με τη **στάθμη** του G



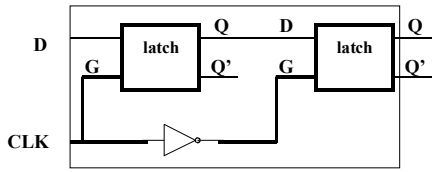
D	G	Q	Q'
x	0	Q	Q'
0	1	0	1
1	1	1	0

D flip-flop

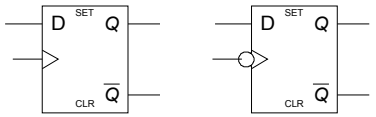
- Ακολουθιακή λογική
- Στοιχεία μνήμης

;

Σε ποια ακμή του CLK αποθηκεύεται η είσοδος D;



Αποθήκευση εισόδου στην ακμή του σήματος ρολογιού

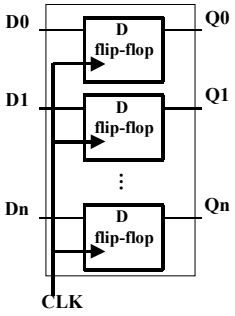


Καταχωρητές (registers)

- Ακολουθιακή λογική
- Στοιχεία μνήμης

;

Ποιο το μήκος της λέξης που αποθηκεύεται σε έναν καταχωρητή;



Ομάδα στοιχείων flip-flop με κοινό σήμα ρολογιού

Register file

- Ακολουθιακή λογική
- Στοιχεία μνήμης
- Register File

i

Οι σύγχρονες ΚΜΕ διαθέτουν τουλάχιστον 32 καταχωρητές γενικού σκοπού

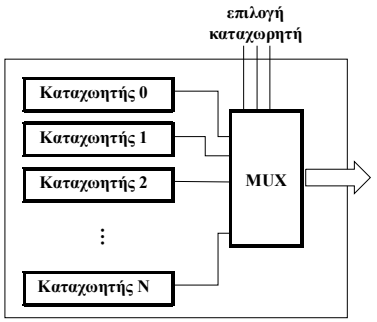
- **Ομάδα καταχωρητών, βασικό στοιχείο μιας ΚΜΕ**
- **Ανάγνωση και εγγραφή**
- **Δυνατότητα ταυτόχρονης λειτουργίας σε περισσότερους από έναν καταχωρητές**

Register file: ανάγνωση

- Ακολουθιακή λογική
- Στοιχεία μνήμης
- Register File

;

Πώς μπορούμε να διαβάσουμε ταυτόχρονα 2 καταχωρητές (σε διαφορετικές εξόδους);

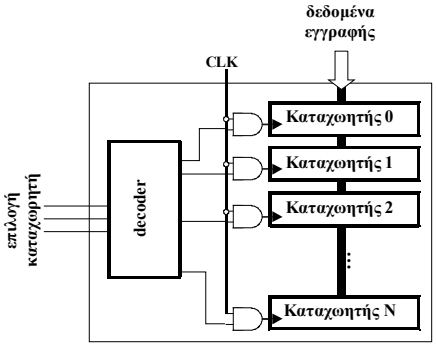


Register file: εγγραφή

- Ακολουθιακή λογική
- Στοιχεία μνήμης
- Register File

!

Τα δεδομένα προς εγγραφή θα πρέπει να είναι σταθερά κατά τους χρόνους setup και hold!



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Αρχιτεκτονικές Συνόλου Εντολών

(Instruction Set Architectures - ISA)

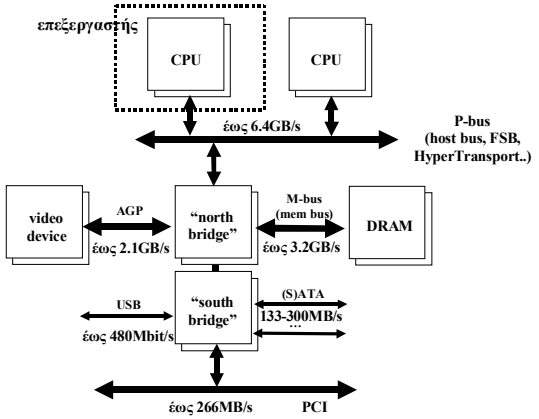
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Ένα τυπικό υπολογιστικό σύστημα σήμερα

• Ο επεξεργαστής

•**Επεξεργαστής**
•Μνήμη
•Συσκευές Ε/Ε

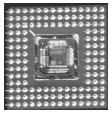


Ο (μικρο)επεξεργαστής

• Ο επεξεργαστής

;

Τι περιέχεται στη συσκευασία ενός μικροεπεξεργαστή σήμερα;



- (Micro)processor
  - Ψηφιακό σύστημα που εκτελεί υπολογισμούς σε πολλαπλά βήματα
    - Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)
    - Central Processing Unit (CPU)
  - Μέρος ενός ευρύτερου υπολογιστικού συστήματος (ή "υπολογιστή")
    - **γενικού** ή ειδικού σκοπού
  - Μπορεί να περιέχει σήμερα πολλαπλές υπομονάδες επεξεργασίας

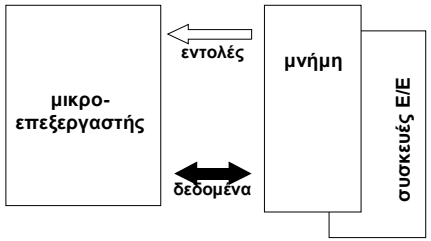
Είδη μικροεπεξεργαστών

• Ο επεξεργαστής

- Γενικού σκοπού
  - Οι περισσότερο γνωστοί επεξεργαστές (PCs)
- Συνεπεξεργαστές (co-processors)
  - Ειδικές λειτουργίες
- Μικροελεγκτές (microcontrollers)
  - Συστήματα ελέγχου
- Μέσα σε Systems-on-Chip
  - Επεξεργαστικοί πυρήνες

Το μοντέλο von Neumann

• Ο επεξεργαστής



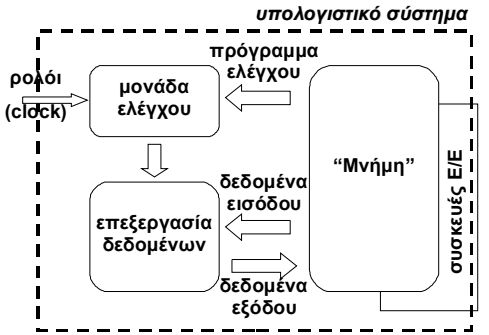
;

Σε ποια μορφή αποθηκεύονται οι εντολές;

- Το πρόγραμμα εκτέλεσης, όπως και τα δεδομένα, αποθηκεύονται στη μνήμη του υπολογιστή
  - **Stored-program computer**

Εκτέλεση ακολουθίας λειτουργιών

• Ο επεξεργαστής



;

Ποιος ο ρόλος του σήματος ρολογιού;

•

Εκτέλεση εντολών: ο κύκλος μηχανής

• Ο επεξεργαστής

;

Τι συμβαίνει με τις εξωτερικές διακοπές (**interrupts**);

Ο χρόνος εκτέλεσης είναι ο ίδιος για όλες τις εντολές;

για πάντα

Fetch: Φέρε την επόμενη εντολή προς εκτέλεση από τη μνήμη

⇩

Decode: Αποκωδικοποίησε την εντολή (προετοίμασε τα σήματα ελέγχου και τις πηγές των δεδομένων)

⇩

Execute: Εκτέλεσε την απαιτούμενη πράξη/λειτουργία

⇩

Store: Αποθήκευσε τα αποτελέσματα (εάν απαιτείται)

Εκτέλεση εντολών

• Ο επεξεργαστής

- Επόμενη εντολή προς εκτέλεση
  - Program Counter (PC): η διεύθυνση της θέσης μνήμης όπου περιέχεται η επόμενη εντολή
    - Σειριακή αύξηση διεύθυνσης μετά την εκτέλεση εντολής
    - Ή μεταπήδηση σε νέα θέση μνήμης (διακλάδωση)
- Εκκίνηση εκτέλεσης
  - Με την εφαρμογή τάσης ο PC παίρνει μια προκαθορισμένη τιμή
    - BIOS
- Τερματισμός επανάληψης κύκλου μηχανής
  - Συμβατικά, ποτέ!
  - Περιπτώσεις SLEEP για μείωση κατανάλωσης ενέργειας

Αρχιτεκτονική Συνόλου Εντολών

- Ο επεξεργαστής
- ISA

- Instruction Set Architecture (ISA)
  - Το ορατό μέρος ενός υπολογιστικού συστήματος για τον προγραμματιστή (μεταγλωττιστή)
  - Δεκαετία 60-70: συνώνυμο του όρου αρχιτεκτονική Η/Υ
    - « η δομή ενός υπολογιστή, την οποία ο προγραμματιστής πρέπει να γνωρίζει για να γράψει ένα σωστό (χρονικά ανεξάρτητο) πρόγραμμα σε γλώσσα μηχανής για τον υπολογιστή αυτόν» (IBM)

Η διεπαφή ISA στην ιεραρχία επιπέδων

- Ο επεξεργαστής
- ISA

;

Τι ακριβώς περιγράφει η διεπαφή ISA;

- Αρχιτεκτονική Εντολών (ISA)
  - Η διεπαφή υλικού-λογισμικού

Αρχιτεκτονική Συνόλου Εντολών

- Ο επεξεργαστής
- ISA

- Τι περιγράφει;
  - Διαθέσιμες πράξεις/λειτουργίες
    - Κωδικοποίηση λειτουργιών
  - Μορφή των δεδομένων εισόδου-εξόδου
    - Έντελα (operands)
  - Μέθοδοι προσπέλασης μνήμης
    - Προέλευση των δεδομένων
  - Χώροι προσωρινής αποθήκευσης
    - Καταχωρητές
  - Διακοπές και καταστάσεις σφάλματος
    - Ποια η "αντίδραση" του επεξεργαστή

Σχεδιασμός Συνόλου Εντολών

- Ο επεξεργαστής
- ISA

- Συμβιβασμός μεταξύ:
  - Κόστους και απόδοσης υλικού
  - Βαθμού υποστήριξης λογισμικού
  - Άλλων παραγόντων όπως η κατανάλωση ενέργειας
    - Το υλικό καταναλώνει ενέργεια, υπό τον έλεγχο του λογισμικού όμως!
  - Μη επικάλυψης λειτουργιών εντολών (orthogonality)

Κωδικοποίηση Εντολών

- Ο επεξεργαστής
- ISA

;

Γιατί είναι ταχύτερη η λήψη και αποκωδικοποίηση των εντολών σταθερού μήκους;

opcode	operand1	operand2	..	operandN

- Σειρά δυαδικών ψηφίων
  - Μεταβλητού μήκους
    - Περισσότερο συμπαγή προγράμματα
    - Πολυπλοκότερο υλικό!
  - Σταθερού μήκους
    - Απλούστερη και ταχύτερη λήψη-αποκωδικοποίηση
    - Μεγαλύτερα προγράμματα
    - Μέθοδοι συμπίεσης

Κωδικοποίηση Εντολών

- Ο επεξεργαστής
- ISA

opcode	operand1	operand2	..	operandN

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: κατηγορίες λειτουργιών

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

- Βασικές κατηγορίες
  - Αριθμητικές και λογικές πράξεις
  - Μεταφορά δεδομένων
    - Από-πρός Καταχωρητές και Μνήμη
  - Έλεγχος ροής εκτέλεσης
    - Διακλαδώσεις και κλήσεις ρουτινών
- Άλλες κατηγορίες
  - Ειδικές εντολές συστήματος
    - ΛΣ, ιδεατή μνήμη
  - Επεξεργασία πολλαπλών δεδομένων
    - Χρήσιμο για γραφικά, σειρές χαρακτήρων, multimedia

Αριθμητικές εντολές και μεταφορά δεδομένων

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

;

Τι συμβολίζουν τα R1, R2 ..; Πώς αναπαρίστανται μέσα στην εντολή;

- Αριθμητικές-λογικές πράξεις
  - Πηγές δεδομένων και προορισμός
    - add R1, R2, R3 // R3 = R1+R2

add	R1	R2	R3

- Μεταφορά δεδομένων
  - Πηγή δεδομένων και προορισμός
  - Μήκος μεταφερόμενης λέξης (ενδεχομένως)
    - load R1, 0x7FF0 // R1 = mem[0x7FF0]

load	R1	0x7FF0

Εντολές διακλάδωσης

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

- Διακλάδωση
  - Με ή χωρίς συνθήκη
    - bne R1, R2, +8 // branch if not R1==R2
  - Σε απόλυτη διεύθυνση
    - jump 0xFF97DE00
  - Σχετικά ως προς την τρέχουσα θέση (offset)
    - jump +130 // offset = +130
    - **Ο παραγόμενος κώδικας μπορεί να τοποθετηθεί οπουδήποτε στη μνήμη**

bne	R1	R2	+8

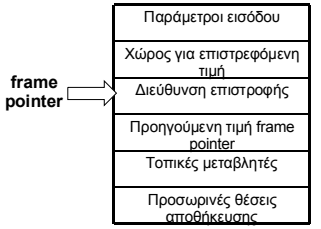
Εντολές διακλάδωσης

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

;

Πού θα μπορούσε να αποθηκευτεί η διεύθυνση επιστροφής μετά από κλήση συνάρτησης;

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



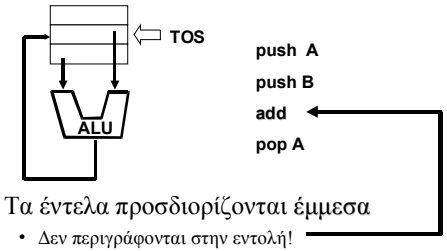
Προέλευση και αποθήκευση δεδομένων

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων

- Προέλευση δεδομένων – αποθήκευση αποτελεσμάτων μιας πράξης
  - **Operand addressing**
  - Εξαρτάται από την αρχιτεκτονική του επεξεργαστή
  - Στους πρώτους επεξεργαστές
    - Stack (σωρός-στοίβα)
    - Accumulator (συσσωρευτής)
  - Μεταγενέστεροι υπολογιστές
    - Έντελα Καταχωρητή-Μνήμης
    - Έντελα Καταχωρητή-Καταχωρητή (load-store)

Αρχιτεκτονική σωρού (stack)

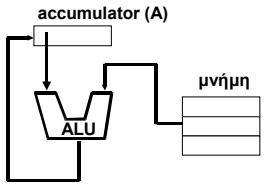
- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων



- Τα έντελα προσδιορίζονται έμμεσα
  - Δεν περιγράφονται στην εντολή!
  - 0-address architecture
  - Δημοφιλές σχήμα κατά τη δεκαετία του 60
  - Δύσκολη προσπέλαση σωρού, απαιτούνται πολλαπλές αντιμεταθέσεις και αντιγραφές
  - Το σχήμα επιζεί στην αρχιτεκτονική της Java VM

Αρχιτεκτονική συσσωρευτή (accumulator)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων



add X

(A = A + X)

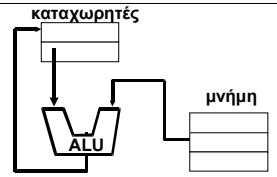
- Ένα έντελο - θέση αποθήκευσης του αποτελέσματος είναι πάντα ο συσσωρευτής
  - 1-address architecture
  - Αρχιτεκτονική των πρώτων υπολογιστών!

Αρχιτεκτονικές με καταχωρητές (registers)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων

!

Καταχωρητές: προσωρινές θέσεις αποθήκευσης αποτελεσμάτων, η γενίκευση της ιδέας του συσσωρευτή.



add R2, R1,mem(100)

(R2 = R1 + mem[100])

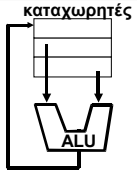
- Memory-register
  - Οποιαδήποτε εντολή μπορεί να προσπελάσει τη μνήμη
- Όμως:
  - Πολλαπλές προσπελάσεις μνήμης
  - Λήψη εντολής – Λήψη εντέλων
  - Συνωστισμός στον δίαυλο επικοινωνίας με μνήμη

Αρχιτεκτονικές με καταχωρητές (registers)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων

;

Είναι επιθυμητός ένα πολύ μεγάλος αριθμός καταχωρητών;



add R1, R2,R3

(R1 = R2 + R3)

- Register-register (load-store)
  - Μόνο εντολές load-store μπορούν να προσπελάσουν τη μνήμη
- Η αρχιτεκτονική των σύγχρονων επεξεργαστών
  - Οι καταχωρητές προσπελαύνονται πολύ γρήγορα
  - Χρειάζονται λιγότερα bits για να επιλεγούν
  - Οι μεταγλωττιστές αναθέτουν μεταβλητές σε καταχωρητές

Μέθοδοι προσπέλασης μνήμης

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

- Τουλάχιστον κάποιες εντολές προσπελαύνουν 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

Μέθοδοι προσπέλασης μνήμης

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

;

Πώς κωδικοποιούνται οι μέθοδοι προσπέλασης μνήμης μέσα στην εντολή;

- Στο σχηματισμό της διεύθυνσης μνήμης μπορούν να συμμετέχουν:
  - Απόλυτες τιμές διεύθυνσης
  - Καταχωρητές
  - Σταθερές τιμές μετατόπισης (offsets)

		πιθανή χρήση
displacement	**mem[offs+reg]**	**τοπικές**
register indirect	**mem[reg]**	**δείκτες**
indexed	**mem[reg1+reg2]**	**πίνακες**
direct	**mem[addr]**	**στατικές**
memory indirect	**mem[mem[reg]]**	***δείκτες**
auto-increment	**mem[reg++]**	**πίνακες**
scaled	**mem[offs+reg1+reg2*d]**	**πίνακες**

Η εξέλιξη της αρχιτεκτονικής εντολών

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Εξέλιξη αρχιτεκτονικής

- Οι πρώτοι υπολογιστές (.. - '60)
  - Αρχιτεκτονική συσσωρευτή και αργότερα σωρού
    - Ικανοποιητική λύση λόγω της απλής τεχνολογίας των μεταγλωττιστών
- Πολύπλοκες αρχιτεκτονικές ('70 - ..)
  - Ενσωμάτωση σύνθετων μορφών εντολών και μεθόδων προσπέλασης μνήμης
    - Προσπάθεια υποστήριξης υψηλών γλωσσών προγραμματισμού – μείωσης κόστους λογισμικού
    - Πολλά χαρακτηριστικά μένουν αχρησιμοποίητα!
  - Complex Instruction Set Computers (CISC)

Η εξέλιξη της αρχιτεκτονικής εντολών

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Εξέλιξη αρχιτεκτονικής

- Reduced Instruction Set Computers (RISC) ('80 - …)
  - Απλούστερες και φθηνότερες load-store αρχιτεκτονικές με σταθερό μήκος εντολών
  - Μεγαλύτερη απόδοση – ταχύτερη εκτέλεση εντολών
  - Ευνοείται από την αφθονία υλικού χαμηλού κόστους και την προηγμένη τεχνολογία των μεταγλωττιστών
  - Ακόμα και ο μοναδικός επιζών επεξεργαστής με εντολές CISC (αρχιτεκτονική x86), μεταφράζει εσωτερικά σε εντολές RISC…

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Αρχιτεκτονικές Συνόλου Εντολών (ΙΙ)

(Δομή Εντολών και Παραδείγματα)

http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Αρχιτεκτονική Συνόλου Εντολών

- ISA

- Τι περιγράφει;
  - Διαθέσιμες πράξεις/λειτουργίες
    - Κωδικοποίηση λειτουργιών
  - Μορφή των δεδομένων εισόδου-εξόδου
    - Έντελα (operands)
  - Μέθοδοι προσπέλασης μνήμης
    - Προέλευση των δεδομένων
  - Χώροι προσωρινής αποθήκευσης
    - Καταχωρητές
  - Διακοπές και καταστάσεις σφάλματος
    - Ποια η "αντίδραση" του επεξεργαστή

Σχεδιασμός Συνόλου Εντολών

- ISA

- Συμβιβασμός μεταξύ:
  - Κόστους και απόδοσης υλικού
  - Βαθμού υποστήριξης λογισμικού
  - Άλλων παραγόντων όπως η κατανάλωση ενέργειας
    - Το υλικό καταναλώνει ενέργεια, υπό τον έλεγχο του λογισμικού όμως!
  - Μη επικάλυψης λειτουργιών εντολών (orthogonality)

Κωδικοποίηση Εντολών Μηχανής

- ISA

;

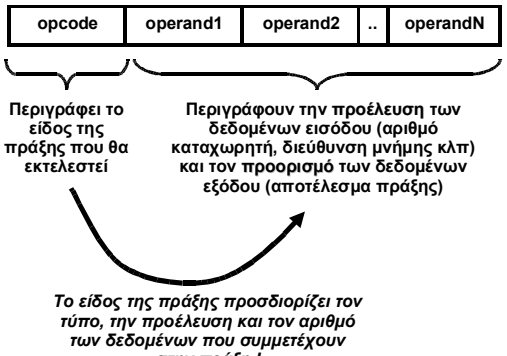
Γιατί είναι ταχύτερη η λήψη και αποκωδικοποίηση των εντολών σταθερού μήκους;

opcode	operand1	operand2	..	operandN

- Σειρά δυαδικών ψηφίων
  - Μεταβλητού μήκους
    - Περισσότερο συμπαγή προγράμματα
    - Πολυπλοκότερο υλικό!
  - Σταθερού μήκους
    - Απλούστερη και ταχύτερη λήψη-αποκωδικοποίηση
    - Μεγαλύτερα προγράμματα
    - Μέθοδοι συμπίεσης

Κωδικοποίηση Εντολών Μηχανής

• ISA



Εντολές Μηχανής και Assembly

• ISA

!

Κάθε διαφορετική αρχιτεκτονική επεξεργαστών έχει διαφορετική γλώσσα assembly!

- 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
  - Προσοχή:
    - MOV R1,R2 // π.χ. R1 => R2
    - MOV R1,#100 // π.χ. R1 => mem[100]
    - παρόλο που το μνημονικό MOV είναι ίδιο, δεν πρόκειται για την ίδια εντολή!

Τα παραδείγματά μας: Intel IA-32 (και 64)

• ISA

i

Και η ανταγωνίστρια εταιρία AMD παράγει επεξεργαστές συμβατούς με την αρχιτεκτονική αυτή

- Αρχιτεκτονική επεξεργαστών Intel IA-32 και Intel 64
  - Μία πολύπλοκη αρχιτεκτονική επεξεργαστών
    - 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 στην Intel έρευνα και ανάπτυξη για διαρκείς βελτιώσεις απόδοσης
    - "Τελικά η αρχιτεκτονική αυτή είναι βέλτιστη μέσα στην τεράστια πολυπλοκότητά της"

Τα παραδείγματά μας: Intel IA-32 (και 64)

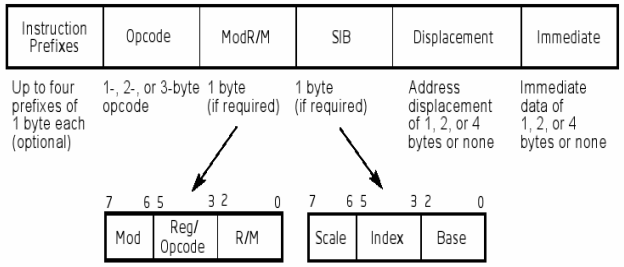
• ISA

- Οικογένειες επεξεργαστών
  - Pentium
  - P6
    - Pentium Pro, II, III
  - Netburst
    - Pentium 4, D (IA-32 – σε νεώτερες εκδόσεις και Intel 64)
  - Pentium M
    - Intel Core, Core Duo
  - Intel Core (IA-32 και Intel 64)
    - Intel Core 2 Duo

Intel® 64 and IA-32 Architectures Software Developer's Manuals
http://www.intel.com/products/processor/manuals/index.htm

Intel IA-32: η γενική μορφή των εντολών

- Μεταβλητός αριθμός bytes ανά εντολή (1-17)!



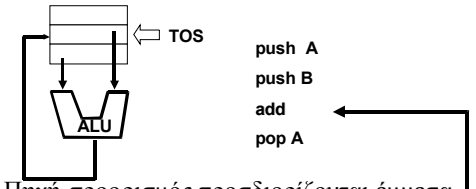
Προέλευση και αποθήκευση δεδομένων

- ISA
- Προέλευση δεδομένων

- Προέλευση δεδομένων – αποθήκευση αποτελεσμάτων μιας πράξης
  - **Operand addressing**
  - Εξαρτάται από την αρχιτεκτονική του επεξεργαστή
  - Στους πρώτους επεξεργαστές
    - Stack (σωρός-στοίβα)
    - Accumulator (συσσωρευτής)
  - Μεταγενέστεροι υπολογιστές
    - Καταχωρητές – Μνήμη (register-memory)
    - Κυρίως καταχωρητές (register-register ή load-store)
      - μόνο ανάγνωση και εγγραφή μνήμης

Αρχιτεκτονική σωρού (stack)

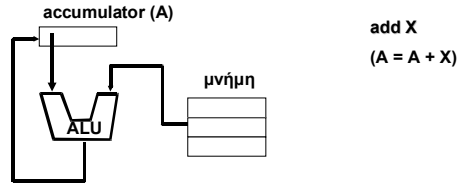
- ISA
- Προέλευση δεδομένων



- Πηγή-προορισμός προσδιορίζονται έμμεσα
  - Δεν περιγράφονται στην εντολή!
  - 0-address architecture
  - Δημοφιλές σχήμα κατά τη δεκαετία του 60
  - Δύσκολη προσπέλαση σωρού, απαιτούνται πολλαπλές αντιμεταθέσεις και αντιγραφές
  - Το σχήμα επιζεί στην αρχιτεκτονική της Java VM

Αρχιτεκτονική συσσωρευτή (accumulator)

- ISA
- Προέλευση δεδομένων



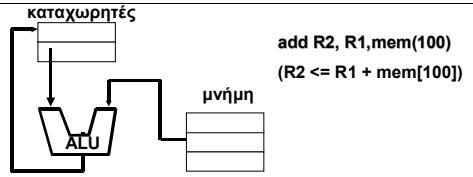
- Μία πηγή - θέση αποθήκευσης του αποτελέσματος είναι πάντα ο συσσωρευτής
  - 1-address architecture
  - Αρχιτεκτονική των πρώτων υπολογιστών!

Αρχιτεκτονικές με καταχωρητές (1)

- ISA
- Προέλευση δεδομένων

!

Καταχωρητές: προσωρινές θέσεις αποθήκευσης αποτελεσμάτων, η γενίκευση της ιδέας του συσσωρευτή.



add R2, R1,mem(100)

(R2 <= R1 + mem[100])

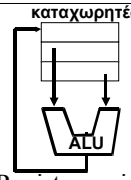
- Memory-register
  - Οποιαδήποτε εντολή μπορεί να προσπελάσει τη μνήμη
- Όμως:
  - Πολλαπλές προσπελάσεις μνήμης
  - Λήψη εντολής – Λήψη εντέλων
  - Συνωστισμός στον δίαυλο επικοινωνίας με μνήμη

Αρχιτεκτονικές με καταχωρητές (2)

- ISA
- Προέλευση δεδομένων

;

Είναι επιθυμητός ένα πολύ μεγάλος αριθμός καταχωρητών;



add R1, R2,R3

(R1 <= R2 + R3)

- Register-register (load-store)
  - Μόνο εντολές load-store μπορούν να προσπελάσουν τη μνήμη
- Η αρχιτεκτονική των σύγχρονων επεξεργαστών
  - Οι καταχωρητές προσπελαύνονται πολύ γρήγορα
  - Χρειάζονται λιγότερα bits για να επιλεγούν
  - Οι μεταγλωττιστές αναθέτουν μεταβλητές σε καταχωρητές

Η εξέλιξη της αρχιτεκτονικής εντολών

- ISA
- Προέλευση δεδομένων

- Οι πρώτοι υπολογιστές (.. - '60)
  - Αρχιτεκτονική συσσωρευτή και αργότερα σωρού
    - Ικανοποιητική λύση λόγω της απλής τεχνολογίας των μεταγλωττιστών
- Πολύπλοκες αρχιτεκτονικές ('70 - ..)
  - Ενσωμάτωση σύνθετων μορφών εντολών και μεθόδων προσπέλασης μνήμης
    - Προσπάθεια υποστήριξης υψηλών γλωσσών προγραμματισμού – μείωσης κόστους λογισμικού
    - Πολλά χαρακτηριστικά μένουν αχρησιμοποίητα!
  - Complex Instruction Set Computers (CISC)

Η εξέλιξη της αρχιτεκτονικής εντολών

- ISA
- Προέλευση δεδομένων

- Reduced Instruction Set Computers (RISC) ('80 - …)
  - Απλούστερες και φθηνότερες load-store αρχιτεκτονικές με σταθερό μήκος εντολών
  - Μεγαλύτερη απόδοση – ταχύτερη εκτέλεση εντολών
  - Ευνοείται από την αφθονία υλικού χαμηλού κόστους και την προηγμένη τεχνολογία των μεταγλωττιστών
  - Ακόμα και ο μοναδικός επιζών επεξεργαστής με εντολές CISC (αρχιτεκτονική x86), μεταφράζει εσωτερικά σε εντολές RISC…

IA-32 (64): memory-register

- ISA
- Προέλευση δεδομένων

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

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..	

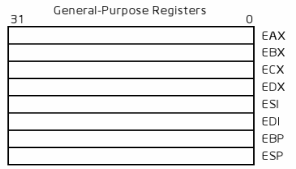
IA-32: καταχωρητές

- ISA
- Προέλευση δεδομένων

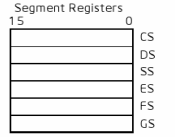
i

Εκτός από τους καταχωρητές γενικού σκοπού, οι επεξεργαστές IA-32 έχουν πρόσθετους καταχωρητές για πράξεις κινητής υποδιαστολής και πράξεις streaming και multimedia.

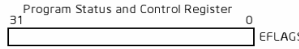
γενικού σκοπού και σχηματισμός διευθύνσεων



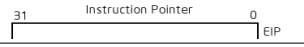
τμηματοποίηση (σε επόμενα μαθηματα)



καταχωρητής κατάστασης



"program counter"



Μέθοδοι προσπέλασης μνήμης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

- Τουλάχιστον κάποιες εντολές προσπελαύνουν 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

Μέθοδοι προσπέλασης μνήμης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

;

Πώς κωδικοποιούνται οι μέθοδοι προσπέλασης μνήμης μέσα στην εντολή;

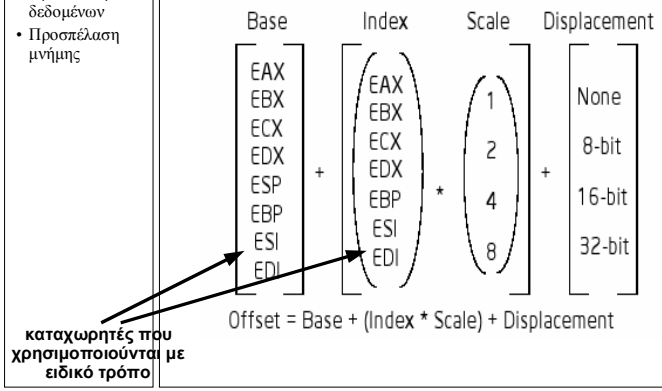
- Στο σχηματισμό της διεύθυνσης μνήμης μπορούν να συμμετέχουν:
  - Απόλυτες τιμές διεύθυνσης
  - Καταχωρητές
  - Σταθερές τιμές μετατόπισης (offsets)

πιθανή χρήση

displacement	**mem[offs+reg]**	**τοπικές**
register indirect	**mem[reg]**	**δείκτες**
indexed	**mem[reg1+reg2]**	**πίνακες**
direct	**mem[addr]**	**στατικές**
memory indirect	**mem[mem[reg]]**	***δείκτες**
auto-increment	**mem[reg++]**	**πίνακες**
scaled	**mem[offs+reg1+reg2*d]**	**πίνακες**

IA-32 (64): Σχηματισμός διευθύνσεων

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης



Εντολές: κατηγορίες λειτουργιών

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

- Βασικές κατηγορίες
  - Αριθμητικές και λογικές πράξεις
  - Μεταφορά δεδομένων
    - Από-πρός Καταχωρητές και Μνήμη
  - Έλεγχος ροής εκτέλεσης
    - Διακλαδώσεις και κλήσεις ρουτινών
- Άλλες κατηγορίες
  - Ειδικές εντολές συστήματος
    - ΛΣ, ιδεατή μνήμη
  - Επεξεργασία πολλαπλών δεδομένων
    - Χρήσιμο για γραφικά, σειρές χαρακτήρων, multimedia

Αριθμητικές εντολές και μεταφορά δεδομένων

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

;

Τι συμβολίζουν τα R1, R2 ..; Πώς αναπαρίστανται μέσα στην εντολή;

- Αριθμητικές-λογικές πράξεις
  - Πηγές δεδομένων και προορισμός
    - add R1, R2, R3 // R3 = R1+R2

add	R1	R2	R3

- Μεταφορά δεδομένων
  - Πηγή δεδομένων και προορισμός
  - Μήκος μεταφερόμενης λέξης (ενδεχομένως)
    - load R1, 0x7FF0 // R1 = mem[0x7FF0]

load	R1	0x7FF0

IA-32: εντολές μεταφοράς δεδομένων

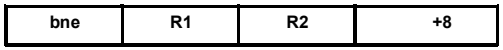
Table 7-1. Move Instruction Operations

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

Εντολές διακλάδωσης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

- Διακλάδωση
  - Με ή χωρίς συνθήκη
    - bne R1, R2, +8 // branch if not R1==R2
  - Σε απόλυτη διεύθυνση
    - jump 0xFF97DE00
  - Σχετικά ως προς την τρέχουσα θέση (offset)
    - jump +130 // offset = +130
    - **Ο παραγόμενος κώδικας μπορεί να τοποθετηθεί οπουδήποτε στη μνήμη**



IA-32: Εντολές διακλάδωσης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

- Με ή χωρίς συνθήκη
  - Σε απόλυτη ή σχετική διεύθυνση
    - 16 ή 32 bits απόλυτη διεύθυνση
      - πιθανώς σε μνήμη ή καταχωρητή
    - 8, 16 ή 32 bits σχετική διεύθυνση
  - 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

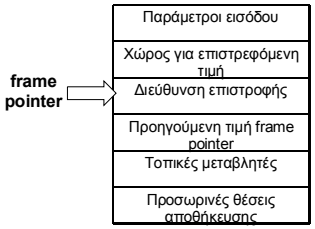
Εντολές διακλάδωσης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

;

Πού θα μπορούσε να αποθηκευτεί η διεύθυνση επιστροφής μετά από κλήση συνάρτησης;

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



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Κεντρική Μονάδα Επεξεργασίας

(Σχεδιασμός και λειτουργία μιας απλής ΚΜΕ)

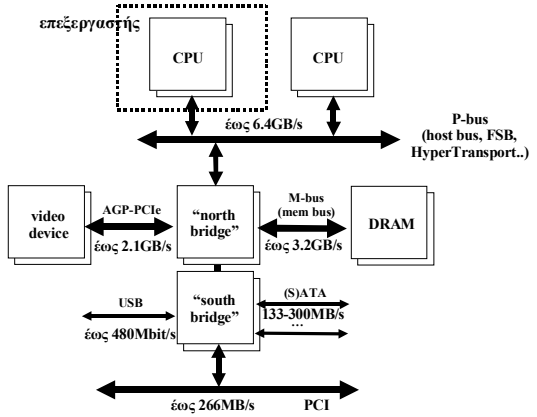
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Η υπολογιστική μηχανή: ο επεξεργαστής

- Ο επεξεργαστής

- Επεξεργαστής
- Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)



Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)

- Ο επεξεργαστής
- ΚΜΕ

- (Μικρο)επεξεργαστής
  - Ψηφιακό σύστημα που εκτελεί υπολογισμούς σε πολλαπλά βήματα
- Κεντρική Μονάδα Επεξεργασίας
  - Συνώνυμο του επεξεργαστή (σχεδόν!)
  - Στην ίδια συσκευασία
    - Κρυφή Μνήμη (1ου και 2ου επιπέδου)
    - Ελεγκτές Ε/Ε

Σχεδιασμός μιας απλής ΚΜΕ

- Ο επεξεργαστής
- ΚΜΕ

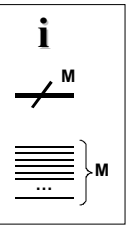
i

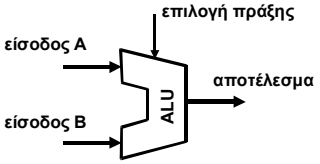
Όλες οι σύγχρονες ΚΜΕ είναι πολυπλοκότερες!

- Παραδοχές
  - Μοντέλο load-store
    - Αριθμητικές-λογικές πράξεις
    - Ανάγνωση-εγγραφή από/στη μνήμη
    - Εντολές διακλάδωσης
  - Ν καταχωρητές γενικού σκοπού
  - Απλουστευμένο μοντέλο μνήμης
    - Ως "κουτί" που δέχεται διεύθυνση και επιστρέφει δεδομένα και εντολές
    - Αντιπροσωπεύει στην πραγματικότητα μια ιεραρχία μνήμης!

Εκτέλεση πράξεων: ΑΛΜ

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ





- Αριθμητική-Λογική Μονάδα (ΑΛΜ)
  - Arithmetic-Logic Unit (ALU)
    - Εκτέλεση αριθμητικών και λογικών πράξεων
    - Εκτέλεση συγκρίσεων (για διακλάδωση υπό συνθήκη)
    - Υπολογισμός διευθύνσεων (για προσπέλαση μνήμης)

Εντολές για αριθμητικές – λογικές πράξεις

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ

R3 ← R1 op R2

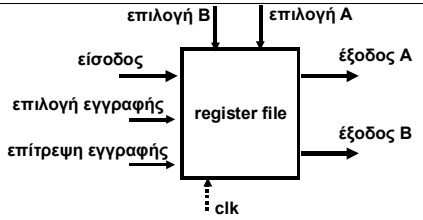
- Σε κάθε εκτέλεση εντολής
  - Ανάγνωση από 2 καταχωρητές
  - Εγγραφή σε 1 καταχωρητή
- Ίδια ροή δεδομένων
  - διαφορετική επιλεγόμενη πράξη

Συστοιχία καταχωρητών (register file)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές

!

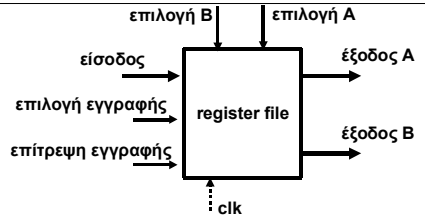
Τα σήματα στο σχήμα αυτό (και στα επόμενα) πιθανόν να αποτελούνται από πολλά bits το καθένα!



- Ανάγνωση: Επιλογή A και Επιλογή B
  - Ποιων καταχωρητών το περιεχόμενο θα εμφανιστεί στην αντίστοιχη έξοδο

Συστοιχία καταχωρητών (register file)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές



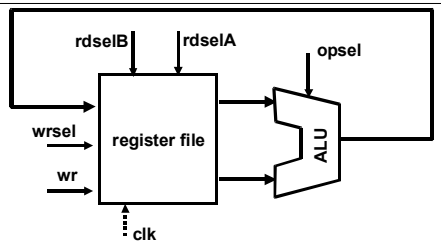
- Εγγραφή: στην (θετική) ακμή του clk
  - Στον καταχωρητή που ορίζει η **επιλογή εγγραφής**, εάν το επιτρέπει η **επίτρεψη εγγραφής**

Ένα απλό μονοπάτι δεδομένων (datapath)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath

;

Ποιος οδηγεί τα σήματα ελέγχου;



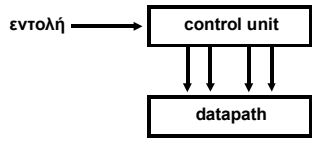
- Εκτέλεση μιας εντολής ανά κύκλο ρολογιού
- **Ανάγνωση και εγγραφή στον ίδιο κύκλο;**
  - Ναι, η ανάγνωση θα δώσει το αποτέλεσμα εγγραφής προηγούμενου κύκλου
  - Ενώ η εγγραφή θα είναι διαθέσιμη για ανάγνωση στον επόμενο κύκλο

Η μονάδα ελέγχου (control unit)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου

;

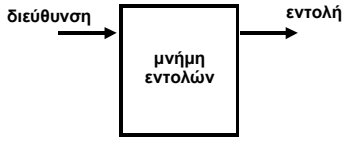
Από πού προέρχονται οι εντολές;



- Αποκωδικοποίηση bits της εντολής
  - Επιλογή καταχωρητών προέλευσης δεδομένων
  - Επιλογή καταχωρητή αποθήκευσης αποτελέσματος
  - Επιλογή λειτουργίας ΑΛΜ
  - Το δυσκολότερο στη σχεδίαση τμήμα της ΚΜΕ

Η "μνήμη εντολών"

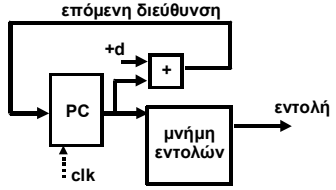
- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές



- Μια χρήσιμη αφαίρεση
  - Στην πραγματικότητα
    - Αντιπροσωπεύει μια ιεραρχία μνήμης
    - Από την κρυφή μνήμη 1ου επιπέδου μέχρι την κύρια μνήμη
    - Πιθανόν να μην είναι δυνατή η ανάκτηση εντολής ανά κύκλο ρολογιού

Η διεύθυνση της επόμενης εντολής

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές



- Program Counter (PC)
  - Περιέχει τη διεύθυνση στη μνήμη της επόμενης εντολής
    - Αυξάνεται κατά την εκτέλεση κάθε εντολής
    - Επόμενη διεύθυνση = τρέχουσα + d

Ο κύκλος μηχανής

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές

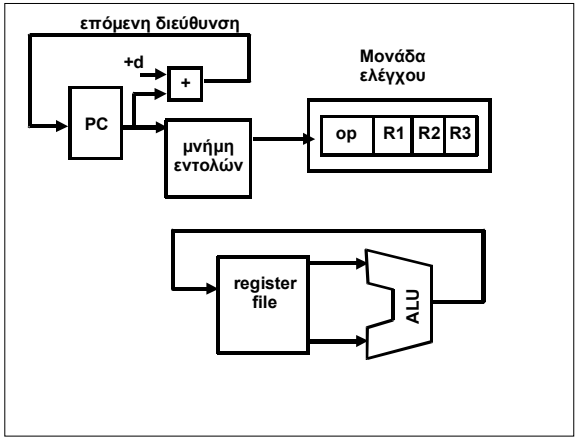
;

Ποια στάδια χρησιμοποιούν οι εντολές για αριθμητικές και λογικές πράξεις;

- Στάδια εκτέλεσης εντολής
  - Instruction Fetch (IF)
    - Ανάκτηση εντολής προς εκτέλεση
  - Instruction Decode (ID)
    - Επιλογή δεδομένων και λειτουργιών ανάλογα με εντολή
  - Execute (EX)
    - Εκτέλεση υπολογισμού στην ΑΛΜ
  - Data Memory Access (DM)
    - Προσπέλαση μνήμης δεδομένων
  - Write Back (WB)
    - Αποθήκευση αποτελεσμάτων σε καταχωρητή

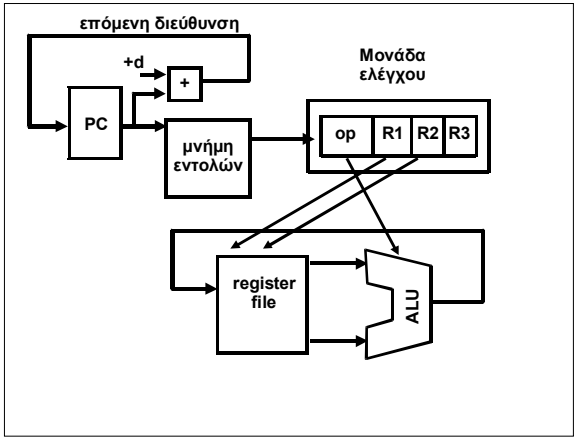
Instruction Fetch (IF)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές



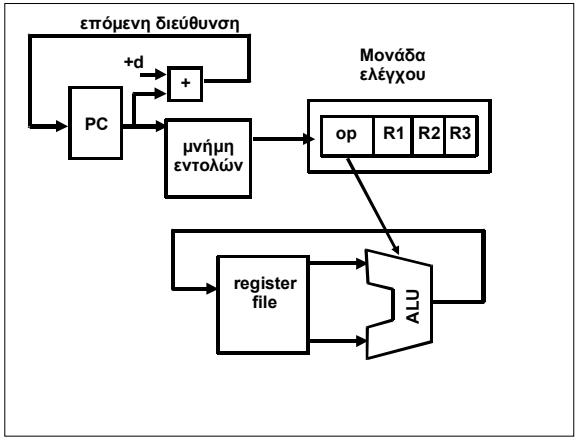
Instruction Decode (ID)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές



Execute (EX)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές

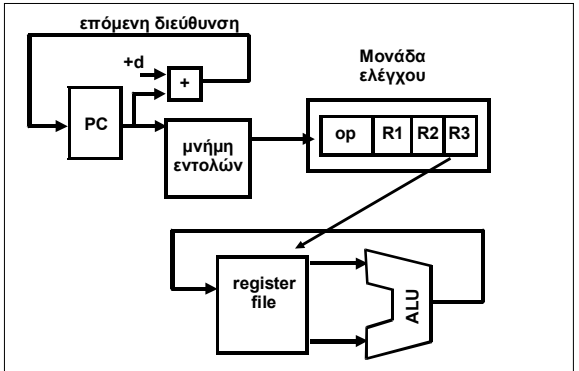


Write Back (WB)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές

;

Τι χρειάζεται να αλλάξει για να προστεθεί η εισαγωγή σταθερών;



Εντολές διακλάδωσης (branch ή jump)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις

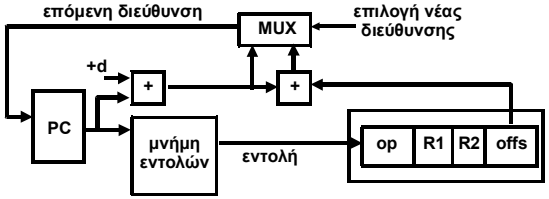
- Παράδειγμα

if R1 == R2 then PC ← PC + offset
"branch if equal"

- Διακλάδωση υπό συνθήκη
  - Βασισμένο σε αποτέλεσμα πράξης ΑΛΜ
- Σχετική διεύθυνση
- Υπολογισμός συνθήκης – διακλάδωση
  - Στην ίδια εντολή

Διακλαδώσεις και ενημέρωση PC

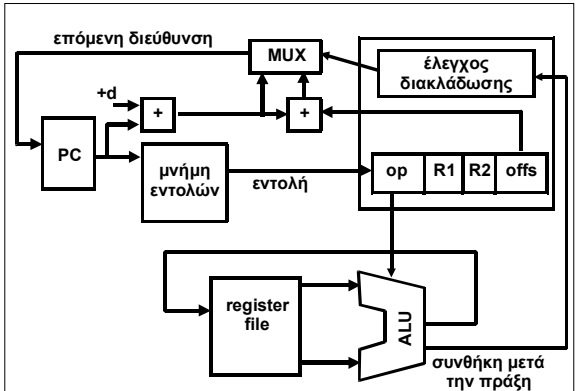
- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις



- Επιλογή νέας επόμενης διεύθυνσης
  - Η νέα τιμή του PC
  - Από τη μονάδα ελέγχου
  - Σύμφωνα με τιμές από πράξη στην ΑΛΜ
    - Π.χ. σύγκριση 2 καταχωρητών

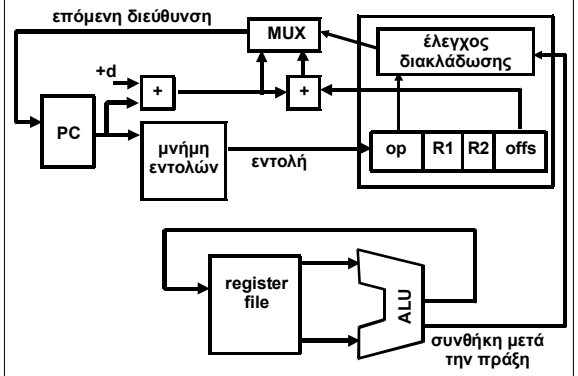
Execute (EX)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις



Ενημέρωση PC (“DM”)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις



Εντολές προσπέλασης μνήμης (load-store)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης

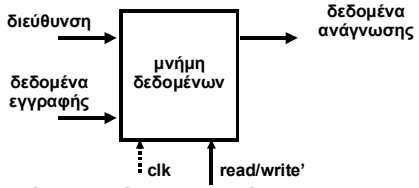
- Παράδειγμα

R1 ← mem[R2 + offset]

  - Μόνο οι εντολές load και store προσπελαύνουν τη μνήμη δεδομένων
  - Η διεύθυνση μνήμης παράγεται από την ΑΛΜ
    - Ως άθροισμα καταχωρητή + σταθεράς

Η “μνήμη δεδομένων”

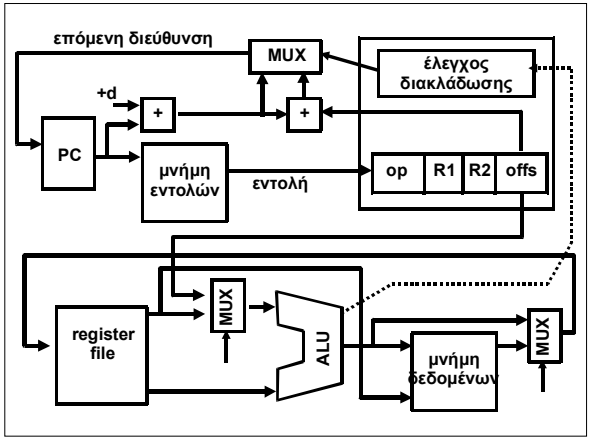
- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



- Μια επίσης χρήσιμη αφαίρεση
  - Ή εγγραφή ή ανάγνωση
    - clk για την εγγραφή
  - Στην πραγματικότητα
    - Αντιπροσωπεύει μια ιεραρχία μνήμης

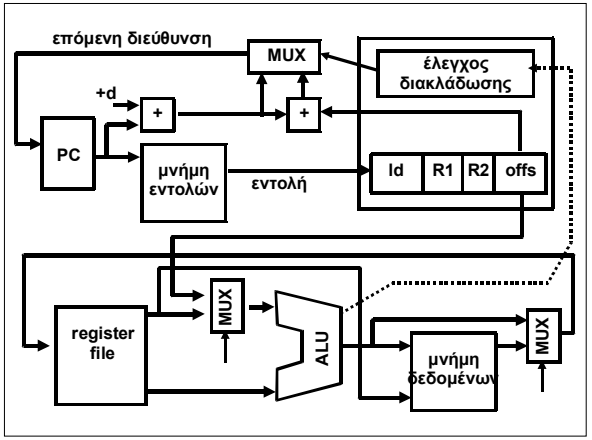
Προσπέλαση μνήμης δεδομένων

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



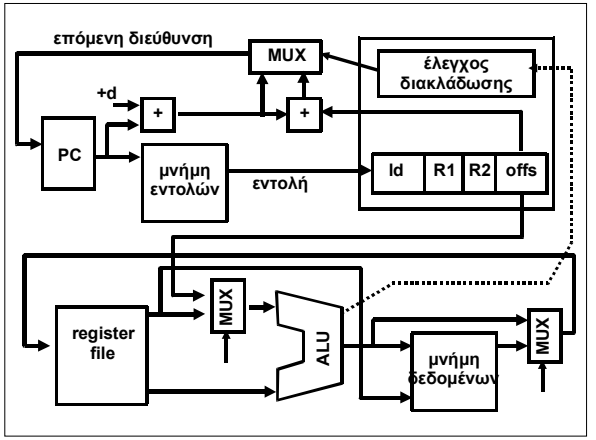
Execute (EX)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



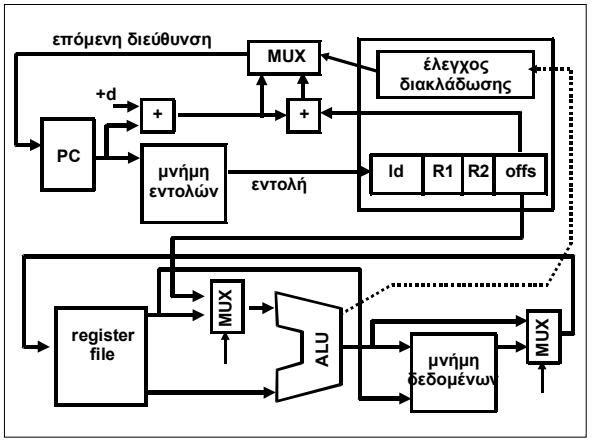
Data Memory Access (DM)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



Write Back (WB)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



Σύνοψη λειτουργίας ΚΜΕ

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης
- Σύνοψη

- Κύκλοι ρολογιού ανά εντολή
  - Clocks per Instruction (CPI)
    - Στο σημερινό παράδειγμα ΚΜΕ: 1 κύκλος ανά εντολή
- Περίοδος κύκλου ρολογιού
  - Clock Cycle (Period) (CC)
- Στο σημερινό παράδειγμα ΚΜΕ
  - CC σταθερό ανεξάρτητα από είδος εντολής
  - CC εξαρτάται από την πιο χρονοβόρα πράξη!
  - Όχι αποδοτικό σχήμα
  - Υπάρχει τεχνική για βελτίωση της απόδοσης;
    - (στο επόμενο μάθημα)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

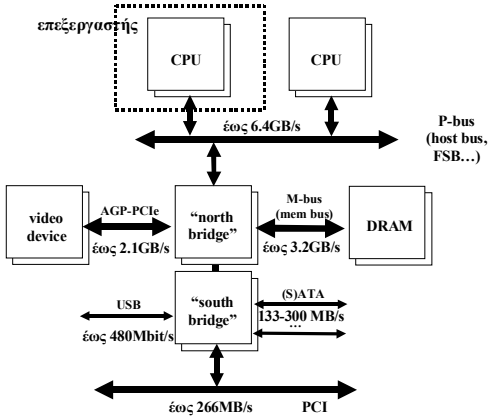
Απόδοση ΚΜΕ

(Μέτρηση και τεχνικές βελτίωσης απόδοσης)

http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Κεντρική Μονάδα Επεξεργασίας



•Επεξεργαστής ή...
•Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)

Απόδοση ΚΜΕ

- Υπολογιστικό σύστημα
  - Η απόδοση εξαρτάται από όλα τα επιμέρους τμήματά του
    - Υλικό και λογισμικό
- Κεντρική Μονάδα Επεξεργασίας
  - Πόσο γρήγορα εκτελείται ένα πρόγραμμα;
    - Επίσης σε αρχιτεκτονικές ειδικού σκοπού: αξιοπιστία, κατανάλωση ενέργειας
  - Πώς επηρεάζει η αρχιτεκτονική την απόδοση;
    - Πόσο γρηγορότερα εκτελείται ένα πρόγραμμα μετά από μια αρχιτεκτονική αλλαγή;

Εκτέλεση προγραμμάτων

- Χρόνος εκτέλεσης (execution time)
  - Επίσης: response time
    - Ο συνολικός χρόνος εκτέλεσης ενός προγράμματος
    - Μαζί με επιβάρυνση λειτουργικού συστήματος
  - Αύξηση απόδοσης ⇔ Μείωση χρόνου εκτέλεσης
  - Για υπολογιστή Χ:

$$\text{Απόδοση(X)} = \frac{1}{\text{Χρόνος Εκτέλεσης(X)}}$$

i

Throughput:
Ο ρυθμός ολοκλήρωσης έργου σε συγκεκριμένο χρόνο

Συγκρίνοντας δύο υπολογιστές

- Απόδοση ΚΜΕ

;

Ο Χ εκτελεί ένα πρόγραμμα σε 10 sec και ο Υ σε 15 sec. Πόσο πιο γρήγορος είναι ο Χ;

- Συγκρίνοντας αποδόσεις
  - Για υπολογιστές Χ και Υ
  - Εάν:

  Απόδοση(Χ) > Απόδοση(Υ)
  - Τότε (και αντίστροφα):

  Χρόνος Εκτέλεσης(Χ) < Χρόνος Εκτέλεσης(Υ)

$$\frac{\text{Απόδοση(X)}}{\text{Απόδοση(Y)}} = \frac{\text{Χρόνος Εκτέλεσης(Y)}}{\text{Χρόνος Εκτέλεσης(X)}} = n$$

Ο Χ είναι *n* φορές γρηγορότερος από τον Υ

Απόδοση ΚΜΕ

- Απόδοση ΚΜΕ

- Χρόνος εκτέλεσης προγράμματος
  - Εξαρτάται από όλα τα επιμέρους τμήματα του υπολογιστικού συστήματος
- Χρόνος εκτέλεσης ΚΜΕ
  - Ο χρόνος για τον οποίο η ΚΜΕ εκτελεί εντολές του προγράμματος
    - Όχι χρόνος για Ε/Ε ή άλλες διεργασίες
  - Χρόνος προγράμματος χρήστη
    - Για το πρόγραμμα καθεαυτό
  - Χρόνος συστήματος
    - Λειτουργίες ΛΣ για την εξυπηρέτηση του προγράμματος

Βασικά μεγέθη μέτρησης απόδοσης ΚΜΕ

- Απόδοση ΚΜΕ

;

Πώς υπολογίζεται το CPI όταν υπάρχουν διαφορετικοί τύποι εντολών;

- Κύκλος ρολογιού (περίοδος)
  - Clock Cycle (CC)
    - Η διάρκεια ενός κύκλου ρολογιού κατά τον οποίο η ΚΜΕ εκτελεί τις μικρότερες βασικές λειτουργίες
    - Σταθερό μέγεθος
- Κύκλοι ρολογιού ανά εντολή
  - Clocks Per Instruction (CPI)
    - Οι απαιτούμενοι κύκλοι ρολογιού για την ολοκλήρωση μιας εντολής
    - Ενδεχομένως διαφορετικό μέγεθος ανά τύπο εντολής
- Αριθμός εντολών
  - Instruction Count (IC)
    - Ο αριθμός των εντολών
    - Διάφοροι τύποι εντολών

Συσχέτιση με λογισμικό

- Απόδοση ΚΜΕ

!

Το υλικό (ISA) καθορίζει και τα τρία μεγέθη (IC, CPI και CC)

- Αλγόριθμος
  - Καθορίζει το IC
  - Ενδεχομένως καθορίζει το CPI, ευνοώντας ορισμένους τύπους εντολών (π.χ. κινητής υποδιαστολής)
- Γλώσσα προγραμματισμού
  - Καθορίζει το IC (μετάφραση εντολών υψηλού επιπέδου)
  - Καθορίζει το CPI απαιτώντας συγκεκριμένους τύπους εντολών
- Μεταγλωττιστής
  - Καθορίζει το IC
  - Καθορίζει το CPI

Χρόνος εκτέλεσης ΚΜΕ (χρήστη)

- Χρόνος Εκτέλεσης ΚΜΕ για ένα πρόγραμμα

  ExecTime = IC * CPI * CC

- Τι μπορεί να κάνει ο σχεδιαστής ΚΜΕ για να βελτιώσει την απόδοση;
  - **Να μειώσει τον κύκλο ρολογιού (CC)**
  - **Να μειώσει τον αριθμό κύκλων ανά εντολή (CPI)**
  - Οι εντολές παραμένουν ως έχουν

Παράδειγμα

Τύπος εντολής	A	B	C
CPI	1	2	3

Ακολουθία κώδικα	A	B	C
1	2	1	2
2	4	1	1

- Επιλογή μεταξύ 2 ακολουθιών εντολών
  - **Ποια ακολουθία εκτελεί τις περισσότερες εντολές;**
  - **Ποια είναι ταχύτερη;**
  - **Ποιο το μέσο CPI σε κάθε περίπτωση;**

  [Patterson-Hennessy "Computer Organization and Design", 3rd ed]

Μετροπρογράμματα

- Benchmarks
  - Για τη μέτρηση της απόδοσης
  - Και τη σύγκριση μεταξύ υπολογιστών
  - Θα πρέπει να αντιπροσωπεύουν τις πραγματικές εφαρμογές
  - Υπό ρεαλιστικές συνθήκες εκτέλεσης και δεδομένα εισόδου
  - Χωρίς "εσωτερικές" ειδικές βελτιστοποιήσεις
  - Δυνατότητα επανάληψης μέτρησης
  - Διαφορετικά για ανόμοιες κλάσεις υπολογιστών
    - PCs, servers, embedded systems…

Ο νόμος του Amdahl

;

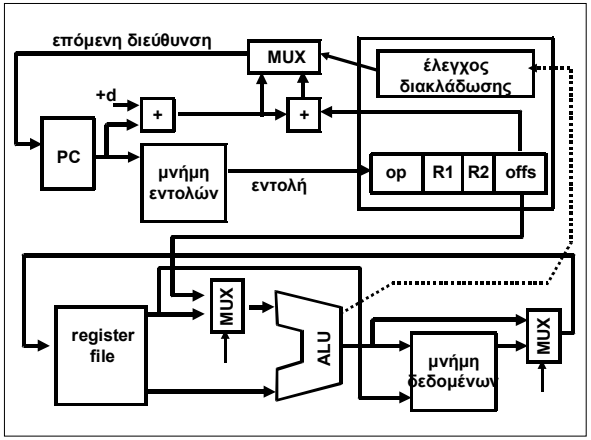
Ένα πρόγραμμα τρέχει για 100 sec σε έναν υπολογιστή και εκτελεί πολλαπλασιασμούς για 80 sec. Πόσο πρέπει να βελτιώσω τη ταχύτητα του πολλαπλασιασμού για να πενταπλασιάσω την απόδοση;

- "Η βελτίωση της συνολικής απόδοσης ενός συστήματος μέσω της εισαγωγής ενός νέου χαρακτηριστικού, περιορίζεται από το βαθμό χρήσης αυτού του νέου χαρακτηριστικού"
- Οι περισσότερο χρησιμοποιούμενες περιπτώσεις πρέπει να είναι γρήγορες!
- Όχι βελτιστοποίηση των σπάνιων περιπτώσεων!

ΚΜΕ ενός κύκλου (single-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Κάθε εντολή ολοκληρώνεται σε έναν κύκλο ρολογιού



ΚΜΕ ενός κύκλου (single-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

- CPI = 1
  - Σε κάθε έναν κύκλο ρολογιού ολοκληρώνεται μια εντολή ή
  - κάθε εντολή απαιτεί έναν κύκλο ρολογιού
- Πόσο πρέπει να είναι το CC;
  - Ίσο με τη διάρκεια της μεγαλύτερης λειτουργίας!
  - Μη αποδοτικό σχήμα
  - Όλες οι εντολές δεν απαιτούν τον ίδιο χρόνο

Παράδειγμα

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Εντολή	IF	ID	EX	DM	WB	Σύνολο
Αριθμητική	200	50	100	0	50	**400 ps**
Διακλάδωση	200	50	100	0	0	**350 ps**
Ανάγνωση μνήμης	200	50	100	200	50	**600 ps**
Εγγραφή μνήμης	200	50	100	200	0	**550 ps**

- CC πρέπει να είναι 600 ps (single cycle CPU)
  - Αν ήταν δυνατή η χρήση με μεταβλητό CC (πρακτικά αδύνατο!)
  - Ποια η βελτίωση της απόδοσης;
  - 25% ανάγνωση, 10% εγγραφή, 45% αριθμητικές, 20% διακλάδωσης

ΚΜΕ πολλαπλών κύκλων (multi-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

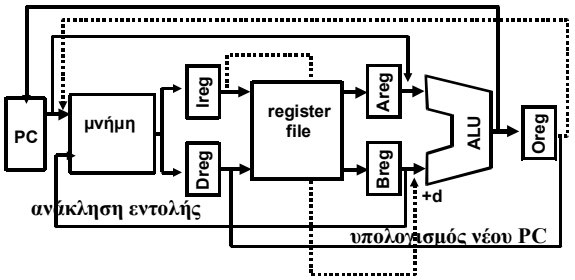
- CPI > 1
  - Κάθε εντολή χωρίζεται σε έναν μεταβλητό αριθμό βημάτων
  - Κάθε βήμα απαιτεί έναν κύκλο ρολογιού
- Πόσο πρέπει να είναι το CC;
  - Ίσο με τη διάρκεια του μεγαλύτερου βήματος
    - Τα βήματα επιλέγονται με μικρή διάρκεια
  - Μέρη της ΚΜΕ μπορούν να χρησιμοποιηθούν για περισσότερες από μία φορές κατά την εκτέλεση μιας εντολής
    - Αρκεί να ανήκουν σε διαφορετικό βήμα
    - Καταχωρητές για τη συγκράτηση αποτελεσμάτων μεταξύ βημάτων

Παράδειγμα: Εντολή load (βήμα IF)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

!

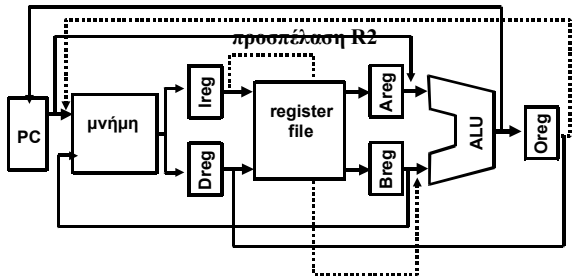
Η εντολή load απαιτεί τα περισσότερα βήματα για να εκτελεστεί!



R1 ← mem[R2 + offset]

Παράδειγμα: Εντολή load (βήμα ID)

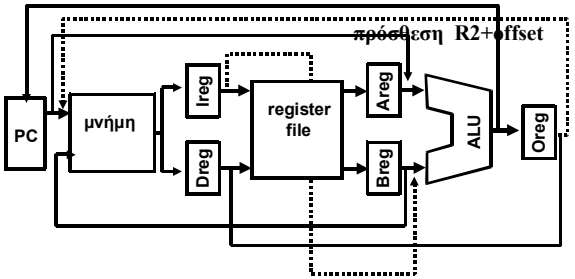
- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



R1 ← mem[R2 + offset]

Παράδειγμα: Εντολή load (βήμα EX)

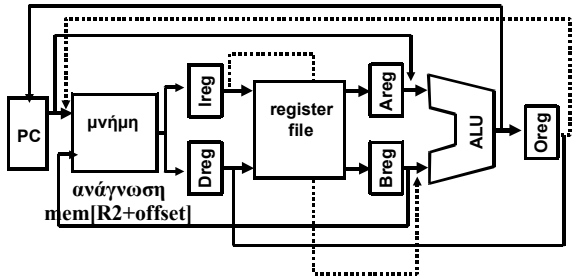
- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



R1 ← mem[R2 + offset]

Παράδειγμα: Εντολή load (βήμα DM)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



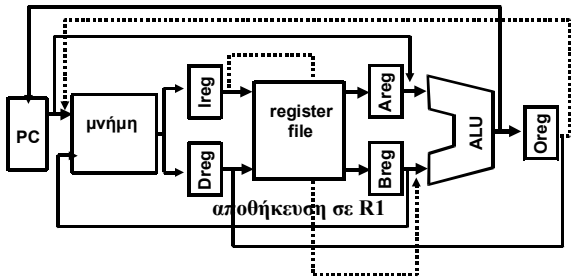
R1 ← mem[R2 + offset]

Παράδειγμα: Εντολή load (βήμα WB)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

;

Σε κάθε βήμα, μερικά τμήματα μένουν ανενεργά. Πώς θα μπορούσαμε να τα εκμεταλλευτούμε;



R1 ← mem[R2 + offset]

Μονάδα Ελέγχου ΚΜΕ πολλαπλών κύκλων

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

- Πολυπλοκότητα σημάτων ελέγχου
  - Σε κάθε βήμα εκτέλεσης της εντολής
  - Ανάλογα με το είδος της εντολής
    - Διατήρηση προηγούμενης κατάστασης
- Μέθοδοι υλοποίησης
  - Αυτόματα πεπερασμένων καταστάσεων
    - Ακολουθιακά λογικά κυκλώματα
    - Παραγωγή σημάτων ελέγχου ανάλογα με εισόδους και τρέχουσα κατάσταση
  - Μικροπρόγραμμα
    - Καθορισμός σημάτων μέσω μικροεντολών
    - Εσωτερικά στην ΚΜΕ (δεν αλλάζουν μετά τη σχεδίαση)
    - Για υλοποίηση σύνθετων εντολών με πολλά βήματα και πολλαπλά περάσματα από το datapath

Απόδοση ΚΜΕ πολλαπλών κύκλων

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

- Πλεονεκτήματα
  - Δεν απαιτείται ο μέγιστος χρόνος για το CC
  - Μέρη της ΚΜΕ χρησιμοποιούνται με πολλαπλό τρόπο κατά την εκτέλεση μιας εντολής
- Μειονεκτήματα
  - Η μονάδα ελέγχου γίνεται πολυπλοκότερη
    - Η πολυπλοκότητα πιθανόν να ακυρώνει τα πλεονεκτήματα!
- Σήμερα
  - Οι ΚΜΕ υψηλής απόδοσης χρησιμοποιούν πρόσθετες τεχνικές παραλληλισμού σε επίπεδο εντολών (instruction level parallelism – ILP)
    - (στο επόμενο μάθημα...)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Παραλληλισμός σε επίπεδο εντολών

(Pipelining και άλλες τεχνικές αύξησης απόδοσης)

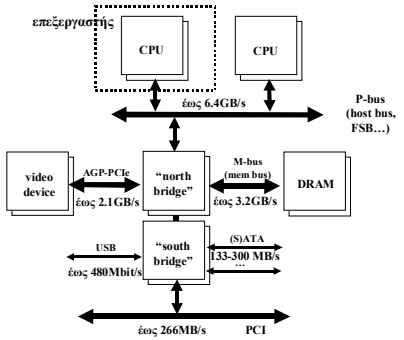
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Κεντρική Μονάδα Επεξεργασίας

- Απόδοση ΚΜΕ

- Επεξεργαστής ή...
- Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)



Επανάληψη: Απόδοση ΚΜΕ

- Απόδοση ΚΜΕ

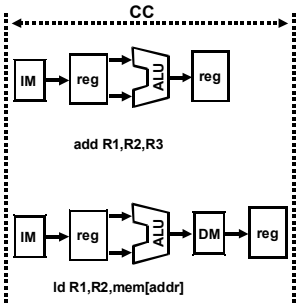
- Υπολογιστικό σύστημα
  - Η απόδοση εξαρτάται από όλα τα επιμέρους τμήματά του
    - Υλικό και λογισμικό
- Κεντρική Μονάδα Επεξεργασίας
  - Πόσο γρήγορα εκτελείται ένα πρόγραμμα;

ExecTime = IC * CPI * CC

ΚΜΕ ενός κύκλου (single-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Κάθε εντολή ολοκληρώνεται σε έναν κύκλο ρολογιού
CPI = 1
CC = διάρκεια αργότερης εντολής



Παράδειγμα

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Εντολή	IF	ID	EX	DM	WB	Σύνολο
Αριθμητική	200	50	100	0	50	400 ps
Διακλάδωση	200	50	100	0	0	350 ps
Ανάγνωση μνήμης	200	50	100	200	50	600 ps
Εγγραφή μνήμης	200	50	100	200	0	550 ps

- CC πρέπει να είναι 600 ps
  - Αν ήταν δυνατή η χρήση με μεταβλητό CC (πρακτικά αδύνατο!)
  - Ποια η βελτίωση της απόδοσης;
  - 25% ανάγνωση, 10% εγγραφή, 45% αριθμητικές, 20% διακλάδωσης

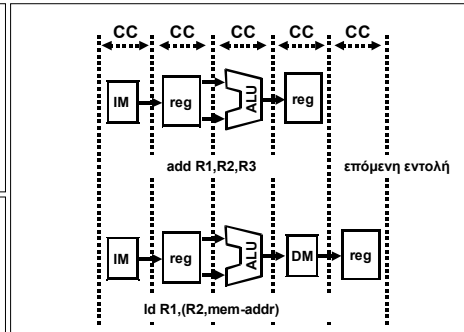
ΚΜΕ πολλαπλών κύκλων (multi-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

Κάθε εντολή ολοκληρώνεται σε μεταβλητό αριθμό κύκλων ρολογιού

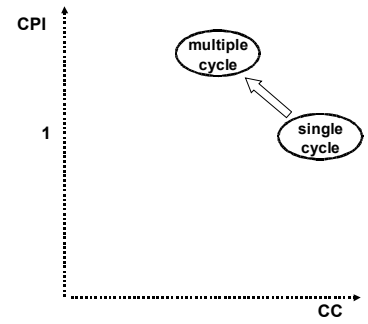
CPI > 1

CC = διάρκεια αργότερου σταδίου



Απόδοση ΚΜΕ πολλαπλών κύκλων

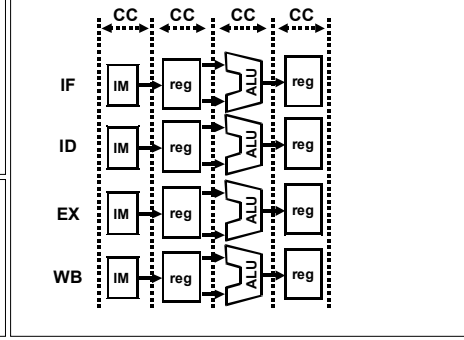
- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



Πιθανότητα περαιτέρω βελτίωσης

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

Σε κάθε κύκλο, ένα μέρος της ΚΜΕ δεν χρησιμοποιείται!



Παραλληλισμός σε επίπεδο εντολών

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP

- Instruction Level Parallelism (ILP)
  - Παράλληλη (ταυτόχρονη) εκτέλεση
  - Μεταξύ εντολών
- Pipelining
  - Επικάλυψη εκτέλεσης πολλαπλών εντολών
    - Την ίδια στιγμή
    - Βαθμίδες pipeline
    - Ρύθμιση του clock cycle
  - Βασικό στοιχείο αύξησης απόδοσης σύγχρονων επεξεργαστών
  - Στόχος
    - **Να μειωθεί το CC, αλλά ταυτόχρονα**
    - **το CPI να παραμείνει 1**

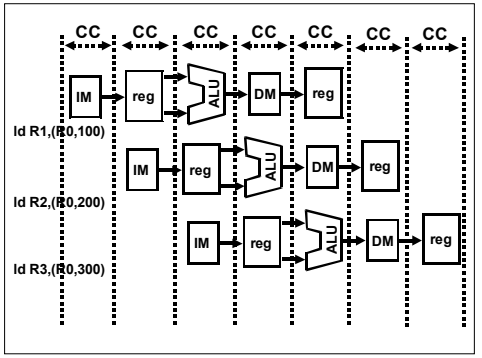
Αρχές τεχνικής pipelining

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

;

Ποιος ο χρόνος εκτέλεσης μιας εντολής;

Κάθε πότε ολοκληρώνεται μια εντολή;



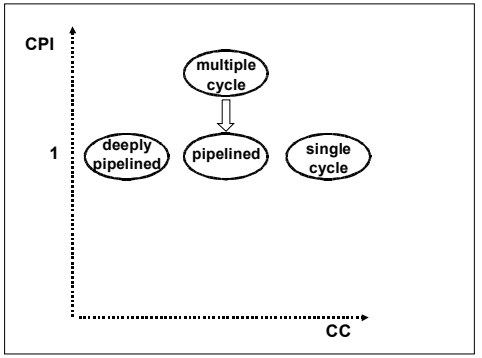
Απόδοση τεχνικής pipelining

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

- Χρόνος ολοκλήρωσης μιας εντολής
  - (latency)
  - Δεν αλλάζει!
- Ρυθμός ολοκλήρωσης εντολών
  - Σε κάθε έναν κύκλο ρολογιού
    - Ολοκληρώνεται μια εντολή
    - Μια νέα ξεκινά την εκτέλεσή της
  - Η ιδανική περίπτωση...
    - Απόδοση συστήματος μνήμης;
    - Χρήση ίδιων βαθμίδων pipeline;
    - Αλληλεξαρτήσεις δεδομένων εντολών;
    - Διακλαδώσεις;

Απόδοση ιδανικού pipeline

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining



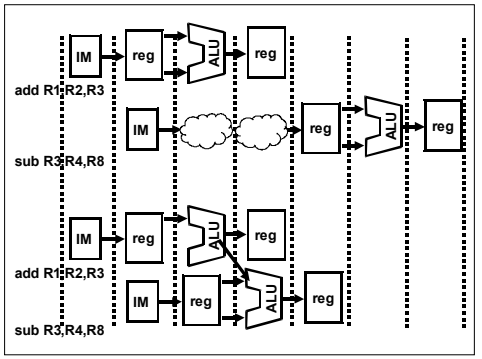
Δυσκολίες στην εφαρμογή του pipelining

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

- Αδυναμία χρήσης μιας βαθμίδας
  - Structural hazard
  - Η βαθμίδα χρησιμοποιείται από άλλη εντολή
  - Αποφεύγεται με σωστή σχεδίαση
- Αλληλεξαρτήσεις δεδομένων
  - Data hazard
  - Τα αναγκαία δεδομένα για την εκτέλεση μιας εντολής δεν είναι διαθέσιμα
    - Υπολογίζονται από προηγούμενη εντολή
  - Αντιμετωπίζεται με πρόσθετο υλικό
    - Forwarding - bypassing

Forwarding - bypassing

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining



!

Pipeline **bubbles**

Δεν αποφεύγονται πάντοτε!

Δυσκολίες στην εφαρμογή του pipelining

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

- Διακλαδώσεις
  - Control (branch) hazard
  - Η ροή της εκτέλεσης αλλάζει
    - Οι εντολές που έχουν μπει στο pipeline δεν είναι οι σωστές!
  - Πρόβλεψη διακλαδώσεων
    - Δυναμικές μέθοδοι (διατήρηση ιστορίας προηγούμενων διακλαδώσεων)
    - Συσχέτιση με αποφάσεις για άλλες διακλαδώσεις
    - >90% σωστές προβλέψεις
  - Σε περίπτωση λάθους πρόβλεψης
    - Απόρριψη όλων των εντολών στο pipeline
    - Χωρίς να λαμβάνονται υπ'όψη τα αποτελέσματά τους

Pipelining και διακοπές-σφάλματα

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

- Interrupts - Exceptions
  - Ποια εντολή στο pipeline προκάλεσε τη διακοπή;
  - Διακοπή εκτέλεσης εντολής
  - Ολοκλήρωση προηγούμενων εντολών στο pipeline
  - Άδειασμα pipeline (flush) από τυχόν επόμενες εντολές
  - Μετάβαση σε προκαθορισμένη διεύθυνση
    - Λειτουργικό Σύστημα

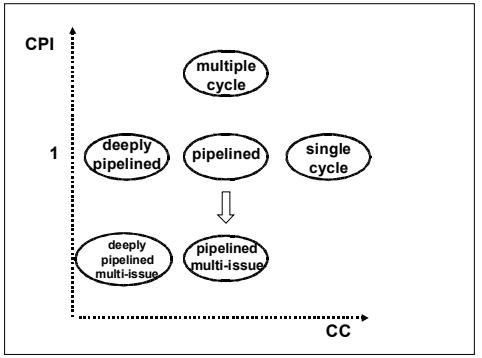
Ακόμα μεγαλύτερος βαθμός ILP

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

- Ο στόχος: CPI < 1
  - Η αλλιώς: ολοκλήρωση πολλών εντολών ανά κύκλο ρολογιού
    - Πολλαπλές βαθμίδες εκτέλεσης
    - Παράλληλο υλικό (hardware)
    - Σε κάθε κύκλο ξεκινά η εκτέλεση **πολλαπλών** νέων εντολών
  - Πολλαπλή εκτέλεση (multiple-issue)
  - Στατική πολλαπλή εκτέλεση
    - Ο μεταγλωττιστής αποφασίζει κατά κύριο λόγο ποιες εντολές θα εκτελεστούν ταυτόχρονα
  - Δυναμική πολλαπλή εκτέλεση
    - Οι αποφάσεις εκτέλεσης λαμβάνονται από την ΚΜΕ

Απόδοση multiple-issue επεξεργαστών

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP



Εικαζόμενη εκτέλεση

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

- Speculative Execution
  - Κλειδί για την αύξηση του βαθμού παραλληλίας σε επίπεδο εντολών
    - Πολλαπλές βαθμίδες εκτέλεσης
  - Πρόβλεψη αποτελεσμάτων για την επίτρεψη εκτέλεσης επόμενων εντολών με αλληλεξαρτήσεις
    - Από τον μεταγλωττιστή ή τον επεξεργαστή
    - Διόρθωση λάθους πρόβλεψης
    - Προσωρινή αποθήκευση έως ότου τα δεδομένα να μην είναι πλέον εικαζόμενα
  - Επηρεάζει απόδοση επεξεργαστή

Είδη multiple-issue επεξεργαστών

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

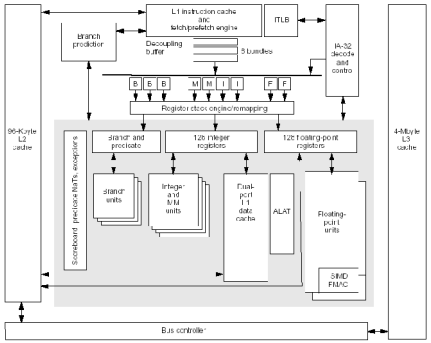
- Επεξεργαστές VLIW
  - Very Long Instruction Word
  - Πακέτα πολλαπλών εντολών
    - Παράλληλη εκτέλεση
    - Δεν είναι δυνατοί όλοι οι συνδυασμοί εντολών στο ίδιο πακέτο
  - Ο μεταγλωττιστής συγκροτεί τα πακέτα εντολών
  - Ελέγχει σε (μεγάλο βαθμό)
    - Αλληλεξαρτήσεις – αλληλουχίες εντολών
    - Πρόβλεψη διακλαδώσεων
  - Intel IA-64 (Itanium), Trimedia CPUs…

Intel Itanium processor

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

i

Η Intel ονομάζει την αρχιτεκτονική αυτή **EPIC** (explicitly parallel instruction computer)



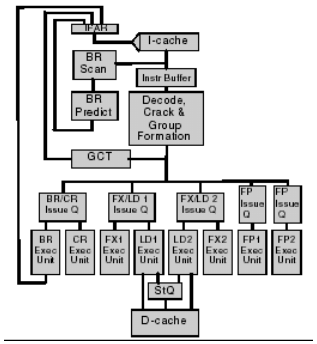
Είδη multiple-issue επεξεργαστών

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

- Επεξεργαστές superscalar
  - Ο επεξεργαστής ελέγχει σε κάθε κύκλο αν μπορεί να ξεκινήσει την εκτέλεση μιας η περισσότερων εντολών
    - Πολλαπλές μονάδες εκτέλεσης
    - Δυναμική απόφαση ανάθεσης εντολών σε μονάδες
    - Εκτέλεση εκτός σειράς (out-of-order execution)
  - Ο κώδικας εκτελείται πάντα σωστά
    - Ανεξάρτητα από μεταγλώττιση
  - Intel IA-32 (Pentium), PowerPC…

IBM Power4 processor

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Τεχνολογίες Κύριας Μνήμης

(και η ανάγκη για χρήση ιεραρχιών μνήμης)

http://www.ionio.gr/~mistral/tp/comparch/

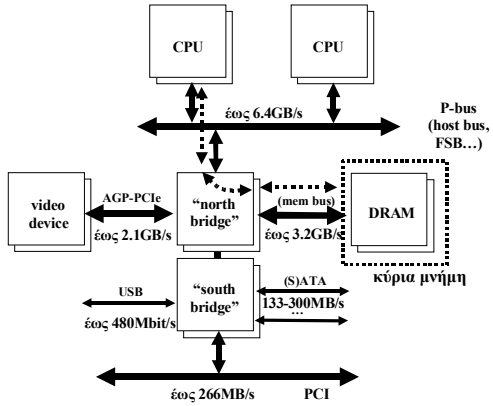
Μ.Στεφανιδάκης

Κύρια Μνήμη

• Κύρια Μνήμη

i

Στα σύγχρονα συστήματα η κύρια μνήμη δεν συνδέεται απευθείας με τον επεξεργαστή



Τεχνολογίες Κύριας Μνήμης

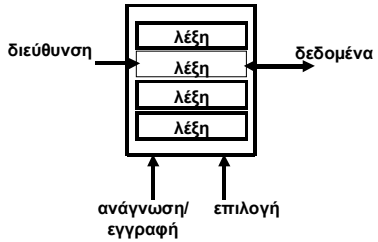
• Κύρια Μνήμη

- Στους πρώτους υπολογιστές
  - Ιστορικά, η κατασκευή κύριας μνήμης ήταν πολύ πιο δύσκολη από την κατασκευή των πρώτων υπολογιστών!
- Αρχικές τεχνολογίες
  - Flip-flop με λυχνίες κενού
  - Γραμμές καθυστέρησης υδραργύρου
  - Μαγνητικές μνήμες (core memories - 1950)
    - Η πρώτη αξιόπιστη και σχετικά φθηνή τεχνολογία
    - Κυριάρχησε για 20 περίπου χρόνια
- Ημιαγωγικές μνήμες (Intel – 1970)
  - 1Kbit DRAM - "core killer"

Το μοντέλο της Μνήμης Τυχαίας Προσπέλασης

• Κύρια Μνήμη
• RAM

•Η λέξη είναι η μικρότερη προσπελάσιμη ομάδα bits.
•Το εύρος των μεταφερόμενων δεδομένων σε κάθε ανάγνωση ή εγγραφή ισούται με το εύρος της λέξης μνήμης



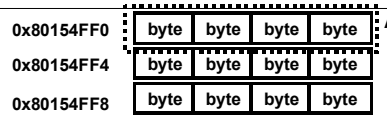
- Random Access Memory (RAM)
  - Λέξη μνήμης (**word**) με εύρος **M** bits
  - Διεύθυνση (**address**) επιλογής λέξης, **N** bits
  - Μέγεθος (χωρητικότητα) μνήμης 2^N**xM** bits

Διευθυνσιοδότηση μνήμης RAM

- Κύρια Μνήμη
- RAM

;

Πώς προσπελαύνω bytes όταν η λέξη μνήμης έχει πολλαπλάσιο εύρος;



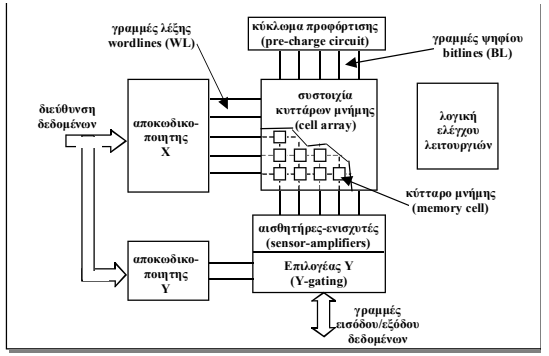
- Byte addressing
  - Οι διευθύνσεις μνήμης που παράγει ο επεξεργαστής αυξάνονται ανά byte
    - **Ακόμα κι όταν η λέξη μνήμης έχει πολλαπλάσιο εύρος!**
    - **Μετάφραση από ελεγκτή μνήμης**
  - Ευθυγραμμισμένη προσπέλαση:
    - **Αντικείμενο των s bytes στη διεύθυνση A ⇒ *A mod s = 0***
- Εναλλακτικά: word addressing
  - Υπερυπολογιστές – εδώ η προσπέλαση ανά byte είναι σπάνια

Οργάνωση Μνήμης Τυχαίας Προσπέλασης

- Κύρια Μνήμη
- RAM

i

Οι μεγαλύτερες μνήμες RAM διαθέτουν πολλαπλές συστοιχίες κυττάρων μνήμης



Ταχύτητα Προσπέλασης RAM

- Κύρια Μνήμη
- RAM

- Access Time (χρόνος προσπέλασης)
  - Ο απαιτούμενος χρόνος για την ολοκλήρωση μιας αίτησης προς τη μνήμη RAM
    - Ανάγνωση - Εγγραφή
- Cycle Time (χρόνος κύκλου προσπέλασης)
  - Ο ελάχιστος απαιτούμενος χρόνος μεταξύ διαδοχικών αιτήσεων προς τη μνήμη RAM
    - Πρόβλεψη ενδιάμεσων λειτουργιών
- Ταχύτητα διαφορετική για ανάγνωση - εγγραφή!

Τύποι Μνήμης Τυχαίας Προσπέλασης

- Κύρια Μνήμη
- RAM
- SRAM

Ο χρόνος προσπέλασης μιας μνήμης SRAM βρίσκεται μεταξύ 0,5 και 5 ns

Το κόστος ενός GB SRAM είναι $4000-$10.000 [Patterson, Hennessy]

- Στατική Μνήμη RAM (SRAM)
  - Κάθε bit αποθηκεύεται σε κύτταρο 6 τρανζίστορ
    - **Ανάλογο ενός flip-flop**
  - Διατήρηση όσο υπάρχει τροφοδοσία της μνήμης
  - Η προσπέλαση είναι γρήγορη
  - Μεγαλύτερο κόστος
  - Πολυπλοκότερο κύκλωμα
    - **Δεν επιτρέπει μεγάλη ολοκλήρωση**
    - **Μικρής χωρητικότητας**
  - Μεγαλύτερη κατανάλωση ενέργειας
    - **Αλλά: ελάχιστη κατανάλωση σε standby mode**
  - Χρησιμοποιείται στις κρυφές μνήμες (caches)

Τύποι Μνήμης Τυχαίας Προσπέλασης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM

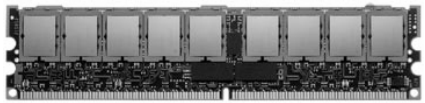
Ο χρόνος προσπέλασης μιας μνήμης DRAM βρίσκεται μεταξύ 50 και 70 ns

Το κόστος ενός GB DRAM είναι $100-$200 [Patterson, Hennessy]

- Δυναμική Μνήμη RAM (DRAM)
  - Κάθε bit αποθηκεύεται ως φορτίο
    - **Ανάλογο ενός πυκνωτή**
  - Διατήρηση μόνο με συχνή **ανανέωση** του φορτίου
    - Κάθε 16 έως 128 ms (5% συνολικού χρόνου)
  - Απλούστερο κύκλωμα – μεγάλη ολοκλήρωση
    - **Πολύ μεγάλες χωρητικότητες (1Gbit/ολοκληρωμένο κυκλ.)**
  - Η προσπέλαση είναι αργή
    - **Αρχιτεκτονικές βελτιώσεις για αύξηση ρυθμού μεταφοράς δεδομένων**
  - Χρησιμοποιείται για τη συγκρότηση της κύριας μνήμης όλων των σύγχρονων υπολογιστικών συστημάτων

Τμήματα μνήμης DRAM

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM

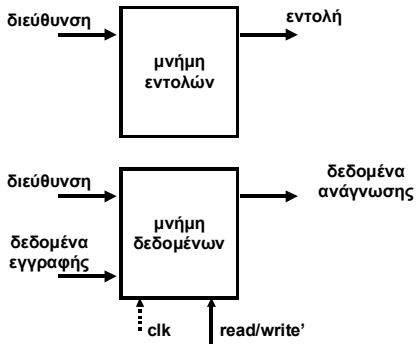


- 64 έως 72 bits δεδομένων (έως 4GB)
- Σύγχρονες μεταφορές στις 2 ακμές ρολογιού
- Σήματα ανίχνευσης και αναγνώρισης
- Διατάξεις βελτίωσης ηλεκτρικών χαρακτηριστικών σημάτων
- Ρυθμός μεταφοράς > 3.2GB/s

Η "ιδανική μνήμη"

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

Πόσο απέχει η ιδανική εικόνα από την πραγματικότητα;



- **Ολοκλήρωση ανάγνωσης-εγγραφής σε έναν κύκλο ρολογιού...**

Η πραγματική εικόνα

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

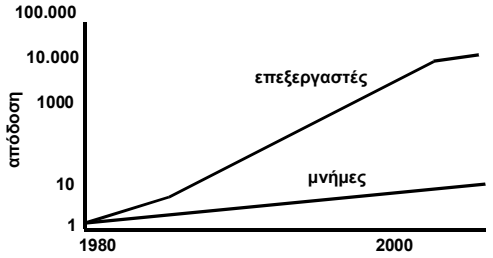
Η ιδανική μνήμη είναι πρακτικά αδύνατο να υλοποιηθεί. Ποια η πιθανή λύση;

- Ένας σύγχρονος superscalar επεξεργαστής
  - με ρολόι 3 GHz
  - και έναρξη εκτέλεσης έως και 8 εντολών ανά κύκλο
  - απαιτεί από τη μνήμη 24G εντολές/sec!
- Η "ιδανική μνήμη" θα έπρεπε να είναι
  - Πολύ γρήγορη
  - Πολύ φθηνή
  - Με πολύ μεγάλη χωρητικότητα

Το χάσμα απόδοσης μεταξύ επεξεργαστή-μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

Οι μνήμες ακολουθούν τον νόμο του Moore στην αύξηση της χωρητικότητάς τους, όχι όμως και στην απόδοση



- Επεξεργαστές: αύξηση απόδοσης 35%-55% /έτος
- Μνήμες: αύξηση απόδοσης 7% /έτος

[Patterson-Hennessy]

Η αρχή της τοπικότητας

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

"ένα πρόγραμμα εκτελεί το 90% των εντολών του μέσα στο 10% του κώδικά του"

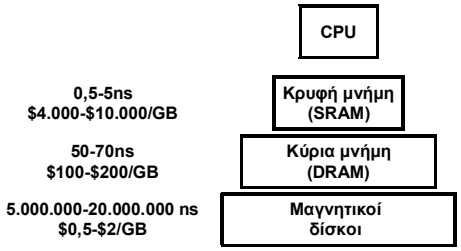
- Χρονική Τοπικότητα
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστεί ξανά στο άμεσο μέλλον
  - Π.χ. για εντολές ενός βρόχου (loop)
- Χωρική Τοπικότητα
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστούν και οι γειτονικές θέσεις στο άμεσο μέλλον
  - Εντολές προγραμμάτων
  - Δεδομένα σε πίνακες κλπ

Ιεραρχίες Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

Και οι δικτυακές τοποθεσίες μπορούν να θεωρηθούν μέρος της ιεραρχίας μνήμης (το χαμηλότερο)

- Πολλαπλά επίπεδα μνήμης
  - Διαφορετικής τεχνολογίας
  - Με διαφορετική ταχύτητα και μέγεθος
  - Γρηγορότερη μνήμη κοντά στον επεξεργαστή

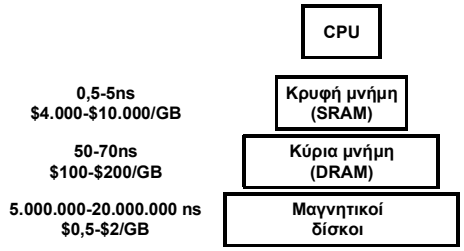


Σκοπός της Ιεραρχίας Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

Για να επιτύχει τον σκοπό της η ιεραρχία μνήμης εκμεταλλεύεται την αρχή της **τοπικότητας**

- Προσέγγιση της ιδανικής μνήμης
  - Ο επεξεργαστής να βλέπει "μνήμη"
  - Με την ταχύτητα του υψηλότερου επιπέδου
  - Και το μέγεθος του χαμηλότερου



Αποθήκευση δεδομένων στην Ιεραρχία Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

- **Αποθήκευση δεδομένων**
  - Τα υψηλότερα επίπεδα είναι υποσύνολα των χαμηλότερων
  - Όλα τα δεδομένα αποθηκεύονται τελικά στο χαμηλότερο επίπεδο
- **Μεταφορά δεδομένων**
  - Αντιγραφή από επίπεδο σε επίπεδο
  - Το ελάχιστο σύνολο δεδομένων που μεταφέρεται μεταξύ δύο επιπέδων ονομάζεται μπλοκ
    - Πολλαπλά bytes

Αναζήτηση δεδομένων στην Ιεραρχία Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

- **Αναζήτηση δεδομένων**
  - Ο επεξεργαστής ζητά πάντοτε τα δεδομένα από το κοντινότερο σε αυτόν επίπεδο
  - Τα δεδομένα υπάρχουν στο επίπεδο αυτό: hit
  - Τα δεδομένα δεν βρίσκονται στο επίπεδο αυτό: miss
    - Η αίτηση προωθείται στο επόμενο (χαμηλότερο) επίπεδο
    - Και το μπλοκ που περιέχει τα δεδομένα αντιγράφεται στο ανώτερο επίπεδο

Μετρήσεις απόδοσης στην Ιεραρχία Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

- **Hit Rate**
  - Ποσοστό προσπελάσεων μνήμης, όπου τα δεδομένα βρίσκονται στο ανώτερο επίπεδο
- **Miss Rate**
  - Ποσοστό προσπελάσεων μνήμης, όπου τα δεδομένα δεν βρίσκονται στο ανώτερο επίπεδο
    - (1-hit rate)
- **Hit Time**
  - Ο χρόνος για την προσπέλαση δεδομένων σε hit
- **Miss Penalty**
  - Ο χρόνος για την προσπέλαση, μεταφορά και τοποθέτηση των δεδομένων miss από το χαμηλότερο στο ανώτερο επίπεδο

Εισαγωγή στις κρυφές μνήμες (caches)

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης
- Κρυφές Μνήμες

- **Κρυφή μνήμη**
  - Μεταξύ του επεξεργαστή και της κύριας μνήμης
  - Εμφάνιση στη δεκαετία του 60
  - Σήμερα δεν υπάρχει υπολογιστικό σύστημα χωρίς κρυφή μνήμη
- **Αποθήκευση δεδομένων στην κρυφή μνήμη**
  - Όχι ανά λέξη μνήμης ή ανά byte…
  - …αλλά ανά μπλοκ (64-512bits)
  - Μεταφορά δεδομένων από την κύρια προς την κρυφή μνήμη σε ριπές (bursts)
  - Το σύστημα κύριας μνήμης έχει βελτιστοποιηθεί αρχιτεκτονικά για αυτού του τύπου τις μεταφορές

Θέματα κρυφών μνημών

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης
- Κρυφές Μνήμες

- Πού αποθηκεύεται ένα μπλοκ στην κρυφή μνήμη;
- Πώς εντοπίζεται ένα μπλοκ στην κρυφή μνήμη;
- Ποιο μπλοκ θα αντικατασταθεί όταν χρειαστεί;
- Τι συμβαίνει στην εγγραφή νέων δεδομένων;
- Πώς υπολογίζεται η απόδοση της ιεραρχίας μνήμης;

(στο επόμενο μάθημα..)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Κρυφές Μνήμες

(οργάνωση, λειτουργία και απόδοση)

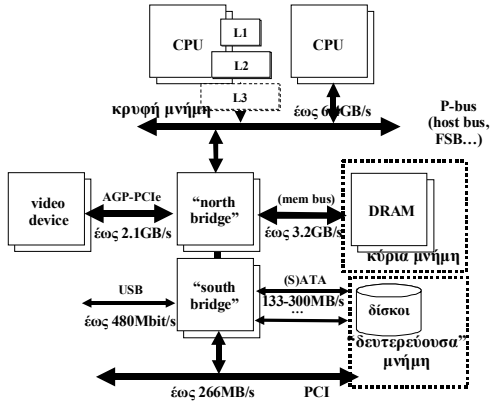
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης

Για βελτίωση της απόδοσης του συστήματος μνήμης (και του συνολικού υπολογιστικού συστήματος)

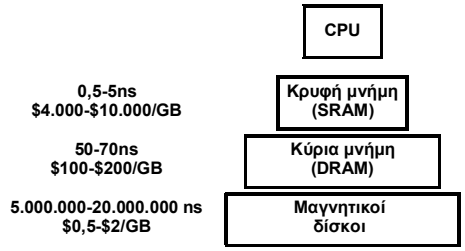


Σκοπός της Ιεραρχίας Μνήμης

- Ιεραρχία Μνήμης

Για να επιτύχει τον σκοπό της η ιεραρχία μνήμης εκμεταλλεύεται την αρχή της **τοπικότητας**

- Προσέγγιση της ιδανικής μνήμης
  - Ο επεξεργαστής να βλέπει "μνήμη"
  - Με την ταχύτητα του υψηλότερου επιπέδου
  - Και το μέγεθος του χαμηλότερου επιπέδου



Ιεραρχία μνήμης και τοπικότητα

- Ιεραρχία Μνήμης

"ένα πρόγραμμα εκτελεί το 90% των εντολών του μέσα στο 10% του κώδικά του"

- Χρονική Τοπικότητα
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστεί ξανά στο άμεσο μέλλον
    - Παράδειγμα: οι εντολές ενός βρόχου (loop)
- Εφαρμογή:
  - Δεδομένα – εντολές που βρίσκονται ήδη κοντύτερα στον επεξεργαστή (π.χ. στην κρυφή μνήμη) θα προσπελαστούν πολύ γρηγορότερα

Ιεραρχία μνήμης και τοπικότητα

- Ιεραρχία Μνήμης

"ένα πρόγραμμα εκτελεί το 90% των εντολών του μέσα στο 10% του κώδικά του"

- Χωρική Τοπικότητα
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστούν και οι γειτονικές θέσεις στο άμεσο μέλλον
    - Εντολές προγραμμάτων
    - Δεδομένα σε πίνακες κλπ
- Εφαρμογή:
  - Εάν προσπελαστεί μια θέση μνήμης, μεταφέρονται και οι διπλανές της λέξεις στη μνήμη του υψηλότερου επιπέδου
    - Μεταφορά σε μπλοκ (πολλαπλές λέξεις μνήμης)
  - Γρηγορότερη προσπέλαση

Κρυφές μνήμες

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

- Σημαντικό τμήμα στην ιεραρχία μνήμης
- Εξέλιξη συστημάτων κρυφής μνήμης
  - 1962: οι πρώτες ιεραρχίες μνήμης (Atlas)
    - Όχι όμως κρυφή μνήμη
  - 1965: η πρώτη περιγραφή κρυφής μνήμης (Wilkes)
    - Ο πρώτος υπολογιστής με κρυφή μνήμη (IBM 360/85)
  - 1968: η πρώτη χρησιμοποίηση του όρου "cache memory"
  - Μελέτες απόδοσης κρυφής μνήμης
  - Πολλαπλά επίπεδα κρυφής μνήμης (L1, L2, L3)
  - Βελτιωμένες αρχιτεκτονικές κρυφής μνήμης

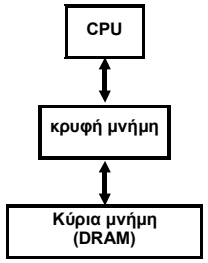
Απλό μοντέλο ιεραρχίας μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Οι αρχές λειτουργίας της απλής ιεραρχίας μπορούν να επεκταθούν σε πολλαπλά επίπεδα

Η διαχείριση της κρυφής μνήμης γίνεται από το υλικό

διαφανώς προς τις εφαρμογές



- Τα δεδομένα βρίσκονται αρχικά στην κύρια μνήμη
- Η κρυφή μνήμη περιέχει υποσύνολο των δεδομένων
- Μεταφορά μεταξύ επιπέδων μνήμης σε μπλοκ λέξεων

Αποθήκευση δεδομένων στην Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

- Αποθήκευση δεδομένων
  - Τα υψηλότερα επίπεδα είναι υποσύνολα των χαμηλότερων
  - Όλα τα δεδομένα αποθηκεύονται τελικά στο χαμηλότερο επίπεδο
- Μεταφορά δεδομένων
  - Αντιγραφή από επίπεδο σε επίπεδο
  - Το ελάχιστο σύνολο δεδομένων που μεταφέρεται μεταξύ δύο επιπέδων ονομάζεται μπλοκ
    - Πολλαπλά bytes

Αναζήτηση δεδομένων στην Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

- Αναζήτηση δεδομένων
  - Ο επεξεργαστής ζητά πάντοτε τα δεδομένα από το κοντινότερο σε αυτόν επίπεδο
  - Τα δεδομένα υπάρχουν στο επίπεδο αυτό: hit
  - Τα δεδομένα δεν βρίσκονται στο επίπεδο αυτό: miss
    - Η αίτηση προωθείται στο επόμενο (χαμηλότερο) επίπεδο
    - Όταν βρεθεί, το μπλοκ που περιέχει τα δεδομένα αντιγράφεται στο ανώτερο επίπεδο

Μπλοκ (γραμμές) κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

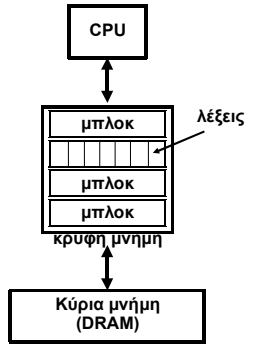
Οι σύγχρονοι επεξεργαστές διαθέτουν κρυφές μνήμες (L1) με μέγεθος μπλοκ έως 64 bytes

•Για την εκμετάλλευση της τοπικότητας

•Όταν πρέπει να μεταφερθεί μια λέξη, μεταφέρεται το μπλοκ που την περιέχει

•Το μέγεθος του μπλοκ είναι καθοριστικό για την απόδοση της ιεραρχίας μνήμης

•Το σύστημα κύριας μνήμης έχει βελτιστοποιηθεί αρχιτεκτονικά για μεταφορές μπλοκ



Τοποθέτηση ενός μπλοκ

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

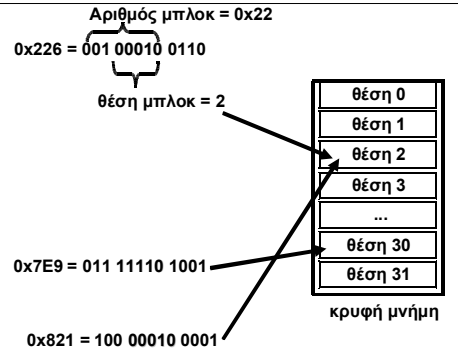
- Η κύρια μνήμη περιέχει πολύ περισσότερα μπλοκ από όσα χωρούν στην κρυφή μνήμη
  - Συνεπώς, στην ίδια θέση της κρυφής μνήμης πρέπει να τοποθετηθούν περισσότερα από ένα μπλοκ
    - Σύγκρουση μπλοκ;
- Πώς αποφασίζεται η θέση ενός μπλοκ στην κρυφή μνήμη;
  - Η απλή λύση: άμεση απεικόνιση (direct mapped caches)
  - Κάθε μπλοκ πηγαίνει σε μία μόνο θέση

  (αριθμός μπλοκ) mod (θέσεις στην κρυφή μνήμη)
    - Υπολογίζεται πολύ εύκολα αν οι θέσεις είναι δύναμη του 2

Άμεση απεικόνιση θέσης μπλοκ

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

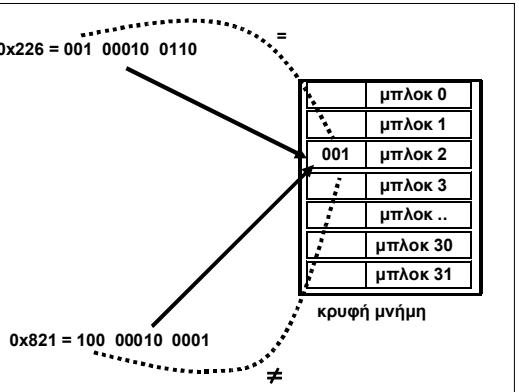
Παράδειγμα:
Block = 16 bytes
Cache = 32 blocks



Ποιο μπλοκ βρίσκεται σε κάθε θέση;

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

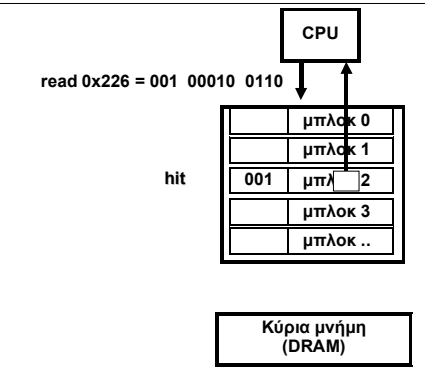
Είναι η θέση κατειλημμένη από κάποιο μπλοκ;
valid bit (V)



Ανάγνωση: Cache Hit

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Σε περίπτωση εύρεσης των δεδομένων στην κρυφή μνήμη, η ΚΜΕ μπορεί να τα λάβει ακόμα και σε 1 κύκλο

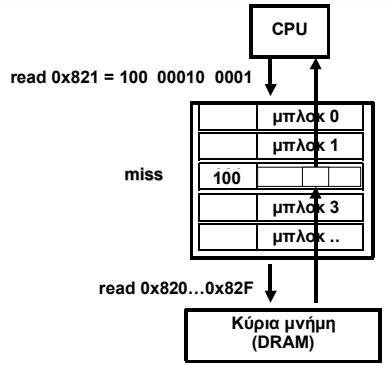


•**Αιτήσεις για ανάγνωση: εντολές και δεδομένα**

Ανάγνωση: Cache Miss

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Miss penalty:
ο χρόνος για την μεταφορά του μπλοκ από κύρια μνήμη και επιστροφή δεδομένων στον επεξεργαστή



•**Αιτήσεις για ανάγνωση: εντολές και δεδομένα**

Εγγραφή στην κρυφή μνήμη

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Συνοχή δεδομένων:
Πώς επηρεάζουν οι κρυφές μνήμες τη σχεδίαση πολυ-επεξεργαστικών συστημάτων;

- Μόνο για δεδομένα
- Write Hit – Ενημέρωση κρυφής μνήμης
  - Στη συνέχεια:
  - Είτε ενημερώνω αμέσως την κύρια μνήμη (write-through)
    - Επιβάρυνση σε κάθε εγγραφή (χρήση buffers)
  - Είτε μόνο όταν το μπλοκ πρέπει να φύγει από την κρυφή μνήμη (write-back)
    - Απαιτείται πολύπλοκη λογική για τον έλεγχο της σωστής αποθήκευσης των δεδομένων
- Write Miss
  - Πρέπει το μπλοκ να έρθει πρώτα στην κρυφή μνήμη

Τι δημιουργεί cache misses;

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Η πρώτη φορά προσπέλασης ενός μπλοκ
  - Μπλοκ που δεν βρέθηκαν ποτέ μέχρι τώρα στην κρυφή μνήμη
- Λόγω χωρητικότητας της κρυφής μνήμης
  - Η κρυφή μνήμη δεν χωράει όλα τα μπλοκ (ταυτόχρονα)
- Λόγω σύγκρουσης μπλοκ στην ίδια θέση
  - Μπλοκ που τοποθετούνται στην ίδια θέση στην κρυφή μνήμη, συναγωνίζονται για τη θέση αυτή
    - ανάλογα με τη μέθοδο τοποθέτησης
    - ακόμα κι αν μέρος της κρυφής μνήμης είναι ελεύθερο...

Χαρακτηριστικά απόδοσης κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Hit Rate
  - Ποσοστό προσπελάσεων μνήμης, όπου τα δεδομένα βρίσκονται στην κρυφή μνήμη
- Miss Rate
  - Ποσοστό προσπελάσεων μνήμης, όπου τα δεδομένα δεν βρίσκονται στην κρυφή μνήμη
    - (1-hit rate)
- Hit Time
  - Ο χρόνος για την προσπέλαση δεδομένων σε hit
- Miss Penalty
  - Ο χρόνος για την προσπέλαση, μεταφορά και τοποθέτηση των δεδομένων miss από την κύρια στην κρυφή μνήμη και στον επεξεργαστή

Το κόστος των cache misses

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Χαμένοι κύκλοι ρολογιού
  - Σε αναμονή για προσπέλαση κύριας μνήμης

  Κύκλοι Αναμονής =

  Προσπελάσεις μνήμης * Miss Rate * Miss Penalty
- Απλουστευμένο μοντέλο
  - Miss Rate ανά κατηγορίες εντολών
  - Διαφορετική συμπεριφορά για ανάγνωση-εγγραφή
  - Σύνθετη ανάλυση για εκτέλεση εκτός σειράς
- Βελτίωση της απόδοσης
  - Μείωση του miss rate
  - Μείωση του miss penalty

Τεχνικές μείωσης miss rate

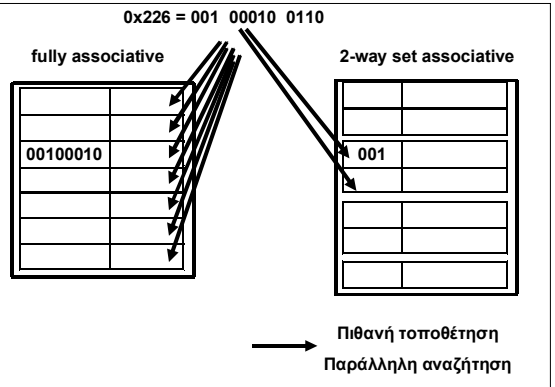
- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Αντιμετώπιση αιτιών που προκαλούν misses
- Αύξηση χωρητικότητας κρυφής μνήμης
  - Αλλά: μια μεγάλη κρυφή μνήμη μπορεί να είναι πιο αργή! (αύξηση hit time)
- Αύξηση του μεγέθους του μπλοκ
  - Προσπάθεια εκμετάλλευσης της τοπικότητας
  - Αλλά: αυξάνει το miss penalty
  - Πιθανόν να αυξάνει τελικά το miss rate, λόγω λιγότερων μπλοκ στην κρυφή μνήμη
- Ευέλικτες τεχνικές τοποθέτησης των μπλοκ
  - Ώστε να παραμένουν περισσότερο στην κρυφή μνήμη

Ευέλικτες τεχνικές τοποθέτησης μπλοκ

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Πιθανή η αύξηση του hit time λόγω πολύπλοκου κυκλώματος



•**Αντικατάσταση: least recently used (LRU)**

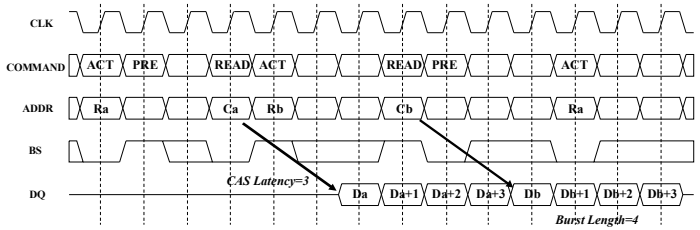
Τεχνικές μείωσης miss penalty

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

Οι σύγχρονοι επεξεργαστές έχουν τουλάχιστον L1 και L2 cache μέσα στο ίδιο το chip τους

- Μείωση των χρόνων μεταφοράς μπλοκ
- Βελτιστοποιήσεις στην επικοινωνία με την κύρια μνήμη
  - Έτσι ώστε ένα ολόκληρο μπλοκ να μεταφέρεται με τη μικρότερη δυνατή καθυστέρηση (bursts)
- Πολυεπίπεδες ιεραρχίες κρυφής μνήμης
  - Μείωση miss penalty πρώτου επιπέδου (L1)
  - L1: μικρότερο μέγεθος, μεγαλύτερη ταχύτητα
    - Μεγαλύτερο miss rate αλλά miss penalty μικρότερο
  - L2: μεγαλύτερο μέγεθος, μικρότερη ταχύτητα
    - Αργότερη αλλά δεν επηρεάζει hit time επεξεργαστή

Ανάγνωση από σύγχρονη κύρια μνήμη



Πολυεπίπεδη οργάνωση κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

Οι σύγχρονοι επεξεργαστές έχουν ξεχωριστή κρυφή μνήμη L1 για εντολές και δεδομένα. Ποια τα πλεονεκτήματα-μειονεκτήματα;

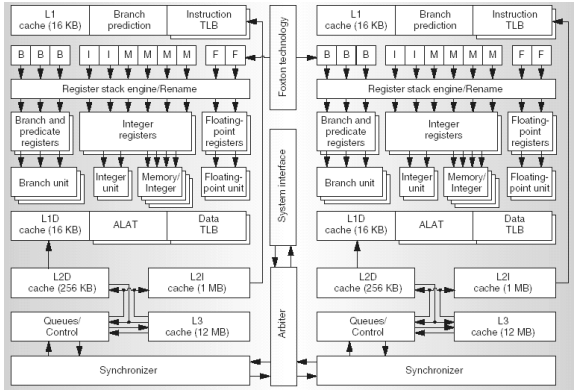
Τύπος	Μέγεθος	Χρόνος προσπέλασης	Ρυθμός μεταφοράς
L1	έως 64KB	4ns	50GB/s
L2	έως 8MB	10ns	25GB/s
L3	έως 64MB	20ns	10GB/s

- Παράδειγμα: Pentium4
  - L1 cache: 4 κύκλοι ρολογιού (pipelined: 1)
  - L2 cache: 20 κύκλοι ρολογιού
  - Προσπέλαση στη μνήμη: >100 κύκλοι ρολογιού

Intel "Montecito": Επίπεδα κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

Intel Montecito:
1,72 δις τρανζίστορ
2 επεξεργαστές Itanium2
1.8GHz @ 100W

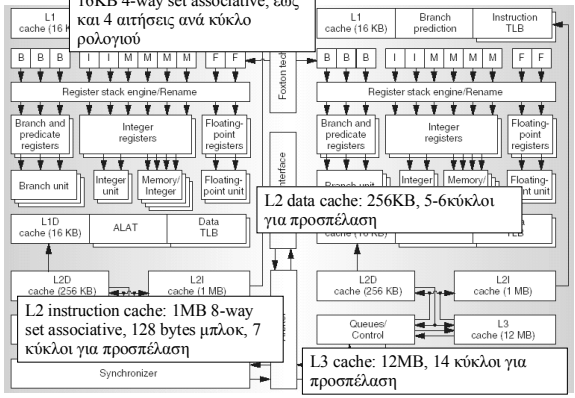


Intel "Montecito": Επίπεδα κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

Intel Montecito:
συνολικά 27MB κρυφή μνήμη μέσα στο chip

L1 instruction & data caches: 16KB 4-way set associative, έως και 4 αιτήσεις ανά κύκλο ρολογιού



L2 data cache: 256KB, 5-6κύκλοι για προσπέλαση

L2 instruction cache: 1MB 8-way set associative, 128 bytes μπλοκ, 7 κύκλοι για προσπέλαση

L3 cache: 12MB, 14 κύκλοι για προσπέλαση

Βελτιστοποίηση απόδοσης κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Αρχιτεκτονικές βελτιώσεις
  - Pipelining
  - Non-blocking – εξυπηρέτηση πολλαπλών αιτήσεων
  - Πολλαπλά επίπεδα κρυφής μνήμης στο chip του επεξεργαστή
- Ο ρόλος του λογισμικού (μεταγλωττιστές)
  - Αναδιοργάνωση προγραμμάτων για αύξηση της τοπικότητας (κυρίως στους βρόχους επανάληψης)
  - Prefetching: μετακίνηση δεδομένων στην κρυφή μνήμη πριν αυτά χρειαστούν στον επεξεργαστή!

Η απόδοση της κρυφής μνήμης συνοπτικά

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Καθοριστική για τα σύγχρονα υπολογιστικά συστήματα
- Μείωση του miss rate ή του miss penalty
  - Όμως: η συμπεριφορά της ιεραρχίας μνήμης επηρεάζεται από πολλούς παράγοντες!
- Η πραγματική συμπεριφορά
  - Είναι σύνθετη – απαιτούνται εξομοιώσεις πριν τη σχεδίαση νέων συστημάτων
  - Είναι διαφορετική ανά εφαρμογή – δεν υπάρχει ένα μόνο αντιπροσωπευτικό πρόγραμμα!
  - Είναι διαφορετική ανά υπολογιστικό σύστημα – desktop, server ή embedded

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Εικονική Μνήμη

(και ο ρόλος της στην ιεραρχία μνήμης)

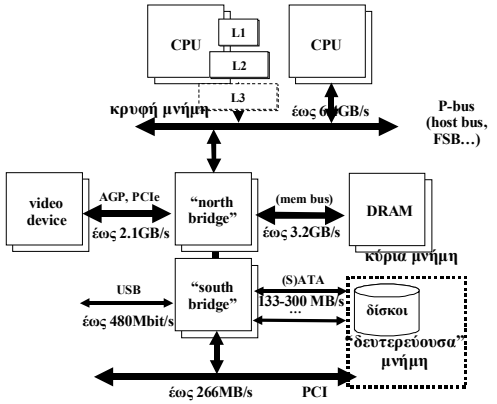
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης

"Δευτερεύουσα μνήμη": Το τελευταίο επίπεδο στην ιεραρχία εντός του υπολογιστικού συστήματος

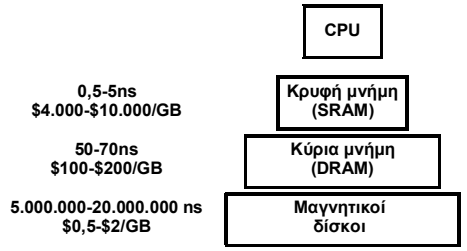


Επεκτείνοντας την Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης

Επέκταση ιεραρχίας εκτός του συστήματος: δικτυακές θέσεις αποθήκευσης

- Εννοιολογικά:
  - Κάθε υψηλότερο επίπεδο δρα ως "κρυφή μνήμη" για το αμέσως χαμηλότερο
    - Η **κύρια μνήμη** λειτουργεί ως "κρυφή μνήμη" των **μαγνητικών δίσκων**
    - Τα περιεχόμενα στην **κύρια μνήμη** είναι υποσύνολο εκείνων στους **δίσκους** του συστήματος



Εικονική μνήμη (virtual memory)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

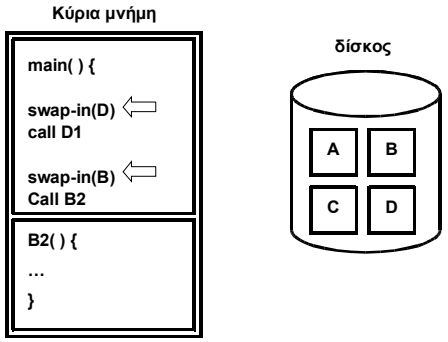
Το πρώτο σύστημα εικονικής μνήμης παρουσιάστηκε το 1962 (Atlas computer)

- Για ποιον λόγο εμφανίστηκε;
  - Στους πρώτους υπολογιστές το μέγεθος της κύριας μνήμης ήταν περιορισμένο
  - Ακόμα και στην περίπτωση του μονοπρογραμματισμού η κύρια μνήμη ήταν ανεπαρκής
    - Τεχνικές με overlays
  - Εμφάνιση ΛΣ με υποστήριξη πολυπρογραμματισμού: αδυναμία ταυτόχρονης διατήρησης πολλών προγραμμάτων στην κύρια μνήμη
  - Η λύση: εικονική μνήμη
    - Μέρος των δεδομένων βρίσκεται στους δίσκους του συστήματος
    - Μεταφορά στην κύρια μνήμη όταν χρειαστεί
    - Πιθανότατα αντικαθιστώντας άλλα τμήματα δεδομένων
    - Τα τελευταία μεταφέρονται πίσω στους δίσκους

Πριν την εικονική μνήμη: overlays

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η τεχνική των overlays απαιτούσε από τον προγραμματιστή να χειρίζεται τις λεπτομέρειες (π.χ διευθύνσεις!) φόρτωσης και κλήσης των υπο-προγραμμάτων!



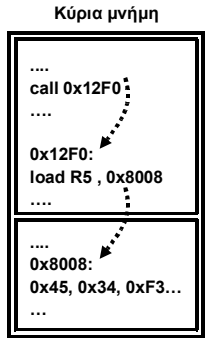
- Καταλληλότερο για στατικά δεδομένα
  - υποπρογράμματα

Χώρος διευθύνσεων προγράμματος

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Ποιες οι διευθύνσεις που δημιουργούσε ο μεταγλωττιστής; Τι συνέβαινε αν το πρόγραμμα δεν φορτωνόταν πάντοτε στον ίδιο χώρο μνήμης;

- Address Space
  - Εκτελούμενο πρόγραμμα στη μνήμη:
  - Διευθύνσεις κώδικα
    - Εντολές διακλάδωσης
  - Διευθύνσεις δεδομένων
    - Εντολές load-store
- Πριν την εικονική μνήμη:
  - Φυσικές διευθύνσεις
  - Άμεση αντιστοιχία με διευθύνσεις κύριας μνήμης

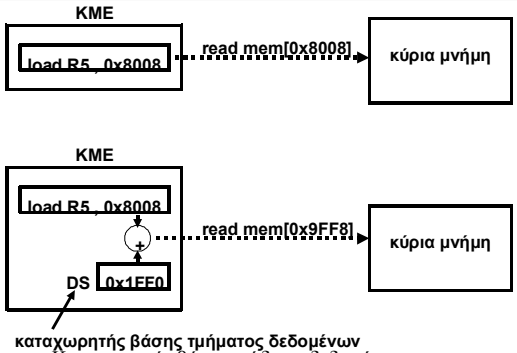


Πολυπρογραμματισμός πριν την εικονική μνήμη

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Χρήση καταχωρητών βάσης (χωρίς άλλη υποστήριξη) σε χαμηλού κόστους επεξεργαστές (π.χ. Intel 8086).

Διευκόλυνση τοποθέτησης προγραμμάτων οπουδήποτε στη μνήμη.



- Καταχωρητές βάσης κώδικα, δεδομένων
- Αλλαγή τιμής καταχωρητών βάσης ανά πρόγραμμα

Μεταβαίνοντας σε εικονικές διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

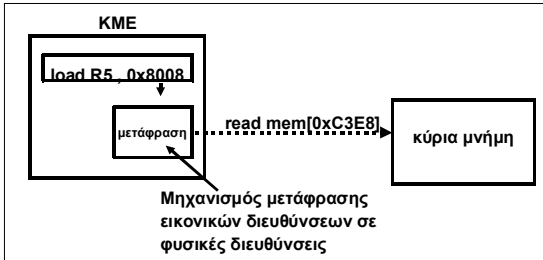
- Το προηγούμενο σχήμα
  - Εισήγαγε την αποσύνδεση των λογικών διευθύνσεων των προγραμμάτων από τις φυσικές διευθύνσεις κύριας μνήμης
  - Με απλή αντιστοιχία:

    φυσική διεύθυνση = λογική διεύθυνση + καταχωρητής βάσης
  - Απαιτείται υποστήριξη από το υλικό (KME)
  - Το πρόγραμμα μπορεί να φορτωθεί σε οποιαδήποτε θέση μνήμης (relocation)
    - Δεν περιέχει αναφορές σε φυσικές διευθύνσεις
  - Εισάγεται η έννοια των ξεχωριστών χώρων διευθύνσεων (κώδικα, δεδομένων...) ανά πρόγραμμα
    - χωρίς περαιτέρω υποστήριξη όμως!

Εικονικές Διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Ο κύριος ρόλος της εικονικής μνήμης στα σημερινά υπολογιστικά συστήματα, με την άφθονη κύρια μνήμη, είναι η **υποστήριξη και προστασία των** εικονικών χώρων διευθύνσεων ανά πρόγραμμα



- Μετάφραση εικονικών διευθύνσεων προγραμμάτων
- Προστασία χώρων διευθύνσεων
  - Προνόμια προσπέλασης χώρων μνήμης
- Απαιτείται συνδυασμένη υποστήριξη από ΚΜΕ, κρυφή μνήμη και λειτουργικό σύστημα

Η εικονική μνήμη (ξανά)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

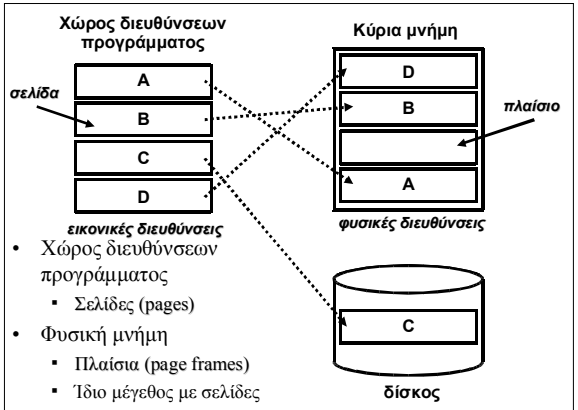
Η διαχείριση της εικονικής μνήμης έχει ομοιότητες με τη διαχείριση κρυφής-κύριας μνήμης. Τα δύο επίπεδα εδώ όμως (μνήμη-δίσκοι) έχουν σημαντικές διαφορές στα χαρακτηριστικά τους!

- Ποιος ο ρόλος της;
  1. Η εικονική χρήση μνήμης πέρα από όση είναι πραγματικά διαθέσιμη
  2. Η αντιστοίχιση εικονικών διευθύνσεων σε φυσικές και η διαχείριση των προνομίων προσπέλασης
     1. Σημαντικότερο σήμερα!
- Ποιος τη διαχειρίζεται;
  - Αυτόματη διαχείριση από το λειτουργικό σύστημα
  - Υποστήριξη από το υλικό (ΚΜΕ/κρυφή μνήμη)
- Πώς υλοποιείται;
  - Μετακίνηση τμημάτων μνήμης από/προς τους δίσκους
  - Εκμετάλλευση αρχής τοπικότητας
    - Μερικά μέρη μόνο των προγραμμάτων είναι "ενεργά" κάθε στιγμή

Βασικό σχήμα Εικονικής Μνήμης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Σελιδοποίηση (paging):
Το μέγεθος των σελίδων καθορίζεται από τους μηχανισμούς του υλικού.
Συνήθη μεγέθη 4KB έως 16KB



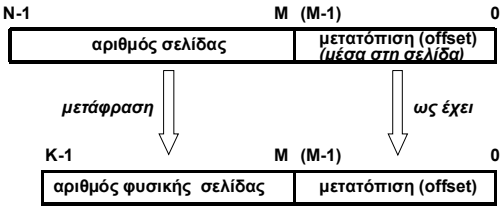
- Χώρος διευθύνσεων προγράμματος
  - Σελίδες (pages)
- Φυσική μνήμη
  - Πλαίσια (page frames)
  - Ίδιο μέγεθος με σελίδες

Σελίδες και εικονικές διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η μετάφραση γίνεται μέσα στην ΚΜΕ, άρα εκεί (δηλ. στο υλικό) καθορίζεται το μέγεθος της σελίδας

Ο επεξεργαστής παράγει εικονική διεύθυνση



Στη μνήμη στέλνεται φυσική διεύθυνση

- Ο επεξεργαστής μπορεί να παράγει έως και 2^N εικονικές διευθύνσεις
- Η φυσική μνήμη μπορεί να έχει έως 2^K διευθύνσεις
- Το μέγεθος σελίδας είναι 2^M bytes

"Σελίδες" εναντίον "Τμημάτων"

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η πολυπλοκότητα του μηχανισμού χρήσης τμημάτων (segmentation) είναι ο λόγος που η τεχνική αυτή σπάνια υποστηρίζεται ή χρησιμοποιείται σήμερα...

- Σελίδες (pages)
  - Σταθερό μέγεθος (4KB-16KB)
  - Μία λέξη εικονικής διεύθυνσης (page+offset)
  - Ευκολία τοποθέτησης και αντικατάστασης σελίδων στην κύρια μνήμη
  - Βέλτιστο μέγεθος για μεταφορά από-προς δίσκο
  - Ενδεχομένως μέρος σελίδας μένει αχρησιμοποίητο
- Τμήματα (segments)
  - Μεταβλητό μέγεθος (από 1 byte(!) έως 2^{32} bytes)
  - Δύο λέξεις ανά διεύθυνση (segment, offset)
  - Αναγκαίος ο έλεγχος ορίων
  - Δυσκολία τοποθέτησης στη μνήμη (εύρεση θέσης)
  - "ταιριάζει καλύτερα στα χαρακτηριστικά των εφαρμογών" (;)

Σελιδοποίηση κατ'απαίτηση

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η μεγάλη επιβάρυνση για τη μετακίνηση των σελίδων από και προς τον δίσκο είναι καθοριστική για την απόδοση ενός συστήματος

- Οι σελίδες των προγραμμάτων (κώδικας-δεδομένα) βρίσκονται αρχικά μόνο στον δίσκο
- Το ΛΣ τις σημειώνει ως "απούσες" από τη μνήμη
- Όταν προσπελαστεί μια "απούσα" σελίδα, δημιουργείται ένα σφάλμα εκτέλεσης (page fault)...
- ...και το ΛΣ τη φορτώνει σε ένα πλαίσιο στη μνήμη
- Ενδεχομένως εκτοπίζοντας πίσω στον δίσκο μια άλλη σελίδα από τη μνήμη
  - Η τελευταία σημειώνεται ως "απούσα"
- Page faults: μεγάλο κόστος σε κύκλους αναμονής
  - 1-10Μκύκλοι

Κρίσιμα σημεία στη σχεδίαση εικονικής μνήμης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

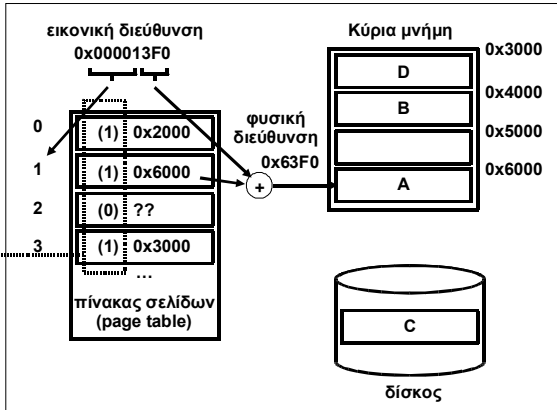
Ακόμα και μικρή μείωση στην εμφάνιση page faults μπορεί να έχει σημαντικό όφελος για την απόδοση του συστήματος

- Οι σελίδες πρέπει να έχουν ικανό μέγεθος
  - Για εξισορρόπηση του κόστους προσπέλασης του δίσκου
- Η μείωση των page faults είναι επιβεβλημένη
  - Οι σελίδες τοποθετούνται οπουδήποτε μέσα στη μνήμη
    - Σχήμα ανάλογο των **fully-associative** κρυφών μνημών
- Η διαχείριση της εικονικής μνήμης μπορεί να γίνεται από λογισμικό (ΛΣ)
  - Μικρή επιβάρυνση συγκρινόμενη με χρόνο μετακίνησης σελίδων στους δίσκους
  - Δυνατότητα χρήσης πολυπλοκότερων αλγορίθμων για τοποθέτηση-αντικατάσταση σελίδων στη μνήμη
- Δεν είναι δυνατή η ενημέρωση στο δίσκο με κάθε εγγραφή νέων δεδομένων στη σελίδα

Μετάφραση εικονικών διευθύνσεων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Presence bit: βρίσκεται η σελίδα στην κύρια μνήμη;



Πίνακας σελίδων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Τι συμβαίνει στην περίπτωση πολλαπλών προγραμμάτων;

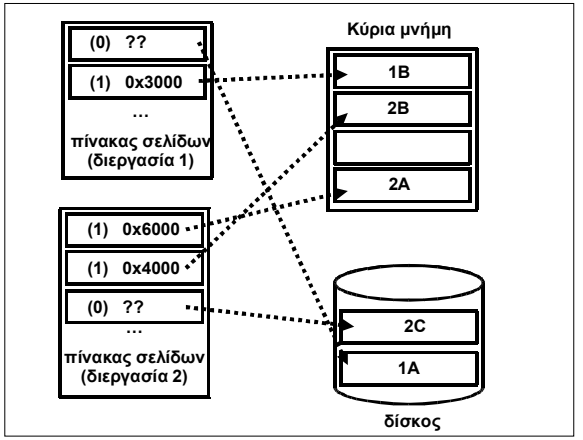
- Πού βρίσκεται;
  - Στην κύρια μνήμη!
- Πόσες θέσεις διαθέτει;
  - Θεωρητικά: ίσες με τον μέγιστο αριθμό σελίδων
    - Για 2^{20} σελίδες με 4 bytes ανά γραμμή, απαιτούνται 4MB
  - Πρακτικά: πίνακες πολλαπλών επιπέδων
  - Πιθανόν: σελιδοποίηση πινάκων (!)
    - Αρκεί να υπάρχει πάντα στη μνήμη ένα μέρος του πίνακα
- Ποια πρόσθετη πληροφορία χρειάζεται;
  - Βρίσκεται η σελίδα στη μνήμη; Έχει αλλάξει;
  - Αν όχι, σε ποιο σημείο του δίσκου είναι;
    - Συχνά μέσα στον πίνακα σελίδων
  - Ποια σελίδα βρίσκεται σε κάθε πλαίσιο μνήμης;
  - Πόσο πρόσφατα χρησιμοποιήθηκε;
    - Δομές του ΛΣ για αλγόριθμο αντικατάστασης σελίδων

Πίνακες σελίδων και πολλαπλά προγράμματα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Κάθε διεργασία έχει τους δικούς του πίνακες σελίδων!

Κατά την εναλλαγή διεργασιών αλλάζει και ο καταχωρητής-δείκτης στους πίνακες



Translation-Lookaside Buffer

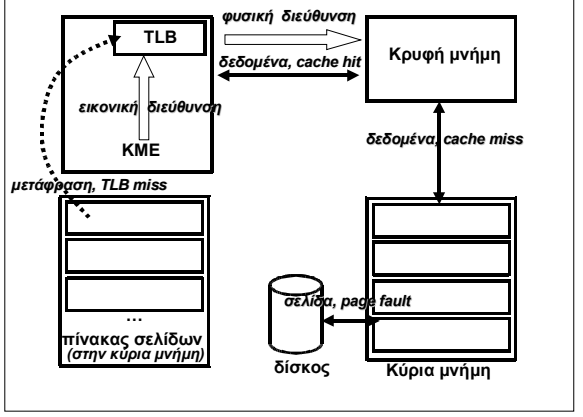
- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- Το πρόβλημα με τους πίνακες σελίδων
  - Βρίσκονται στην κύρια μνήμη
  - Για κάθε προσπέλαση μνήμης απαιτείται μια δεύτερη!
    - Για τον πίνακα σελίδων
    - Μη αποδεκτή χρονική επιβάρυνση!
- Translation-Lookaside Buffer (TLB)
  - Μικρή κρυφή μνήμη για πρόσφατες μεταφράσεις εικονικών διευθύνσεων
    - Μέσα στην ΚΜΕ
    - 16-512 θέσεις, 1-2 γραμμές του πίνακα σελίδων ανά θέση
    - Προσπέλαση < 1 κύκλο ρολογιού
    - Διάφορες οργανώσεις κρυφής μνήμης
    - Ενημέρωση από υλικό ή λογισμικό
    - Παρατηρούμενο Miss rate: 0.01% - 1%

Προσπέλαση μνήμης: η συνολική εικόνα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Στο σχήμα η κρυφή μνήμη δέχεται φυσικές διευθύνσεις. Για λόγους απόδοσης υπάρχουν αρχιτεκτονικές με εικονική κρυφή μνήμη ή συνδυασμό εικονικής-φυσικής κρυφής μνήμης



Εικονική Μνήμη και Προστασία Προσπέλασης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- Προστασία προσπέλασης σελίδων
  - Με διαφορετικούς πίνακες σελίδων ανά διεργασία είναι αδύνατη η προσπέλαση "ξένων" σελίδων
  - Δικαιώματα προσπέλασης ανά σελίδα
  - User mode και Supervisor Mode
    - Σε user mode δεν είναι δυνατή η προσπέλαση του TLB, του πίνακα σελίδων και των αντίστοιχων καταχωρητών συστήματος
    - Υπάρχουν αρχιτεκτονικές με περισσότερα από 2 επίπεδα προνομίων
  - Ελεγχόμενη προσπέλαση συναρτήσεων ΛΣ
    - Call gates: ελεγχόμενη εκτέλεση συναρτήσεων ΛΣ από χρήστη, στο επίπεδο όμως των προνομίων του χρήστη → δεν είναι δυνατή η προσπέλαση "ξένων" δεδομένων!

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Διασύνδεση Εισόδου-Εξόδου (I)

(δίαυλοι και μέθοδοι επικοινωνίας)

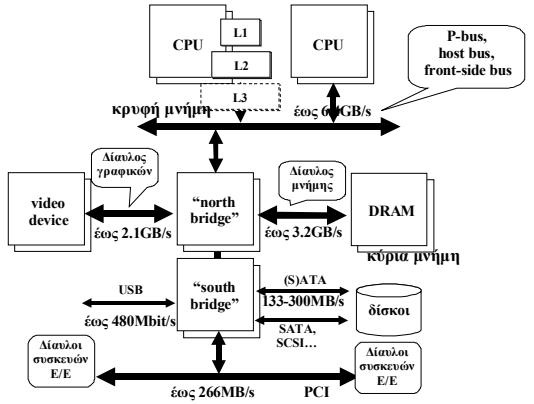
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Διασυνδετικοί Δίαυλοι

- Διασυνδετικοί δίαυλοι

Διασυνδετικοί δίαυλοι: Αναλαμβάνουν την μεταφορά των δεδομένων σε ένα υπολογιστικό σύστημα



Μεταφορά δεδομένων

- Διασυνδετικοί δίαυλοι

Ποιο θα μπορούσε να είναι το ζητούμενο από έναν διασυνδετικό δίαυλο εκτός της απόδοσης;

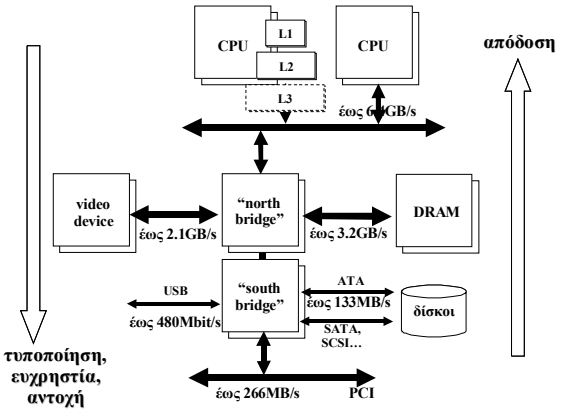
- Σε ένα υπολογιστικό σύστημα
  - Μεταφορά δεδομένων επεξεργασίας
    - Μεταξύ ΚΜΕ, κύριας μνήμης και συσκευών Ε/Ε
    - Τον ρόλο αναλαμβάνουν οι διασυνδετικοί δίαυλοι
  - Απόδοση των "συστημάτων μεταφοράς"
    - Σημαντική παράμετρος της συνολικής απόδοσης
    - Αλλά όχι το μόνο ζητούμενο!
- Χαρακτηριστικά διαύλων
  - Υπάρχει μεγάλη ποικιλία διαύλων
    - Με διαφορετικά λειτουργικά χαρακτηριστικά
  - Ανάλογα με τον ρόλο του καθενός διαύλου
    - Σε ένα υπολογιστικό σύστημα τα διασυνδεόμενα μέρη (και ιδίως οι συσκευές Ε/Ε) έχουν τελείως διαφορετικές ανάγκες μεταφοράς δεδομένων

Ιεραρχία διασυνδετικών διαύλων

- Διασυνδετικοί δίαυλοι

Το ζητούμενο από έναν διασυνδετικό δίαυλο εξαρτάται από τη θέση του στην ιεραρχία

Ο αριθμός των εναλλακτικών προϊόντων αυξάνεται στα χαμηλότερα επίπεδα της ιεραρχίας



Τι διασυνδέει ένας δίαυλος;

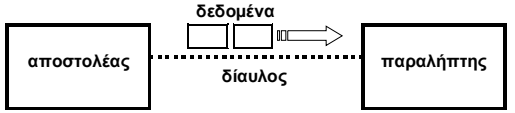
- Διασυνδετικοί δίαυλοι

Η μεταφορά των δεδομένων μέσα σε έναν υπολογιστή επιτυγχάνεται ηλεκτρικά μέσω αγωγών χαλκού. Νέες φωτονικές τεχνολογίες υπόσχονται πολύ μεγαλύτερες ταχύτητες (>10Gbps ανά αγωγό)

- Μέσα στο ολοκληρωμένο κύκλωμα
  - π.χ. ΚΜΕ-κρυφή μνήμη
    - Μέσα στο ολοκληρωμένο κύκλωμα επιτυγχάνεται η μέγιστη ταχύτητα μεταφοράς των δεδομένων, μέσω αγωγών μετάλλου "τυπωμένων" στην επιφάνεια πυριτίου
- Πάνω στο ίδιο τυπωμένο κύκλωμα
  - π.χ μεταξύ κύριας μνήμης και ελεγκτή μνήμης
    - Αγωγοί σχεδιασμένοι με ακρίβεια πάνω στο τυπωμένο κύκλωμα
- Μεταξύ τυπωμένων κυκλωμάτων
  - π.χ η διασύνδεση με τις μονάδες δίσκου
    - Διασύνδεση μέσω εύκαμπτων καλωδίων σε αποστάσεις της τάξης του ενός μέτρου
- Μεταξύ υπολογιστικών συστημάτων
  - π.χ η (τοπική) διασύνδεση της κάρτας δικτύου

Βασικά χαρακτηριστικά διαύλων

- Διασυνδετικοί δίαυλοι



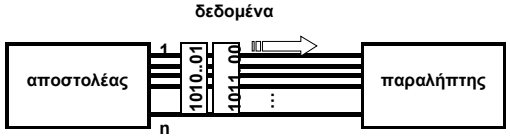
Μεταφορά δεδομένων: η γενική εικόνα

- Πόσα bits μεταφέρονται ταυτόχρονα;
- Πότε ο παραλήπτης θα διαβάσει την είσοδο;
- Ποια η τοπολογία του διαύλου;
- Πώς οργανώνεται η μεταφορά;

Παράλληλοι δίαυλοι

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου

Ο ρυθμός μεταφοράς δεδομένων μετριέται πάντοτε σε μονάδες με βάση το 10: ρυθμός 1Mb/s = 10^6 b/s

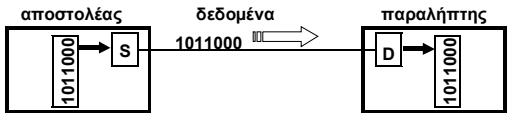


εύρος διαύλου = n bits

- Πολλαπλασιασμός των bits που μεταφέρονται ταυτόχρονα
  - Συχνά ο ρυθμός μετριέται σε μεταφορές/s (T/s)
  - Π.χ. 10MT/s για δίαυλο εύρους 64 bits → 640Mb/s
- Πολύ υψηλοί ρυθμοί μεταφοράς
- Αλλά και μεγάλος ηλεκτρονικός "θόρυβος"
  - Απαιτείται ακριβής σχεδιασμός για να διατηρηθεί η ποιότητα του σήματος και ο συγχρονισμός στη μεταφορά δεδομένων
  - Χρησιμοποιείται στα υψηλότερα επίπεδα της ιεραρχίας διαύλων

Σειριακοί δίαυλοι

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου

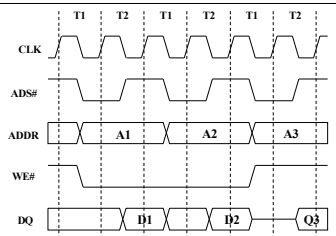


- Οι λέξεις (words) μετατρέπονται σε σειρά από bits πριν την αποστολή - και αντίστροφα μετά την παραλαβή
  - Serializer – deserializer (SerDes)
- Γιατί χρησιμοποιούνται;
  - Μεγαλύτερη ανοσία στον ηλεκτρονικό θόρυβο
  - Δυνατότητα μεταφοράς σε μεγαλύτερη απόσταση
  - Μικρότερες διαστάσεις αγωγού
  - Χρησιμοποιείται στα χαμηλότερα επίπεδα της ιεραρχίας διαύλων

Χρονισμός μεταφοράς

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός

Οι σύγχρονοι δίαυλοι χρησιμοποιούνται στα ανώτερα επίπεδα της ιεραρχίας: π.χ. στο σχήμα φαίνεται η λειτουργία ενός τυπικού διαύλου της ΚΜΕ (FSB)

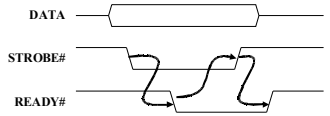


- Πότε ο αποστολέας θα στείλει τα δεδομένα στις εξόδους;
- Πότε ο παραλήπτης θα δειγματοληπτήσει τις εισόδους;
- Γραμμή ρολογιού (clock)
  - Στις ανερχόμενες ή/και κατερχόμενες ακμές
  - Σύγχρονοι δίαυλοι (synchronous)
  - Αλλά: ευαισθησία στις παραμορφώσεις του clock
  - Θόρυβος και κατανάλωση ενέργειας

Άλλα σχήματα χρονισμού

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός

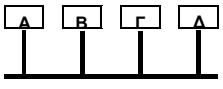
Σειριακοί δίαυλοι με ενσωμάτωση της χρονικής πληροφορίας στα δεδομένα χρησιμοποιούνται στα κατώτερα επίπεδα της ιεραρχίας (π.χ. ο δίαυλος USB)

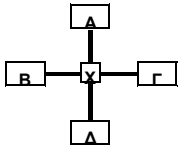


- Ασύγχρονοι δίαυλοι (χωρίς ρολόι)
  - Σήματα συγχρονισμού
  - Ανοσία σε παραμορφώσεις σημάτων
  - Αλλά: αργότερη μεταφορά – πολύπλοκη ανταλλαγή σημάτων
- Ενσωμάτωση ρολογιού στα δεδομένα
  - Δεν υπάρχει ξεχωριστή γραμμή clock
  - Αλλά ο παραλήπτης μπορεί να συμπεράνει για το ρολόι του αποστολέα αν το μεταδιδόμενο σήμα έχει ικανό αριθμό ακμών (εναλλαγές μεταξύ 0 και 1)

Τοπολογίες Διαύλων

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες



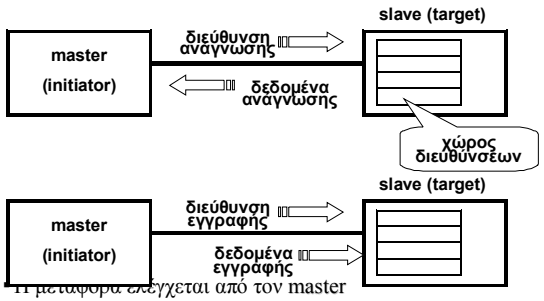


- Μοιραζόμενος δίαυλος
  - Η παλαιότερη λύση
  - Αναγκαία η διαιτησία για την κατοχή του διαύλου
  - Κακή ποιότητα σημάτων → χαμηλότερη ταχύτητα
- Point-to-point
  - Διασύνδεση πάντα μεταξύ δύο σημείων
  - Τμήματα συγκέντρωσης-ανταλλαγής δεδομένων
    - Hubs ή switches
  - Νεώτερες λύσεις για υψηλότερη ταχύτητα μεταφοράς

Οργάνωση μεταφοράς

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς

Η φορά της "ανάγνωσης" και της "εγγραφής" ορίζεται κατά σύμβαση!

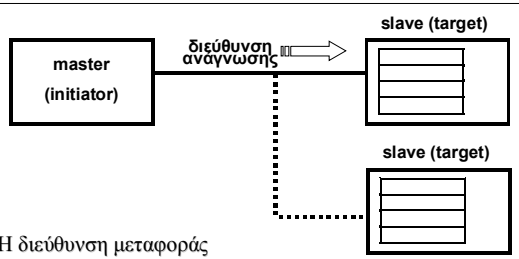


- Η μεταφορά ελέγχεται από τον master
- Ανάγνωση ή εγγραφή από/στον χώρο διευθύνσεων του slave
  - Σε ποιον slave και σε ποιες λέξεις του slave γίνεται η μεταφορά

Διεύθυνση μεταφοράς

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς

Το διπλανό σχήμα υλοποιείται φυσικά (σε μοιραζόμενους διαύλους) ή λογικά (όταν ο δίαυλος είναι point-to-point)



- Η διεύθυνση μεταφοράς
  - Επιλέγει slave
  - Επιλέγει λέξεις μέσα στον χώρο διευθύνσεων του slave
- Κατανομή διευθύνσεων
  - Είτε στατικά (στον σχεδιασμό)
  - Είτε δυναμικά (στην αρχικοποίηση του συστήματος)

Φάσεις μεταφοράς

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς

Λόγω των φάσεων μεταφοράς, η χρήσιμη πληροφορία είναι (max) το 70% της μεταδιδόμενης! Σπάνια επιτυγχάνονται οι ρυθμοί μεταφοράς (peak rates) που αναφέρονται στις προδιαγραφές!

- Σε κάθε μεταφορά σε έναν δίαυλο
- Ορισμένες (ή όλες) από τις εξής φάσεις:
  - Διαιτησία (arbitration)
    - Σε μοιραζόμενους διαύλους, απόκτηση του διαύλου
  - Αίτηση (request)
    - Αποστολή διεύθυνσης, κατεύθυνσης μεταφοράς (read/write), μήκους μεταφερόμενων δεδομένων (σε bytes)
  - Απόκριση (response)
    - Κατάσταση μεταφοράς (ολοκλήρωση, αποτυχία, επανάληψη, αναμονή)
  - Δεδομένα (data)
    - Τα μεταφερόμενα δεδομένα (όταν υπάρχουν)

Πληροφορία μεταφοράς

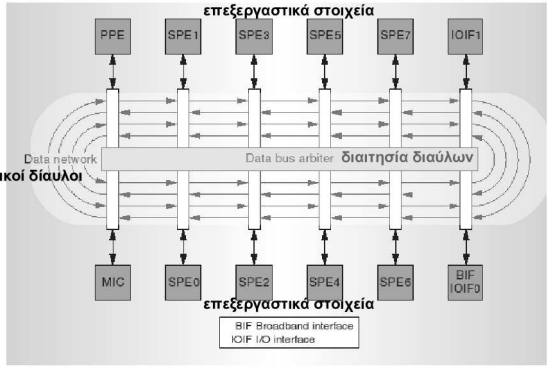
- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς

- Διεύθυνση – δεδομένα – εντολές/κατάσταση
- Παράλληλοι δίαυλοι
  - Συνήθως χρησιμοποιούνται ξεχωριστές γραμμές ανά είδος πληροφορίας
  - Συχνά κάποιες πληροφορίες είναι χρονικά πολυπλεγμένες για οικονομία στο πλήθος αγωγών
    - Π.χ. διεύθυνση-δεδομένα
- Σειριακοί δίαυλοι
  - Τα πακέτα δεδομένων που ανταλλάσσονται περιέχουν και τα 3 είδη πληροφορίας

Παράδειγμα διαύλων: Cell Multiprocessor

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς
- Παράδειγμα

διασυνδετικοί δίαυλοι
peak transfer rate >200GB/s



[IEEE Micro, May-June 2006]

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2007-08

Διασύνδεση Εισόδου-Εξόδου (ΙΙ)

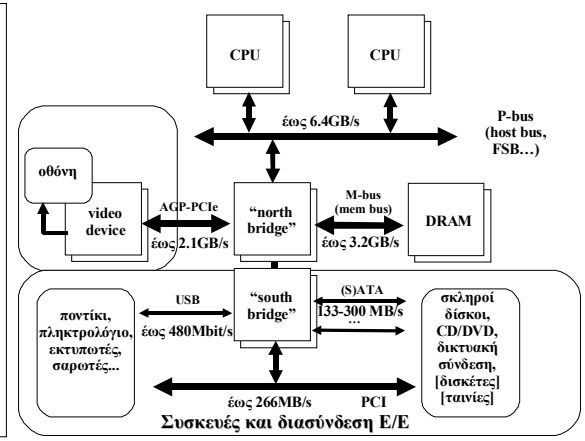
(συσκευές και δίαυλοι Ε/Ε)

http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Διασύνδεση συσκευών εισόδου εξόδου

• Συσκευές Ε/Ε



Συσκευές και διασύνδεση Ε/Ε

Διασύνδεση συσκευών Ε/Ε

• Συσκευές Ε/Ε

- Συσκευές εισόδου-εξόδου (Ε/Ε)
  - Συμβατικά: οτιδήποτε εκτός ΚΜΕ και μνήμη
  - Συσκευές με τελείως διαφορετικούς μεταξύ τους ρόλους και χαρακτηριστικά
    - Λειτουργία: είσοδος, έξοδος ή και τα δύο (αποθήκευση)
    - Συνεργάτης (στην άλλη άκρη): άνθρωπος ή μηχανή
    - Ρυθμός μεταφοράς δεδομένων

Συσκευή	Λειτουργία	Ρυθμός (Mbps)
πληκτρολόγιο	είσοδος	0,0001
ποντίκι	είσοδος	0,0038
εκτυπωτής	έξοδος	3,2
Δίκτυο LAN	είσοδος/έξοδος	100-1000
μαγν. δίσκος	αποθήκευση	240-2565
οθόνη	έξοδος	800-8000

Διασύνδεση συσκευών Ε/Ε

• Συσκευές Ε/Ε

Λόγω της ποικιλότητας και του πλήθους των συσκευών Ε/Ε οι αντίστοιχοι δίαυλοι διασύνδεσης πρέπει να υπακούν σε πρότυπα – πολύ περισσότερο απ'ότι π.χ. το FSB ενός επεξεργαστή!

- Απόδοση συστημάτων Ε/Ε
  - Σύνθετο μέγεθος
  - Εξαρτάται από πολλούς παράγοντες
    - Χαρακτηριστικά συσκευών
    - Χαρακτηριστικά διαύλου διασύνδεσης
    - Ιεραρχία μνήμης
    - Λειτουργικό Σύστημα
  - Χρόνος απόκρισης – Ρυθμός ολοκλήρωσης
- Σημαντικές παράμετροι (εκτός της απόδοσης)
  - Αξιοπιστία
  - Μείωση κόστους
  - Επεκτασιμότητα και διαχείριση διαφορετικότητας συσκευών

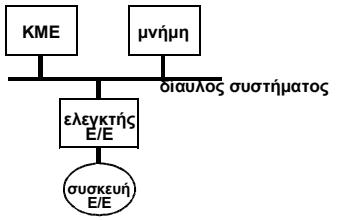
Διασύνδεση συσκευών Ε/Ε

- Συσκευές Ε/Ε

- Κατηγορίες συσκευών Ε/Ε ανάλογα με την διασύνδεσή τους στο σύστημα
  1. Καθορισμός από τη σχεδίαση του συστήματος
     - Βασικά μέρη συστήματος (στη μητρική πλακέτα)
     - Το BIOS γνωρίζει ήδη για τις συσκευές αυτές
  2. Προσθήκη κατά τη σύνθεση του συστήματος
     - Κάρτες επέκτασης
     - Αναγνώριση κατά την εκκίνηση του συστήματος
  3. Προσθήκη κατά τη λειτουργία του συστήματος
     - Συσκευές που συνδέονται εξωτερικά στο σύστημα
     - Αναγνώριση κατά τη λειτουργία του συστήματος (hot-plug)

Οι πρώτες μορφές διασύνδεσης

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε



- Διασύνδεση στον δίαυλο του συστήματος μέσω ενός ελεγκτή Ε/Ε (I/O controller ή adapter)
- Η ΚΜΕ "βλέπει" απευθείας τον ελεγκτή Ε/Ε
- Ο ελεγκτής αναλαμβάνει την επικοινωνία με την πολύ αργότερη συσκευή Ε/Ε

Ο ρόλος ενός ελεγκτή Ε/Ε

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε

Πώς επικοινωνεί η ΚΜΕ με τον ελεγκτή για την υλοποίηση των λειτουργιών αυτών;

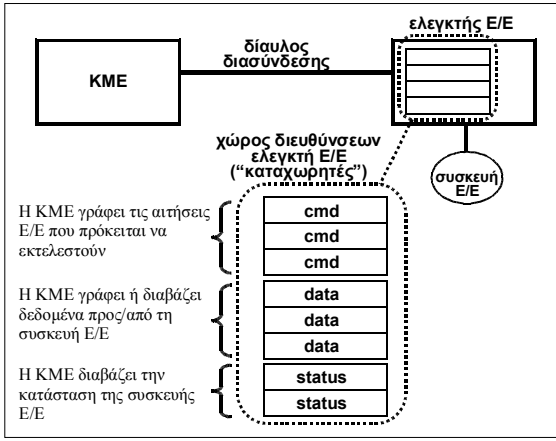
- Αποστολή εντολών προς τη συσκευή Ε/Ε
  - Μετάφραση των αιτήσεων Ε/Ε της ΚΜΕ
    - Μόνο εντολές load-store φτάνουν στον ελεγκτή από την ΚΜΕ!
- Λήψη ένδειξης κατάστασης συσκευής Ε/Ε
  - Δυνατότητα ελέγχου πορείας μιας αίτησης Ε/Ε
    - Οι αιτήσεις Ε/Ε ολοκληρώνονται πολύ αργά σε σχέση με την ταχύτητα επεξεργασίας της ΚΜΕ!
- Ειδοποίηση της ΚΜΕ
  - Μετά την ολοκλήρωση αίτησης Ε/Ε
    - Μηχανισμός interrupts
- Παροχή δεδομένων από συσκευή Ε/Ε
  - Προς την ΚΜΕ ή αυτόνομα προς τη μνήμη
    - Μηχανισμός DMA

Πώς βλέπει η ΚΜΕ έναν ελεγκτή Ε/Ε

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε

"Μοντέλο καταχωρητών" του ελεγκτή Ε/Ε

Οι καταχωρητές βρίσκονται στον χώρο διευθύνσεων του συστήματος



Απεικόνιση διευθύνσεων E/E

- Συσκευές E/E
- Διασύνδεση E/E

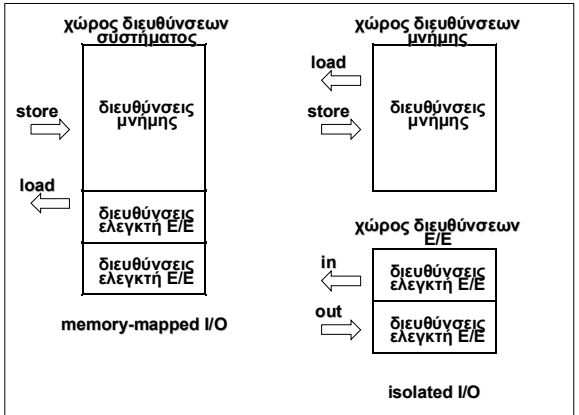
- Η τοποθέτηση των διευθύνσεων των ελεγκτών E/E στον χώρο διευθύνσεων του συστήματος
  - Στατικά
    - Η αρχιτεκτονική του συστήματος ορίζει τις διευθύνσεις
    - Όχι ευέλικτο σχήμα
  - Δυναμικά
    - Ανάθεση κατά την αρχικοποίηση του συστήματος
    - Απαιτείται συνεργασία υλικού και λογισμικού – το περίφημο "plug-n-play"
- Η ΚΜΕ υποστηρίζει ξεχωριστές εντολές E/E;
  - Ναι (isolated I/O): δύο ξεχωριστοί χώροι διευθύνσεων
    - in/out αντί load/store
  - Όχι (memory-mapped I/O): ενιαίος χώρος διευθύνσεων

Χώροι διευθύνσεων για E/E

- Συσκευές E/E
- Διασύνδεση E/E

Memory-mapped I/O: ενιαίος χώρος και διαχείριση των διευθύνσεων συστήματος

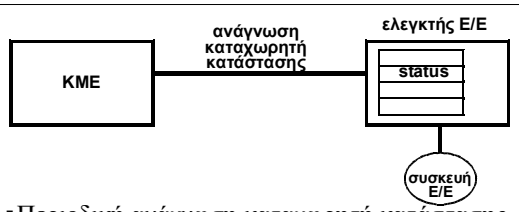
Isolated I/O: εύκολος διαχωρισμός εντολών αν πρέπει E/E να γίνεται μόνο από το ΛΣ



Ανίχνευση ολοκλήρωσης αίτησης E/E

- Συσκευές E/E
- Διασύνδεση E/E
- Εξυπηρέτηση E/E από ΚΜΕ

Το polling παρά την επιβάρυνση χρησιμοποιείται σε συστήματα ελέγχου, όπου ο έλεγχος πρέπει να γίνεται σε προκαθορισμένα διαστήματα



- Περιοδική ανάγνωση καταχωρητή κατάστασης
  - Ολοκληρώθηκε η τρέχουσα αίτηση;
  - Εμφανίστηκε κάποια αλλαγή κατάστασης στη συσκευή E/E;
  - Επαναληπτική διαδικασία polling
  - Επιβάρυνση ΚΜΕ με άσκοπους κύκλους εκτέλεσης – για μεγάλο διάστημα η κατάσταση θα παραμένει σταθερή περιμένοντας τη συσκευή E/E

Ανίχνευση ολοκλήρωσης αίτησης E/E

- Συσκευές E/E
- Διασύνδεση E/E
- Εξυπηρέτηση E/E από ΚΜΕ

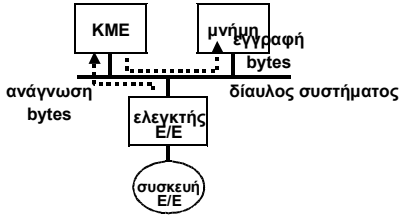
Παλαιότερα ο ελεγκτής ειδοποιούσε την ΚΜΕ με ξεχωριστό σήμα διακοπής. Σήμερα αυτό υλοποιείται με ανταλλαγή μηνυμάτων μέσω του διαύλου διασύνδεσης

- Ασύγχρονη ειδοποίηση ΚΜΕ (interrupts)
  - Ο ελεγκτής ειδοποιεί την ΚΜΕ για αλλαγή κατάστασης της συσκευής E/E
  - Στο μεταξύ η ΚΜΕ μπορεί να εκτελεί άλλη διεργασία
  - Η διακοπή θα προκαλέσει την εκτέλεση ρουτίνας εξυπηρέτησης διακοπής (interrupt service routine – ISR)
  - Ενδεχομένως η ΚΜΕ θα χρειαστεί πρόσθετη πληροφορία από ελεγκτή σχετικά με την αιτία της διακοπής

Μετακίνηση δεδομένων από/προς μνήμη

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ

Εάν η συσκευή Ε/Ε είναι κάρτα δικτύου 100Mbps Ethernet και δημιουργεί διακοπή για κάθε πακέτο με μέσο μήκος 256 bytes, πόσες φορές/sec θα ζητηθεί εξυπηρέτηση από την ΚΜΕ;

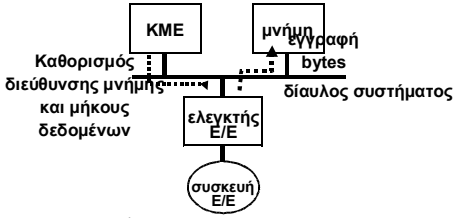


- Μετά από διακοπή ή polling
- Η ΚΜΕ επαναληπτικά διαβάζει δεδομένα από τον ελεγκτή Ε/Ε και τα γράφει στη μνήμη – και το αντίστροφο
- Σημαντική επιβάρυνση της ΚΜΕ όσο
  - το μέγεθος των δεδομένων κάθε αίτησης Ε/Ε αυξάνονται
  - ο ρυθμός ολοκλήρωσης αιτήσεων Ε/Ε αυξάνεται

Direct Memory Access (DMA)

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ

Όταν το σύστημα χρησιμοποιεί εικονική μνήμη, θα πρέπει να ληφθούν τα κατάλληλα μέτρα ώστε κατά τη μεταφορά DMA τα δεδομένα να τοποθετηθούν στη σωστή φυσική διεύθυνση

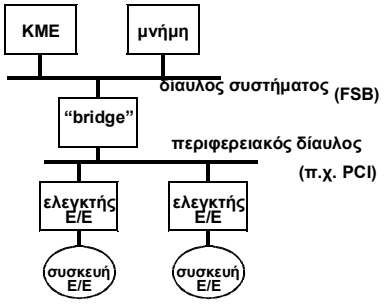


- Η ΚΜΕ απλά θέτει τις παραμέτρους της μεταφοράς
- Ο ελεγκτής διεκδικεί τον δίαυλο του συστήματος και μεταφέρει δεδομένα απευθείας προς/από τη μνήμη
  - Ή ξεχωριστός ελεγκτής DMA
  - Η ΚΜΕ μπορεί να εκτελεί άλλη διεργασία
  - Σύγκρουση στη μνήμη, αλλά ή ΚΜΕ έχει και την κρυφή μνήμη!

Η εμφάνιση του περιφερειακού διαύλου

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ
- Δίαυλοι Ε/Ε

Εμφάνιση της ιεραρχίας διασυνδετικών διαύλων, με διαφορετικούς ρυθμούς μεταφοράς



- Οι συσκευές Ε/Ε (περιφερειακές συσκευές) συνδέονται σε περιφερειακό δίαυλο
- Η ΚΜΕ "βλέπει" μόνο μια "συσκευή": peripheral bus bridge
  - Μέσω της οποίας γίνεται η προσπέλαση των συσκευών Ε/Ε

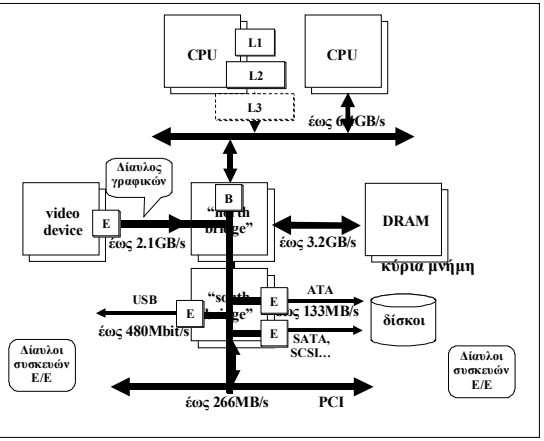
Περιφερειακοί δίαυλοι

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ
- Δίαυλοι Ε/Ε

- Ένας περιφερειακός δίαυλος ορίζεται από συνδυασμό επιπέδων υλικού-λογισμικού
- Δυναμική αρχικοποίηση συσκευών (configuration)
  - Ευέλικτη διασύνδεση
- Οδήγηση συσκευών (drivers)
  - Επεκτασιμότητα
- Ιεραρχική δομή
  - Υπο-δίαυλοι
  - Χρήση bridges

Η φυσική υλοποίηση σήμερα

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ
- Δίαυλοι Ε/Ε



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Εισαγωγή στην Αρχιτεκτονική Η/Υ

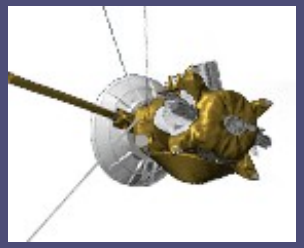
(θεμελιώδεις αρχές λειτουργίας των υπολογιστών)

http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Τι είναι ένα "υπολογιστικό σύστημα";

- Υπολογιστικά Συστήματα (;)



Ενσωματωμένα συστήματα

- Υπολογιστικά Συστήματα

;

Ποια είναι η αρχιτεκτονική του μικροεπεξεργαστή που παράγεται σε μεγαλύτερο αριθμό ανά έτος;

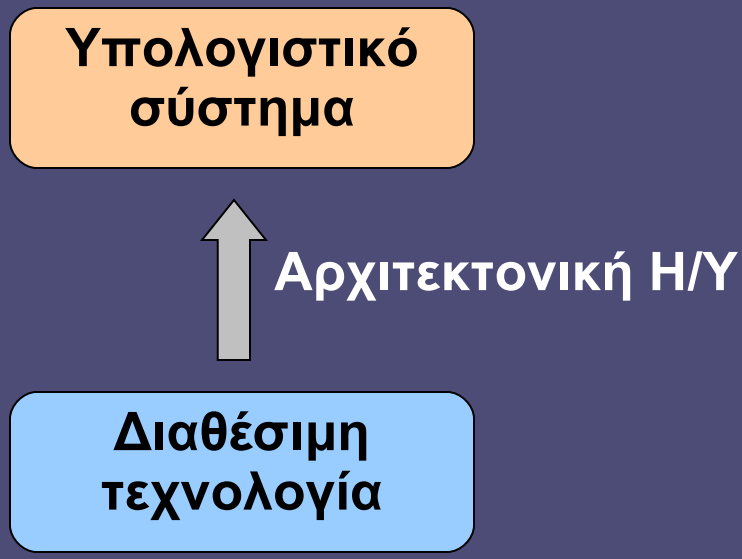
- Υπολογιστικά συστήματα ελέγχου
  - Συστήματα ειδικών απαιτήσεων
    - Ιατρικά, συγκοινωνίες, εργοστάσια, συλλογή πληροφοριών..
  - Καταναλωτικά προϊόντα
- Μια ανερχόμενη αγορά
  - 95% των μικροεπεξεργαστών που πωλούνται ανά έτος καταλήγει σε ένα ενσωματωμένο σύστημα!
- Συστήματα μέσα σε ένα ολοκληρωμένο κύκλωμα
  - Processor cores

Αρχιτεκτονική Η/Υ

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

;

Ανήκει το **λογισμικό** στο διπλανό σχήμα;



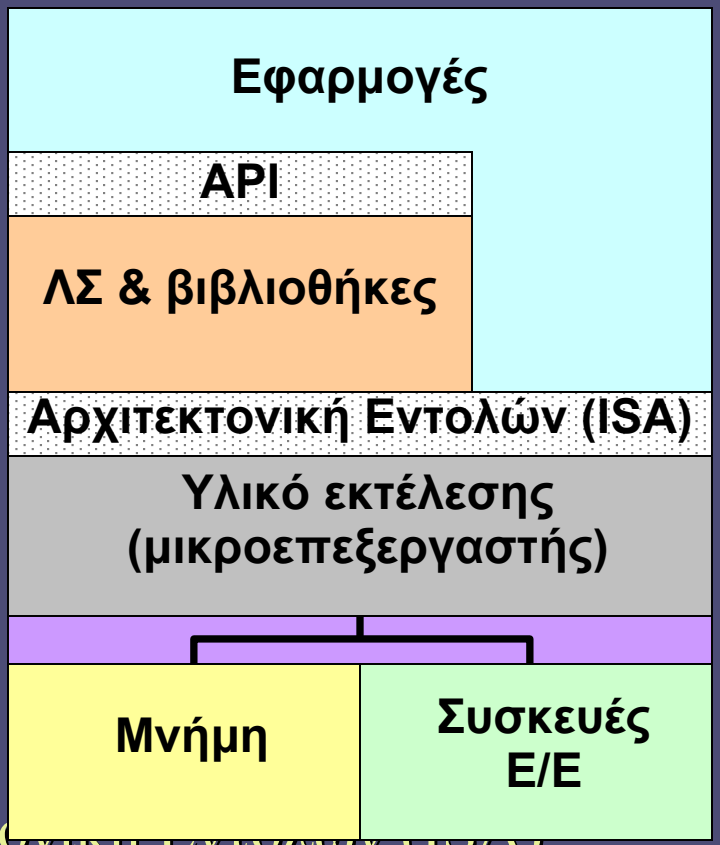
- Η αρχιτεκτονική συγκροτεί υπολογιστικά συστήματα χρησιμοποιώντας την υπάρχουσα τεχνολογία
  - και προδιαγράφει τη μελλοντική

Ο υπολογιστής ως ιεραρχία επιπέδων

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

;

Τι ακριβώς είναι μια **διεπαφή** (interface); Γιατί είναι τόσο σημαντική έννοια;



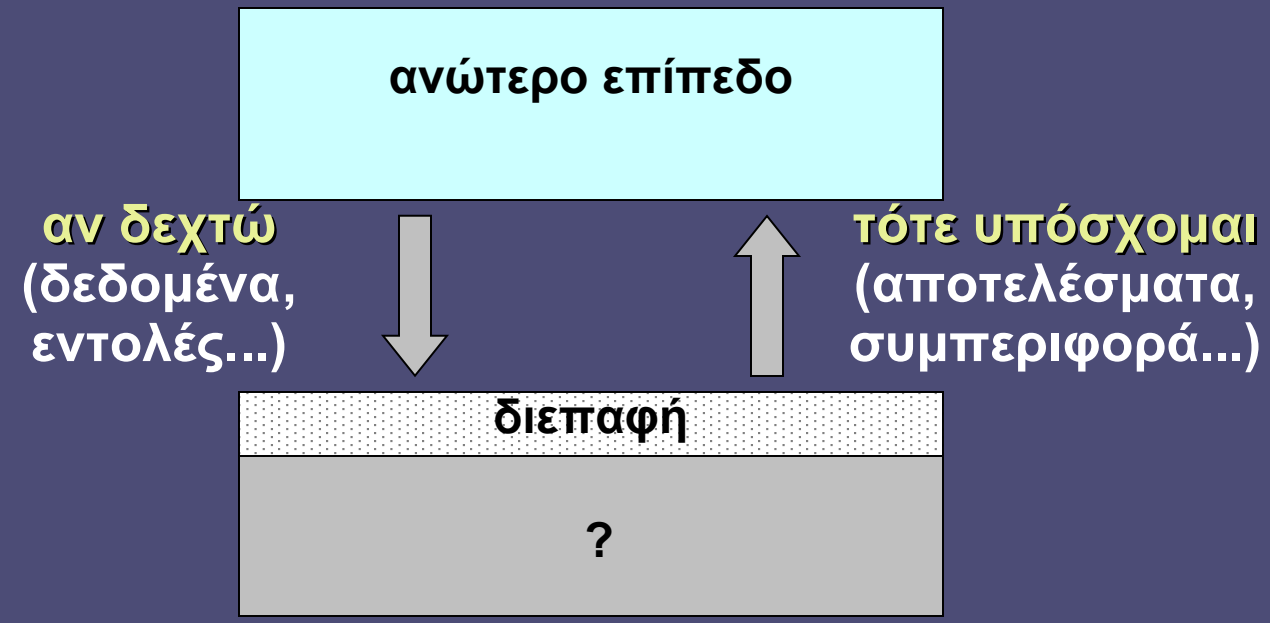
- Αρχιτεκτονική Εντολών (ISA)
  - Η διεπαφή υλικού-λογισμικού

Η σπουδαιότητα των διεπαφών

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

;

Αντιστοιχία με γλώσσες προγραμματισμού;



- Αντιμετώπιση πολυπλοκότητας σχεδιασμού
- Επαναχρησιμοποίηση τμημάτων
- Αξιόπιστος σχεδιασμός

Αρχιτεκτονική: υλικό μόνο ή και λογισμικό;

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

;

Σε ποια συστήματα έχει ιδιαίτερη σημασία η αρχιτεκτονική του λογισμικού;

- Το λογισμικό είναι σημαντικό μέρος ενός υπολογιστικού συστήματος
  - Ορίζει τον τρόπο χρήσης του υλικού
  - Συνεπώς (συν-)διαμορφώνει
    - την απόδοση
    - την κατανάλωση ενέργειας
    - την αξιοπιστία
- "Η αρχιτεκτονική ασχολείται με το υλικό"
  - Η παραδοσιακή αντίληψη
- "Πρέπει να λαμβάνεται υπ'όψη και το λογισμικό"
  - Η σύγχρονη αντίληψη

Αρχιτεκτονική: ο τελικός στόχος

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ

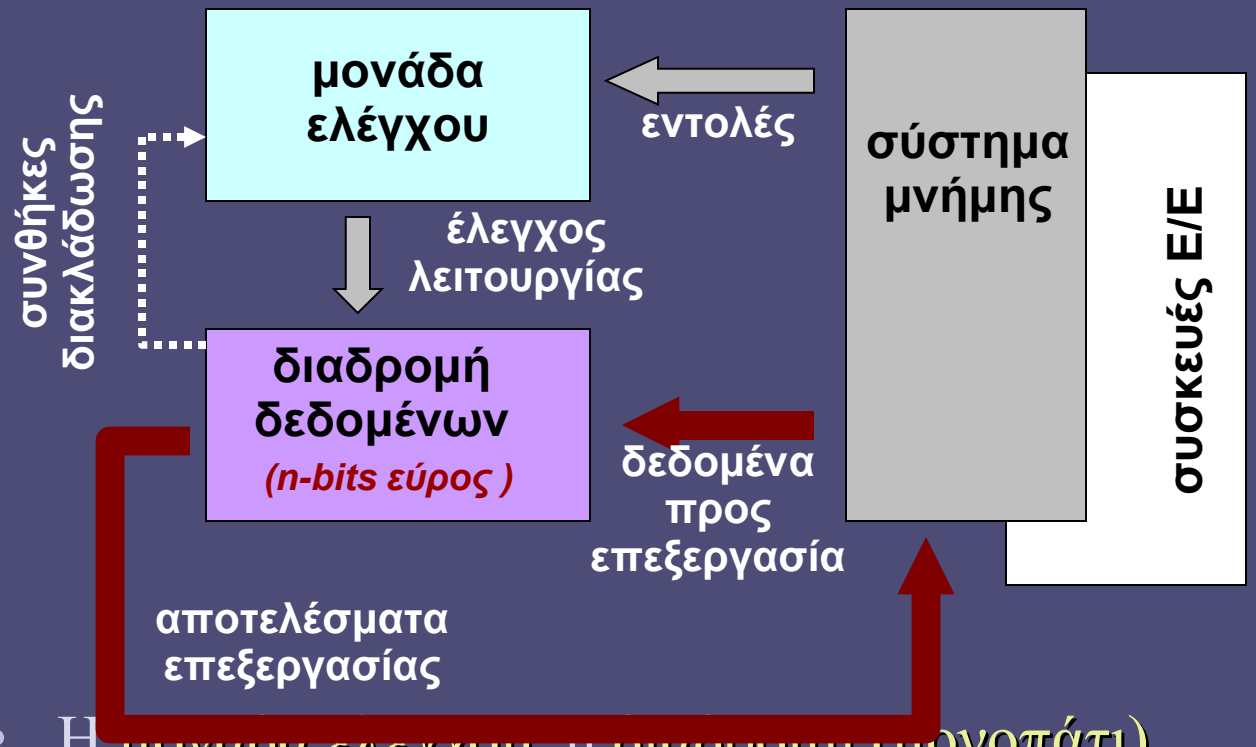
;

Ποια η διαφορά μεταξύ **γρήγορης εκτέλεσης** και **αξιόπιστης εκτέλεσης**;

- Η αξιοποίηση με τον καλύτερο δυνατό τρόπο του υλικού (και λογισμικού) για την ανάπτυξη
  - Αποδοτικών συστημάτων
    - Γρήγορη εκτέλεση προγραμμάτων
    - Υπερυπολογιστές, προσωπικοί υπολογιστές
  - Αξιόπιστων συστημάτων
    - Ασφαλής εκτέλεση προγραμμάτων
    - Συστήματα πραγματικού χρόνου
  - Προσιτών συστημάτων
    - Ικανοποιητική εκτέλεση με μικρό κόστος-ενέργεια
    - Καταναλωτικές συσκευές

Οι βασικές μονάδες κάθε υπολογιστή

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες



- Η μονάδα ελέγχου, η διαδρομή (μονοπάτι) δεδομένων και μέρος του συστήματος μνήμης βρίσκονται σήμερα μέσα στον μικροεπεξεργαστή (ΚΜΕ)

!

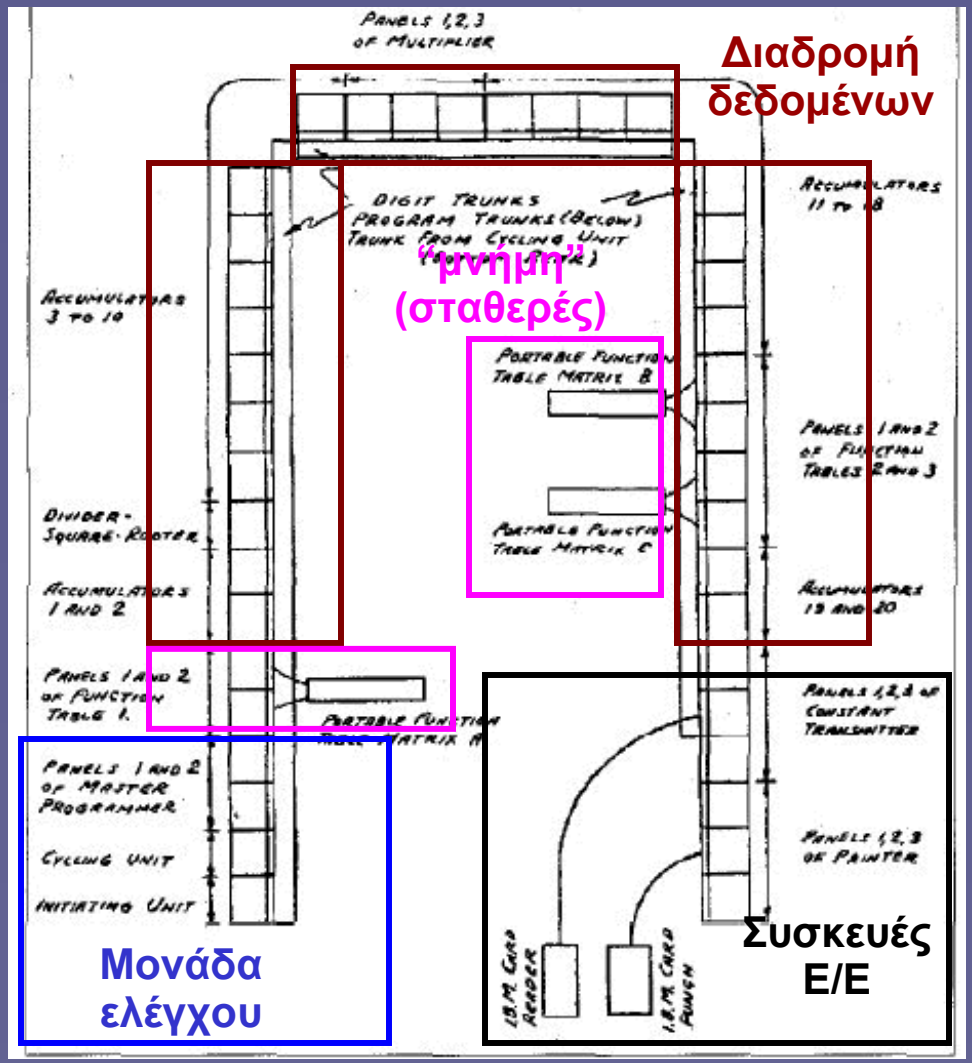
Η διπλανή εικόνα είναι απλοποιημένη!

The Electronic Numerical Integrator and Computer (ENIAC) 1946

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες

i

Ο ENIAC, ο πρώτος ηλεκτρονικός υπολογιστής, διέθετε 18.000 λυχνίες για τα λογικά του κυκλώματα. Δεν υπήρχε μνήμη προγράμματος!

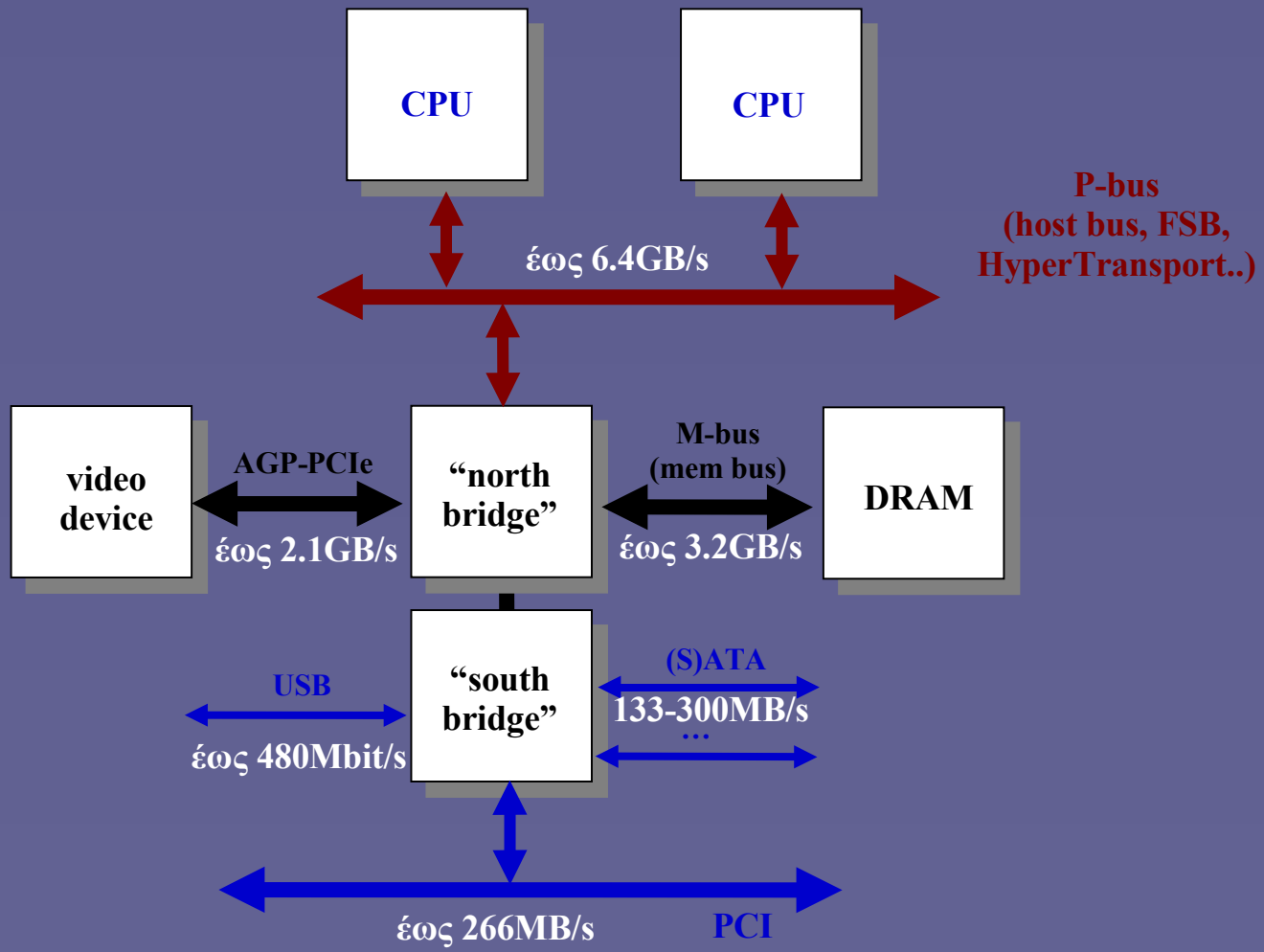


Ένα τυπικό υπολογιστικό σύστημα σήμερα

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες

!

Ο ρυθμός μεταφοράς δεδομένων ανάμεσα στα διάφορα μέρη του υπολογιστή επηρεάζει καθοριστικά τη συνολική απόδοση του συστήματος!



Ηλεκτρονικά κυκλώματα

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

- Σε κάθε υπολογιστή απαιτούνται
  - Λογικά κυκλώματα
    - Για την εκτέλεση πράξεων
    - Για τη σύγκριση και λήψη αποφάσεων
  - Κυκλώματα μνήμης
    - Για την αποθήκευση δεδομένων
    - Για την αποθήκευση εντολών
  - Κυκλώματα διασύνδεσης
    - Για τη μεταφορά δεδομένων μεταξύ τμημάτων

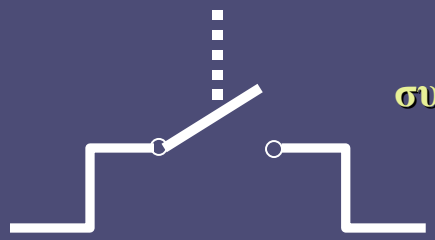
Πριν τα σύγχρονα ηλεκτρονικά

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

- Παλαιότερα χρησιμοποιήθηκαν...
  - Λογικά κυκλώματα
    - Μηχανικά συστήματα (γρανάζια)
    - Ηλεκτρομηχανικά (ρελέ)
    - Λυχνίες κενού
  - Κυκλώματα μνήμης
    - Τα ίδια, αλλά και...
    - Γραμμές υδραργύρου (!) – καθοδική οθόνη (!)
  - Κυκλώματα διασύνδεσης
    - Καλώδια!

Το τρανζίστορ MOS(FET)

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα



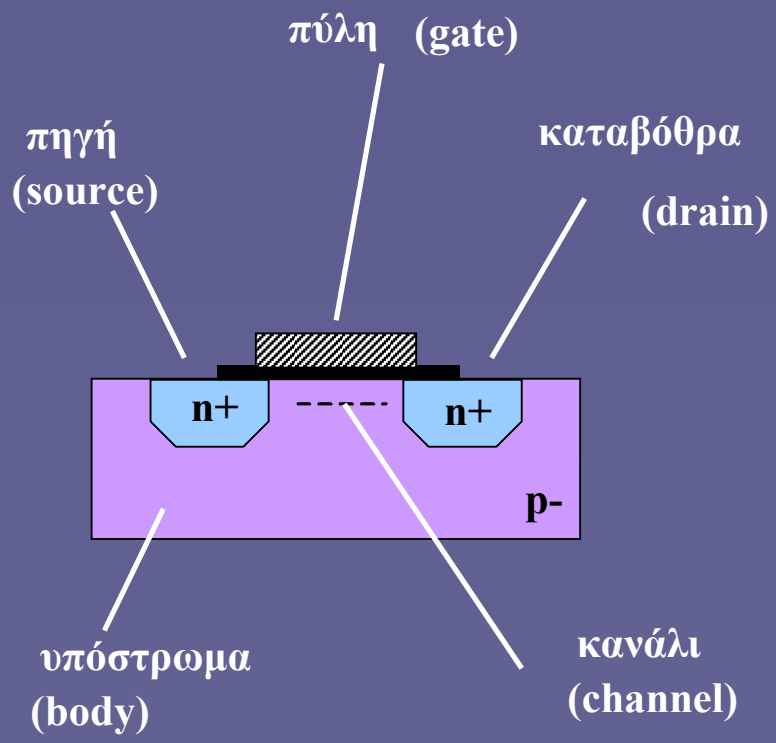
- Ο μικροσκοπικός διακόπτης των σύγχρονων κυκλωμάτων
  - Η θεωρία λειτουργίας του είναι γνωστή από το 1925...
  - ...αλλά τα πρώτα λειτουργικά τρανζίστορ MOS κατασκευάστηκαν στη δεκαετία του 60

Το τρανζίστορ MOS(FET)

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

i

Το τρανζίστορ αυτό είναι τύπου NMOS. Υπάρχει και το συμπληρωματικό PMOS.



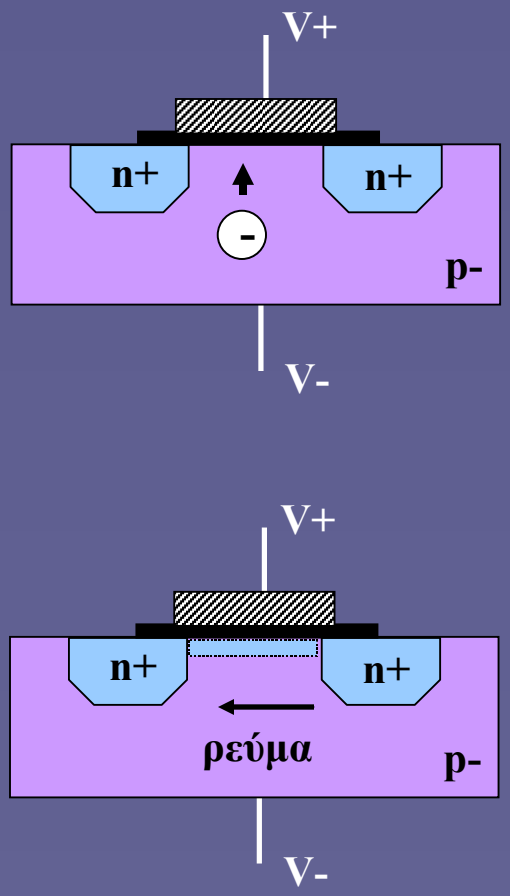
ιδανικό σχήμα..

Λειτουργία του τρανζίστορ MOS(FET)

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

;

Τι συμβαίνει στο τρανζίστορ PMOS;



Κατασκευή ολοκληρωμένων κυκλωμάτων

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

i

Λόγω της απαιτούμενης ακρίβειας, μια γραμμή παραγωγής κοστίζει δισ. $

- Γραμμές παραγωγής
  - Φωτολιθογραφία με μάσκες
  - Διεργασίες στα εκτεθειμένα μέρη
    - Οξείδωση, απόξεση, απόθεση μετάλλου, εμφύτευση ιόντων...
  - Επανάληψη βημάτων
- Ένα σύγχρονο ολοκληρωμένο κύκλωμα μικροεπεξεργαστή
  - Έχει επιφάνεια περίπου 280mm^2
  - Και περιέχει από 100 εκ. έως >1 δισ. τρανζίστορ!

Η συρρίκνωση του τρανζίστορ

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα

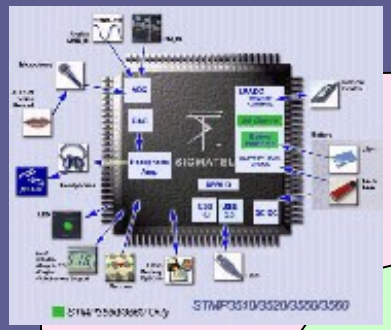
i

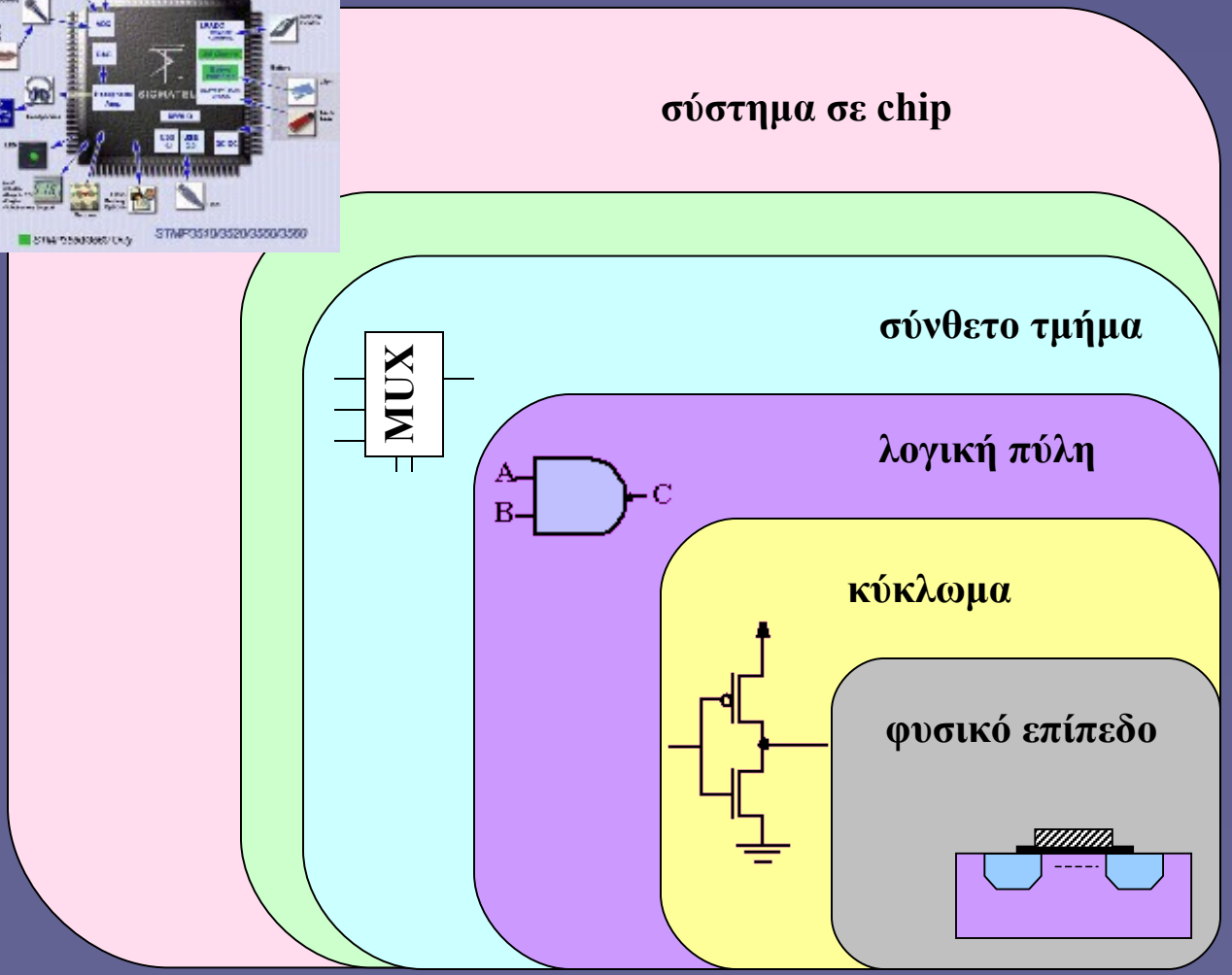
Νόμος του Moore: ο αριθμός των τρανζίστορ ανά ολοκληρωμένο διπλασιάζεται κάθε 1,5-2 χρόνια

- Πλεονεκτήματα
  - Ταχύτερη λειτουργία
    - Πιο γρήγοροι χρόνοι ON-OFF
  - Μικρότερη κατανάλωση ενέργειας
    - Για τον ίδιο αριθμό τρανζίστορ!
  - Μεγαλύτερη ολοκλήρωση
    - Μείωση κόστους παραγωγής και αύξηση λειτουργικότητας
- Τρέχουσα εμπορική τεχνολογία:
  - Μέγεθος (καναλιού) τρανζίστορ = 65nm
- Το μέλλον; 45 και 25 nm

Ψηφιακά Ηλεκτρονικά: Ιεραρχία σχεδίασης

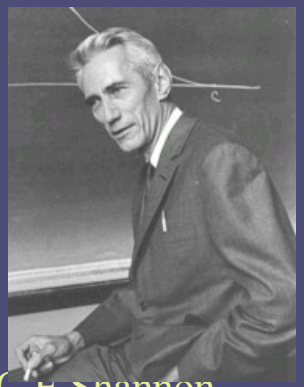
- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα





Ψηφιακά Ηλεκτρονικά και Δυαδική λογική

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα



C.E.Shannon

- Η δυαδική λογική ταιριάζει με την τεχνολογία του τρανζίστορ
  - 2 καταστάσεις: ON-OFF, 1-0
  - Ψηφιακά ηλεκτρονικά (2 στάθμες)
- Δυαδική άλγεβρα Boole
  - Λογική άλγεβρα
  - Συσχέτιση με διακοπτικά κυκλώματα
    - Η εργασία του Shannon (1938)

Άλγεβρα Boole: επανάληψη

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

- $A + B$ (A OR B)
- $A \bullet B$ (ή απλά AB, A AND B)
- $\overline{A}$ (NOT A)
  - $A + 0 = A$ και $A \bullet 1 = A$
  - $A + 1 = 1$ και $A \bullet 0 = 0$
  - $A + \overline{A} = 1$ και $A \bullet \overline{A} = 0$
  - $A + B = B + A$ και $A \bullet B = B \bullet A$
  - $A+(B+C)=(A+B)+C$ και
  - $A(BC)=(AB)C$

Άλγεβρα Boole: επανάληψη

- Υπολογιστικά Συστήματα
- Αρχιτεκτονική Η/Υ
- Οι βασικές μονάδες
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

- $A(B+C) = (AB)+(AC)$ και
- $A+(BC) = (A+B)(A+C)$

- $\overline{(A+B)} = \overline{A} \cdot \overline{B}$ και
- $\overline{(A \cdot B)} = \overline{A} + \overline{B}$ (DeMorgan)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Ψηφιακή Λογική και Σχεδίαση

(σχεδίαση συνδυαστικών κυκλωμάτων)

http://www.ionio.gr/~mistral/tp/comparch/

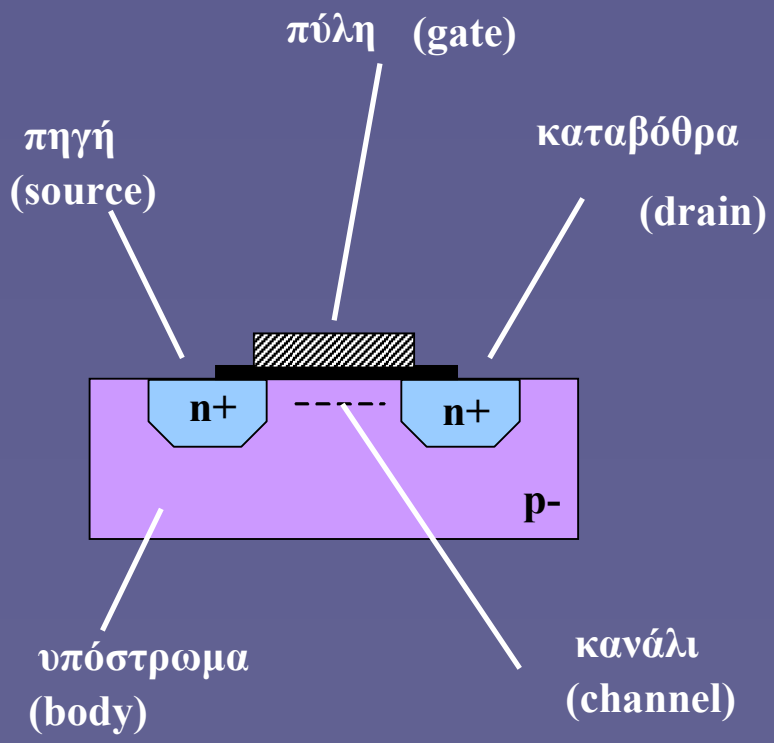
Μ.Στεφανιδάκης

Το τρανζίστορ MOS(FET)

- Ηλεκτρονικά κυκλώματα

i

Το τρανζίστορ αυτό είναι τύπου NMOS. Υπάρχει και το συμπληρωματικό PMOS.

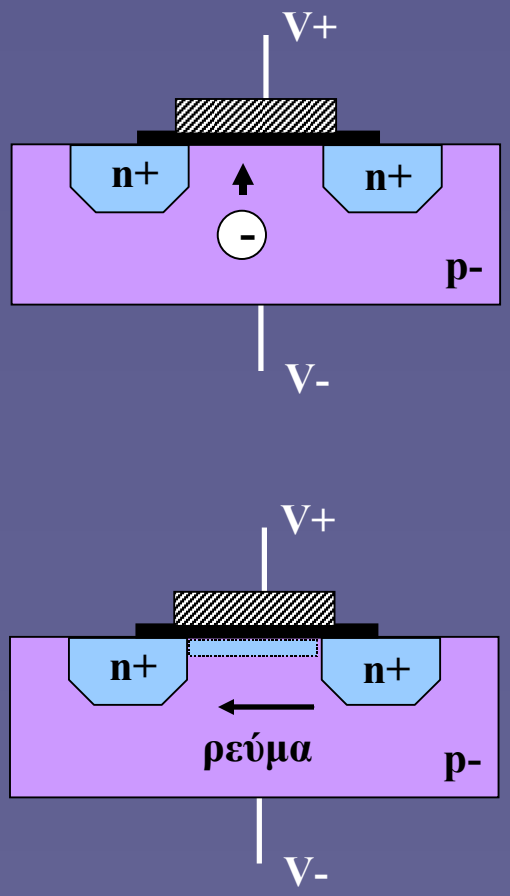


Λειτουργία του τρανζίστορ MOS(FET)

- Ηλεκτρονικά κυκλώματα

;

Τι συμβαίνει στο τρανζίστορ PMOS;



Τρανζίστορ NMOS και PMOS

- Ηλεκτρονικά κυκλώματα

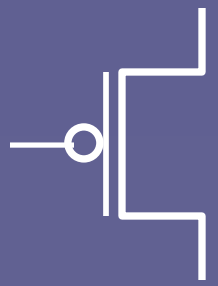
i

Η πλειοψηφία των σημερινών κυκλωμάτων χρησιμοποιεί και τα δύο είδη τρανζίστορ (τεχνολογία CMOS)

NMOS: άγει όταν στην πύλη εφαρμόζεται ‘1’
Περνά ισχυρό ‘0’

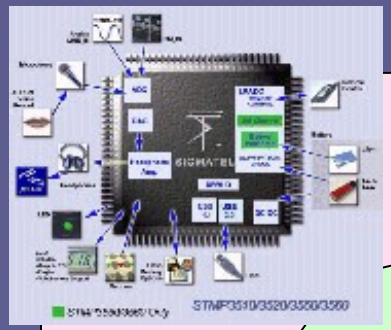
σύμβολα

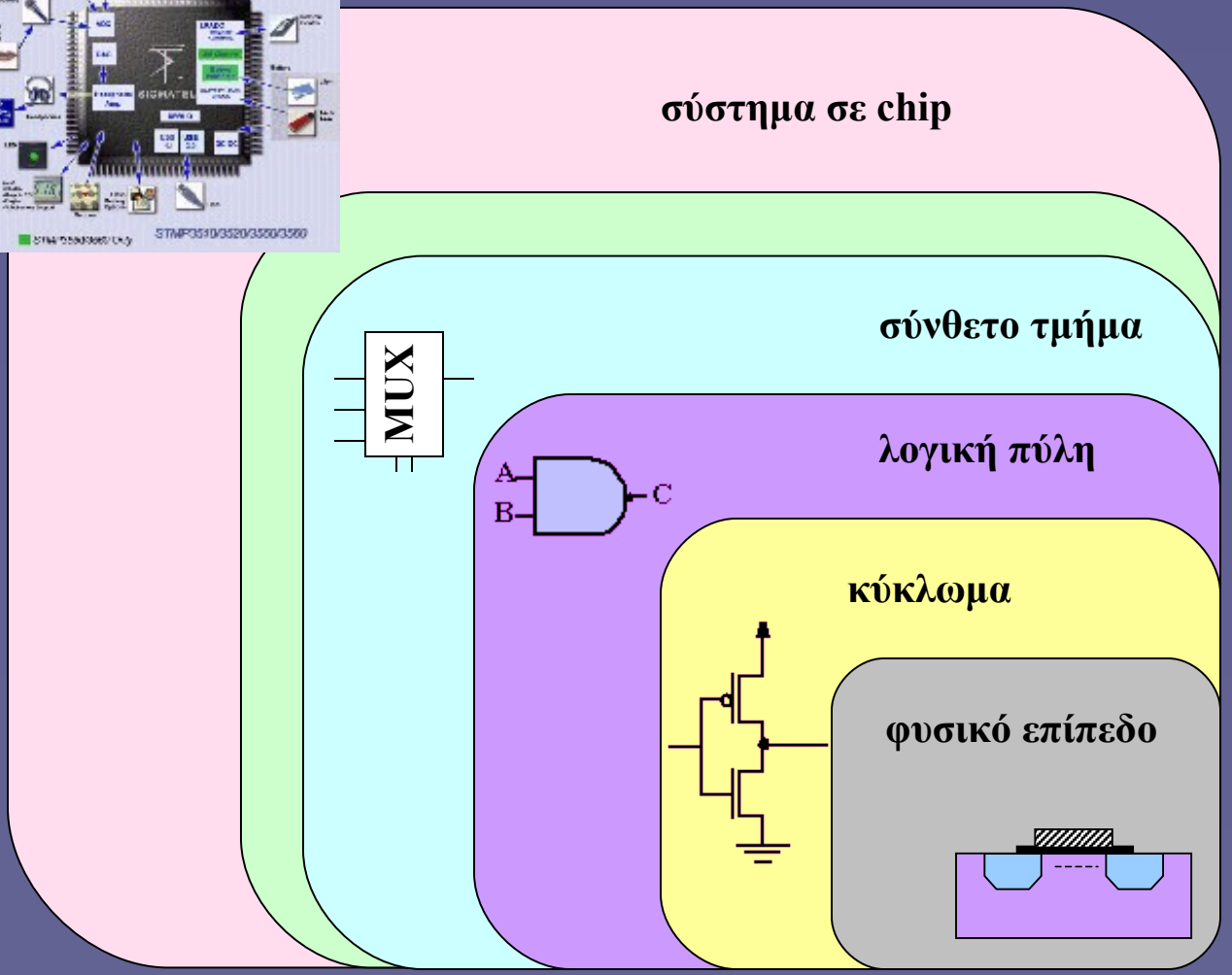
PMOS: άγει όταν στην πύλη εφαρμόζεται ‘0’
Περνά ισχυρό ‘1’



Ψηφιακά Ηλεκτρονικά: Ιεραρχία σχεδίασης

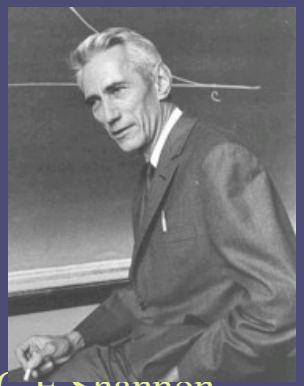
- Ηλεκτρονικά κυκλώματα





Ψηφιακά Ηλεκτρονικά και Δυαδική λογική

- Ηλεκτρονικά κυκλώματα



C.E.Shannon

- Η δυαδική λογική ταιριάζει με την τεχνολογία του τρανζίστορ
  - 2 καταστάσεις: ON-OFF, 1-0
  - Ψηφιακά ηλεκτρονικά (2 στάθμες)
- Δυαδική άλγεβρα Boole
  - Λογική άλγεβρα
  - Συσχέτιση με διακοπτικά κυκλώματα
    - Η εργασία του Shannon (1938)

Άλγεβρα Boole: επανάληψη

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

- $A + B$ (A OR B)
- $A \cdot B$ (ή απλά AB, A AND B)
- $\overline{A}$ (NOT A)
  - $A + 0 = A$ και $A \cdot 1 = A$
  - $A + 1 = 1$ και $A \cdot 0 = 0$
  - $A + \overline{A} = 1$ και $A \cdot \overline{A} = 0$
  - $A + B = B + A$ και $A \cdot B = B \cdot A$
  - $A+(B+C)=(A+B)+C$ και
  - $A(BC)=(AB)C$

Άλγεβρα Boole: επανάληψη

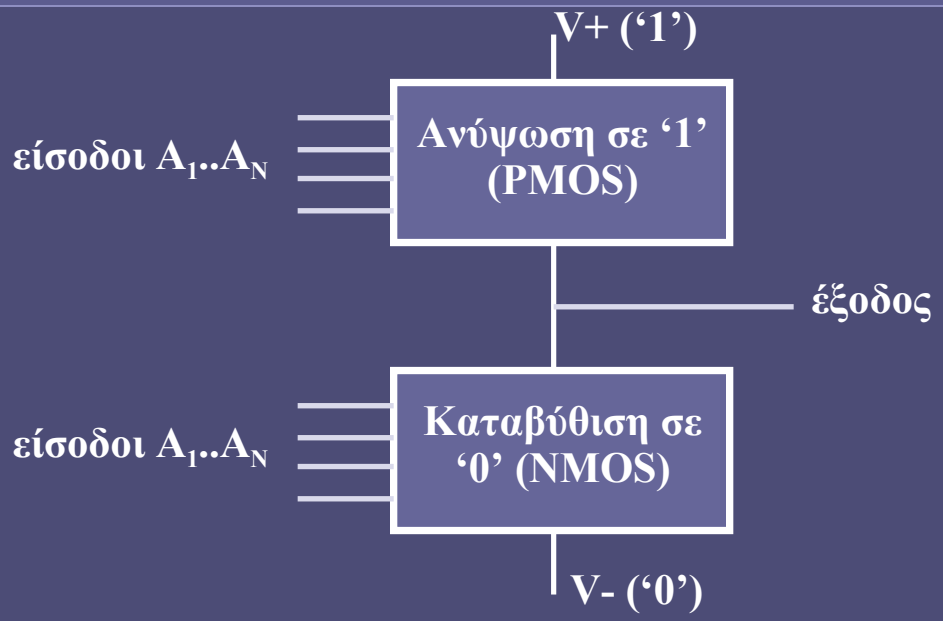
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

- $A(B+C) = (AB)+(AC)$ και
- $A+(BC) = (A+B)(A+C)$

- $\overline{(A+B)} = \overline{A} \cdot \overline{B}$ και
- $\overline{(A \cdot B)} = \overline{A} + \overline{B}$ (DeMorgan)

Απλές λογικές πύλες

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες

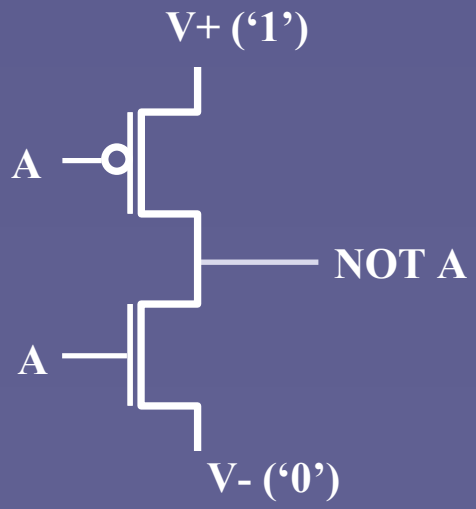


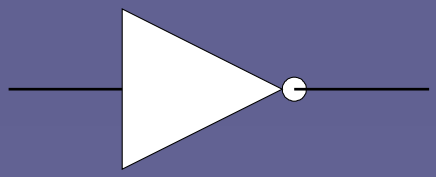
- Στατική τεχνολογία CMOS
  - Αναστρέφουσες συναρτήσεις
    - NOT, NAND, NOR …
  - Αξιόπιστη λειτουργία, εύκολη σχεδίαση
  - Όχι πάντα η αποδοτικότερη λύση

Η πύλη NOT

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες

A	Y
0	1
1	0

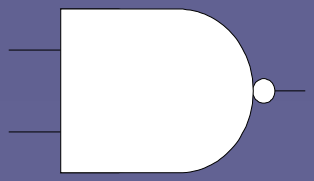


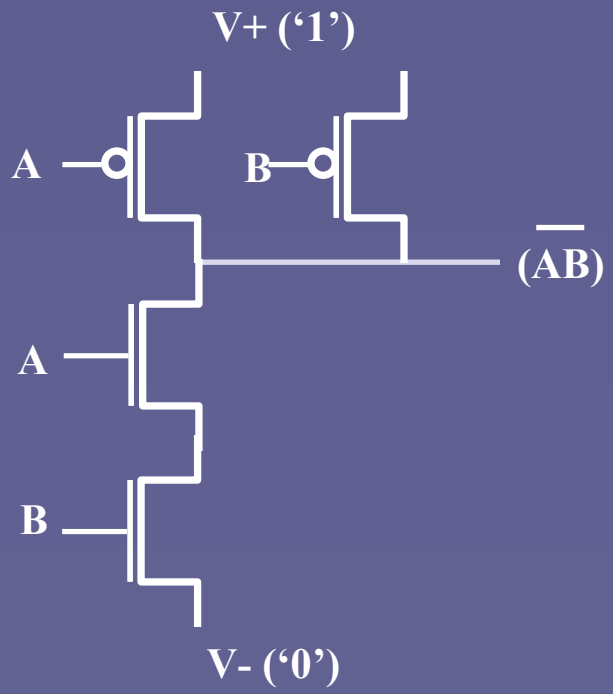


Η πύλη NAND

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

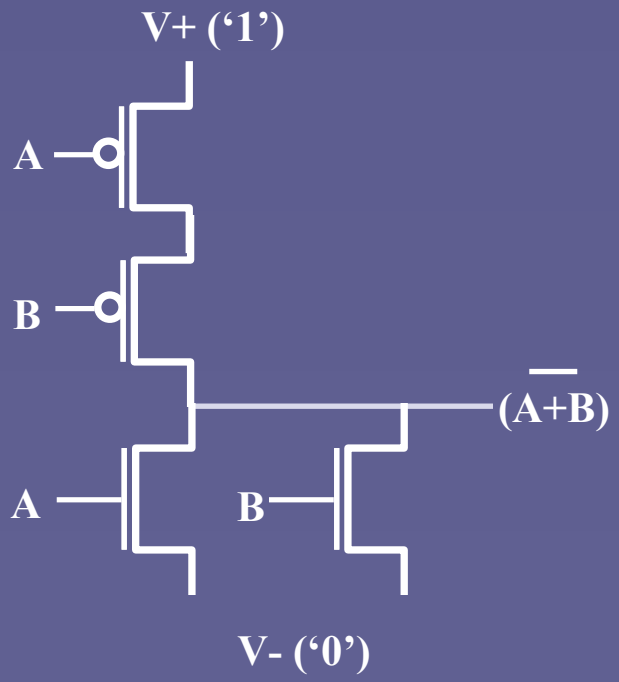


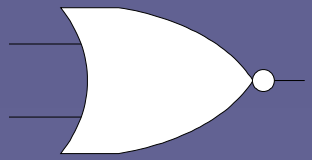


Η πύλη NOR

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες

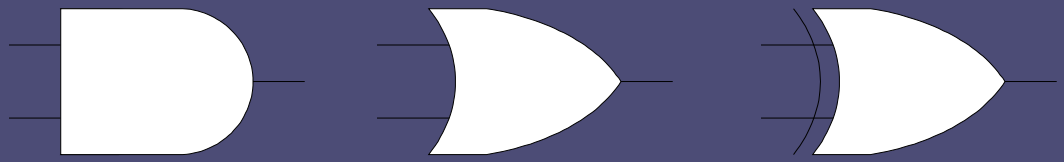
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0





Άλλες λογικές πύλες

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες

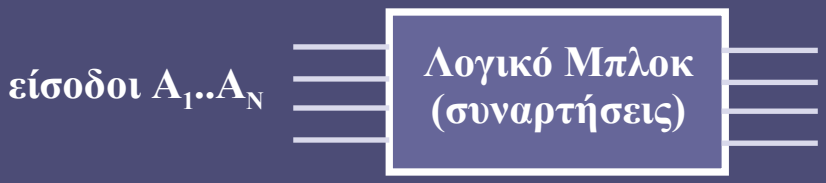


AND OR XOR

- Ως συνδυασμός των βασικών πυλών
  - NOT, NAND, NOR
- Υπάρχουν και εναλλακτικές μέθοδοι σχεδίασης!

Συνδυαστική Λογική

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική



- Μπλοκ λογικών συναρτήσεων
  - Οι έξοδοι εξαρτώνται αποκλειστικά από την τρέχουσα τιμή των εισόδων
  - Δεν υπάρχει μνήμη προηγούμενων καταστάσεων
  - Αλλαγή των εισόδων θα επηρεάσει τις εξόδους μετά από χρονικό διάστημα (καθυστέρηση διάδοσης)
  - Η συνάρτηση που υλοποιεί το μπλοκ μπορεί να εκφραστεί με έναν πίνακα αλήθειας

Υλοποίηση συναρτήσεων

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική

!

Πάντοτε προσπαθούμε να απλοποιήσουμε τις συναρτήσεις

(πίνακες Karnaugh ή άλλες υπολογιστικές μέθοδοι)

A	B	C	Y	Ελαχιστόροι	Μεγιστόροι
0	0	0	1	a'b'c'	a+b+c
0	0	1	0	a'b'c	a+b+c'
0	1	0	0	a'bc'	a+b'+c
0	1	1	1	a'bc	a+b'+c'
1	0	0	0	ab'c'	a'+b+c
1	0	1	0	ab'c	a'+b+c'
1	1	0	1	abc'	a'+b'+c
1	1	1	1	abc	a'+b'+c'

- $Y = a'b'c' + a'bc + abc' + abc$
- $Y = (a+b+c')(a+b'+c)(a'+b+c)(a'+b+c')$

Βασικά συνδυαστικά τμήματα

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

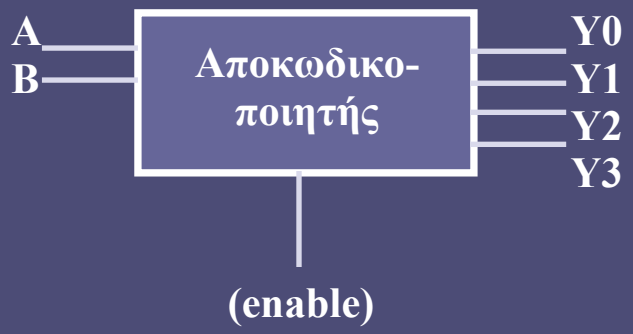
- Αποκωδικοποιητής (decoder)
  - Ν είσοδοι ενεργοποιούν 1 από 2^N εξόδους
- Πολυπλέκτης (multiplexer)
  - Ν είσοδοι επιλέγουν 1 από 2^N εισόδους
- Αθροιστής (adder)
  - Πρόσθεση δυαδικών ψηφίων
  - Διάφοροι τρόποι παραγωγής-διάδοσης κρατουμένου

Αποκωδικοποιητής (decoder)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Πώς υλοποιείται ένας αποκωδικο-ποιητής;



A	B	Y_0	Y_1	Y_2	Y_0
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

- Αποκωδικοποιητής N-σε-2^N
  - N είσοδοι ενεργοποιούν μία από 2^N εξόδους

Πολυπλέκτης (multiplexer)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Πώς υλοποιείται ένας πολυπλέκτης;

S_1	S_0	Y
0	0	A
0	1	B
1	0	C
1	1	D

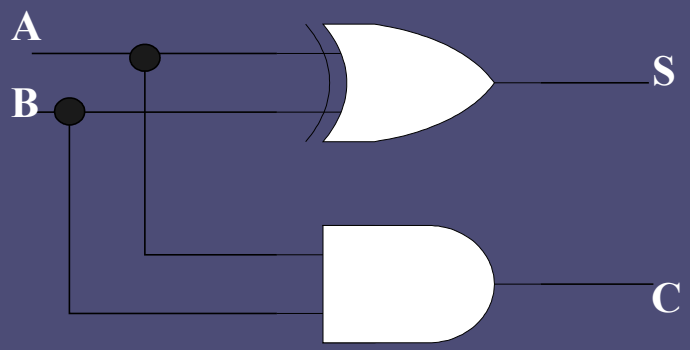
- Πολυπλέκτης 2^N γραμμών σε 1
  - Επιλογή μίας από 2^N εισόδους με τη βοήθεια N σημάτων ελέγχου

Ημιαθροιστής (half-adder)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Αν απαιτείται πρόσθεση αριθμών με περισσότερα bits;



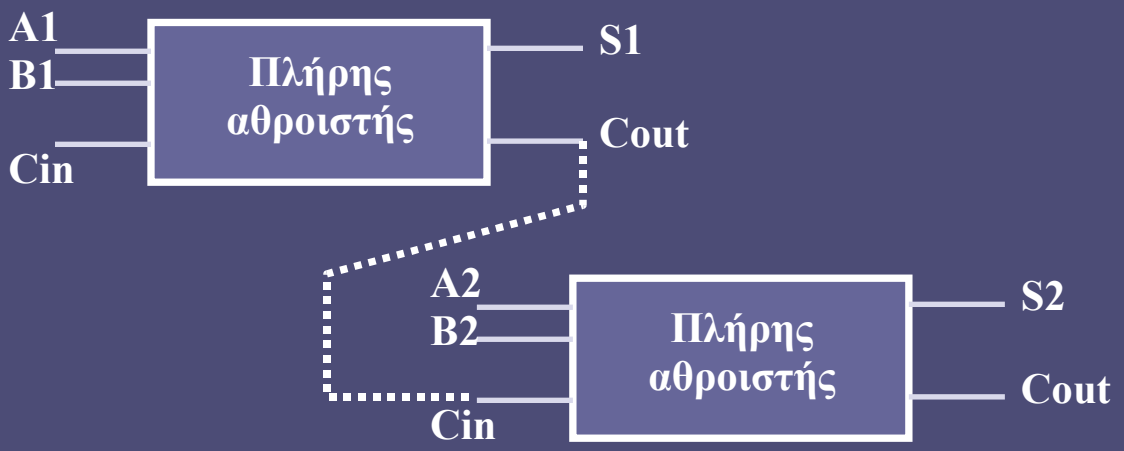
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Πλήρης αθροιστής (full-adder)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Πώς υλοποιείται ένας πλήρης αθροιστής;



- Πολλαπλά τμήματα πλήρη αθροιστή
  - Όμως: πόσο γρήγορα διαδίδεται το κρατούμενο; (ripple carry)
  - Τεχνικές πρόβλεψης κρατουμένου (carry look-ahead)

Πέρα από τη συνδυαστική λογική...

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

- Στο επόμενο μάθημα...
  - Πώς εισάγω την έννοια της κατάστασης ενός λογικού τμήματος;
  - Που φυλάσσεται η κατάσταση;
  - Πότε ενημερώνεται;
  - Πώς συγχρονίζονται τα διάφορα τμήματα λογικής;

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Ψηφιακή Λογική και Σχεδίαση

(στοιχεία μνήμης και μέθοδοι χρονισμού)

http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Πέρα από τη συνδυαστική λογική

- Ακολουθιακή λογική

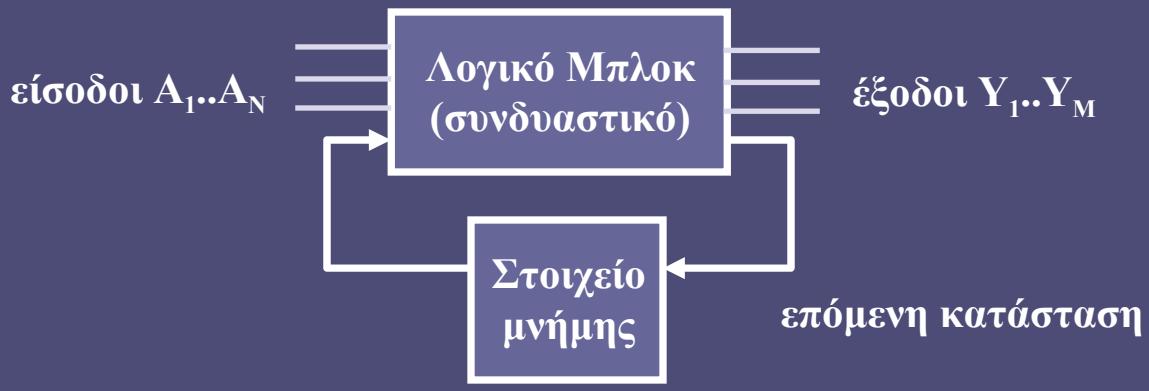
- Βασικά ερωτήματα
  - Πώς εισάγω την έννοια της κατάστασης ενός λογικού τμήματος;
  - Που φυλάσσεται η κατάσταση;
  - Πότε ενημερώνεται;
  - Πώς συγχρονίζονται τα διάφορα τμήματα λογικής;

Ακολουθιακή λογική

- Ακολουθιακή λογική

;

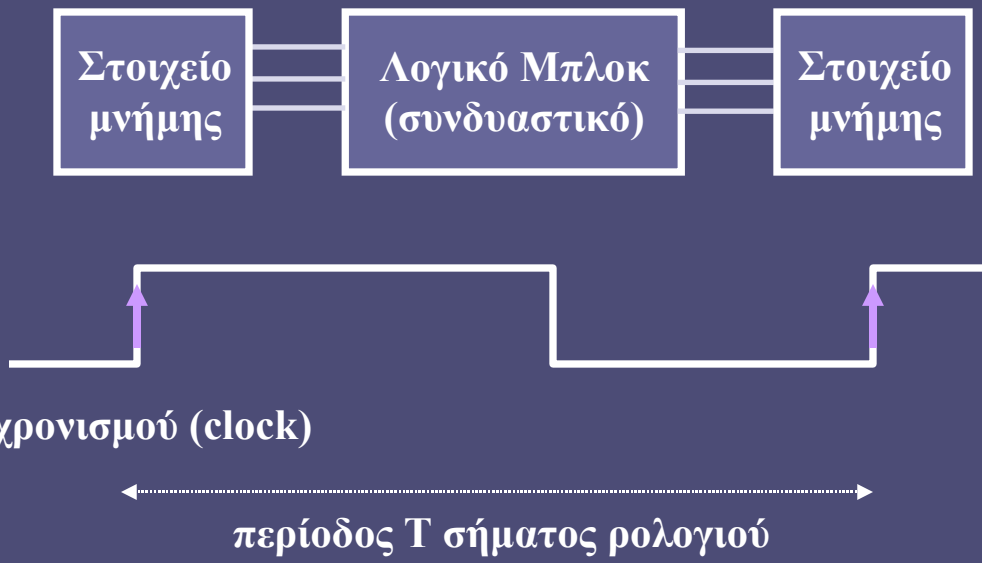
Πότε τελειώνει η τρέχουσα και πότε αρχίζει η επόμενη κατάσταση;



- Η έννοια της κατάστασης: οι έξοδοι εξαρτώνται από τις εισόδους ΚΑΙ από την προηγούμενη κατάσταση

Χρονισμός σε ακολουθιακή λογική

- Ακολουθιακή λογική



- Η πλειοψηφία των ψηφιακών κυκλωμάτων χρησιμοποιεί ένα περιοδικό σήμα συγχρονισμού (ρολόι - clock)

!

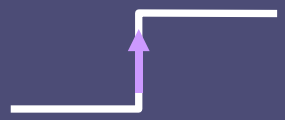
Το clock καθορίζει τη **χρονική στιγμή αποθήκευσης** στα στοιχεία μνήμης

Χρονισμός στις ακμές του ρολογιού

- Ακολουθιακή λογική

!

Η είσοδος πρέπει να είναι σταθερή λίγο **πριν** (setup) και λίγο **μετά** (hold) την ακμή του ρολογιού



Σήμα χρονισμού (clock)

- Η απλούστερη λύση: αποθήκευση στην ανερχόμενη (ή κατερχόμενη) ακμή του ρολογιού
  - Τη στιγμή αυτή η είσοδος αποθηκεύεται στο στοιχείο μνήμης
  - Και εμφανίζεται στην έξοδο μετά από την καθυστέρηση διάδοσης του στοιχείου μνήμης.

Χρονισμός και διάδοση σημάτων

- Ακολουθιακή λογική

;

Μπορεί η περίοδος του ρολογιού να γίνει όσο μικρή θέλουμε;

Οι είσοδοι περνούν στο στοιχείο μνήμης

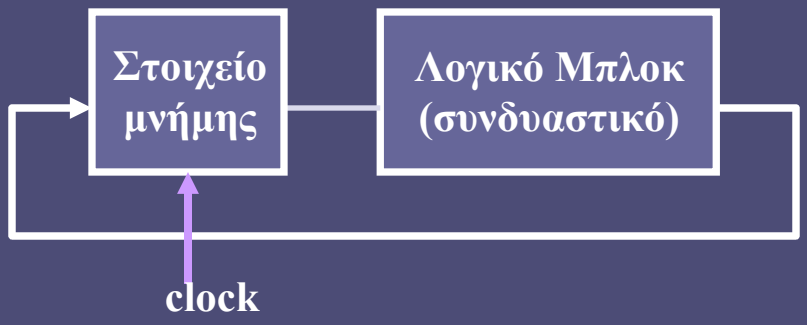
Και εμφανίζονται στις εισόδους του λογικου μπλοκ

Προκαλώντας αλλαγές στις εξόδους του μπλοκ

Στην επόμενη ακμή αποθηκεύονται στο δεύτερο στοιχείο μνήμης

Ανάδραση και στοιχεία μνήμης

- Ακολουθιακή λογική



- Απολύτως επιτρεπτή συνδεσμολογία!
- Λόγω του σήματος ρολογιού δεν δημιουργείται πρόβλημα με την ανατροφοδότηση του σήματος εξόδου στην είσοδο

Συγχρονισμός τμημάτων

- Ακολουθιακή λογική

- Σύγχρονο σύστημα
  - Κοινή πολιτική χρονισμού
  - Κοινό ρολόι
  - Σχεδίαση τήρησης χρόνων διάδοσης
- Τμήματα με διαφορετικά ρολόγια
  - Ή ασύγχρονες είσοδοι
  - Αδυναμία τήρησης χρόνων setup – hold
  - Τα στοιχεία μνήμης μπορούν να περιέλθουν σε απροσδιόριστη κατάσταση (metastability)
  - Ειδικά κυκλώματα συγχρονισμού

Στοιχεία μνήμης

- Ακολουθιακή λογική
- Στοιχεία μνήμης

- Μαζί με λογικά κυκλώματα
  - Για αποθήκευση ενδιάμεσης κατάστασης μεταξύ συνδυαστικών συναρτήσεων
  - Μικρή χωρητικότητα
  - Παράδειγμα: οι καταχωρητές της ΚΜΕ
- Σε μεγάλες συστοιχίες μνήμης
  - Κύρια μνήμη συστήματος
  - Μεγάλη χωρητικότητα
  - Μεγαλύτερη ολοκλήρωση κυκλωμάτων

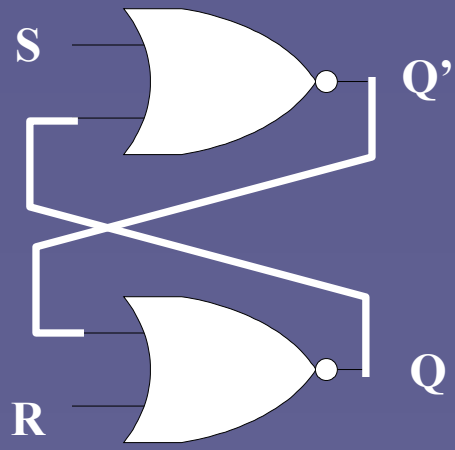
!

Η έξοδος ενός στοιχείου μνήμης εξαρτάται από τις εισόδους του και την τρέχουσα κατάστασή του!

Απλά στοιχεία μνήμης χωρίς ρολόι

- Ακολουθιακή λογική
- Στοιχεία μνήμης

S	R	Q	Q’
0	0	Q	Q’
0	1	0	1
1	0	1	0
~~1~~	~~1~~	~~0~~	~~0~~

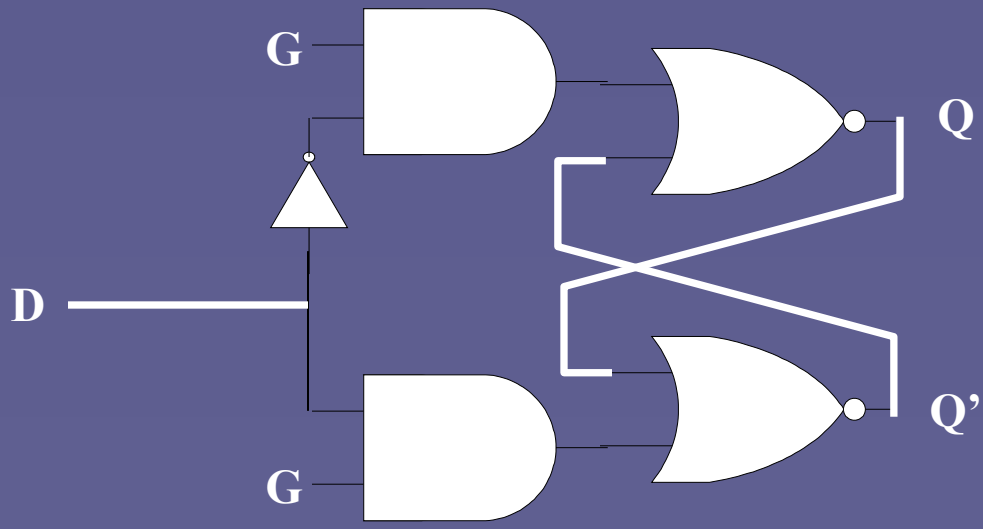


!

Τα S και R δεν επιτρέπεται να είναι ταυτόχρονα ‘1’

Ο "μανδαλωτής" (latch)

- Ακολουθιακή λογική
- Στοιχεία μνήμης



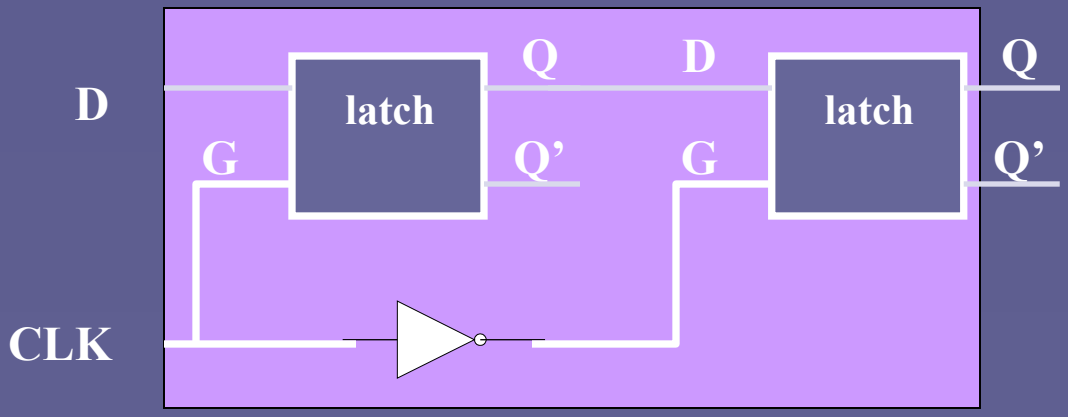
D	G	Q	Q'
x	0	Q	Q'
0	1	0	1
1	1	1	0

!

Το latch περνά την είσοδο στην έξοδο ανάλογα με τη **στάθμη** του G

D flip-flop

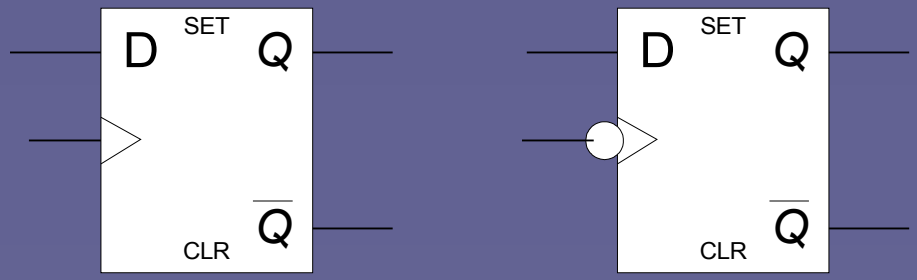
- Ακολουθιακή λογική
- Στοιχεία μνήμης



Αποθήκευση εισόδου στην ακμή του σήματος ρολογιού

;

Σε ποια ακμή του CLK αποθηκεύεται η είσοδος D;

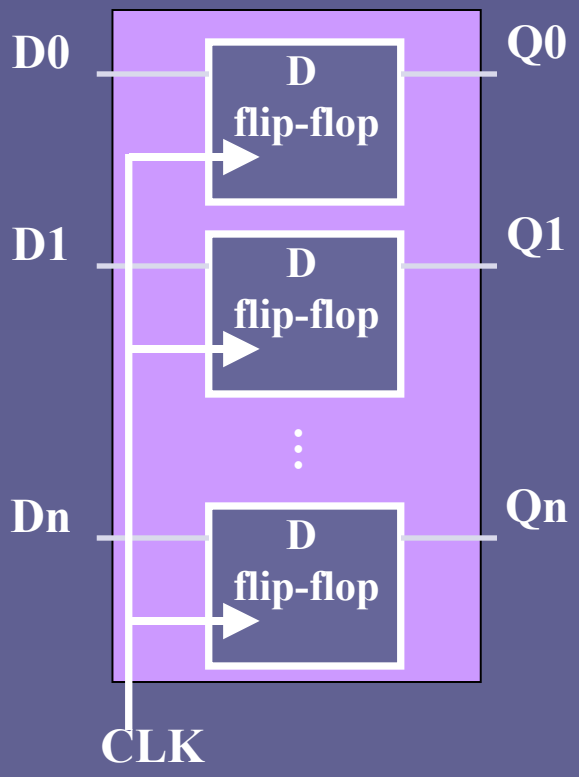


Καταχωρητές (registers)

- Ακολουθιακή λογική
- Στοιχεία μνήμης

;

Ποιο το μήκος της λέξης που αποθηκεύεται σε έναν καταχωρητή;



Ομάδα στοιχείων flip-flop με κοινό σήμα ρολογιού

Register file

- Ακολουθιακή λογική
- Στοιχεία μνήμης
- Register File

- **Ομάδα καταχωρητών, βασικό στοιχείο μιας ΚΜΕ**
- **Ανάγνωση και εγγραφή**
- **Δυνατότητα ταυτόχρονης λειτουργίας σε περισσότερους από έναν καταχωρητές**

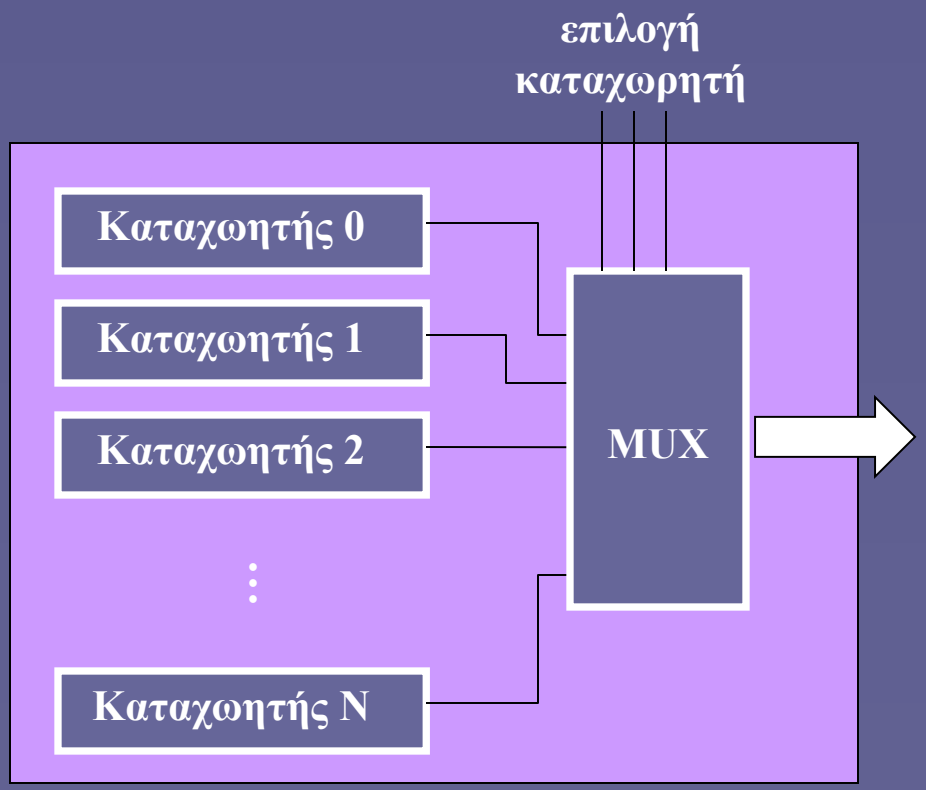
Οι σύγχρονες ΚΜΕ διαθέτουν τουλάχιστον 32 καταχωρητές γενικού σκοπού

Register file: ανάγνωση

- Ακολουθιακή λογική
- Στοιχεία μνήμης
- Register File

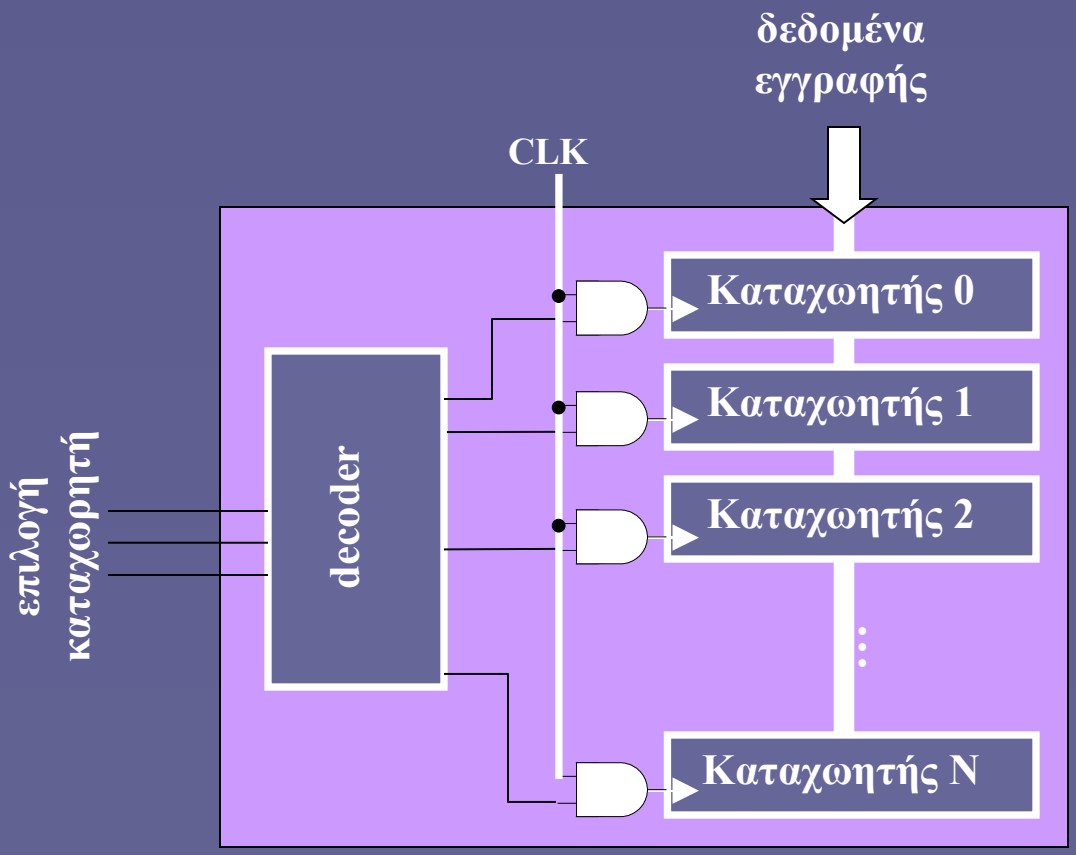
;

Πώς μπορούμε να διαβάσουμε ταυτόχρονα 2 καταχωρητές (σε διαφορετικές εξόδους);



Register file: εγγραφή

- Ακολουθιακή λογική
- Στοιχεία μνήμης
- Register File



!

Τα δεδομένα προς εγγραφή θα πρέπει να είναι σταθερά κατά τους χρόνους setup και hold!

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Αρχιτεκτονικές Συνόλου Εντολών

(Instruction Set Architectures - ISA)

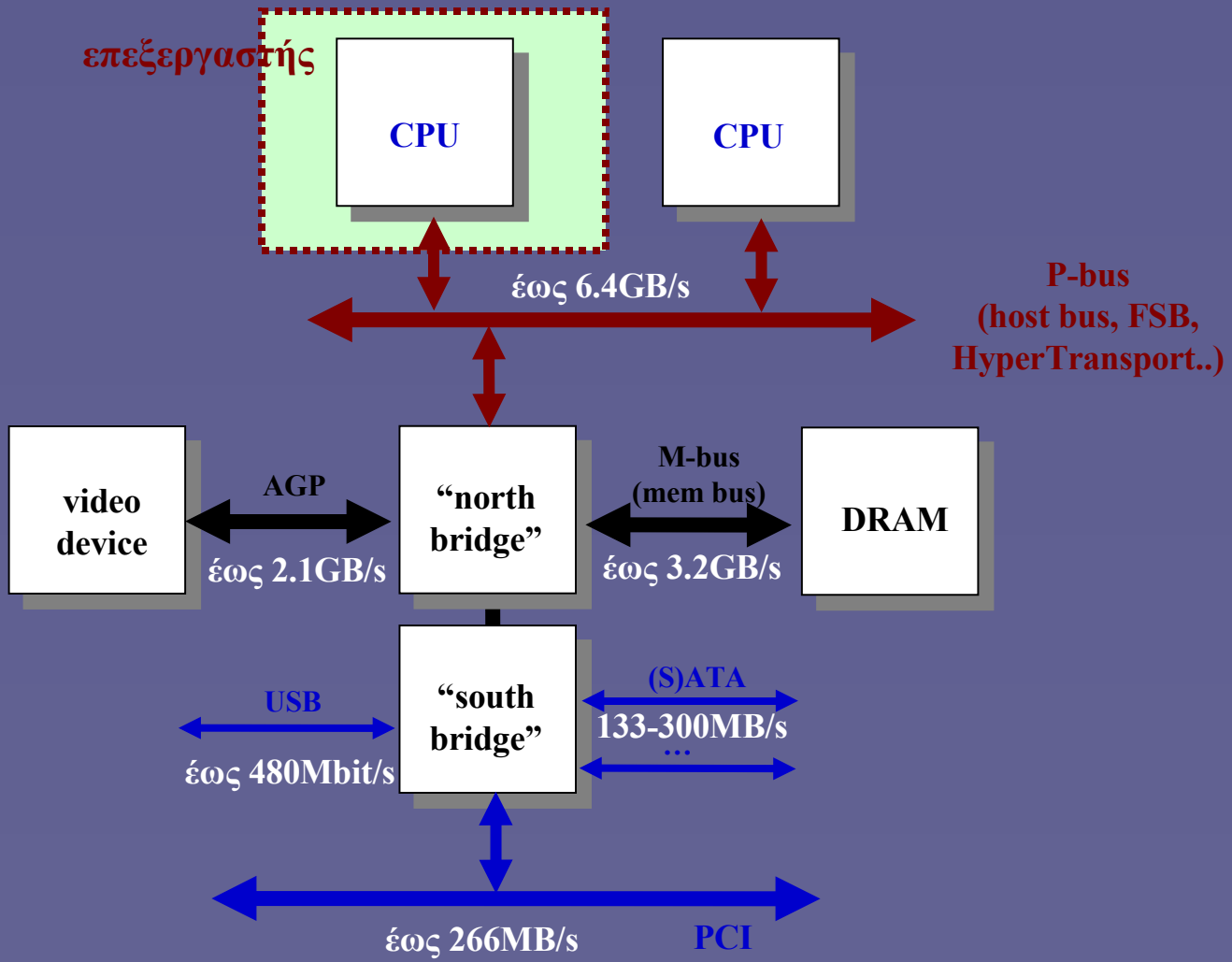
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Ένα τυπικό υπολογιστικό σύστημα σήμερα

- Ο επεξεργαστής

- **Επεξεργαστής**
- Μνήμη
- Συσκευές Ε/Ε

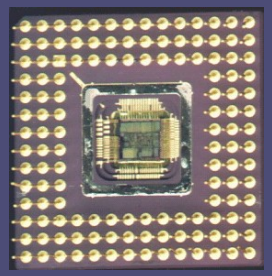


Ο (μικρο)επεξεργαστής

- Ο επεξεργαστής

;

Τι περιέχεται στη συσκευασία ενός μικροεπεξεργαστή σήμερα;



- (Micro)processor
  - Ψηφιακό σύστημα που εκτελεί υπολογισμούς σε πολλαπλά βήματα
    - Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)
    - Central Processing Unit (CPU)
  - Μέρος ενός ευρύτερου υπολογιστικού συστήματος (ή "υπολογιστή")
    - **γενικού** ή ειδικού σκοπού
  - Μπορεί να περιέχει σήμερα πολλαπλές υπομονάδες επεξεργασίας

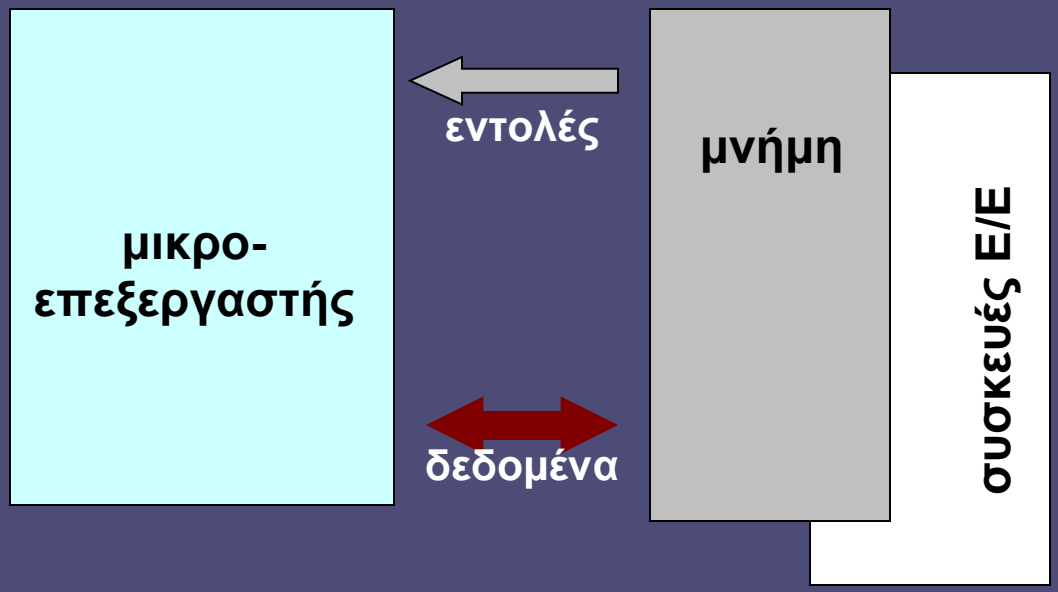
Είδη μικροεπεξεργαστών

- Ο επεξεργαστής

- Γενικού σκοπού
  - Οι περισσότερο γνωστοί επεξεργαστές (PCs)
- Συνεπεξεργαστές (co-processors)
  - Ειδικές λειτουργίες
- Μικροελεγκτές (microcontrollers)
  - Συστήματα ελέγχου
- Μέσα σε Systems-on-Chip
  - Επεξεργαστικοί πυρήνες

Το μοντέλο von Neumann

- Ο επεξεργαστής



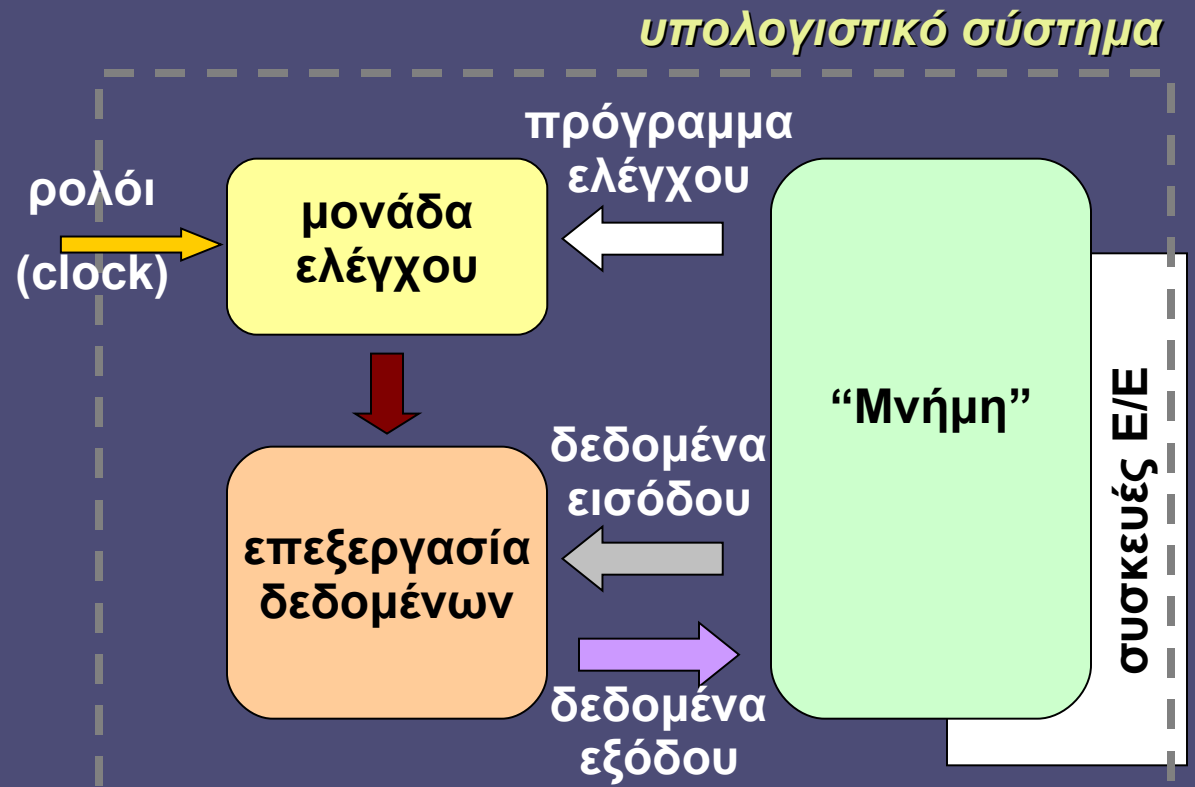
;

Σε ποια μορφή αποθηκεύονται οι εντολές;

- Το πρόγραμμα εκτέλεσης, όπως και τα δεδομένα, αποθηκεύονται στη μνήμη του υπολογιστή
  - **Stored-program computer**

Εκτέλεση ακολουθίας λειτουργιών

- Ο επεξεργαστής



;

Ποιος ο ρόλος του σήματος ρολογιού;

Εκτέλεση εντολών: ο κύκλος μηχανής

- Ο επεξεργαστής

;

Τι συμβαίνει με τις εξωτερικές διακοπές (**interrupts**);

Ο χρόνος εκτέλεσης είναι ο ίδιος για όλες τις εντολές;

για πάντα

Fetch: Φέρε την επόμενη εντολή προς εκτέλεση από τη μνήμη

Decode: Αποκωδικοποίησε την εντολή (προετοίμασε τα σήματα ελέγχου και τις πηγές των δεδομένων)

Execute: Εκτέλεσε την απαιτούμενη πράξη/λειτουργία

Store: Αποθήκευσε τα αποτελέσματα (εάν απαιτείται)

Εκτέλεση εντολών

- Ο επεξεργαστής

- Επόμενη εντολή προς εκτέλεση
  - Program Counter (PC): η διεύθυνση της θέσης μνήμης όπου περιέχεται η επόμενη εντολή
    - Σειριακή αύξηση διεύθυνσης μετά την εκτέλεση εντολής
    - Ή μεταπήδηση σε νέα θέση μνήμης (διακλάδωση)
- Εκκίνηση εκτέλεσης
  - Με την εφαρμογή τάσης ο PC παίρνει μια προκαθορισμένη τιμή
    - BIOS
- Τερματισμός επανάληψης κύκλου μηχανής
  - Συμβατικά, ποτέ!
  - Περιπτώσεις SLEEP για μείωση κατανάλωσης ενέργειας

Αρχιτεκτονική Συνόλου Εντολών

- Ο επεξεργαστής
- ISA

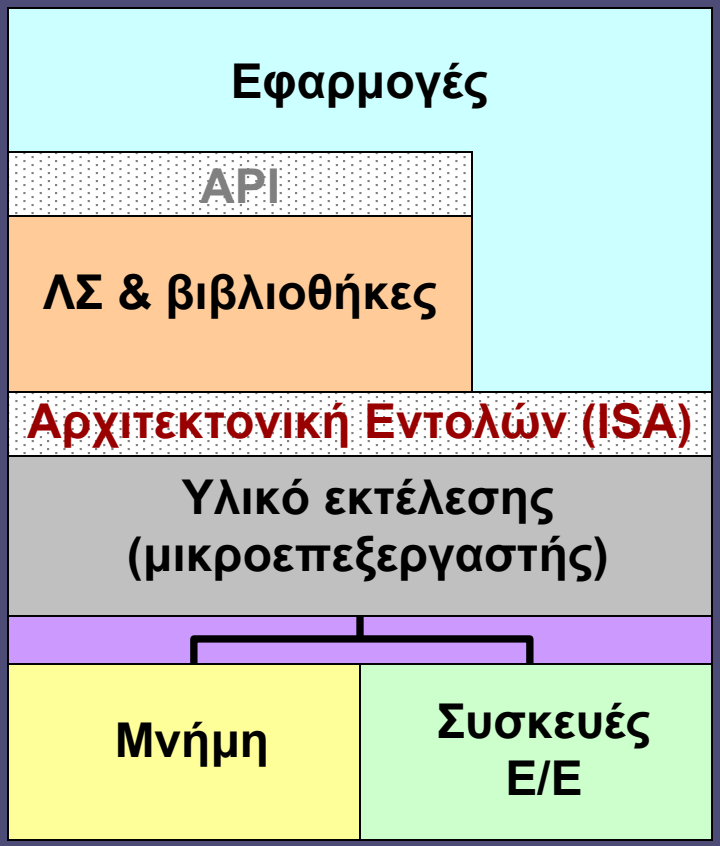
- **Instruction Set Architecture (ISA)**
  - Το ορατό μέρος ενός υπολογιστικού συστήματος για τον προγραμματιστή (μεταγλωττιστή)
  - Δεκαετία 60-70: συνώνυμο του όρου αρχιτεκτονική Η/Υ
    - « η δομή ενός υπολογιστή, την οποία ο προγραμματιστής πρέπει να γνωρίζει για να γράψει ένα σωστό (χρονικά ανεξάρτητο) πρόγραμμα σε γλώσσα μηχανής για τον υπολογιστή αυτόν» (IBM)

Η διεπαφή ISA στην ιεραρχία επιπέδων

- Ο επεξεργαστής
- ISA

;

Τι ακριβώς περιγράφει η διεπαφή ISA;



- Αρχιτεκτονική Εντολών (ISA)
  - Η διεπαφή υλικού-λογισμικού

Αρχιτεκτονική Συνόλου Εντολών

- Ο επεξεργαστής
- ISA

- Τι περιγράφει;
  - Διαθέσιμες πράξεις/λειτουργίες
    - Κωδικοποίηση λειτουργιών
  - Μορφή των δεδομένων εισόδου-εξόδου
    - Έντελα (operands)
  - Μέθοδοι προσπέλασης μνήμης
    - Προέλευση των δεδομένων
  - Χώροι προσωρινής αποθήκευσης
    - Καταχωρητές
  - Διακοπές και καταστάσεις σφάλματος
    - Ποια η "αντίδραση" του επεξεργαστή

Σχεδιασμός Συνόλου Εντολών

- Ο επεξεργαστής
- ISA

- Συμβιβασμός μεταξύ:
  - Κόστους και απόδοσης υλικού
  - Βαθμού υποστήριξης λογισμικού
  - Άλλων παραγόντων όπως η κατανάλωση ενέργειας
    - Το υλικό καταναλώνει ενέργεια, υπό τον έλεγχο του λογισμικού όμως!
  - Μη επικάλυψης λειτουργιών εντολών (orthogonality)

Κωδικοποίηση Εντολών

- Ο επεξεργαστής
- ISA

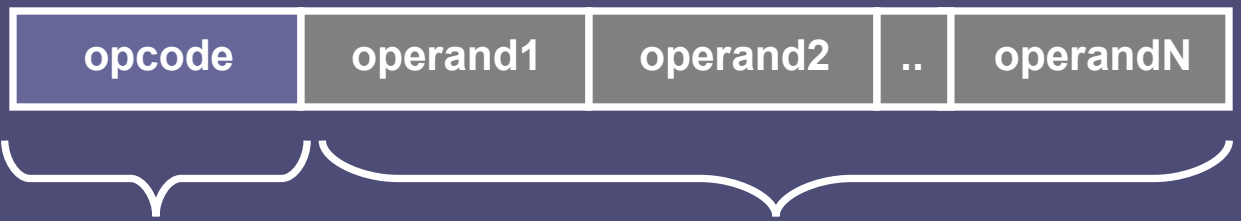
;

Γιατί είναι ταχύτερη η λήψη και αποκωδικοποίηση των εντολών σταθερού μήκους;

- Σειρά δυαδικών ψηφίων
  - Μεταβλητού μήκους
    - Περισσότερο συμπαγή προγράμματα
    - Πολυπλοκότερο υλικό!
  - Σταθερού μήκους
    - Απλούστερη και ταχύτερη λήψη-αποκωδικοποίηση
    - Μεγαλύτερα προγράμματα
    - Μέθοδοι συμπίεσης

Κωδικοποίηση Εντολών

- Ο επεξεργαστής
- ISA



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: κατηγορίες λειτουργιών

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

- Βασικές κατηγορίες
  - Αριθμητικές και λογικές πράξεις
  - Μεταφορά δεδομένων
    - Από-πρός Καταχωρητές και Μνήμη
  - Έλεγχος ροής εκτέλεσης
    - Διακλαδώσεις και κλήσεις ρουτινών
- Άλλες κατηγορίες
  - Ειδικές εντολές συστήματος
    - ΛΣ, ιδεατή μνήμη
  - Επεξεργασία πολλαπλών δεδομένων
    - Χρήσιμο για γραφικά, σειρές χαρακτήρων, multimedia

Αριθμητικές εντολές και μεταφορά δεδομένων

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

;

Τι συμβολίζουν τα R1, R2 ..; Πώς αναπαρίστανται μέσα στην εντολή;

- Αριθμητικές-λογικές πράξεις
  - Πηγές δεδομένων και προορισμός
    - add R1, R2, R3 // R3 = R1+R2

- Μεταφορά δεδομένων
  - Πηγή δεδομένων και προορισμός
  - Μήκος μεταφερόμενης λέξης (ενδεχομένως)
    - load R1, 0x7FF0 // R1 = mem[0x7FF0]

Εντολές διακλάδωσης

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

- **Διακλάδωση**
  - Με ή χωρίς συνθήκη
    - bne R1, R2, +8 // branch if not R1==R2
  - Σε απόλυτη διεύθυνση
    - jump 0xFF97DE00
  - Σχετικά ως προς την τρέχουσα θέση (offset)
    - jump +130 // offset = +130
    - **Ο παραγόμενος κώδικας μπορεί να τοποθετηθεί οπουδήποτε στη μνήμη**

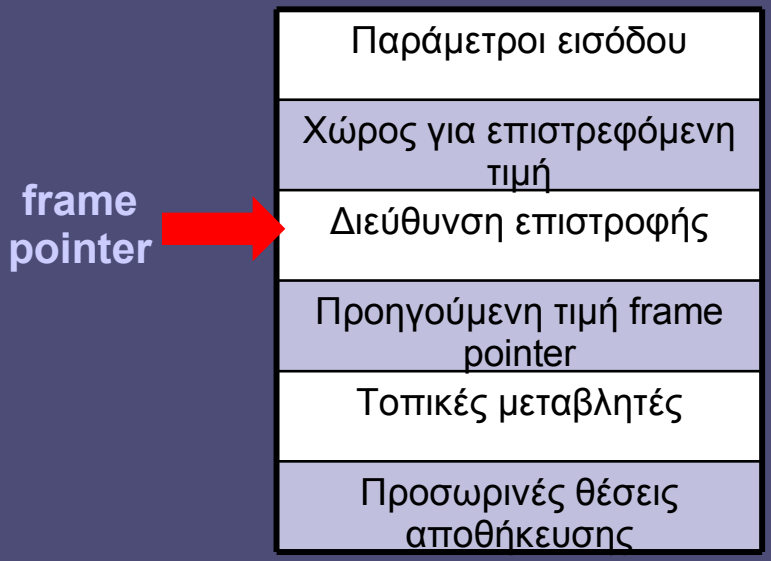
Εντολές διακλάδωσης

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

;

Πού θα μπορούσε να αποθηκευτεί η διεύθυνση επιστροφής μετά από κλήση συνάρτησης;

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



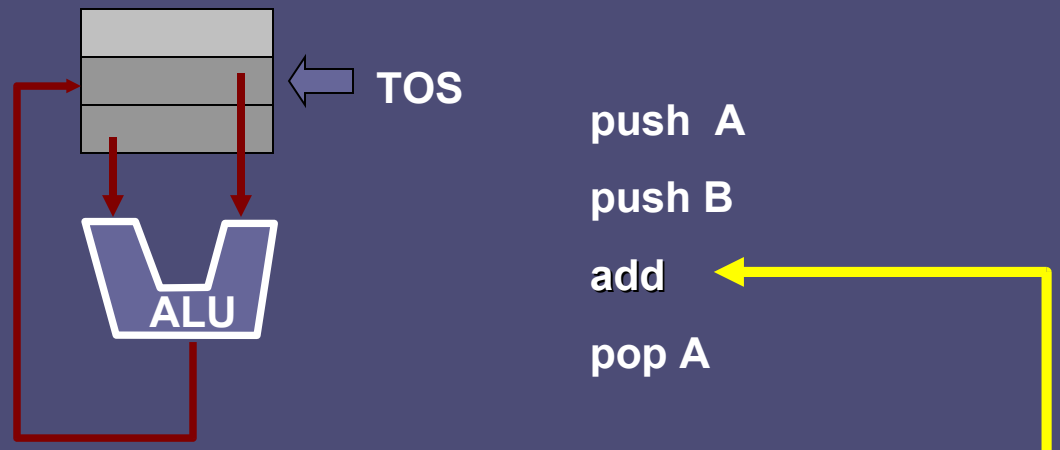
Προέλευση και αποθήκευση δεδομένων

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων

- Προέλευση δεδομένων – αποθήκευση αποτελεσμάτων μιας πράξης
  - **Operand addressing**
  - Εξαρτάται από την αρχιτεκτονική του επεξεργαστή
  - Στους πρώτους επεξεργαστές
    - Stack (σωρός-στοίβα)
    - Accumulator (συσσωρευτής)
  - Μεταγενέστεροι υπολογιστές
    - Έντελα Καταχωρητή-Μνήμης
    - Έντελα Καταχωρητή-Καταχωρητή (load-store)

Αρχιτεκτονική σωρού (stack)

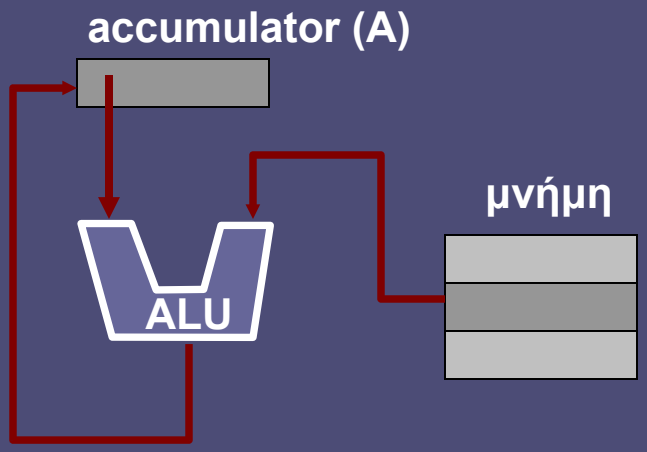
- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων



- Τα έντελα προσδιορίζονται έμμεσα
  - Δεν περιγράφονται στην εντολή!
  - 0-address architecture
  - Δημοφιλές σχήμα κατά τη δεκαετία του 60
  - Δύσκολη προσπέλαση σωρού, απαιτούνται πολλαπλές αντιμεταθέσεις και αντιγραφές
  - Το σχήμα επιζεί στην αρχιτεκτονική της Java VM

Αρχιτεκτονική συσσωρευτή (accumulator)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων



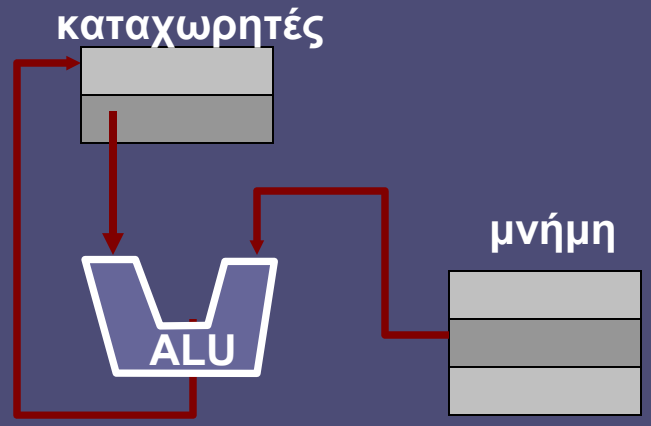
add X

(A = A + X)

- Ένα έντελο - θέση αποθήκευσης του αποτελέσματος είναι πάντα ο συσσωρευτής
  - 1-address architecture
  - Αρχιτεκτονική των πρώτων υπολογιστών!

Αρχιτεκτονικές με καταχωρητές (registers)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων



add R2, R1,mem(100)

(R2 = R1 + mem[100])

- Memory-register
  - Οποιαδήποτε εντολή μπορεί να προσπελάσει τη μνήμη
- Όμως:
  - Πολλαπλές προσπελάσεις μνήμης
  - Λήψη εντολής – Λήψη εντέλων
  - Συνωστισμός στον δίαυλο επικοινωνίας με μνήμη

!

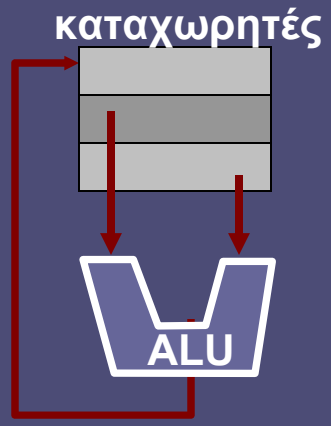
Καταχωρητές: προσωρινές θέσεις αποθήκευσης αποτελεσμάτων, η γενίκευση της ιδέας του συσσωρευτή.

Αρχιτεκτονικές με καταχωρητές (registers)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων

;

Είναι επιθυμητός ένα πολύ μεγάλος αριθμός καταχωρητών;



add R1, R2,R3

(R1 = R2 + R3)

- Register-register (load-store)
  - Μόνο εντολές load-store μπορούν να προσπελάσουν τη μνήμη
- Η αρχιτεκτονική των σύγχρονων επεξεργαστών
  - Οι καταχωρητές προσπελαύνονται πολύ γρήγορα
  - Χρειάζονται λιγότερα bits για να επιλεγούν
  - Οι μεταγλωττιστές αναθέτουν μεταβλητές σε καταχωρητές

Μέθοδοι προσπέλασης μνήμης

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

- Τουλάχιστον κάποιες εντολές προσπελαύνουν 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

Μέθοδοι προσπέλασης μνήμης

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

;

Πώς κωδικοποιούνται οι μέθοδοι προσπέλασης μνήμης μέσα στην εντολή;

- Στο σχηματισμό της διεύθυνσης μνήμης μπορούν να συμμετέχουν:
  - Απόλυτες τιμές διεύθυνσης
  - Καταχωρητές
  - Σταθερές τιμές μετατόπισης (offsets)

		πιθανή χρήση ↓
displacement	**mem[offs+reg]**	**τοπικές**
register indirect	**mem[reg]**	**δείκτες**
indexed	**mem[reg1+reg2]**	**πίνακες**
direct	**mem[addr]**	**στατικές**
memory indirect	**mem[mem[reg]]**	***δείκτες**
auto-increment	**mem[reg++]**	**πίνακες**
scaled	**mem[offs+reg1+reg2*d]**	**πίνακες**

Η εξέλιξη της αρχιτεκτονικής εντολών

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Εξέλιξη αρχιτεκτονικής

- Οι πρώτοι υπολογιστές (.. - ’60)
  - Αρχιτεκτονική συσσωρευτή και αργότερα σωρού
    - Ικανοποιητική λύση λόγω της απλής τεχνολογίας των μεταγλωττιστών
- Πολύπλοκες αρχιτεκτονικές (’70 - ..)
  - Ενσωμάτωση σύνθετων μορφών εντολών και μεθόδων προσπέλασης μνήμης
    - Προσπάθεια υποστήριξης υψηλών γλωσσών προγραμματισμού – μείωσης κόστους λογισμικού
    - Πολλά χαρακτηριστικά μένουν αχρησιμοποίητα!
  - Complex Instruction Set Computers (CISC)

Η εξέλιξη της αρχιτεκτονικής εντολών

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Εξέλιξη αρχιτεκτονικής

- **Reduced Instruction Set Computers (RISC) (’80 - …)**
  - Απλούστερες και φθηνότερες load-store αρχιτεκτονικές με σταθερό μήκος εντολών
  - Μεγαλύτερη απόδοση – ταχύτερη εκτέλεση εντολών
  - Ευνοείται από την αφθονία υλικού χαμηλού κόστους και την προηγμένη τεχνολογία των μεταγλωττιστών
  - Ακόμα και ο μοναδικός επιζών επεξεργαστής με εντολές CISC (αρχιτεκτονική x86), μεταφράζει εσωτερικά σε εντολές RISC…

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Αρχιτεκτονικές Συνόλου Εντολών (ΙΙ)

(Δομή Εντολών και Παραδείγματα)

http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Αρχιτεκτονική Συνόλου Εντολών

- ISA

- Τι περιγράφει;
  - Διαθέσιμες πράξεις/λειτουργίες
    - Κωδικοποίηση λειτουργιών
  - Μορφή των δεδομένων εισόδου-εξόδου
    - Έντελα (operands)
  - Μέθοδοι προσπέλασης μνήμης
    - Προέλευση των δεδομένων
  - Χώροι προσωρινής αποθήκευσης
    - Καταχωρητές
  - Διακοπές και καταστάσεις σφάλματος
    - Ποια η "αντίδραση" του επεξεργαστή

Σχεδιασμός Συνόλου Εντολών

- ISA

- Συμβιβασμός μεταξύ:
  - Κόστους και απόδοσης υλικού
  - Βαθμού υποστήριξης λογισμικού
  - Άλλων παραγόντων όπως η κατανάλωση ενέργειας
    - Το υλικό καταναλώνει ενέργεια, υπό τον έλεγχο του λογισμικού όμως!
  - Μη επικάλυψης λειτουργιών εντολών (orthogonality)

Κωδικοποίηση Εντολών Μηχανής

- ISA

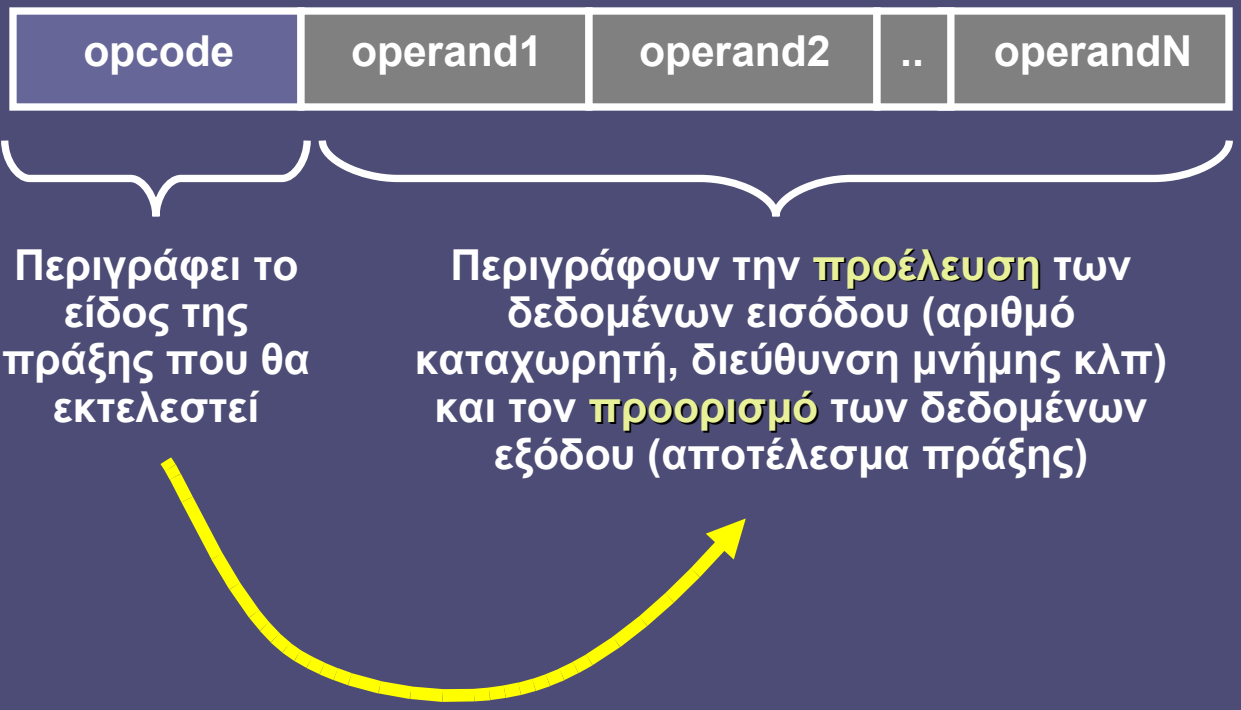
;

Γιατί είναι ταχύτερη η λήψη και αποκωδικοποίηση των εντολών σταθερού μήκους;

- Σειρά δυαδικών ψηφίων
  - Μεταβλητού μήκους
    - Περισσότερο συμπαγή προγράμματα
    - Πολυπλοκότερο υλικό!
  - Σταθερού μήκους
    - Απλούστερη και ταχύτερη λήψη-αποκωδικοποίηση
    - Μεγαλύτερα προγράμματα
    - Μέθοδοι συμπίεσης

Κωδικοποίηση Εντολών Μηχανής

- ISA



Εντολές Μηχανής και Assembly

- ISA

!

Κάθε διαφορετική αρχιτεκτονική επεξεργαστών έχει διαφορετική γλώσσα assembly!

- 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
  - Προσοχή:
    - MOV R1,R2 // π.χ. R1 => R2
    - MOV R1,#100 // π.χ. R1 => mem[100]
    - παρόλο που το μνημονικό MOV είναι ίδιο, δεν πρόκειται για την ίδια εντολή!

Τα παραδείγματά μας: Intel IA-32 (και 64)

- ISA

i

Και η ανταγωνίστρια εταιρία AMD παράγει επεξεργαστές συμβατούς με την αρχιτεκτονική αυτή

- Αρχιτεκτονική επεξεργαστών Intel IA-32 και Intel 64
  - Μία πολύπλοκη αρχιτεκτονική επεξεργαστών
    - 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 στην Intel έρευνα και ανάπτυξη για διαρκείς βελτιώσεις απόδοσης
    - "Τελικά η αρχιτεκτονική αυτή είναι βέλτιστη μέσα στην τεράστια πολυπλοκότητά της"

Τα παραδείγματά μας: Intel IA-32 (και 64)

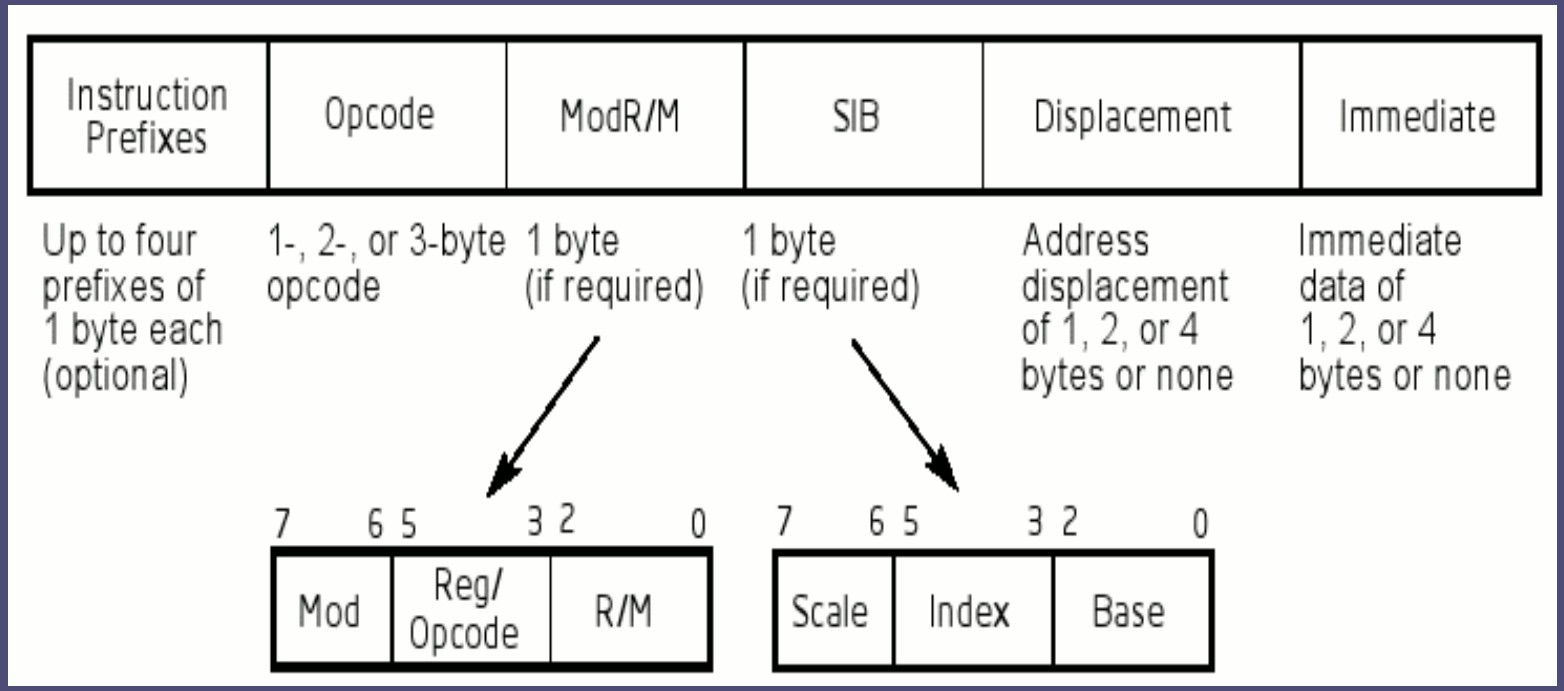
- ISA

- Οικογένειες επεξεργαστών
  - Pentium
  - P6
    - Pentium Pro, II, III
  - Netburst
    - Pentium 4, D (IA-32 – σε νεώτερες εκδόσεις και Intel 64)
  - Pentium M
    - Intel Core, Core Duo
  - Intel Core (IA-32 και Intel 64)
    - Intel Core 2 Duo

Intel® 64 and IA-32 Architectures Software Developer's Manuals
http://www.intel.com/products/processor/manuals/index.htm

Intel IA-32: η γενική μορφή των εντολών

- Μεταβλητός αριθμός bytes ανά εντολή (1-17)!



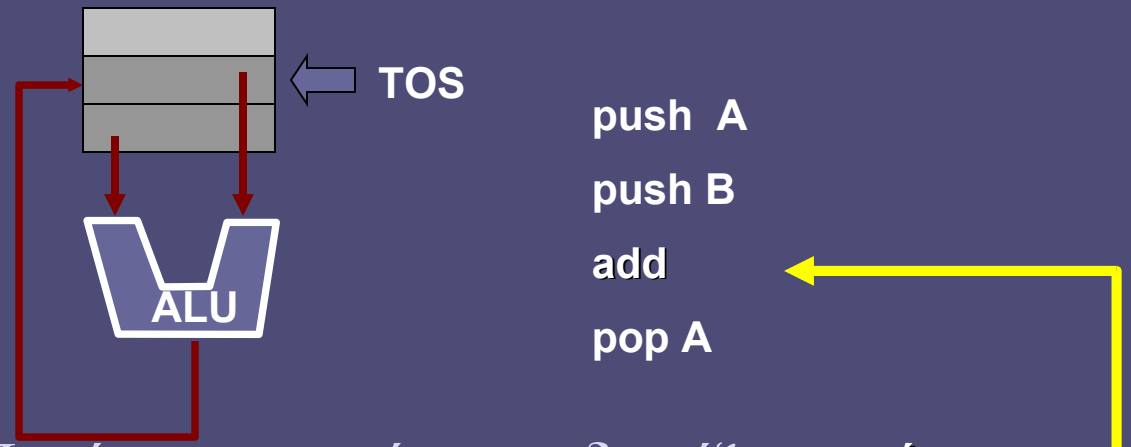
Προέλευση και αποθήκευση δεδομένων

- ISA
- Προέλευση δεδομένων

- Προέλευση δεδομένων – αποθήκευση αποτελεσμάτων μιας πράξης
  - **Operand addressing**
  - Εξαρτάται από την αρχιτεκτονική του επεξεργαστή
  - Στους πρώτους επεξεργαστές
    - Stack (σωρός-στοίβα)
    - Accumulator (συσσωρευτής)
  - Μεταγενέστεροι υπολογιστές
    - Καταχωρητές – Μνήμη (register-memory)
    - Κυρίως καταχωρητές (register-register ή load-store)
      - μόνο ανάγνωση και εγγραφή μνήμης

Αρχιτεκτονική σωρού (stack)

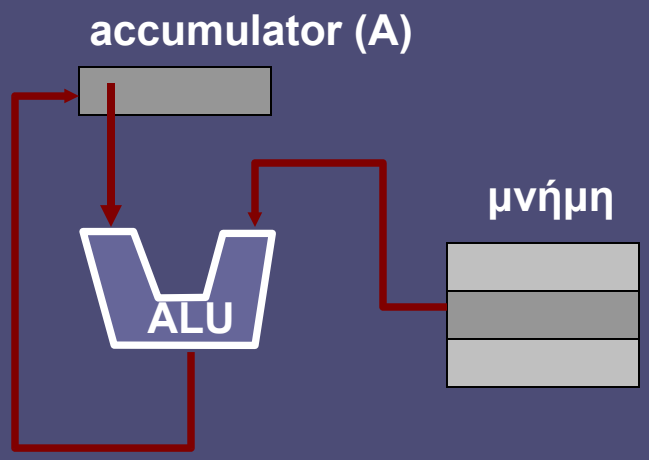
- ISA
- Προέλευση δεδομένων



- Πηγή-προορισμός προσδιορίζονται έμμεσα
  - Δεν περιγράφονται στην εντολή!
  - 0-address architecture
  - Δημοφιλές σχήμα κατά τη δεκαετία του 60
  - Δύσκολη προσπέλαση σωρού, απαιτούνται πολλαπλές αντιμεταθέσεις και αντιγραφές
  - Το σχήμα επιζεί στην αρχιτεκτονική της Java VM

Αρχιτεκτονική συσσωρευτή (accumulator)

- ISA
- Προέλευση δεδομένων



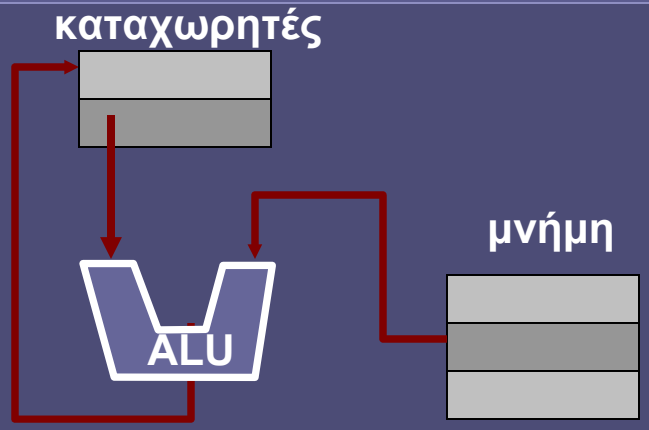
add X

(A = A + X)

- Μία πηγή - θέση αποθήκευσης του αποτελέσματος είναι πάντα ο συσσωρευτής
  - 1-address architecture
  - Αρχιτεκτονική των πρώτων υπολογιστών!

Αρχιτεκτονικές με καταχωρητές (1)

- ISA
- Προέλευση δεδομένων



add R2, R1,mem(100)

(R2 <= R1 + mem[100])

- Memory-register
  - Οποιαδήποτε εντολή μπορεί να προσπελάσει τη μνήμη
- Όμως:
  - Πολλαπλές προσπελάσεις μνήμης
  - Λήψη εντολής – Λήψη εντέλων
  - Συνωστισμός στον δίαυλο επικοινωνίας με μνήμη

!

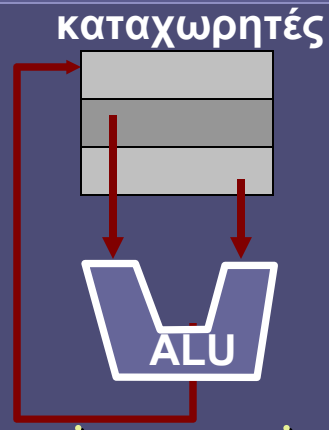
Καταχωρητές: προσωρινές θέσεις αποθήκευσης αποτελεσμάτων, η γενίκευση της ιδέας του συσσωρευτή.

Αρχιτεκτονικές με καταχωρητές (2)

- ISA
- Προέλευση δεδομένων

;

Είναι επιθυμητός ένα πολύ μεγάλος αριθμός καταχωρητών;



add R1, R2,R3

(R1 <= R2 + R3)

- Register-register (load-store)
  - Μόνο εντολές load-store μπορούν να προσπελάσουν τη μνήμη
- Η αρχιτεκτονική των σύγχρονων επεξεργαστών
  - Οι καταχωρητές προσπελαύνονται πολύ γρήγορα
  - Χρειάζονται λιγότερα bits για να επιλεγούν
  - Οι μεταγλωττιστές αναθέτουν μεταβλητές σε καταχωρητές

Η εξέλιξη της αρχιτεκτονικής εντολών

- ISA
- Προέλευση δεδομένων

- Οι πρώτοι υπολογιστές (.. - ’60)
  - Αρχιτεκτονική συσσωρευτή και αργότερα σωρού
    - Ικανοποιητική λύση λόγω της απλής τεχνολογίας των μεταγλωττιστών
- Πολύπλοκες αρχιτεκτονικές (’70 - ..)
  - Ενσωμάτωση σύνθετων μορφών εντολών και μεθόδων προσπέλασης μνήμης
    - Προσπάθεια υποστήριξης υψηλών γλωσσών προγραμματισμού – μείωσης κόστους λογισμικού
    - Πολλά χαρακτηριστικά μένουν αχρησιμοποίητα!
  - Complex Instruction Set Computers (CISC)

Η εξέλιξη της αρχιτεκτονικής εντολών

- ISA
- Προέλευση δεδομένων

- Reduced Instruction Set Computers (RISC) (’80 - …)
  - Απλούστερες και φθηνότερες load-store αρχιτεκτονικές με σταθερό μήκος εντολών
  - Μεγαλύτερη απόδοση – ταχύτερη εκτέλεση εντολών
  - Ευνοείται από την αφθονία υλικού χαμηλού κόστους και την προηγμένη τεχνολογία των μεταγλωττιστών
  - Ακόμα και ο μοναδικός επιζών επεξεργαστής με εντολές CISC (αρχιτεκτονική x86), μεταφράζει εσωτερικά σε εντολές RISC…

IA-32 (64): memory-register

- ISA
- Προέλευση δεδομένων

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

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..

IA-32: καταχωρητές

- ISA
- Προέλευση δεδομένων

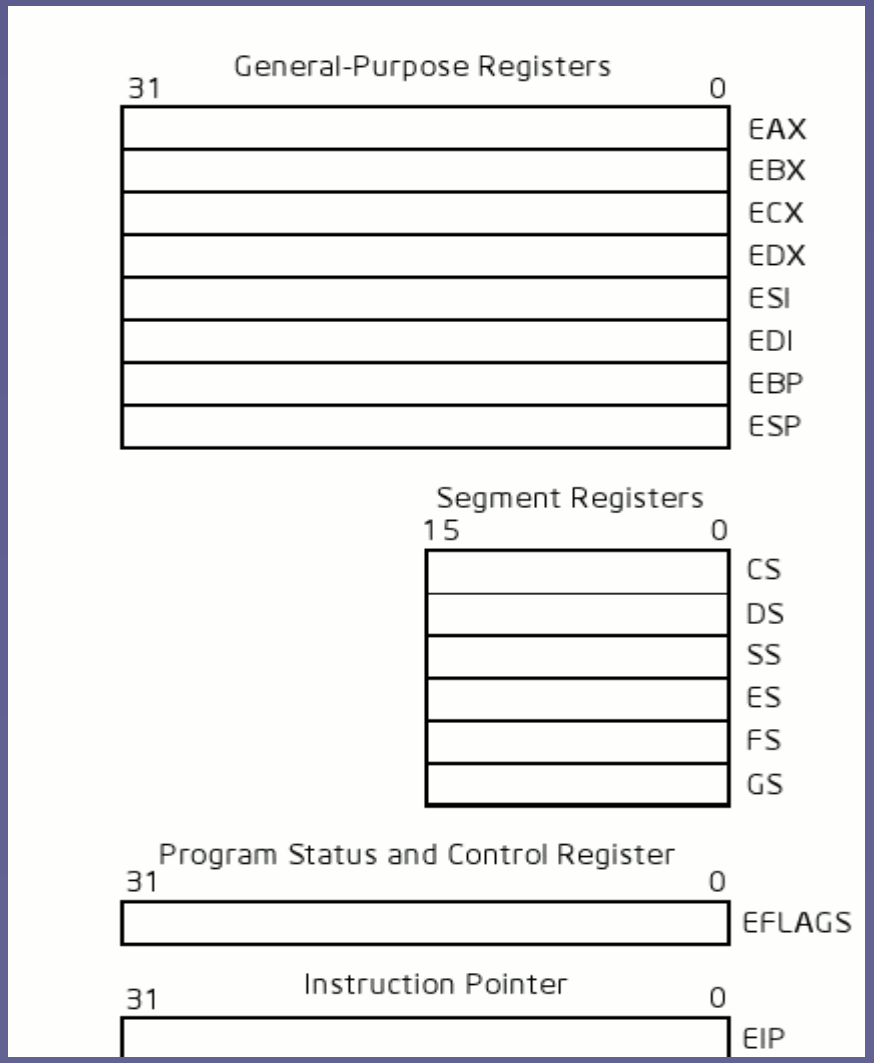
γενικού σκοπού και σχηματισμός διευθύνσεων

τμηματοποίηση

(σε επόμενα μαθηματα)

καταχωρητής κατάστασης

“program counter”



i

Εκτός από τους καταχωρητές γενικού σκοπού, οι επεξεργαστές IA-32 έχουν πρόσθετους καταχωρητές για πράξεις κινητής υποδιαστολής και πράξεις streaming και multimedia.

Μέθοδοι προσπέλασης μνήμης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

- Τουλάχιστον κάποιες εντολές προσπελαύνουν 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

Μέθοδοι προσπέλασης μνήμης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης

;

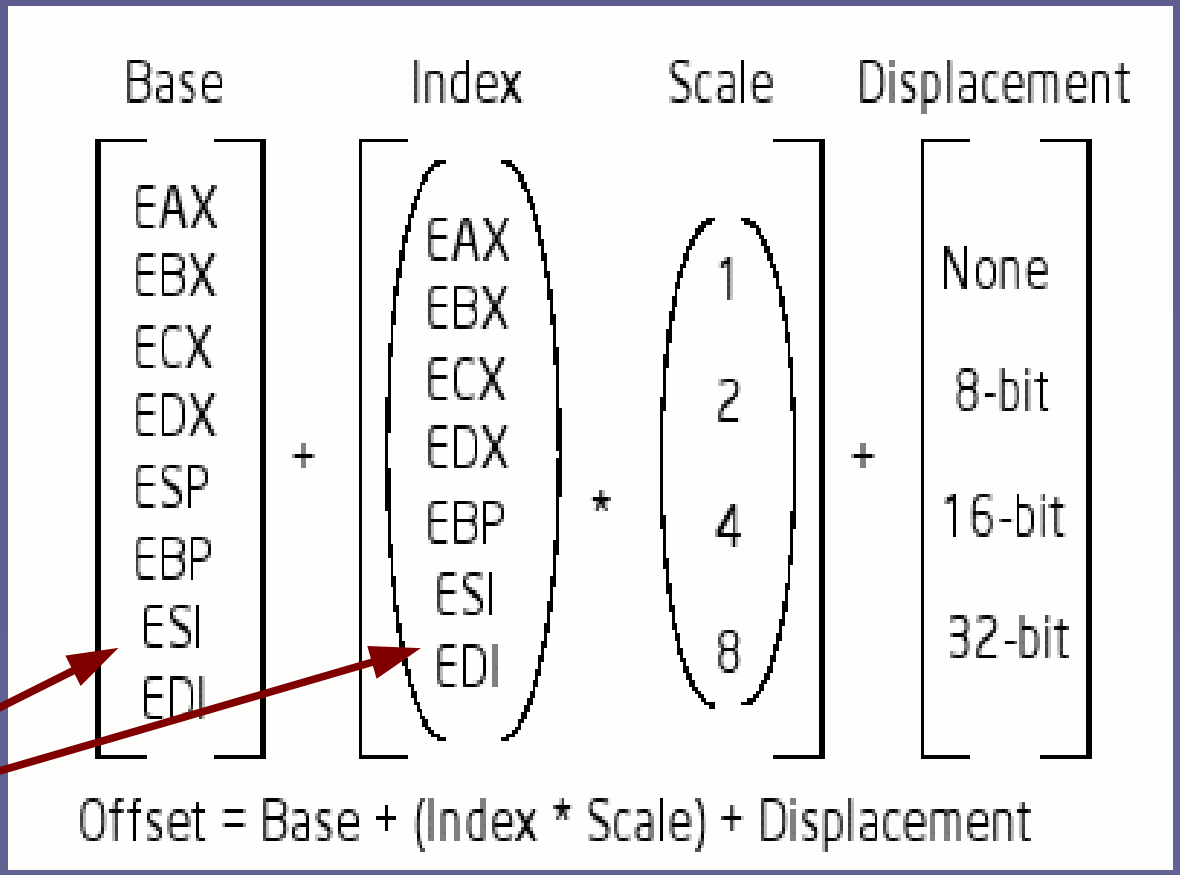
Πώς κωδικοποιούνται οι μέθοδοι προσπέλασης μνήμης μέσα στην εντολή;

- Στο σχηματισμό της διεύθυνσης μνήμης μπορούν να συμμετέχουν:
  - Απόλυτες τιμές διεύθυνσης
  - Καταχωρητές
  - Σταθερές τιμές μετατόπισης (offsets)

		πιθανή χρήση
displacement	**mem[offs+reg]**	**τοπικές**
register indirect	**mem[reg]**	**δείκτες**
indexed	**mem[reg1+reg2]**	**πίνακες**
direct	**mem[addr]**	**στατικές**
memory indirect	**mem[mem[reg]]**	***δείκτες**
auto-increment	**mem[reg++]**	**πίνακες**
scaled	**mem[offs+reg1+reg2*d]**	**πίνακες**

IA-32 (64): Σχηματισμός διευθύνσεων

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης



καταχωρητές που χρησιμοποιούνται με ειδικό τρόπο

Εντολές: κατηγορίες λειτουργιών

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

- Βασικές κατηγορίες
  - Αριθμητικές και λογικές πράξεις
  - Μεταφορά δεδομένων
    - Από-πρός Καταχωρητές και Μνήμη
  - Έλεγχος ροής εκτέλεσης
    - Διακλαδώσεις και κλήσεις ρουτινών
- Άλλες κατηγορίες
  - Ειδικές εντολές συστήματος
    - ΛΣ, ιδεατή μνήμη
  - Επεξεργασία πολλαπλών δεδομένων
    - Χρήσιμο για γραφικά, σειρές χαρακτήρων, multimedia

Αριθμητικές εντολές και μεταφορά δεδομένων

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

;

Τι συμβολίζουν τα R1, R2 ..; Πώς αναπαρίστανται μέσα στην εντολή;

- Αριθμητικές-λογικές πράξεις
  - Πηγές δεδομένων και προορισμός
    - add R1, R2, R3 // R3 = R1+R2

- Μεταφορά δεδομένων
  - Πηγή δεδομένων και προορισμός
  - Μήκος μεταφερόμενης λέξης (ενδεχομένως)
    - load R1, 0x7FF0 // R1 = mem[0x7FF0]

IA-32: 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

Εντολές διακλάδωσης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

- Διακλάδωση
  - Με ή χωρίς συνθήκη
    - bne R1, R2, +8 // branch if not R1==R2
  - Σε απόλυτη διεύθυνση
    - jump 0xFF97DE00
  - Σχετικά ως προς την τρέχουσα θέση (offset)
    - jump +130 // offset = +130
    - **Ο παραγόμενος κώδικας μπορεί να τοποθετηθεί οπουδήποτε στη μνήμη**

IA-32: Εντολές διακλάδωσης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

- Με ή χωρίς συνθήκη
  - Σε απόλυτη ή σχετική διεύθυνση
    - 16 ή 32 bits απόλυτη διεύθυνση
      - πιθανώς σε μνήμη ή καταχωρητή
    - 8, 16 ή 32 bits σχετική διεύθυνση
  - 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

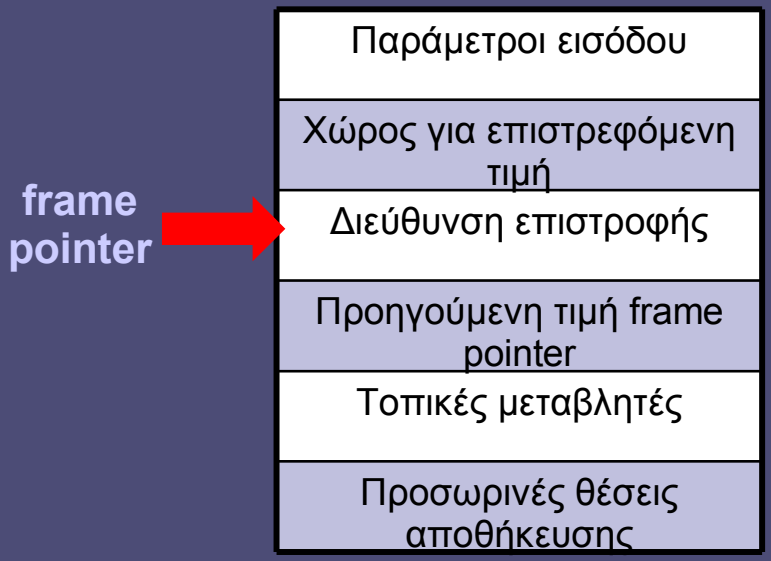
Εντολές διακλάδωσης

- ISA
- Προέλευση δεδομένων
- Προσπέλαση μνήμης
- Κατηγορίες εντολών

;

Πού θα μπορούσε να αποθηκευτεί η διεύθυνση επιστροφής μετά από κλήση συνάρτησης;

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



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Κεντρική Μονάδα Επεξεργασίας

(Σχεδιασμός και λειτουργία μιας απλής ΚΜΕ)

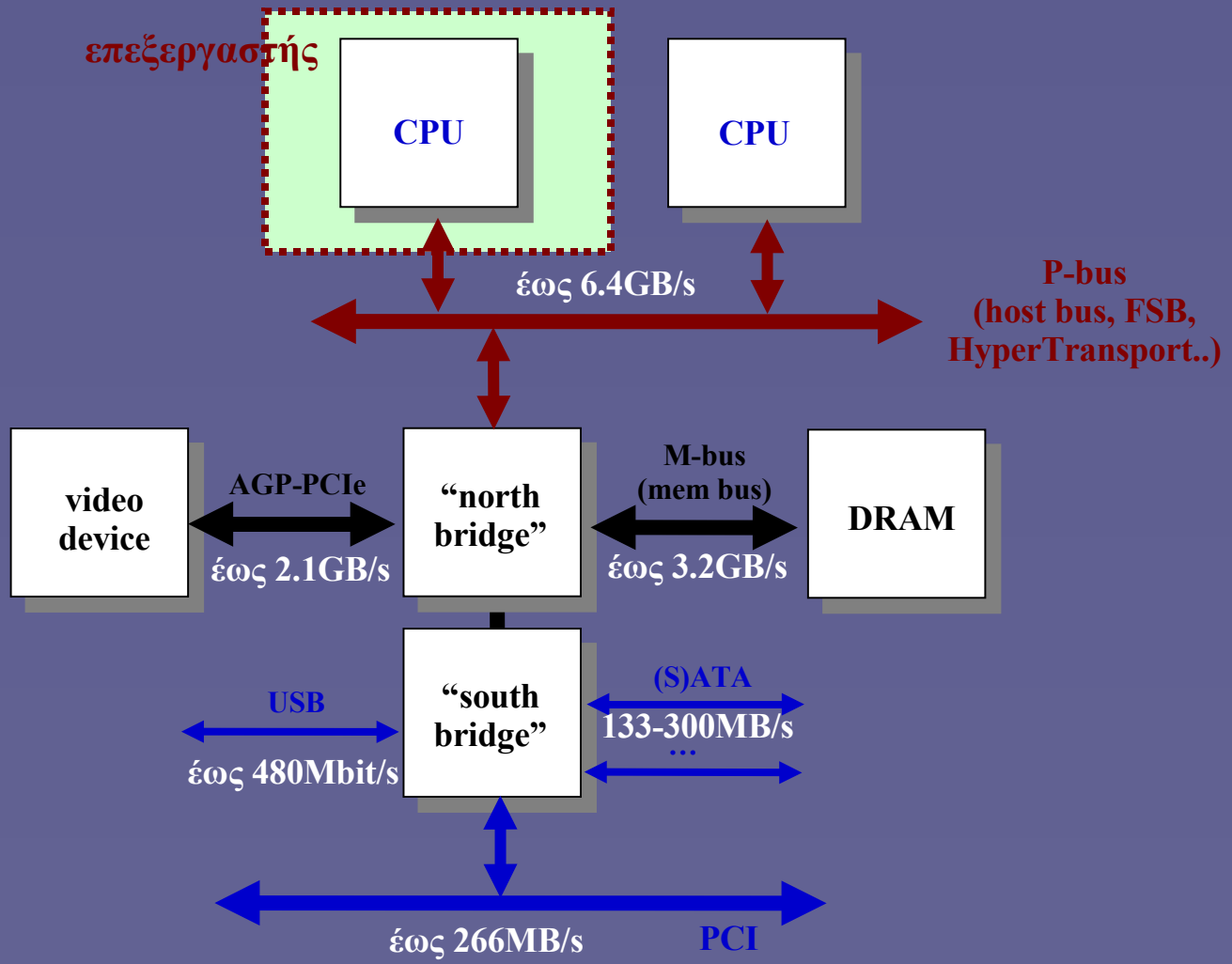
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Η υπολογιστική μηχανή: ο επεξεργαστής

- Ο επεξεργαστής

- Επεξεργαστής
- Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)



Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)

- Ο επεξεργαστής
- ΚΜΕ

- (Μικρο)επεξεργαστής
  - Ψηφιακό σύστημα που εκτελεί υπολογισμούς σε πολλαπλά βήματα
- Κεντρική Μονάδα Επεξεργασίας
  - Συνώνυμο του επεξεργαστή (σχεδόν!)
  - Στην ίδια συσκευασία
    - Κρυφή Μνήμη (1ου και 2ου επιπέδου)
    - Ελεγκτές Ε/Ε

Σχεδιασμός μιας απλής ΚΜΕ

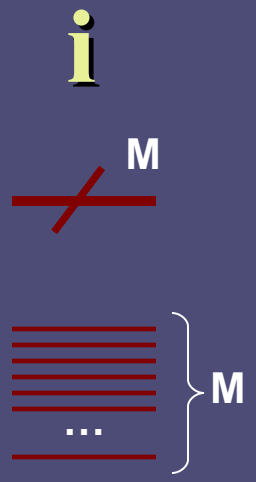
- Ο επεξεργαστής
- ΚΜΕ

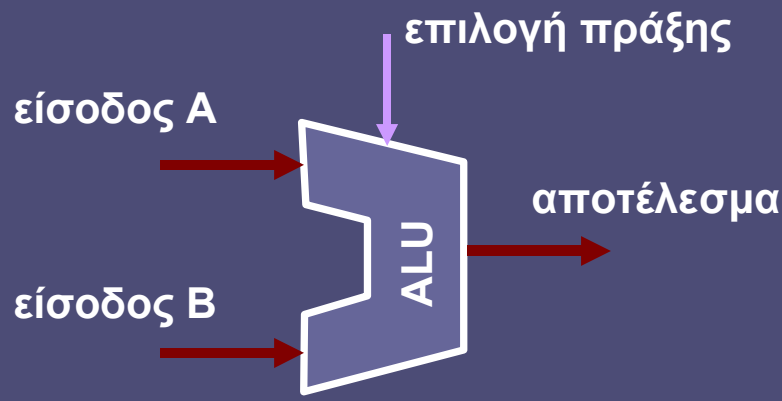
Όλες οι σύγχρονες ΚΜΕ είναι πολυπλοκότερες!

- Παραδοχές
  - Μοντέλο load-store
    - Αριθμητικές-λογικές πράξεις
    - Ανάγνωση-εγγραφή από/στη μνήμη
    - Εντολές διακλάδωσης
  - Ν καταχωρητές γενικού σκοπού
  - Απλουστευμένο μοντέλο μνήμης
    - Ως "κουτί" που δέχεται διεύθυνση και επιστρέφει δεδομένα και εντολές
    - Αντιπροσωπεύει στην πραγματικότητα μια ιεραρχία μνήμης!

Εκτέλεση πράξεων: ΑΛΜ

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ





- Αριθμητική-Λογική Μονάδα (ΑΛΜ)
  - Arithmetic-Logic Unit (ALU)
    - Εκτέλεση αριθμητικών και λογικών πράξεων
    - Εκτέλεση συγκρίσεων (για διακλάδωση υπό συνθήκη)
    - Υπολογισμός διευθύνσεων (για προσπέλαση μνήμης)

Εντολές για αριθμητικές – λογικές πράξεις

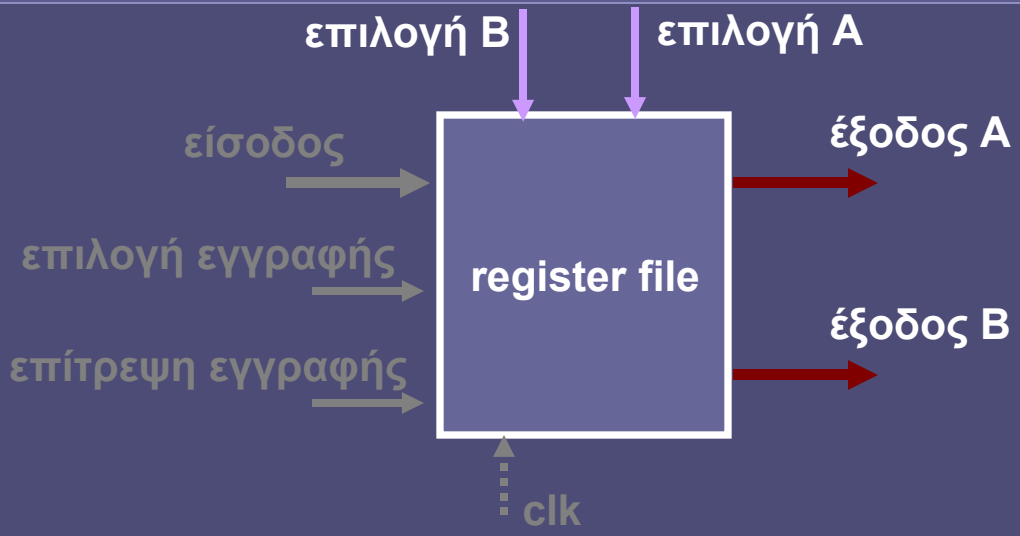
- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ

R3 ← R1 op R2

- Σε κάθε εκτέλεση εντολής
  - Ανάγνωση από 2 καταχωρητές
  - Εγγραφή σε 1 καταχωρητή
- Ίδια ροή δεδομένων
  - διαφορετική επιλεγόμενη πράξη

Συστοιχία καταχωρητών (register file)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές



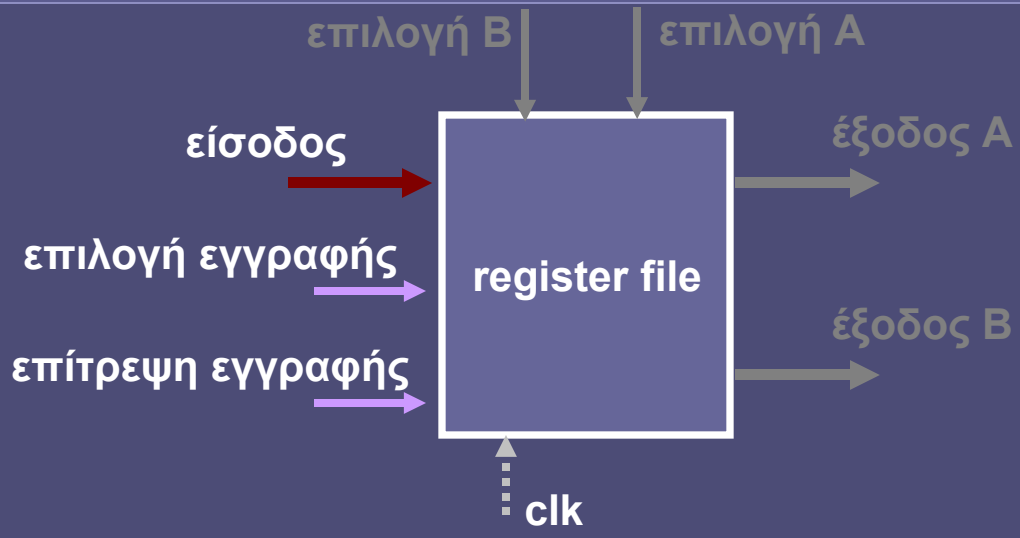
- Ανάγνωση: Επιλογή A και Επιλογή B
  - Ποιων καταχωρητών το περιεχόμενο θα εμφανιστεί στην αντίστοιχη έξοδο

!

Τα σήματα στο σχήμα αυτό (και στα επόμενα) πιθανόν να αποτελούνται από πολλά bits το καθένα!

Συστοιχία καταχωρητών (register file)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές



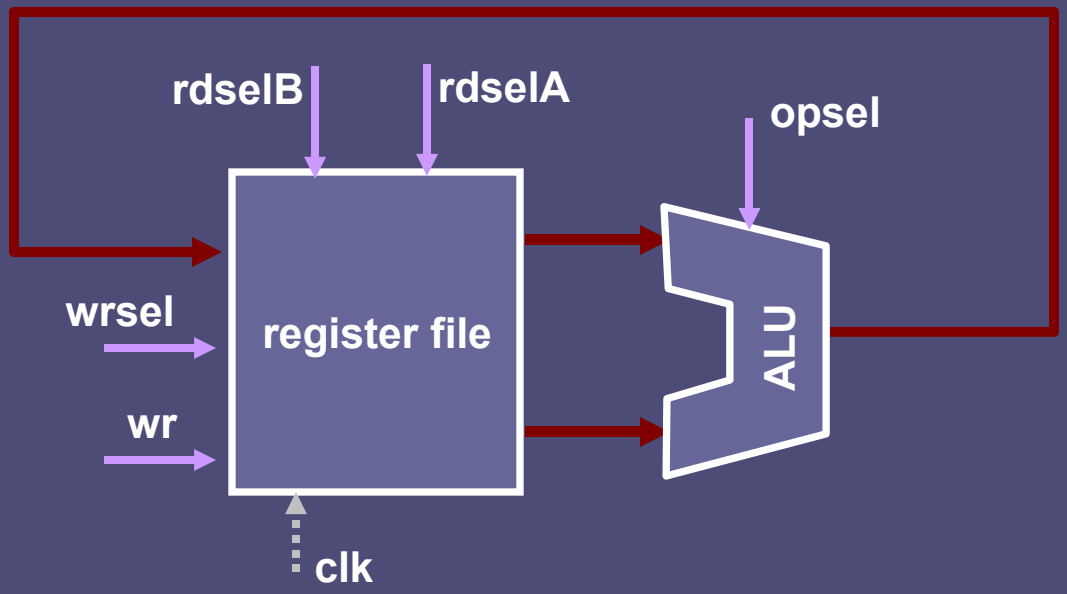
- Εγγραφή: στην (θετική) ακμή του clk
  - Στον καταχωρητή που ορίζει η **επιλογή εγγραφής**, εάν το επιτρέπει η **επίτρεψη εγγραφής**

Ένα απλό μονοπάτι δεδομένων (datapath)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath

;

Ποιος οδηγεί τα σήματα ελέγχου;



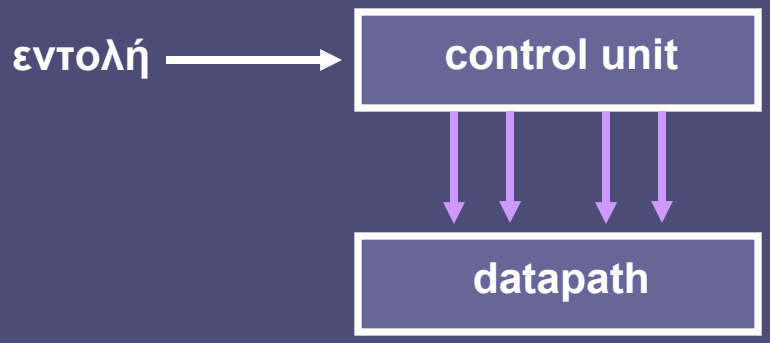
- Εκτέλεση μιας εντολής ανά κύκλο ρολογιού
- **Ανάγνωση και εγγραφή στον ίδιο κύκλο;**
  - Ναι, η ανάγνωση θα δώσει το αποτέλεσμα εγγραφής προηγούμενου κύκλου
  - Ενώ η εγγραφή θα είναι διαθέσιμη για ανάγνωση στον επόμενο κύκλο

Η μονάδα ελέγχου (control unit)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου

;

Από πού προέρχονται οι εντολές;



- Αποκωδικοποίηση bits της εντολής
  - Επιλογή καταχωρητών προέλευσης δεδομένων
  - Επιλογή καταχωρητή αποθήκευσης αποτελέσματος
  - Επιλογή λειτουργίας ΑΛΜ
  - Το δυσκολότερο στη σχεδίαση τμήμα της ΚΜΕ

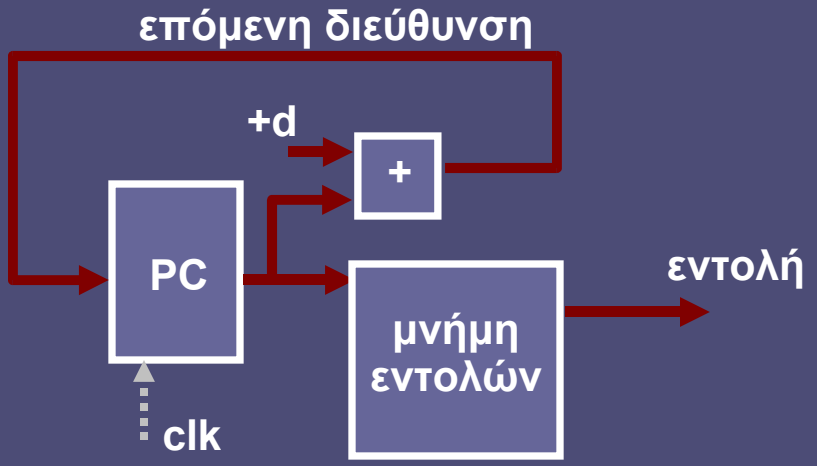
Η "μνήμη εντολών"

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές

- Μια χρήσιμη αφαίρεση
  - Στην πραγματικότητα
    - Αντιπροσωπεύει μια ιεραρχία μνήμης
    - Από την κρυφή μνήμη 1ου επιπέδου μέχρι την κύρια μνήμη
    - Πιθανόν να μην είναι δυνατή η ανάκτηση εντολής ανά κύκλο ρολογιού

Η διεύθυνση της επόμενης εντολής

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές



- Program Counter (PC)
  - Περιέχει τη διεύθυνση στη μνήμη της επόμενης εντολής
    - Αυξάνεται κατά την εκτέλεση κάθε εντολής
    - Επόμενη διεύθυνση = τρέχουσα + d

Ο κύκλος μηχανής

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές

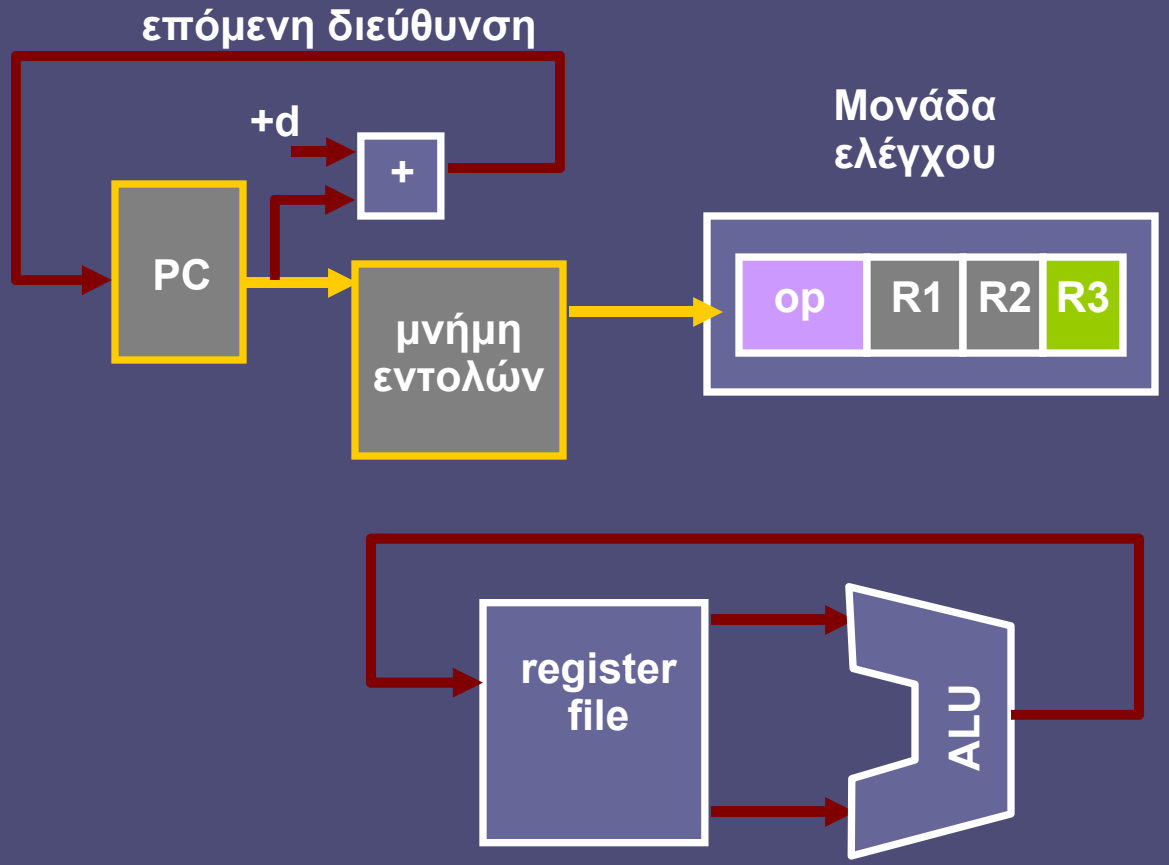
;

Ποια στάδια χρησιμοποιούν οι εντολές για αριθμητικές και λογικές πράξεις;

- **Στάδια εκτέλεσης εντολής**
  - Instruction Fetch (IF)
    - Ανάκτηση εντολής προς εκτέλεση
  - Instruction Decode (ID)
    - Επιλογή δεδομένων και λειτουργιών ανάλογα με εντολή
  - Execute (EX)
    - Εκτέλεση υπολογισμού στην ΑΛΜ
  - Data Memory Access (DM)
    - Προσπέλαση μνήμης δεδομένων
  - Write Back (WB)
    - Αποθήκευση αποτελεσμάτων σε καταχωρητή

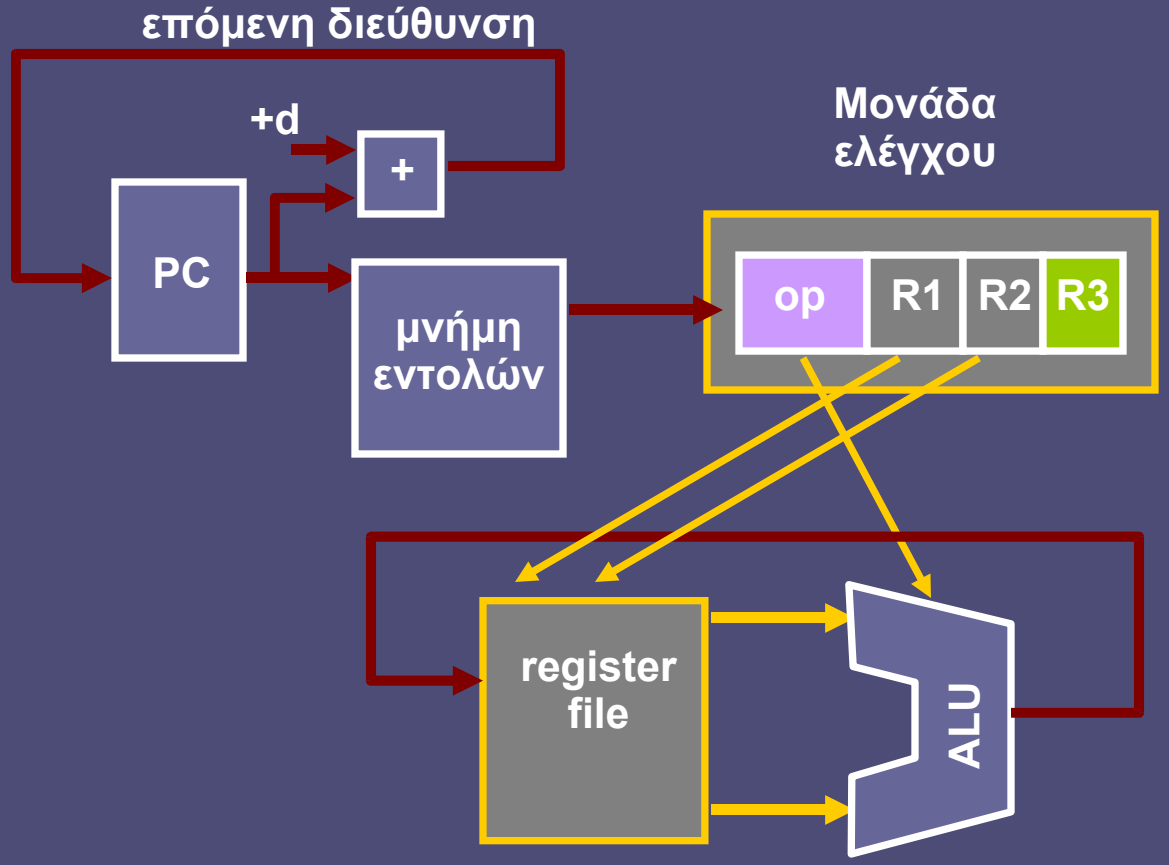
Instruction Fetch (IF)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές



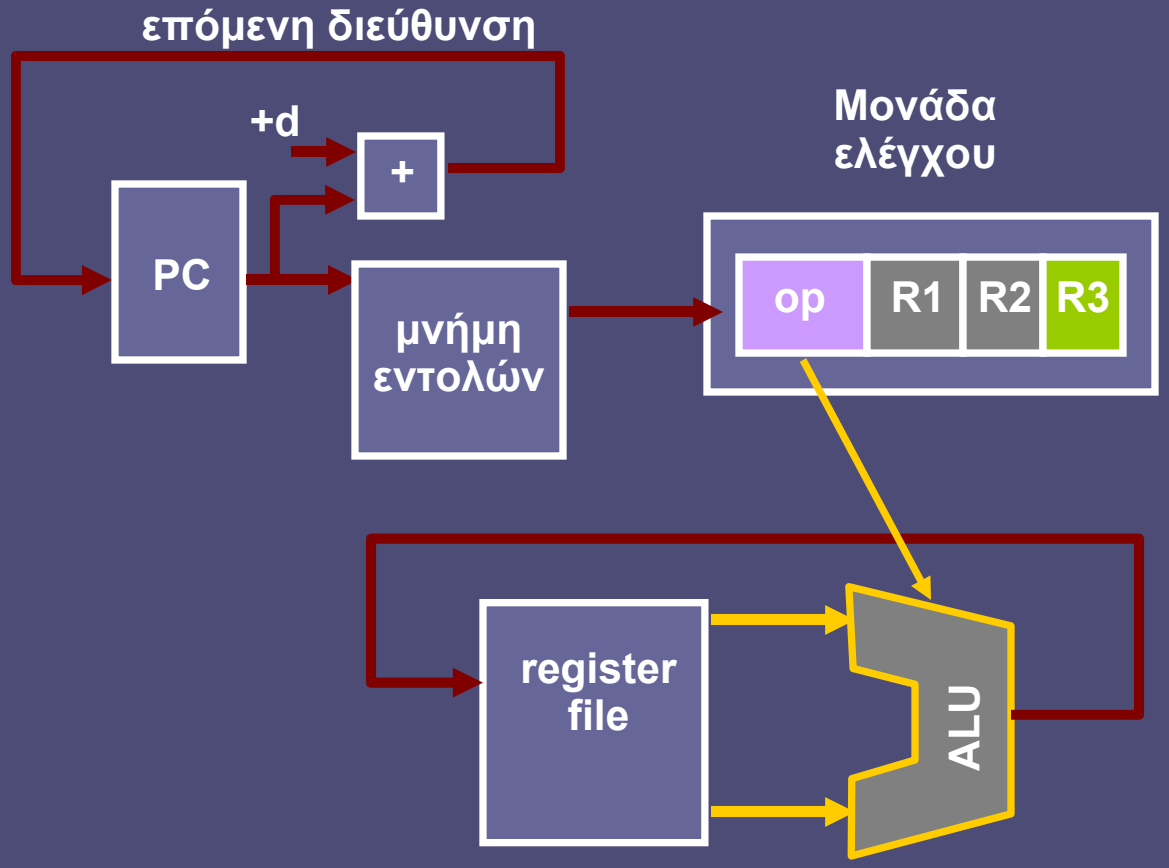
Instruction Decode (ID)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές



Execute (EX)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές

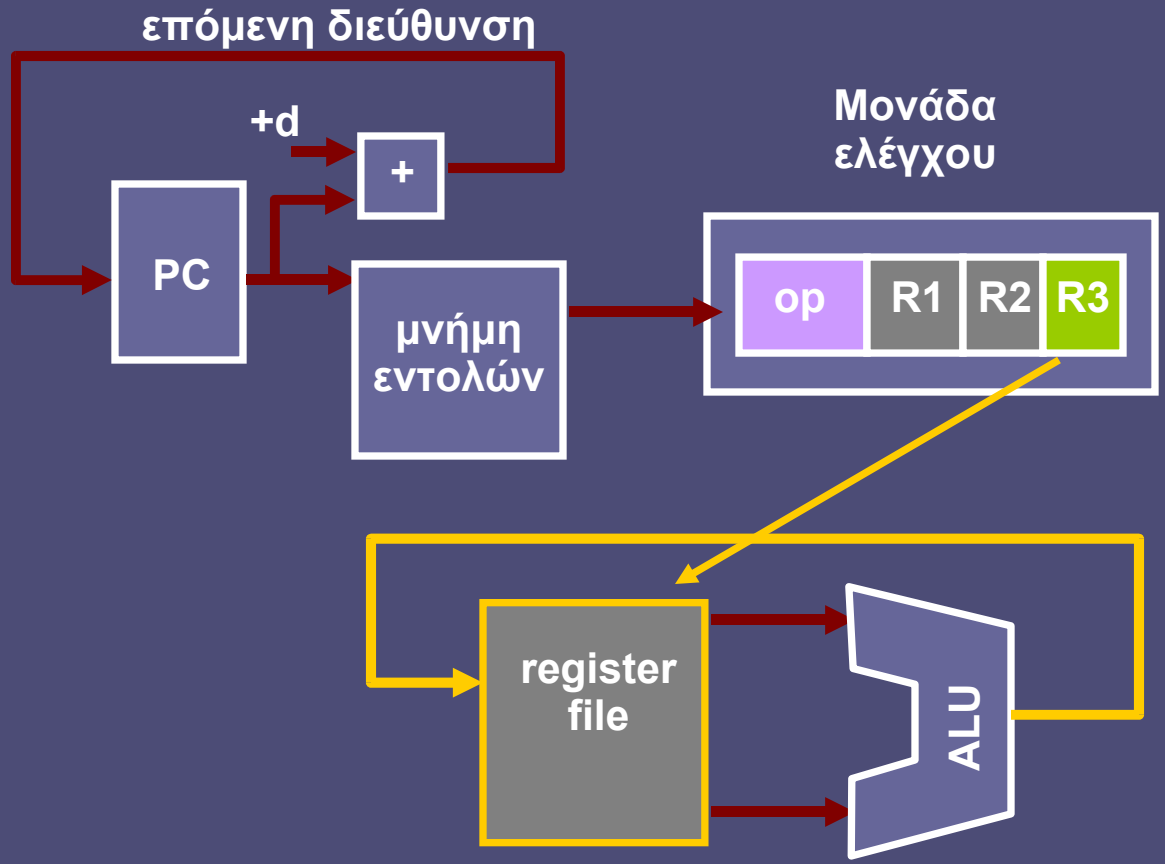


Write Back (WB)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές

;

Τι χρειάζεται να αλλάξει για να προστεθεί η εισαγωγή σταθερών;



Εντολές διακλάδωσης (branch ή jump)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις

- Παράδειγμα

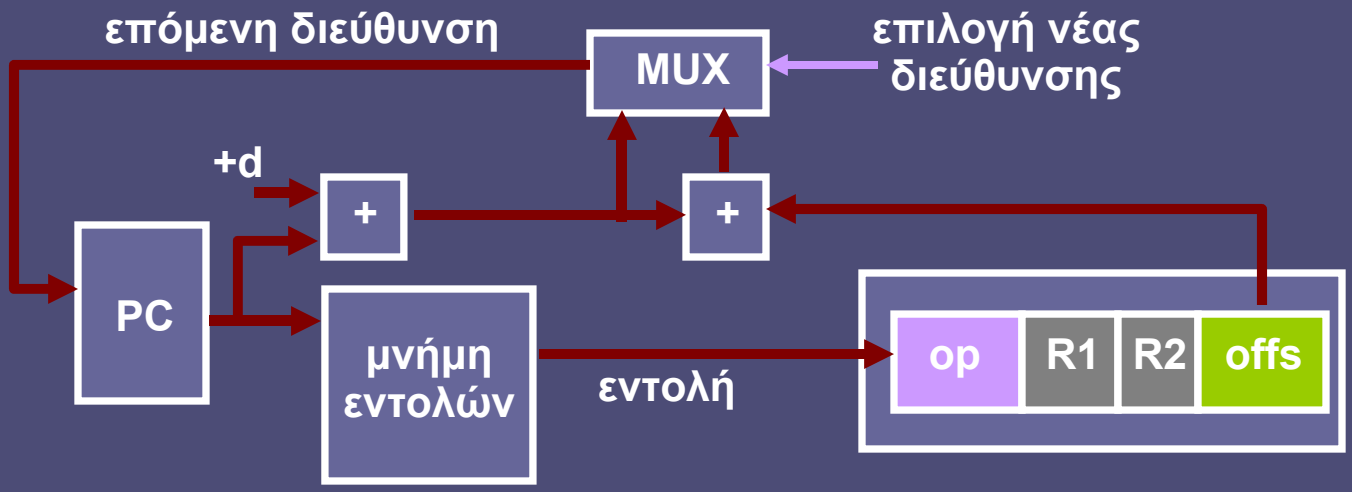
if R1 == R2 then PC ← PC + offset

"branch if equal"

- Διακλάδωση υπό συνθήκη
  - Βασισμένο σε αποτέλεσμα πράξης ΑΛΜ
- Σχετική διεύθυνση
- Υπολογισμός συνθήκης – διακλάδωση
  - Στην ίδια εντολή

Διακλαδώσεις και ενημέρωση PC

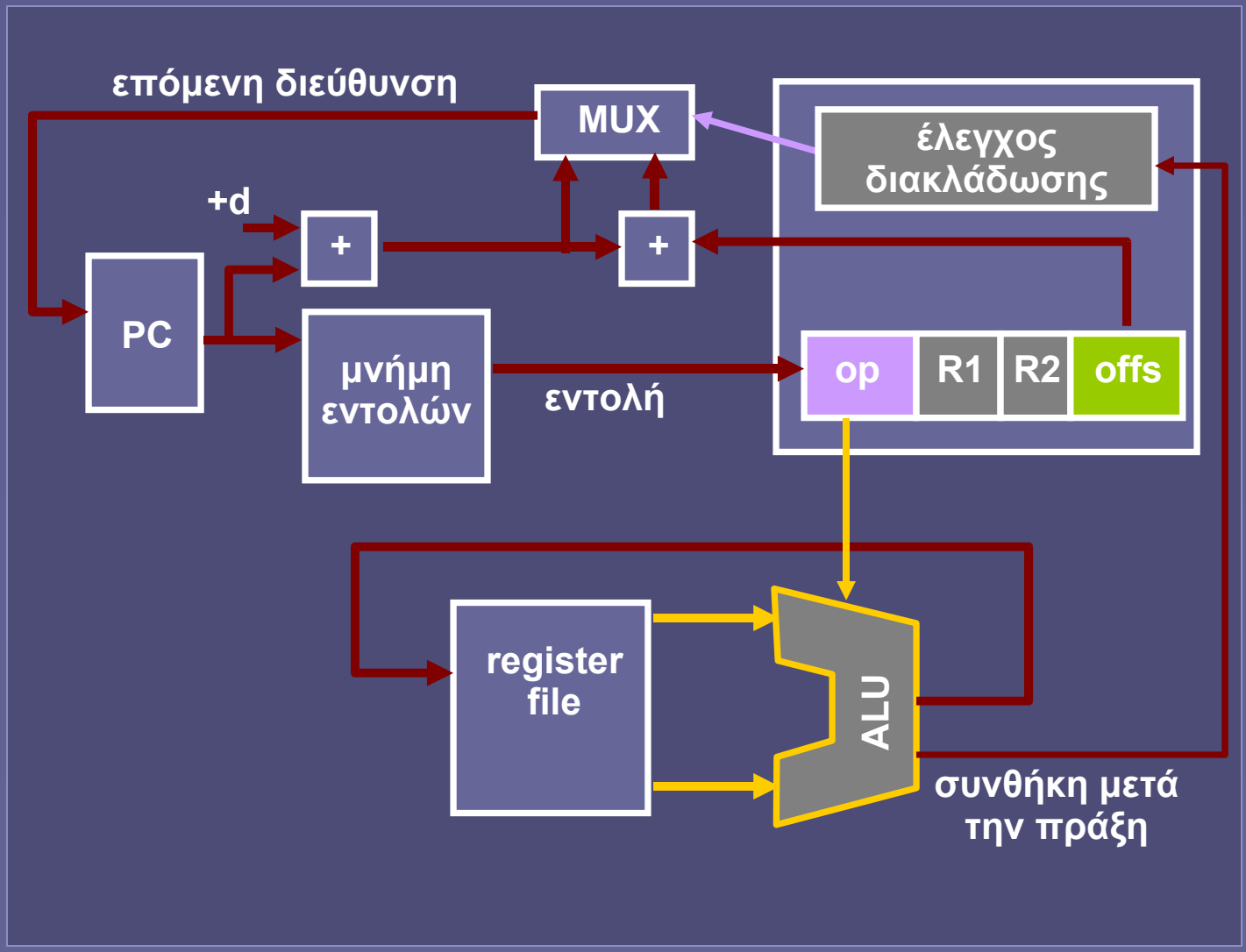
- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις



- Επιλογή νέας επόμενης διεύθυνσης
  - Η νέα τιμή του PC
  - Από τη μονάδα ελέγχου
  - Σύμφωνα με τιμές από πράξη στην ΑΛΜ
    - Π.χ. σύγκριση 2 καταχωρητών

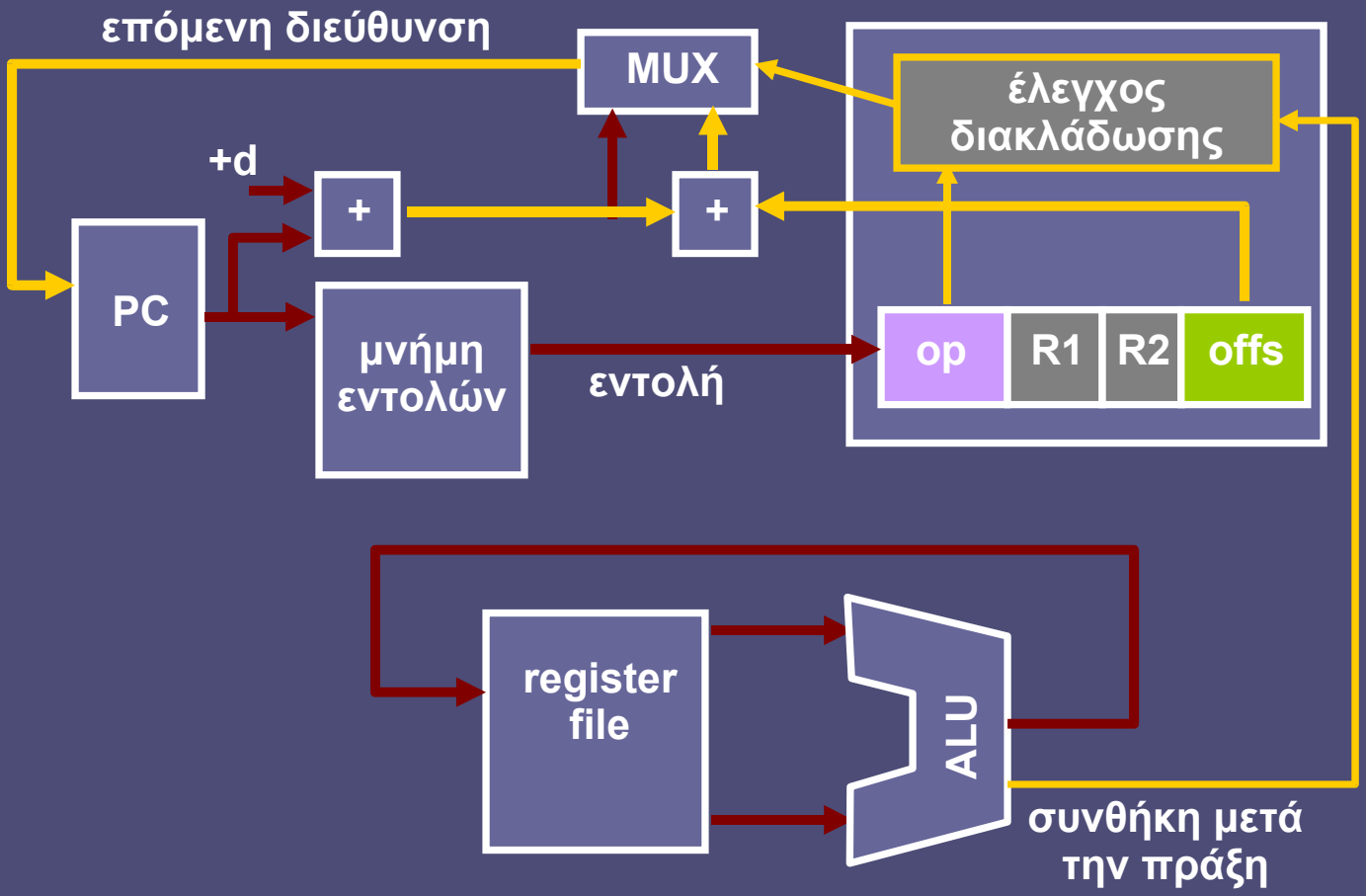
Execute (EX)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις



Ενημέρωση PC ("DM")

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις



Εντολές προσπέλασης μνήμης (load-store)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης

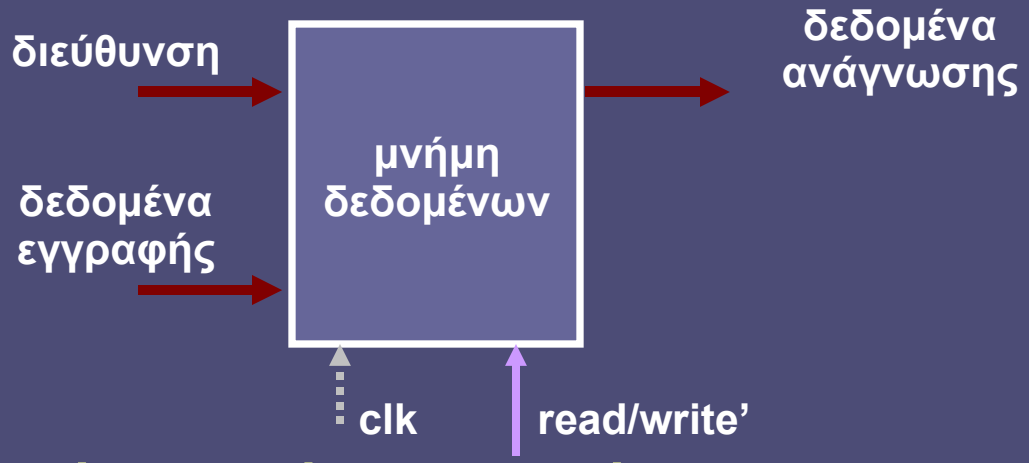
- Παράδειγμα

R1 ← mem[R2 + offset]

- Μόνο οι εντολές load και store προσπελαύνουν τη μνήμη δεδομένων
- Η διεύθυνση μνήμης παράγεται από την ΑΛΜ
  - Ως άθροισμα καταχωρητή + σταθεράς

Η "μνήμη δεδομένων"

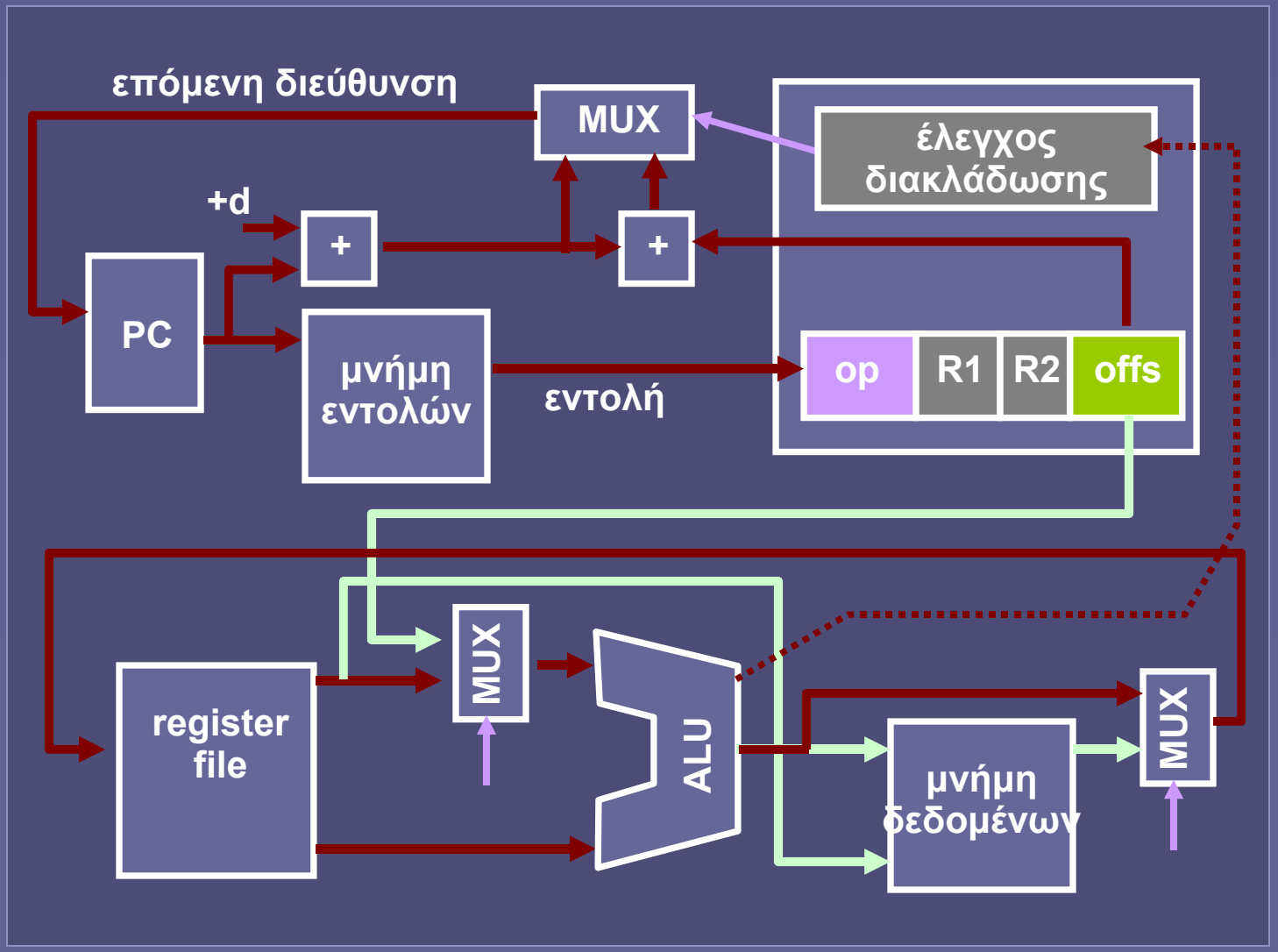
- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



- Μια επίσης χρήσιμη αφαίρεση
  - Ή εγγραφή ή ανάγνωση
    - clk για την εγγραφή
  - Στην πραγματικότητα
    - Αντιπροσωπεύει μια ιεραρχία μνήμης

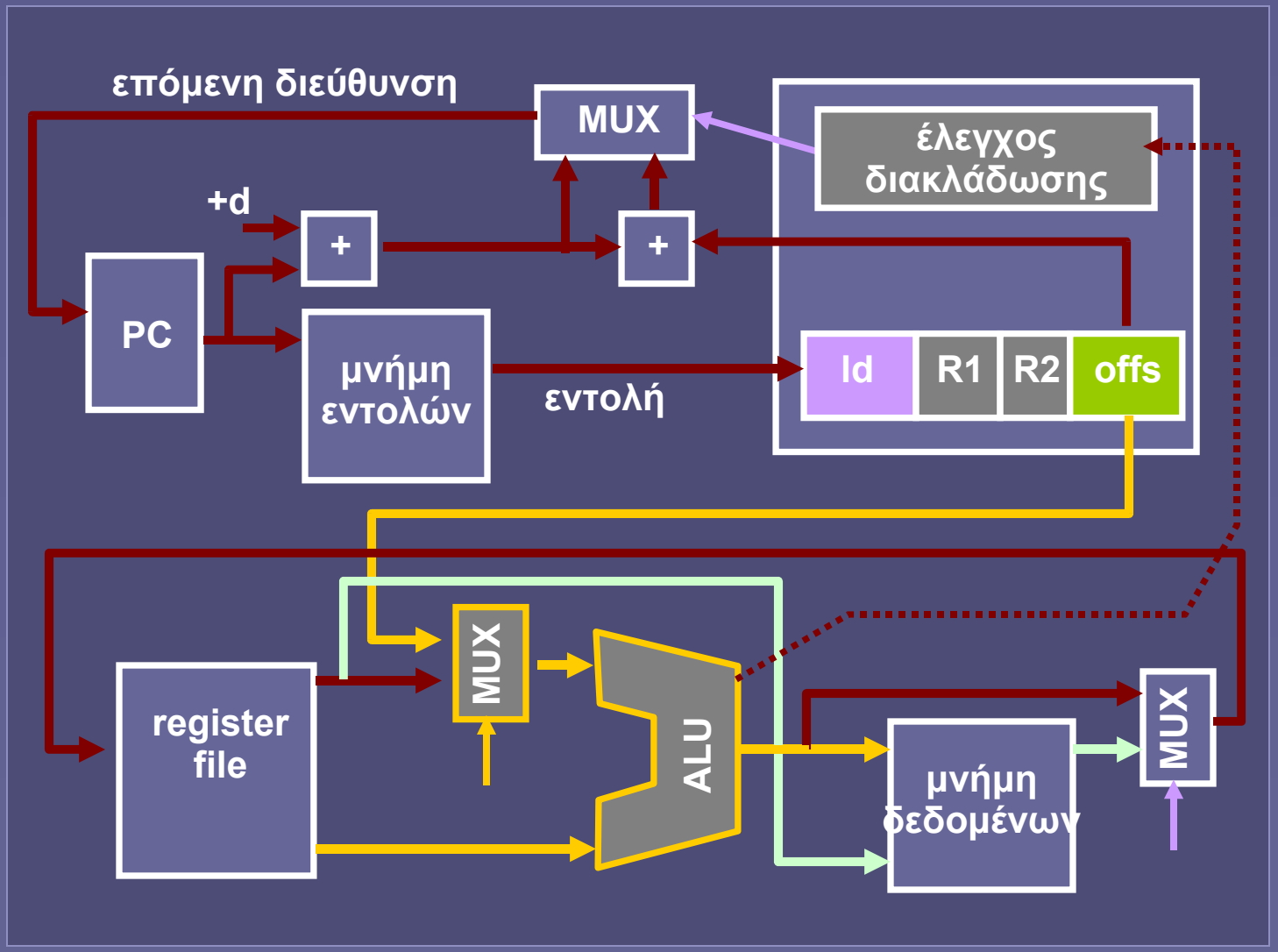
Προσπέλαση μνήμης δεδομένων

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



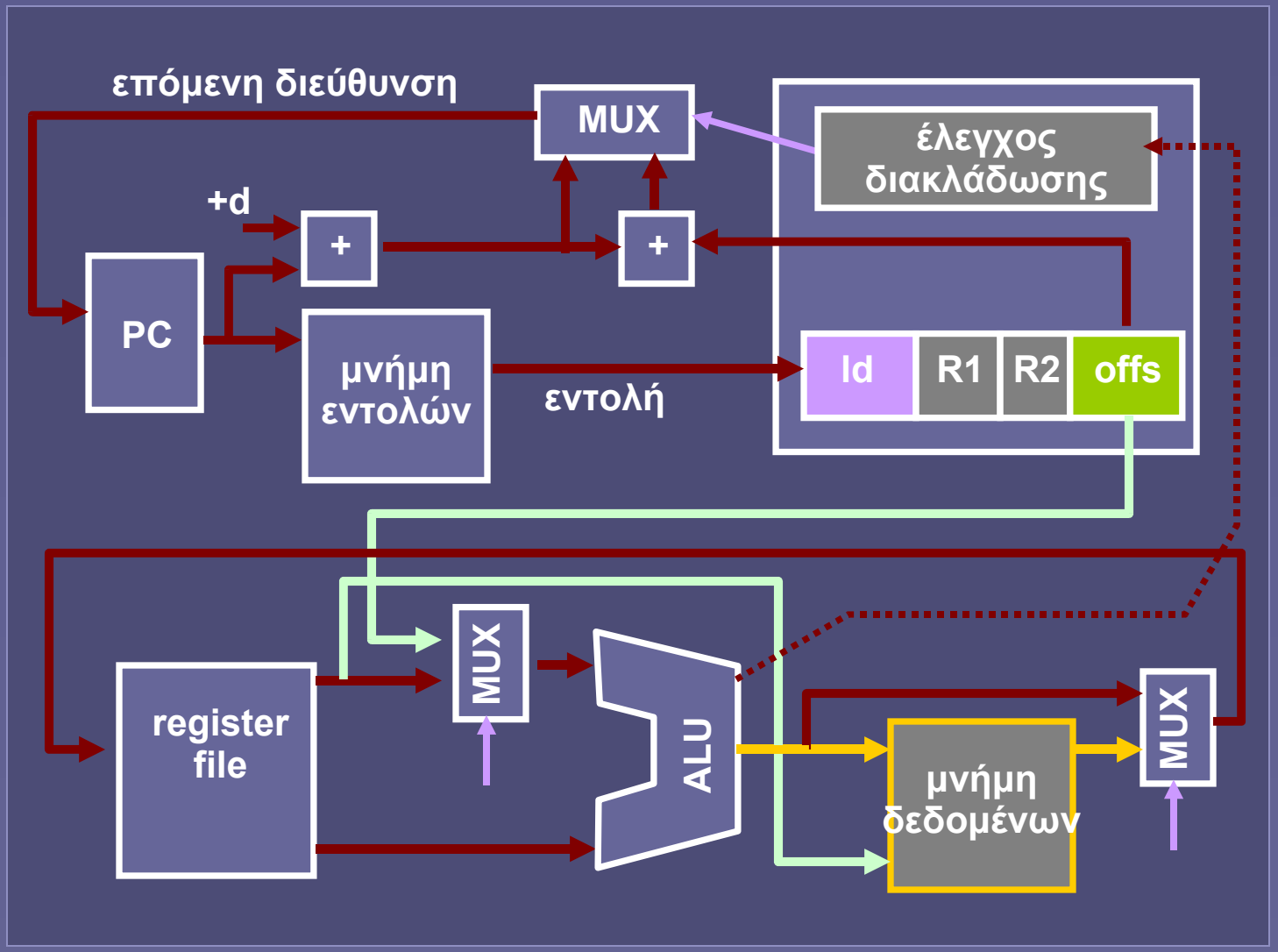
Execute (EX)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



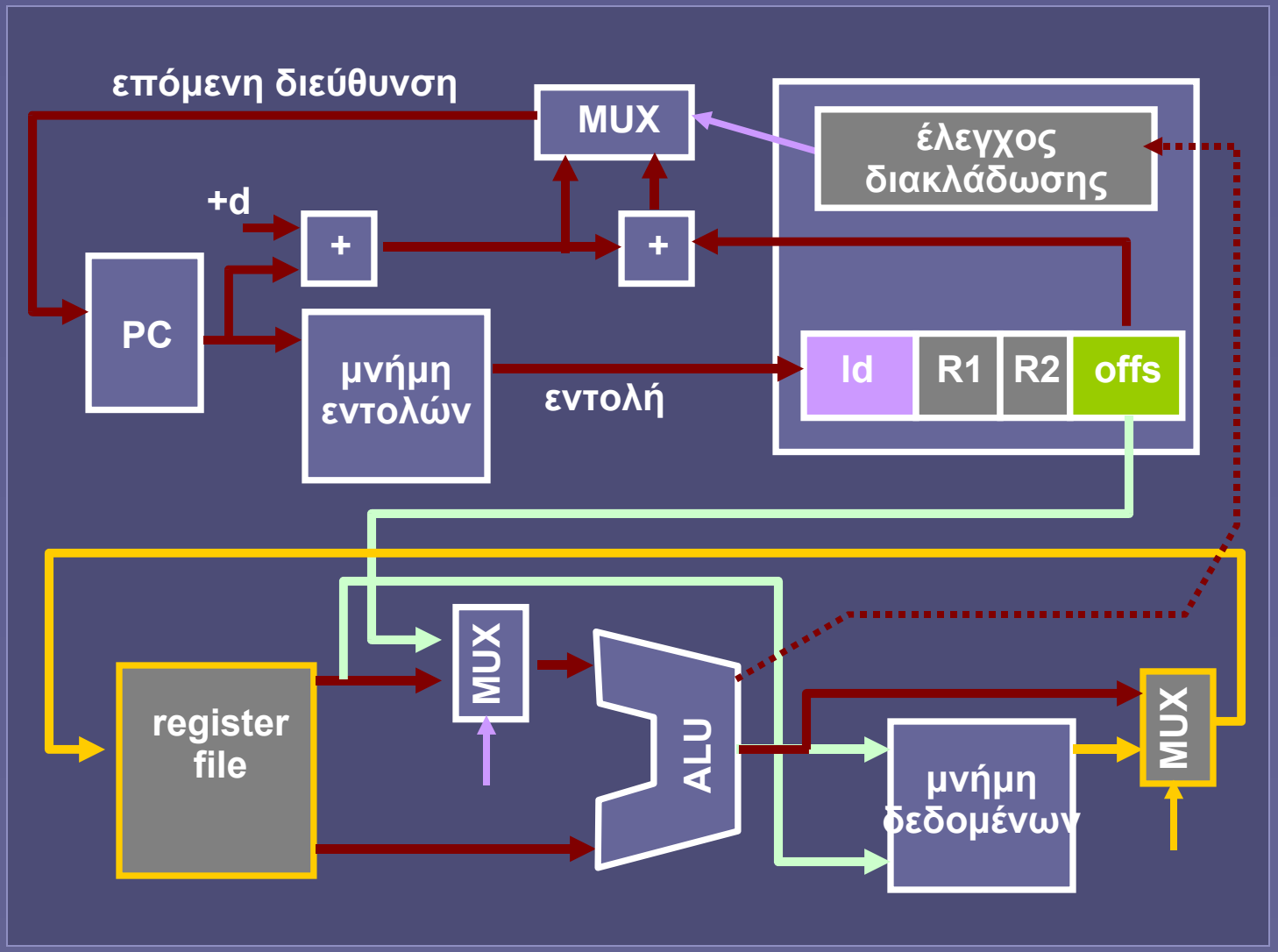
Data Memory Access (DM)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



Write Back (WB)

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης



Σύνοψη λειτουργίας ΚΜΕ

- Ο επεξεργαστής
- ΚΜΕ
- ΑΛΜ
- Καταχωρητές
- Datapath
- Μονάδα Ελέγχου
- Εντολές
- Διακλαδώσεις
- Προσπέλαση μνήμης
- Σύνοψη

- Κύκλοι ρολογιού ανά εντολή
  - Clocks per Instruction (CPI)
    - Στο σημερινό παράδειγμα ΚΜΕ: 1 κύκλος ανά εντολή
- Περίοδος κύκλου ρολογιού
  - Clock Cycle (Period) (CC)
- Στο σημερινό παράδειγμα ΚΜΕ
  - CC σταθερό ανεξάρτητα από είδος εντολής
  - CC εξαρτάται από την πιο χρονοβόρα πράξη!
  - Όχι αποδοτικό σχήμα
  - Υπάρχει τεχνική για βελτίωση της απόδοσης;
    - (στο επόμενο μάθημα)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Απόδοση ΚΜΕ

(Μέτρηση και τεχνικές βελτίωσης απόδοσης)

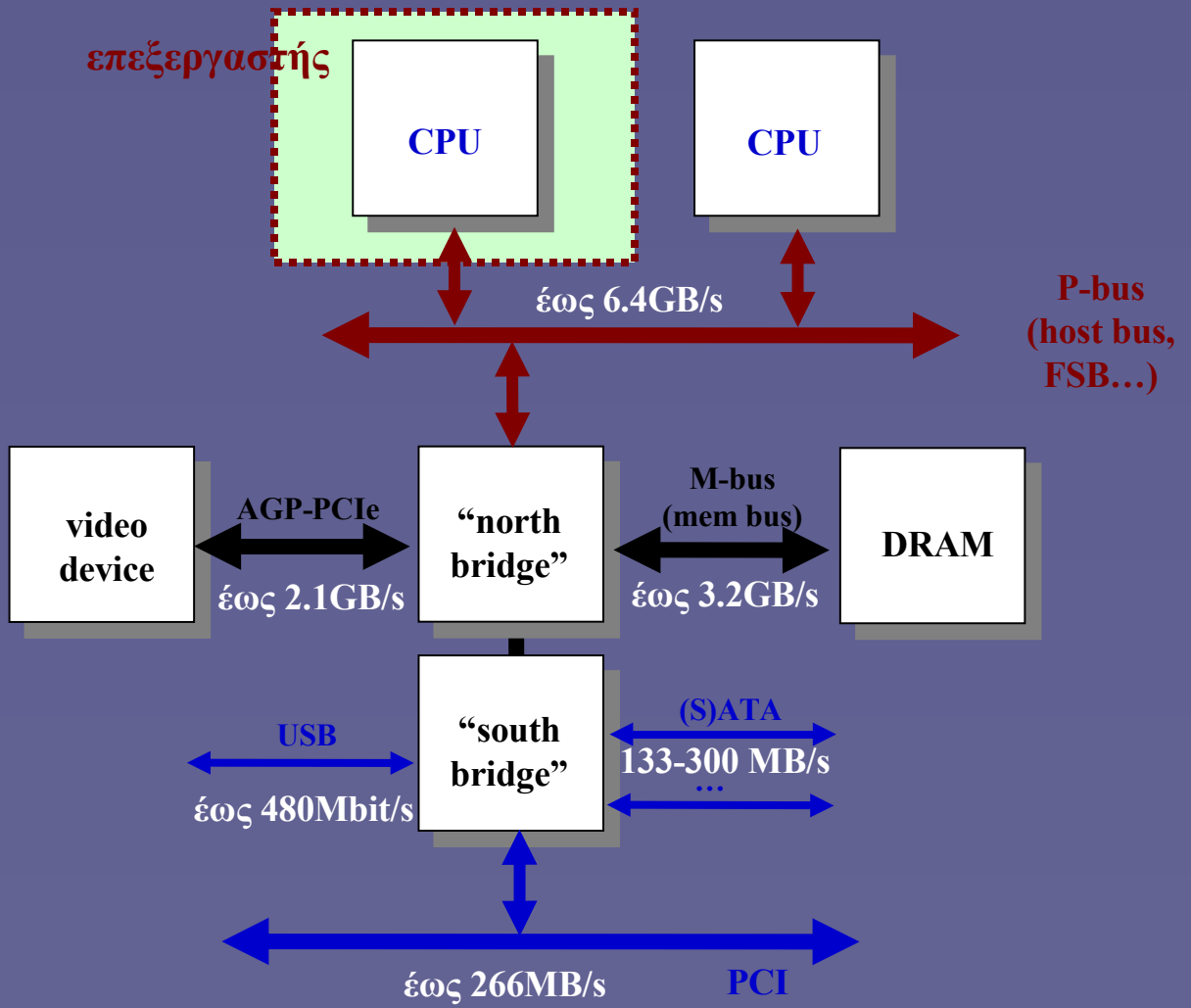
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Κεντρική Μονάδα Επεξεργασίας

- Απόδοση ΚΜΕ

- Επεξεργαστής ή...
- Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)



Απόδοση ΚΜΕ

- Απόδοση ΚΜΕ

- Υπολογιστικό σύστημα
  - Η απόδοση εξαρτάται από όλα τα επιμέρους τμήματά του
    - Υλικό και λογισμικό
- Κεντρική Μονάδα Επεξεργασίας
  - Πόσο γρήγορα εκτελείται ένα πρόγραμμα;
    - Επίσης σε αρχιτεκτονικές ειδικού σκοπού: αξιοπιστία, κατανάλωση ενέργειας
  - Πώς επηρεάζει η αρχιτεκτονική την απόδοση;
    - Πόσο γρηγορότερα εκτελείται ένα πρόγραμμα μετά από μια αρχιτεκτονική αλλαγή;

Εκτέλεση προγραμμάτων

- Απόδοση ΚΜΕ

- Χρόνος εκτέλεσης (execution time)
  - Επίσης: response time
    - Ο συνολικός χρόνος εκτέλεσης ενός προγράμματος
    - Μαζί με επιβάρυνση λειτουργικού συστήματος
  - Αύξηση απόδοσης ⇔ Μείωση χρόνου εκτέλεσης
  - Για υπολογιστή Χ:

$$\text{Απόδοση(X)} = \frac{1}{\text{Χρόνος Εκτέλεσης(X)}}$$

Throughput:
Ο ρυθμός ολοκλήρωσης έργου σε συγκεκριμένο χρόνο

Συγκρίνοντας δύο υπολογιστές

- Απόδοση ΚΜΕ

;

Ο Χ εκτελεί ένα πρόγραμμα σε 10 sec και ο Υ σε 15 sec. Πόσο πιο γρήγορος είναι ο Χ;

- Συγκρίνοντας αποδόσεις
  - Για υπολογιστές Χ και Υ
  - Εάν:

Απόδοση(Χ) > Απόδοση(Υ)

  - Τότε (και αντίστροφα):

Χρόνος Εκτέλεσης(Χ) < Χρόνος Εκτέλεσης(Υ)

$$\frac{\text{Απόδοση(X)}}{\text{Απόδοση(Y)}} = \frac{\text{Χρόνος Εκτέλεσης(Y)}}{\text{Χρόνος Εκτέλεσης(X)}} = n$$

Ο Χ είναι *n* φορές γρηγορότερος από τον Υ

Απόδοση ΚΜΕ

- Απόδοση ΚΜΕ

- Χρόνος εκτέλεσης προγράμματος
  - Εξαρτάται από όλα τα επιμέρους τμήματα του υπολογιστικού συστήματος
- Χρόνος εκτέλεσης ΚΜΕ
  - Ο χρόνος για τον οποίο η ΚΜΕ εκτελεί εντολές του προγράμματος
    - Όχι χρόνος για Ε/Ε ή άλλες διεργασίες
  - Χρόνος προγράμματος χρήστη
    - Για το πρόγραμμα καθεαυτό
  - Χρόνος συστήματος
    - Λειτουργίες ΛΣ για την εξυπηρέτηση του προγράμματος

Βασικά μεγέθη μέτρησης απόδοσης ΚΜΕ

- Απόδοση ΚΜΕ

;

Πώς υπολογίζεται το CPI όταν υπάρχουν διαφορετικοί τύποι εντολών;

- Κύκλος ρολογιού (περίοδος)
  - Clock Cycle (CC)
    - Η διάρκεια ενός κύκλου ρολογιού κατά τον οποίο η ΚΜΕ εκτελεί τις μικρότερες βασικές λειτουργίες
    - Σταθερό μέγεθος
- Κύκλοι ρολογιού ανά εντολή
  - Clocks Per Instruction (CPI)
    - Οι απαιτούμενοι κύκλοι ρολογιού για την ολοκλήρωση μιας εντολής
    - Ενδεχομένως διαφορετικό μέγεθος ανά τύπο εντολής
- Αριθμός εντολών
  - Instruction Count (IC)
    - Ο αριθμός των εντολών
    - Διάφοροι τύποι εντολών

Συσχέτιση με λογισμικό

- Απόδοση ΚΜΕ

- Αλγόριθμος
  - Καθορίζει το IC
  - Ενδεχομένως καθορίζει το CPI, ευνοώντας ορισμένους τύπους εντολών (π.χ. κινητής υποδιαστολής)
- Γλώσσα προγραμματισμού
  - Καθορίζει το IC (μετάφραση εντολών υψηλού επιπέδου)
  - Καθορίζει το CPI απαιτώντας συγκεκριμένους τύπους εντολών
- Μεταγλωττιστής
  - Καθορίζει το IC
  - Καθορίζει το CPI

Το υλικό (ISA) καθορίζει και τα τρία μεγέθη (IC, CPI και CC)

Χρόνος εκτέλεσης ΚΜΕ (χρήστη)

- Απόδοση ΚΜΕ

- Χρόνος Εκτέλεσης ΚΜΕ για ένα πρόγραμμα

  ExecTime = IC * CPI * CC

- Τι μπορεί να κάνει ο σχεδιαστής ΚΜΕ για να βελτιώσει την απόδοση;
  - **Να μειώσει τον κύκλο ρολογιού (CC)**
  - **Να μειώσει τον αριθμό κύκλων ανά εντολή (CPI)**
  - Οι εντολές παραμένουν ως έχουν

Παράδειγμα

- Απόδοση ΚΜΕ

Τύπος εντολής	A	B	C
CPI	1	2	3

Ακολουθία κώδικα	A	B	C
1	2	1	2
2	4	1	1

- Επιλογή μεταξύ 2 ακολουθιών εντολών
  - **Ποια ακολουθία εκτελεί τις περισσότερες εντολές;**
  - **Ποια είναι ταχύτερη;**
  - **Ποιο το μέσο CPI σε κάθε περίπτωση;**

[Patterson-Hennessy "Computer Organization and Design", 3rd ed]

Μετροπρογράμματα

- **Benchmarks**
  - Για τη μέτρηση της απόδοσης
  - Και τη σύγκριση μεταξύ υπολογιστών
  - Θα πρέπει να αντιπροσωπεύουν τις πραγματικές εφαρμογές
  - Υπό ρεαλιστικές συνθήκες εκτέλεσης και δεδομένα εισόδου
  - Χωρίς “εσωτερικές” ειδικές βελτιστοποιήσεις
  - Δυνατότητα επανάληψης μέτρησης
  - Διαφορετικά για ανόμοιες κλάσεις υπολογιστών
    - PCs, servers, embedded systems…

Ο νόμος του Amdahl

- Απόδοση ΚΜΕ

;

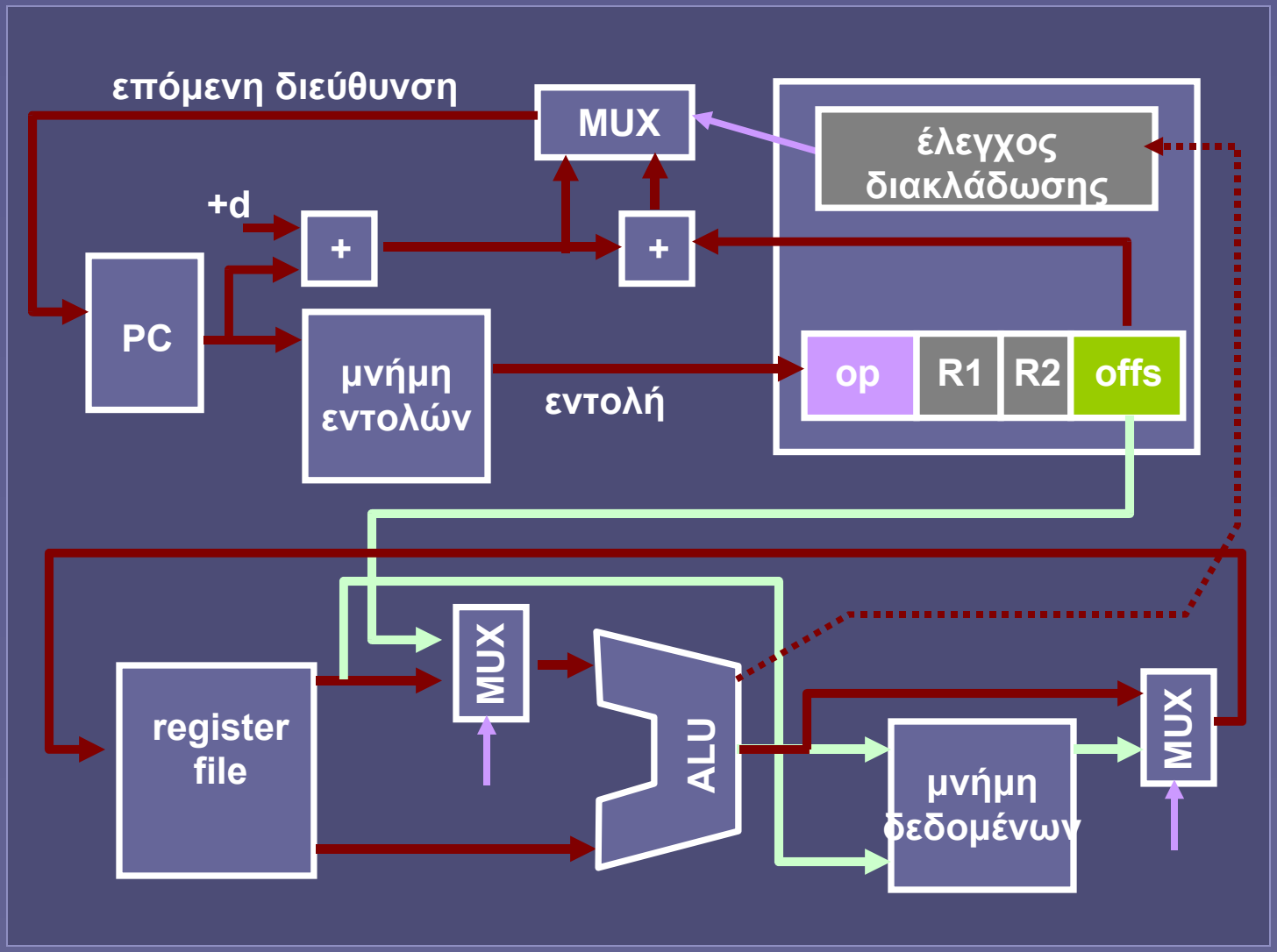
Ένα πρόγραμμα τρέχει για 100 sec σε έναν υπολογιστή και εκτελεί πολλαπλασιασμούς για 80 sec. Πόσο πρέπει να βελτιώσω τη ταχύτητα του πολλαπλασιασμού για να πενταπλασιάσω την απόδοση;

- "Η βελτίωση της συνολικής απόδοσης ενός συστήματος μέσω της εισαγωγής ενός νέου χαρακτηριστικού, περιορίζεται από το βαθμό χρήσης αυτού του νέου χαρακτηριστικού"
- Οι περισσότερο χρησιμοποιούμενες περιπτώσεις πρέπει να είναι γρήγορες!
- Όχι βελτιστοποίηση των σπάνιων περιπτώσεων!

ΚΜΕ ενός κύκλου (single-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Κάθε εντολή ολοκληρώνεται σε έναν κύκλο ρολογιού



ΚΜΕ ενός κύκλου (single-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

- CPI = 1
  - Σε κάθε έναν κύκλο ρολογιού ολοκληρώνεται μια εντολή ή
  - κάθε εντολή απαιτεί έναν κύκλο ρολογιού
- Πόσο πρέπει να είναι το CC;
  - Ίσο με τη διάρκεια της μεγαλύτερης λειτουργίας!
  - Μη αποδοτικό σχήμα
  - Όλες οι εντολές δεν απαιτούν τον ίδιο χρόνο

Παράδειγμα

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Εντολή	IF	ID	EX	DM	WB	Σύνολο
Αριθμητική	200	50	100	0	50	**400 ps**
Διακλάδωση	200	50	100	0	0	**350 ps**
Ανάγνωση μνήμης	200	50	100	200	50	**600 ps**
Εγγραφή μνήμης	200	50	100	200	0	**550 ps**

- CC πρέπει να είναι 600 ps (single cycle CPU)
  - Αν ήταν δυνατή η χρήση με μεταβλητό CC (πρακτικά αδύνατο!)
  - Ποια η βελτίωση της απόδοσης;
  - 25% ανάγνωση, 10% εγγραφή, 45% αριθμητικές, 20% διακλάδωσης

ΚΜΕ πολλαπλών κύκλων (multi-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

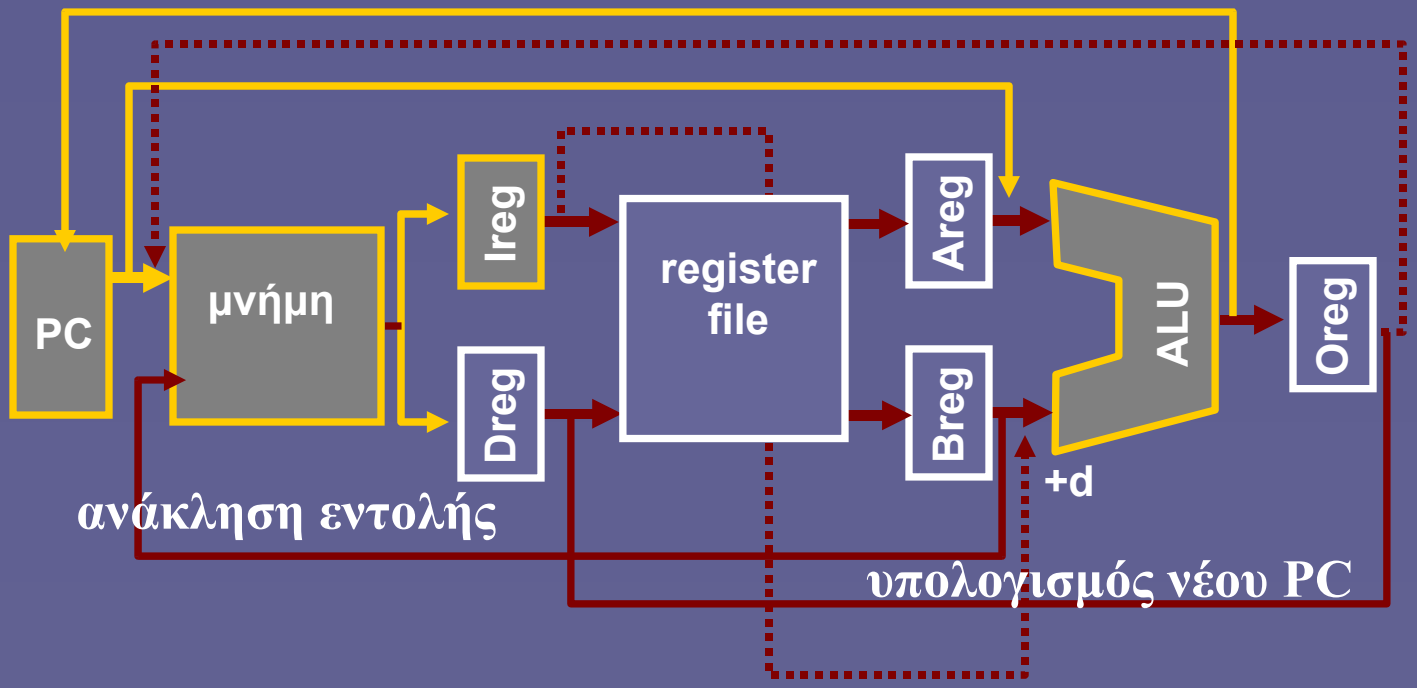
- CPI > 1
  - Κάθε εντολή χωρίζεται σε έναν μεταβλητό αριθμό βημάτων
  - Κάθε βήμα απαιτεί έναν κύκλο ρολογιού
- Πόσο πρέπει να είναι το CC;
  - Ίσο με τη διάρκεια του μεγαλύτερου βήματος
    - Τα βήματα επιλέγονται με μικρή διάρκεια
  - Μέρη της ΚΜΕ μπορούν να χρησιμοποιηθούν για περισσότερες από μία φορές κατά την εκτέλεση μιας εντολής
    - Αρκεί να ανήκουν σε διαφορετικό βήμα
    - Καταχωρητές για τη συγκράτηση αποτελεσμάτων μεταξύ βημάτων

Παράδειγμα: Εντολή load (βήμα IF)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

!

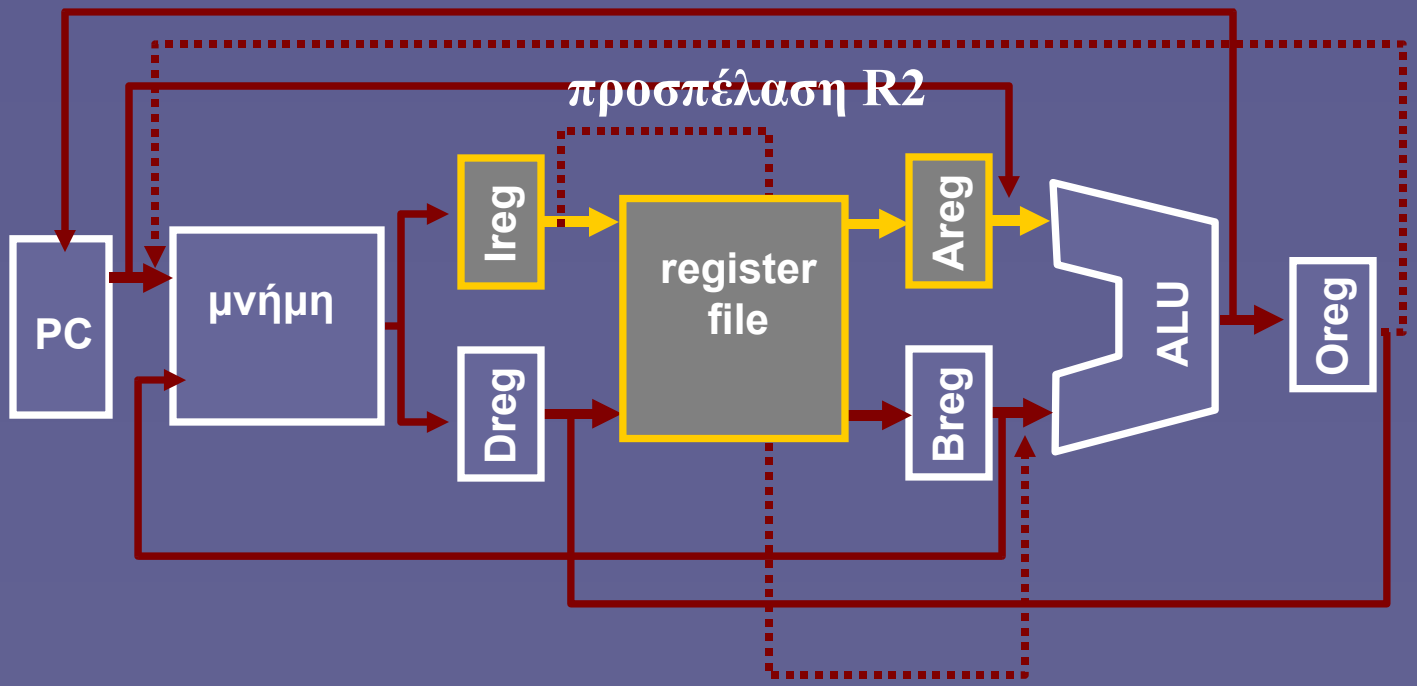
Η εντολή load απαιτεί τα περισσότερα βήματα για να εκτελεστεί!



R1 ← mem[R2 + offset]

Παράδειγμα: Εντολή load (βήμα ID)

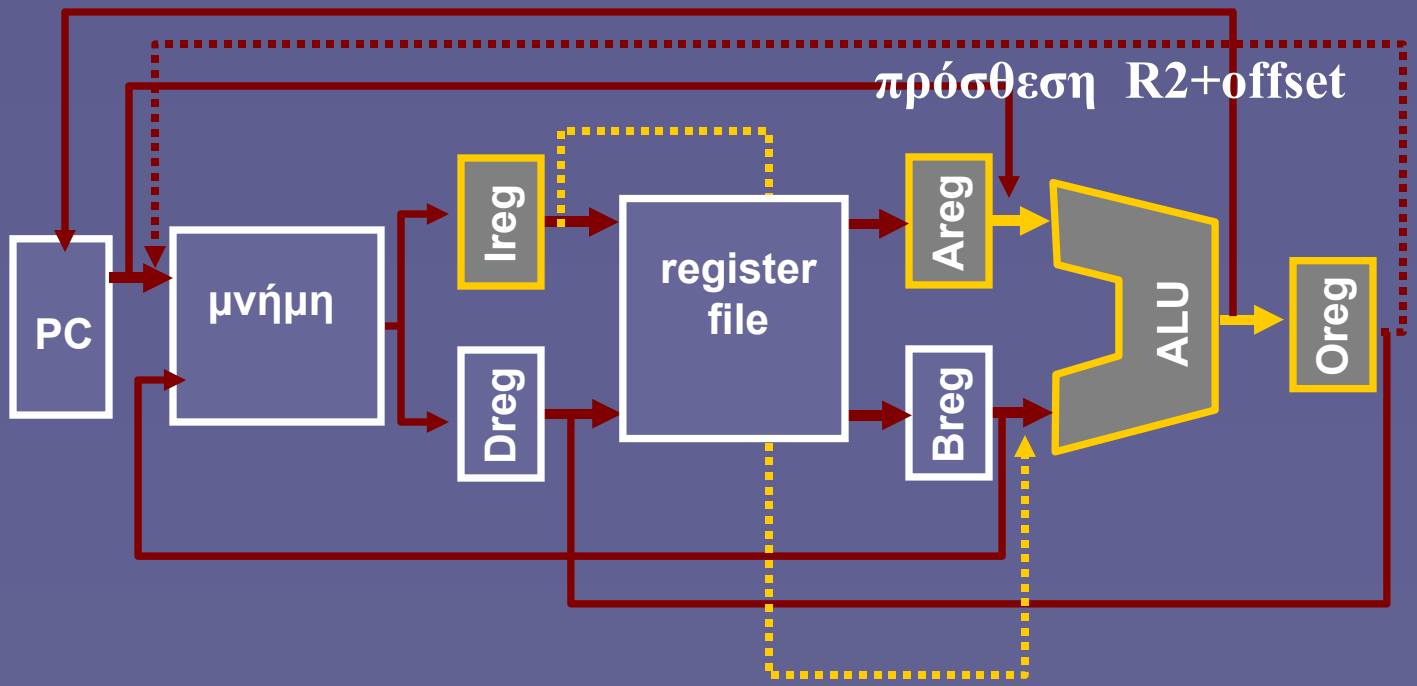
- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



R1 ← mem[R2 + offset]

Παράδειγμα: Εντολή load (βήμα EX)

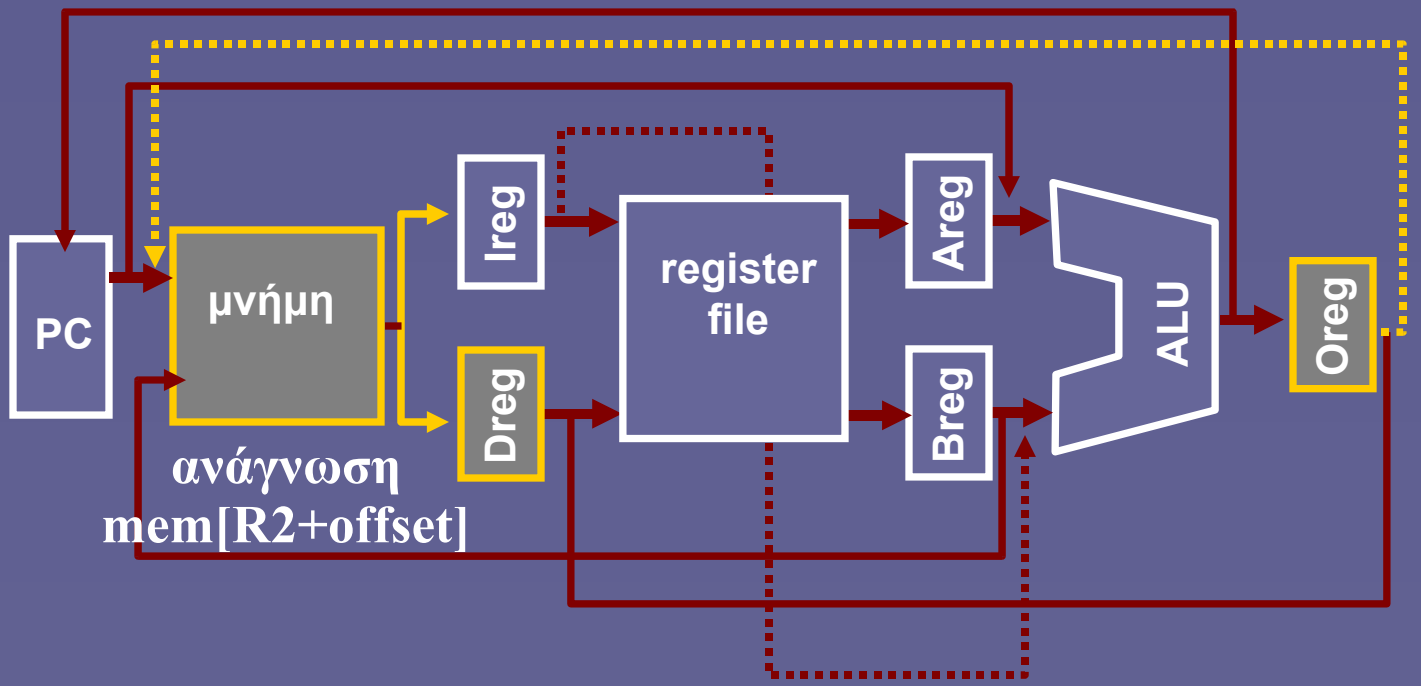
- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



R1 ← mem[R2 + offset]

Παράδειγμα: Εντολή load (βήμα DM)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



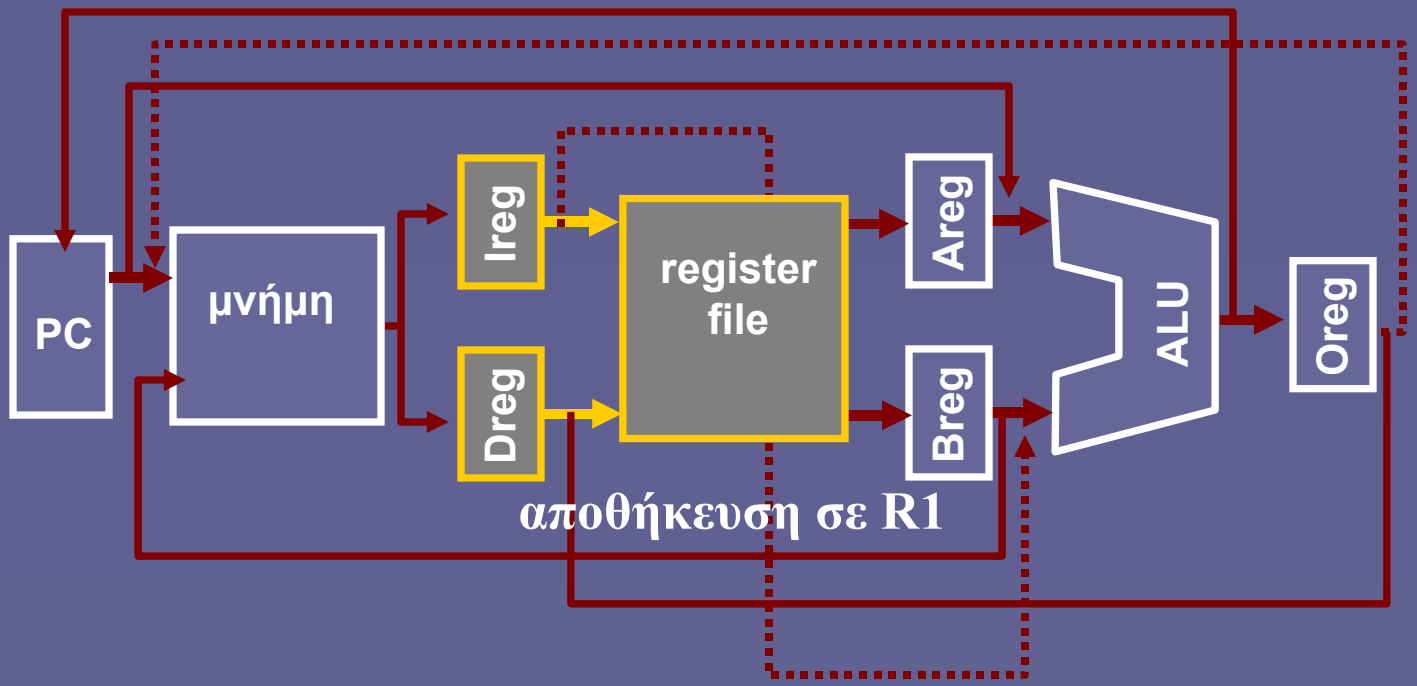
R1 ← mem[R2 + offset]

Παράδειγμα: Εντολή load (βήμα WB)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

;

Σε κάθε βήμα, μερικά τμήματα μένουν ανενεργά. Πώς θα μπορούσαμε να τα εκμεταλλευτούμε;



R1 ← mem[R2 + offset]

Μονάδα Ελέγχου ΚΜΕ πολλαπλών κύκλων

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

- **Πολυπλοκότητα σημάτων ελέγχου**
  - Σε κάθε βήμα εκτέλεσης της εντολής
  - Ανάλογα με το είδος της εντολής
    - Διατήρηση προηγούμενης κατάστασης
- **Μέθοδοι υλοποίησης**
  - Αυτόματα πεπερασμένων καταστάσεων
    - Ακολουθιακά λογικά κυκλώματα
    - Παραγωγή σημάτων ελέγχου ανάλογα με εισόδους και τρέχουσα κατάσταση
  - Μικροπρόγραμμα
    - Καθορισμός σημάτων μέσω μικροεντολών
    - Εσωτερικά στην ΚΜΕ (δεν αλλάζουν μετά τη σχεδίαση)
    - Για υλοποίηση σύνθετων εντολών με πολλά βήματα και πολλαπλά περάσματα από το datapath

Απόδοση ΚΜΕ πολλαπλών κύκλων

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

- **Πλεονεκτήματα**
  - Δεν απαιτείται ο μέγιστος χρόνος για το CC
  - Μέρη της ΚΜΕ χρησιμοποιούνται με πολλαπλό τρόπο κατά την εκτέλεση μιας εντολής
- **Μειονεκτήματα**
  - Η μονάδα ελέγχου γίνεται πολυπλοκότερη
    - Η πολυπλοκότητα πιθανόν να ακυρώνει τα πλεονεκτήματα!
- **Σήμερα**
  - Οι ΚΜΕ υψηλής απόδοσης χρησιμοποιούν πρόσθετες τεχνικές παραλληλισμού σε επίπεδο εντολών (**instruction level parallelism – ILP**)
    - (στο επόμενο μάθημα...)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Παραλληλισμός σε επίπεδο εντολών

(Pipelining και άλλες τεχνικές αύξησης απόδοσης)

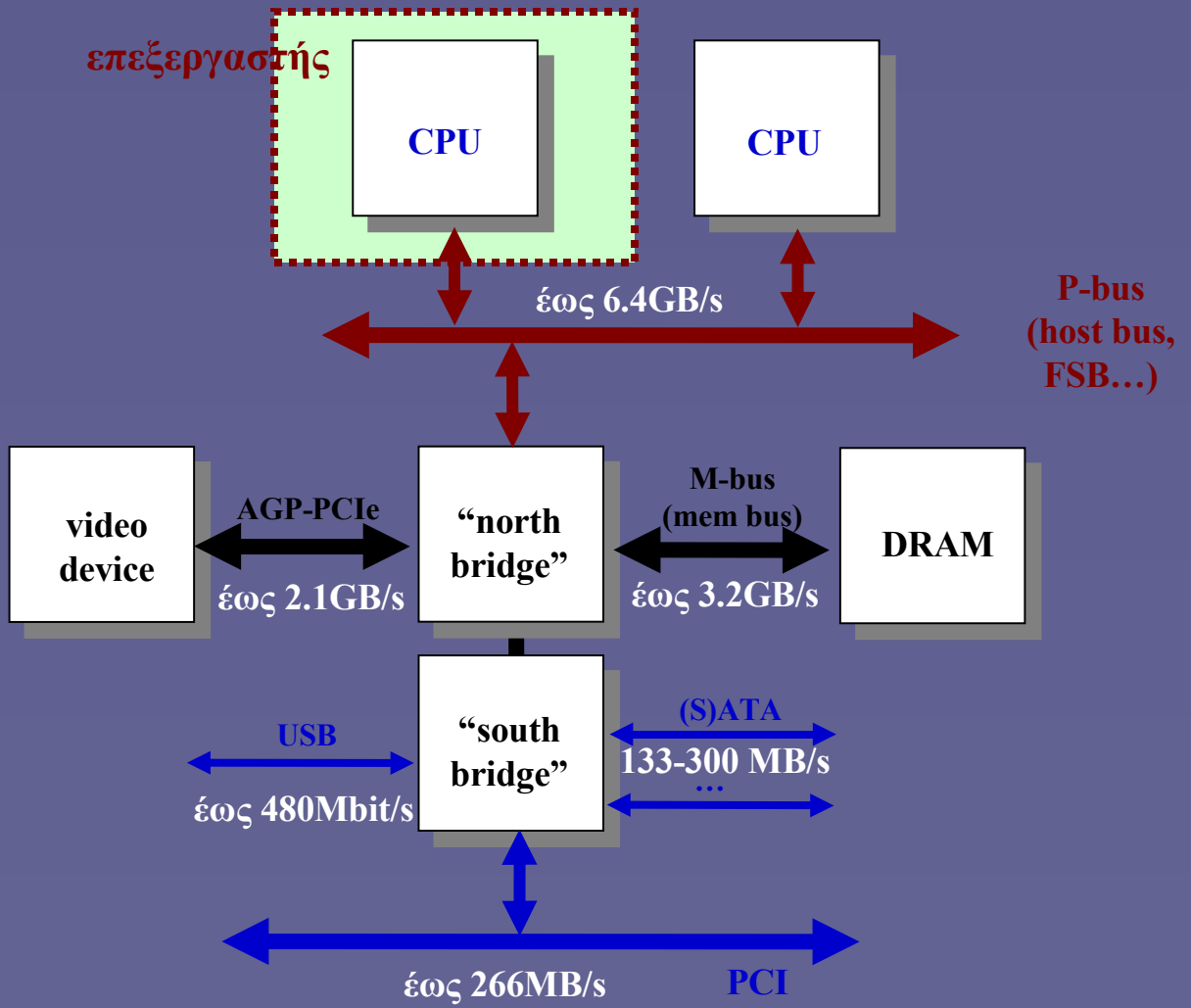
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Κεντρική Μονάδα Επεξεργασίας

- Απόδοση ΚΜΕ

- Επεξεργαστής ή...
- Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)



Επανάληψη: Απόδοση ΚΜΕ

- Απόδοση ΚΜΕ

- Υπολογιστικό σύστημα
  - Η απόδοση εξαρτάται από όλα τα επιμέρους τμήματά του
    - Υλικό και λογισμικό
- Κεντρική Μονάδα Επεξεργασίας
  - Πόσο γρήγορα εκτελείται ένα πρόγραμμα;

ExecTime = IC * CPI * CC

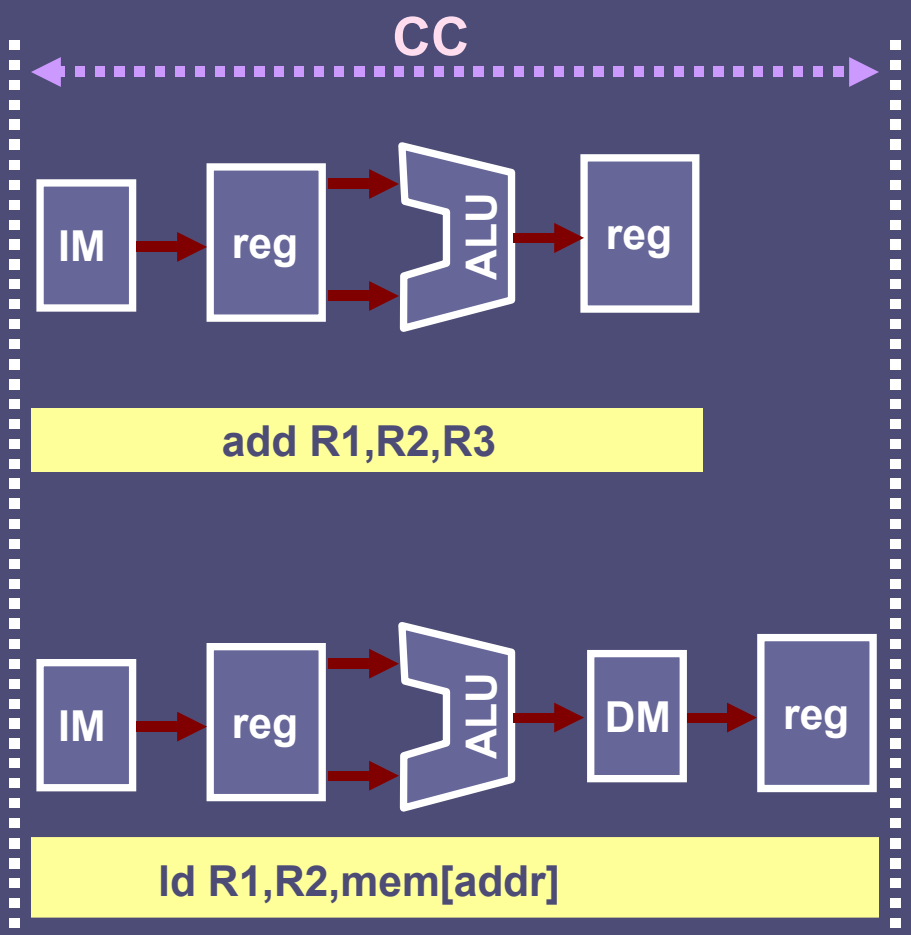
ΚΜΕ ενός κύκλου (single-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Κάθε εντολή ολοκληρώνεται σε έναν κύκλο ρολογιού

CPI = 1

CC = διάρκεια αργότερης εντολής



Παράδειγμα

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Εντολή	IF	ID	EX	DM	WB	Σύνολο
Αριθμητική	200	50	100	0	50	400 ps
Διακλάδωση	200	50	100	0	0	350 ps
Ανάγνωση μνήμης	200	50	100	200	50	600 ps
Εγγραφή μνήμης	200	50	100	200	0	550 ps

- CC πρέπει να είναι 600 ps
  - Αν ήταν δυνατή η χρήση με μεταβλητό CC (πρακτικά αδύνατο!)
  - Ποια η βελτίωση της απόδοσης;
  - 25% ανάγνωση, 10% εγγραφή, 45% αριθμητικές, 20% διακλάδωσης

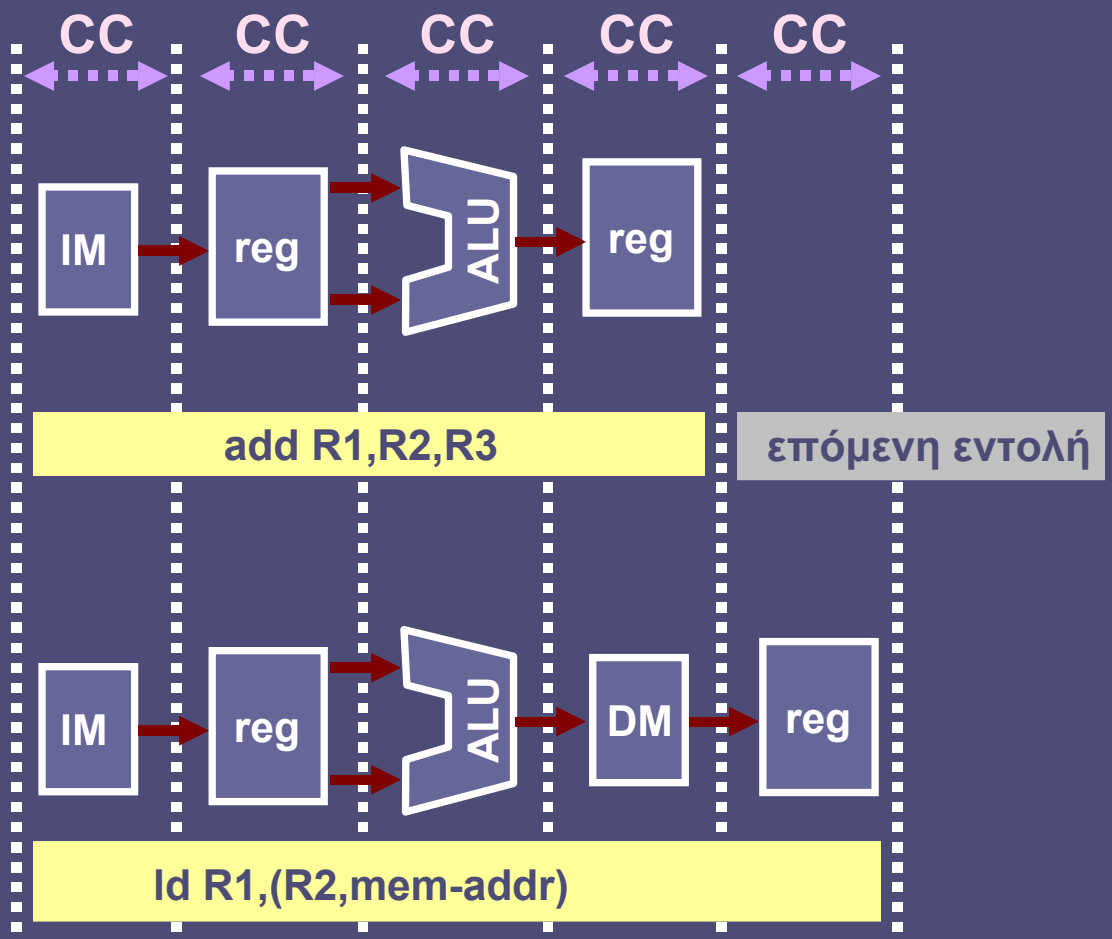
ΚΜΕ πολλαπλών κύκλων (multi-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

Κάθε εντολή ολοκληρώνεται σε μεταβλητό αριθμό κύκλων ρολογιού

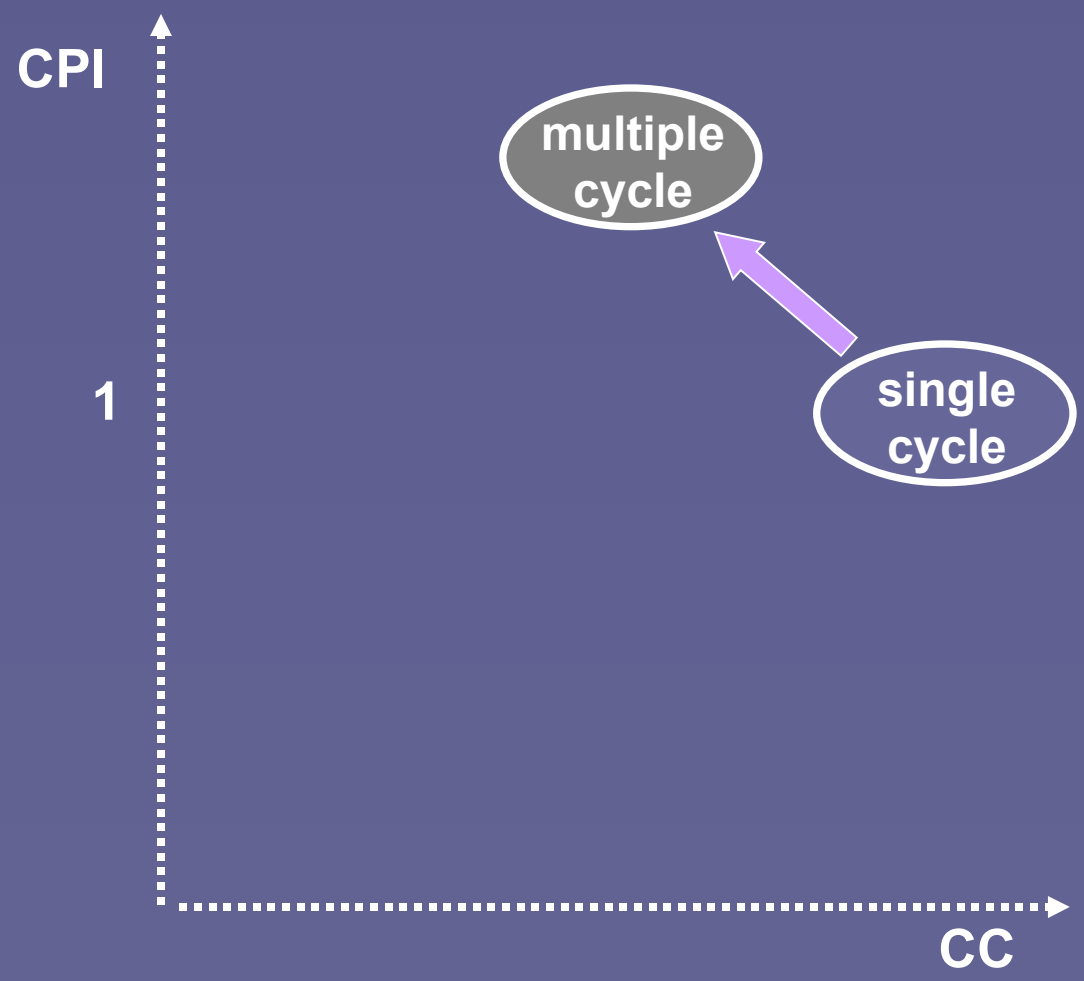
CPI > 1

CC = διάρκεια αργότερου σταδίου



Απόδοση ΚΜΕ πολλαπλών κύκλων

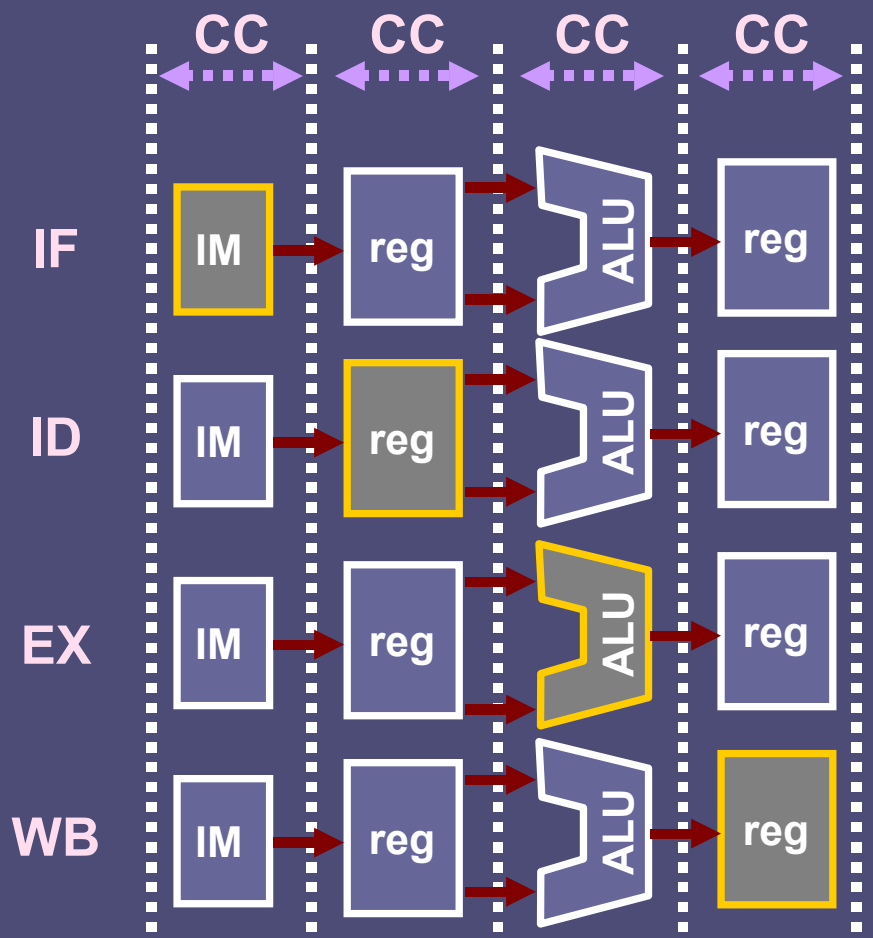
- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



Πιθανότητα περαιτέρω βελτίωσης

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

Σε κάθε κύκλο, ένα μέρος της ΚΜΕ δεν χρησιμοποιείται!



Παραλληλισμός σε επίπεδο εντολών

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP

- **Instruction Level Parallelism (ILP)**
  - Παράλληλη (ταυτόχρονη) εκτέλεση
  - Μεταξύ εντολών
- **Pipelining**
  - Επικάλυψη εκτέλεσης πολλαπλών εντολών
    - Την ίδια στιγμή
    - Βαθμίδες pipeline
    - Ρύθμιση του clock cycle
  - Βασικό στοιχείο αύξησης απόδοσης σύγχρονων επεξεργαστών
  - Στόχος
    - **Να μειωθεί το CC, αλλά ταυτόχρονα**
    - **το CPI να παραμείνει 1**

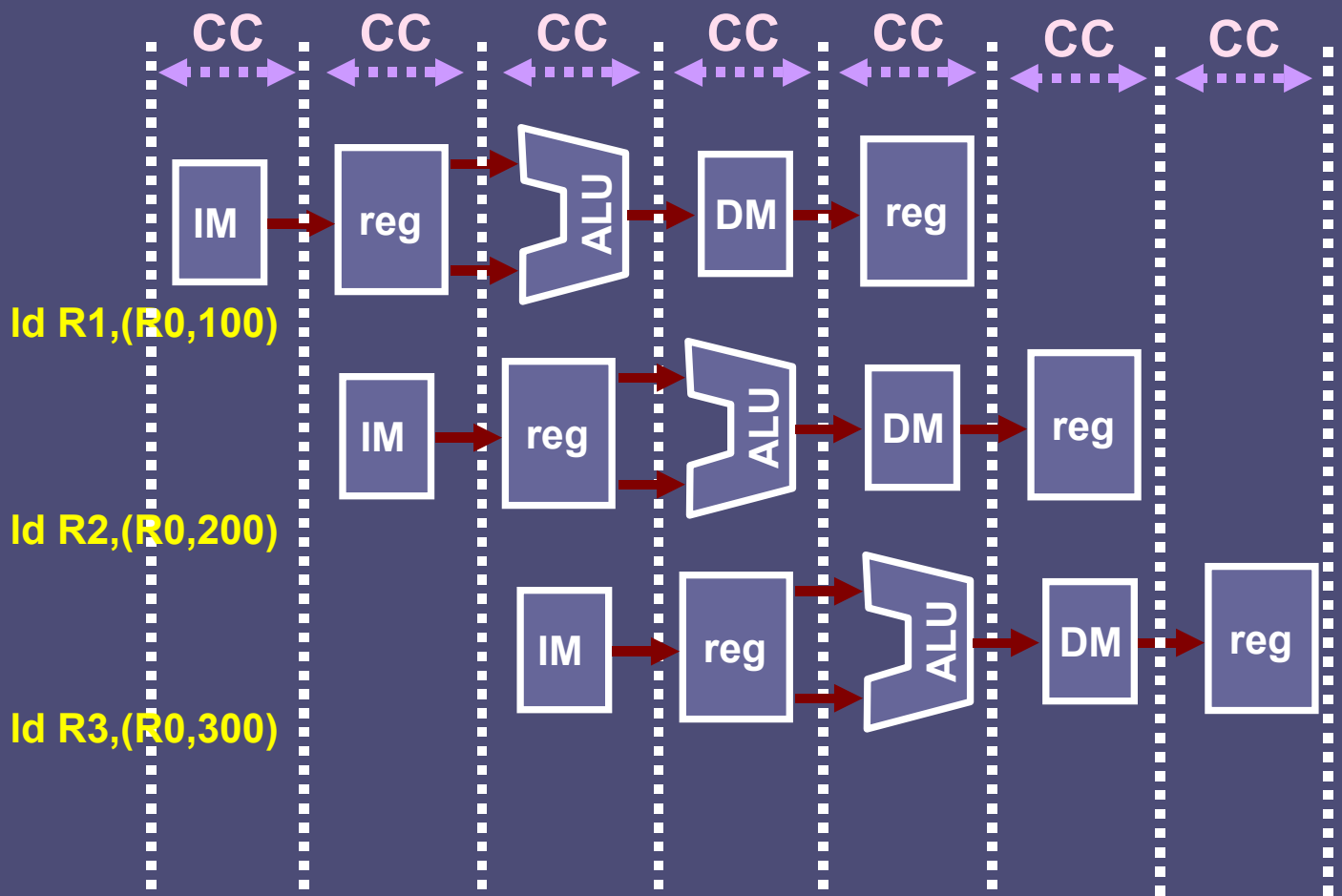
Αρχές τεχνικής pipelining

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

;

Ποιος ο χρόνος εκτέλεσης μιας εντολής;

Κάθε πότε ολοκληρώνεται μια εντολή;



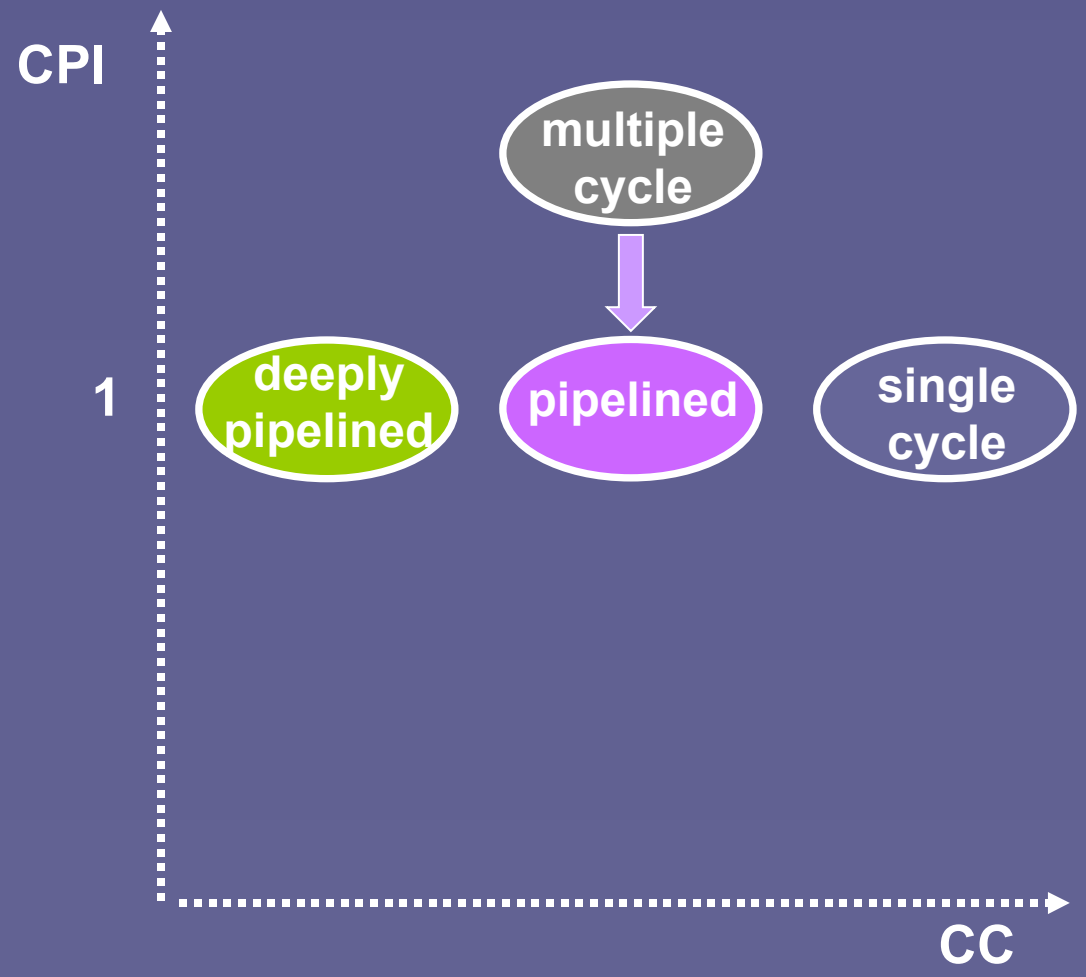
Απόδοση τεχνικής pipelining

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

- Χρόνος ολοκλήρωσης μιας εντολής
  - (latency)
  - Δεν αλλάζει!
- Ρυθμός ολοκλήρωσης εντολών
  - Σε κάθε έναν κύκλο ρολογιού
    - Ολοκληρώνεται μια εντολή
    - Μια νέα ξεκινά την εκτέλεσή της
  - Η ιδανική περίπτωση...
    - Απόδοση συστήματος μνήμης;
    - Χρήση ίδιων βαθμίδων pipeline;
    - Αλληλεξαρτήσεις δεδομένων εντολών;
    - Διακλαδώσεις;

Απόδοση ιδανικού pipeline

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining



Δυσκολίες στην εφαρμογή του pipelining

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

- **Αδυναμία χρήσης μιας βαθμίδας**
  - Structural hazard
  - Η βαθμίδα χρησιμοποιείται από άλλη εντολή
  - Αποφεύγεται με σωστή σχεδίαση
- **Αλληλεξαρτήσεις δεδομένων**
  - Data hazard
  - Τα αναγκαία δεδομένα για την εκτέλεση μιας εντολής δεν είναι διαθέσιμα
    - Υπολογίζονται από προηγούμενη εντολή
  - Αντιμετωπίζεται με πρόσθετο υλικό
    - Forwarding - bypassing

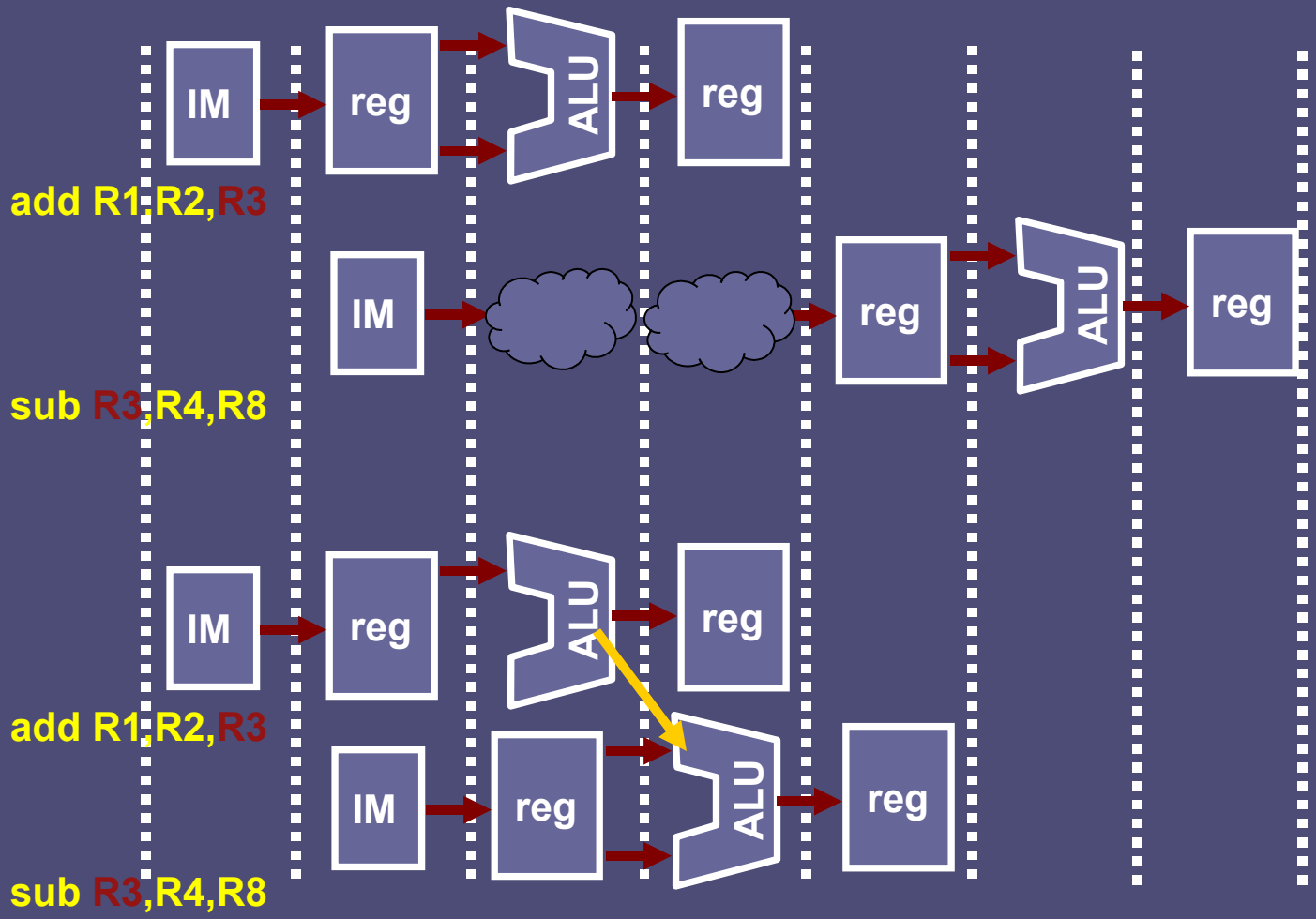
Forwarding - bypassing

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

!

Pipeline **bubbles**

Δεν αποφεύγονται πάντοτε!



Δυσκολίες στην εφαρμογή του pipelining

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

- **Διακλαδώσεις**
  - Control (branch) hazard
  - Η ροή της εκτέλεσης αλλάζει
    - Οι εντολές που έχουν μπει στο pipeline δεν είναι οι σωστές!
  - Πρόβλεψη διακλαδώσεων
    - Δυναμικές μέθοδοι (διατήρηση ιστορίας προηγούμενων διακλαδώσεων)
    - Συσχέτιση με αποφάσεις για άλλες διακλαδώσεις
    - >90% σωστές προβλέψεις
  - Σε περίπτωση λάθους πρόβλεψης
    - Απόρριψη όλων των εντολών στο pipeline
    - Χωρίς να λαμβάνονται υπ'όψη τα αποτελέσματά τους

Pipelining και διακοπές-σφάλματα

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining

- **Interrupts - Exceptions**
  - Ποια εντολή στο pipeline προκάλεσε τη διακοπή;
  - Διακοπή εκτέλεσης εντολής
  - Ολοκλήρωση προηγούμενων εντολών στο pipeline
  - Άδειασμα pipeline (flush) από τυχόν επόμενες εντολές
  - Μετάβαση σε προκαθορισμένη διεύθυνση
    - Λειτουργικό Σύστημα

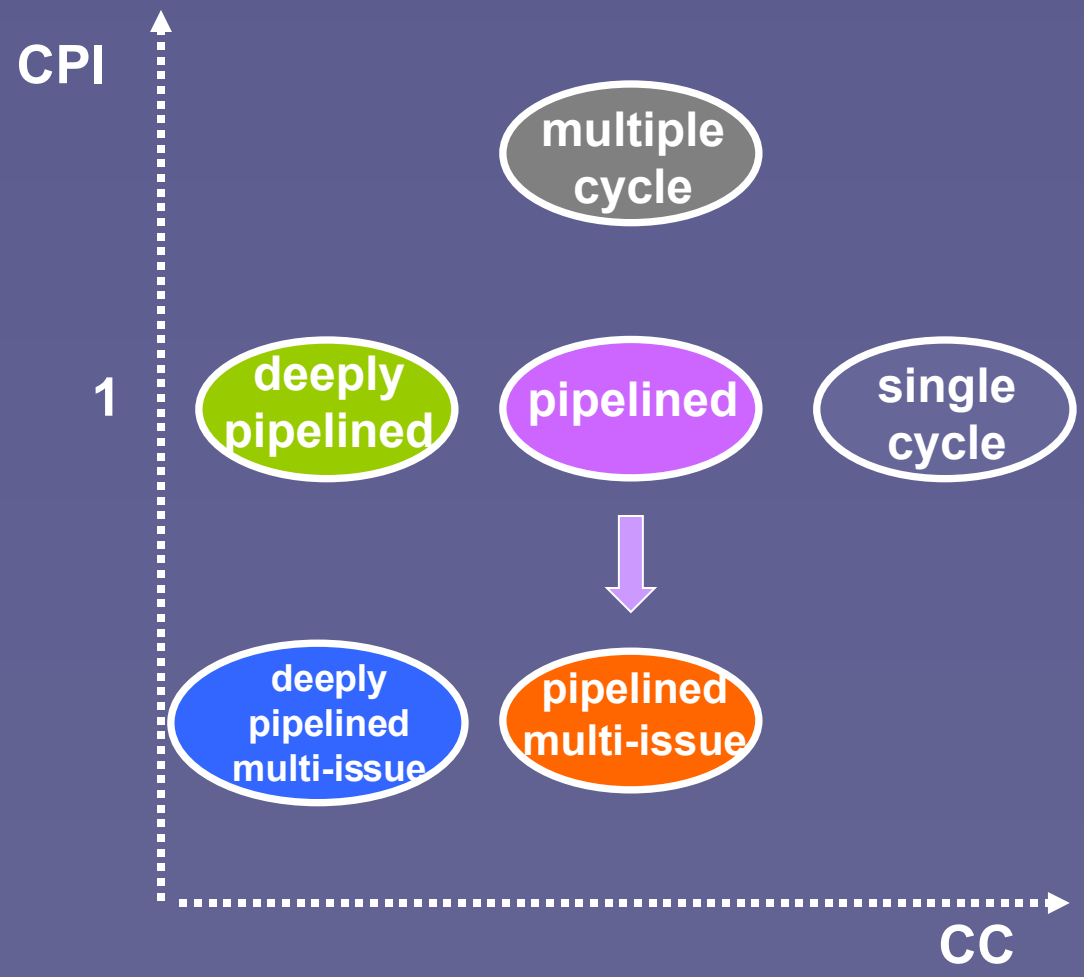
Ακόμα μεγαλύτερος βαθμός ILP

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

- Ο στόχος: CPI < 1
  - Η αλλιώς: ολοκλήρωση πολλών εντολών ανά κύκλο ρολογιού
    - Πολλαπλές βαθμίδες εκτέλεσης
    - Παράλληλο υλικό (hardware)
    - Σε κάθε κύκλο ξεκινά η εκτέλεση **πολλαπλών** νέων εντολών
  - Πολλαπλή εκτέλεση (multiple-issue)
  - Στατική πολλαπλή εκτέλεση
    - Ο μεταγλωττιστής αποφασίζει κατά κύριο λόγο ποιες εντολές θα εκτελεστούν ταυτόχρονα
  - Δυναμική πολλαπλή εκτέλεση
    - Οι αποφάσεις εκτέλεσης λαμβάνονται από την ΚΜΕ

Απόδοση multiple-issue επεξεργαστών

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP



Εικαζόμενη εκτέλεση

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

- **Speculative Execution**
  - Κλειδί για την αύξηση του βαθμού παραλληλίας σε επίπεδο εντολών
    - Πολλαπλές βαθμίδες εκτέλεσης
  - Πρόβλεψη αποτελεσμάτων για την επίτρεψη εκτέλεσης επόμενων εντολών με αλληλεξαρτήσεις
    - Από τον μεταγλωττιστή ή τον επεξεργαστή
    - Διόρθωση λάθους πρόβλεψης
    - Προσωρινή αποθήκευση έως ότου τα δεδομένα να μην είναι πλέον εικαζόμενα
  - Επηρεάζει απόδοση επεξεργαστή

Είδη multiple-issue επεξεργαστών

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

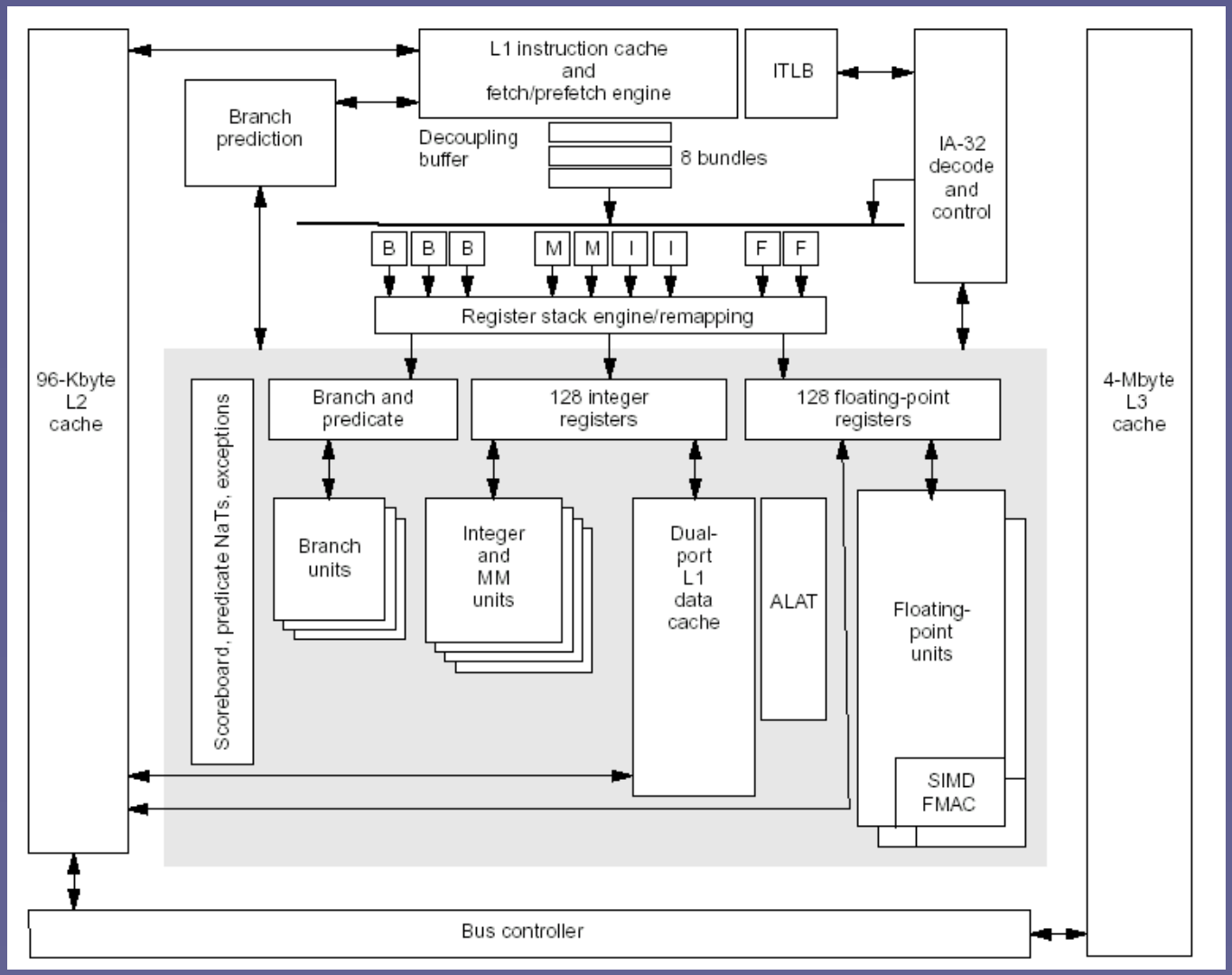
- **Επεξεργαστές VLIW**
  - Very Long Instruction Word
  - Πακέτα πολλαπλών εντολών
    - Παράλληλη εκτέλεση
    - Δεν είναι δυνατοί όλοι οι συνδυασμοί εντολών στο ίδιο πακέτο
  - Ο μεταγλωττιστής συγκροτεί τα πακέτα εντολών
  - Ελέγχει σε (μεγάλο βαθμό)
    - Αλληλεξαρτήσεις – αλληλουχίες εντολών
    - Πρόβλεψη διακλαδώσεων
  - Intel IA-64 (Itanium), Trimedia CPUs…

Intel Itanium processor

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

i

Η Intel ονομάζει την αρχιτεκτονική αυτή **EPIC** (explicitly parallel instruction computer)



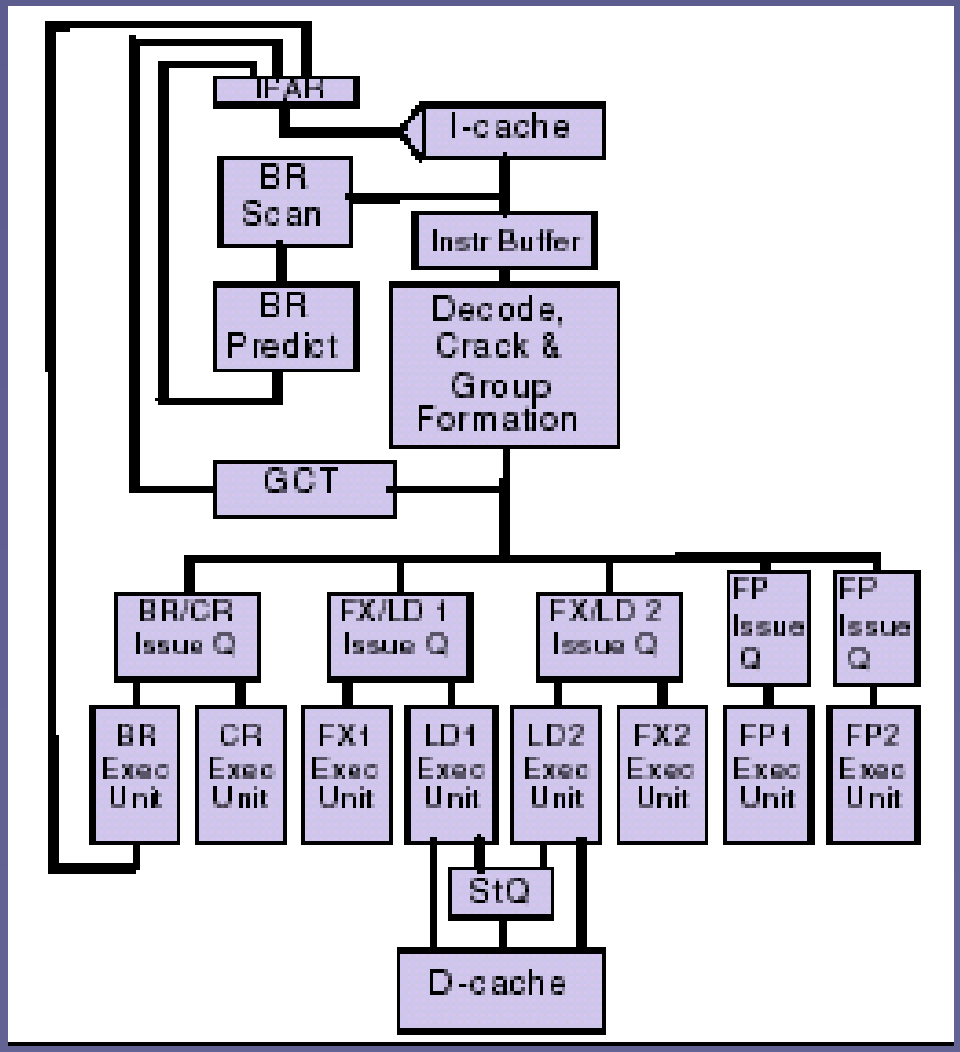
Είδη multiple-issue επεξεργαστών

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP

- **Επεξεργαστές superscalar**
  - Ο επεξεργαστής ελέγχει σε κάθε κύκλο αν μπορεί να ξεκινήσει την εκτέλεση μιας η περισσότερων εντολών
    - Πολλαπλές μονάδες εκτέλεσης
    - Δυναμική απόφαση ανάθεσης εντολών σε μονάδες
    - Εκτέλεση εκτός σειράς (out-of-order execution)
  - Ο κώδικας εκτελείται πάντα σωστά
    - Ανεξάρτητα από μεταγλώττιση
  - Intel IA-32 (Pentium), PowerPC…

IBM Power4 processor

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων
- ILP
- Pipelining
- Προχωρημένες τεχνικές ILP



Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Τεχνολογίες Κύριας Μνήμης

(και η ανάγκη για χρήση ιεραρχιών μνήμης)

http://www.ionio.gr/~mistral/tp/comparch/

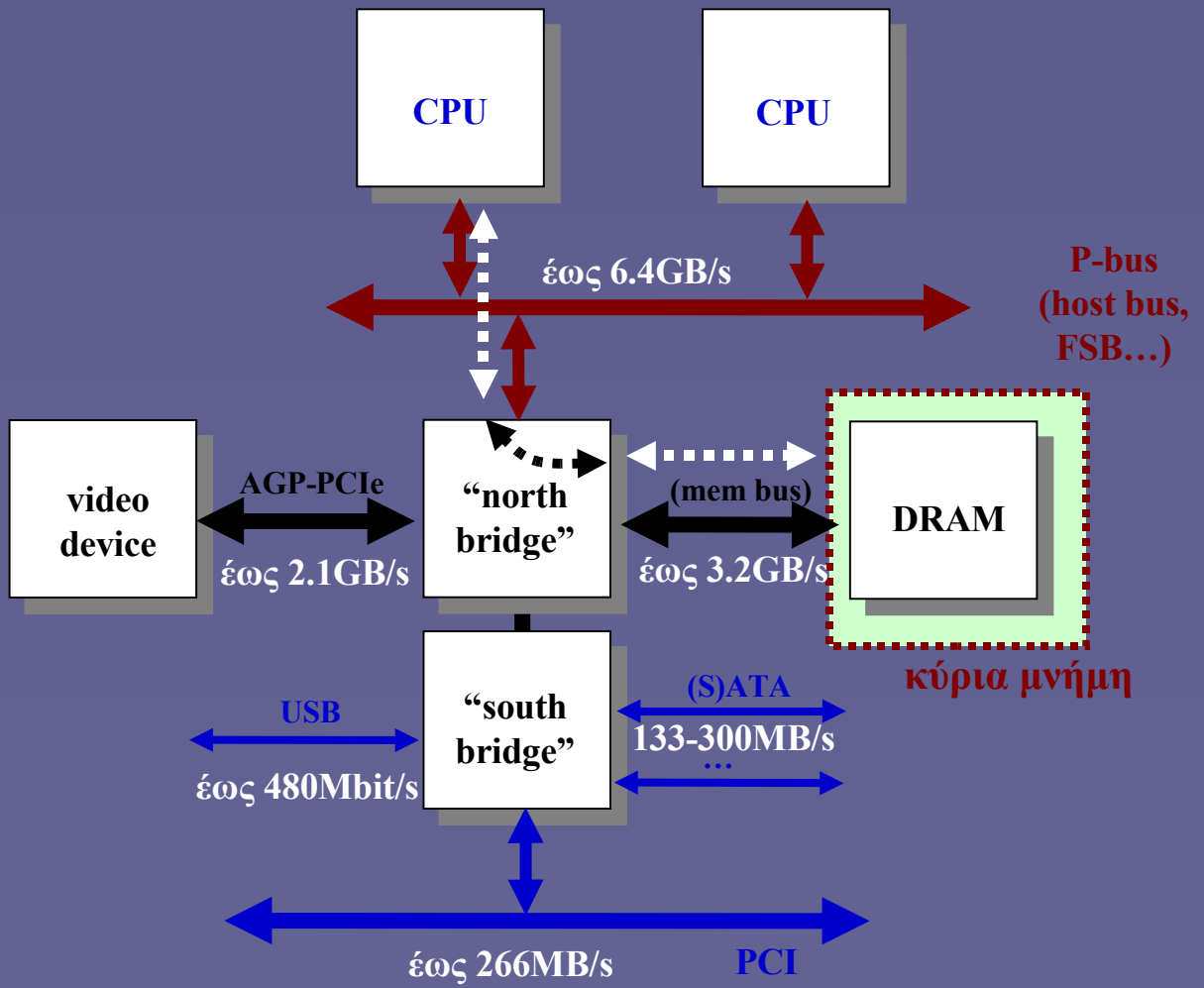
Μ.Στεφανιδάκης

Κύρια Μνήμη

- Κύρια Μνήμη

i

Στα σύγχρονα συστήματα η κύρια μνήμη δεν συνδέεται απευθείας με τον επεξεργαστή



Τεχνολογίες Κύριας Μνήμης

- Κύρια Μνήμη

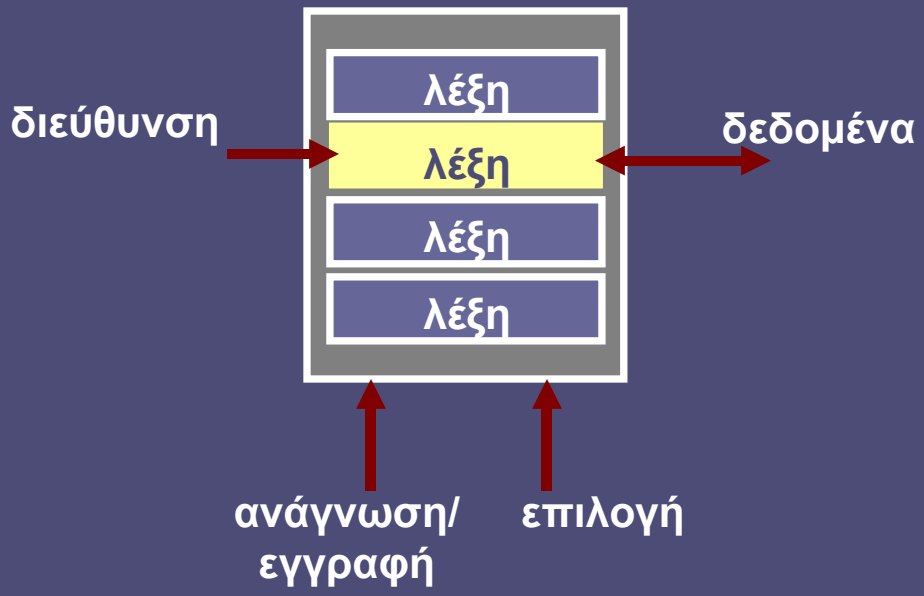
- Στους πρώτους υπολογιστές
  - Ιστορικά, η κατασκευή κύριας μνήμης ήταν πολύ πιο δύσκολη από την κατασκευή των πρώτων υπολογιστών!
- Αρχικές τεχνολογίες
  - Flip-flop με λυχνίες κενού
  - Γραμμές καθυστέρησης υδραργύρου
  - Μαγνητικές μνήμες (core memories - 1950)
    - Η πρώτη αξιόπιστη και σχετικά φθηνή τεχνολογία
    - Κυριάρχησε για 20 περίπου χρόνια
- Ημιαγωγικές μνήμες (Intel – 1970)
  - 1Kbit DRAM - “core killer”

Το μοντέλο της Μνήμης Τυχαίας Προσπέλασης

- Κύρια Μνήμη
- RAM

•Η λέξη είναι η μικρότερη προσπελάσιμη ομάδα bits.

•Το εύρος των μεταφερόμενων δεδομένων σε κάθε ανάγνωση ή εγγραφή ισούται με το εύρος της λέξης μνήμης



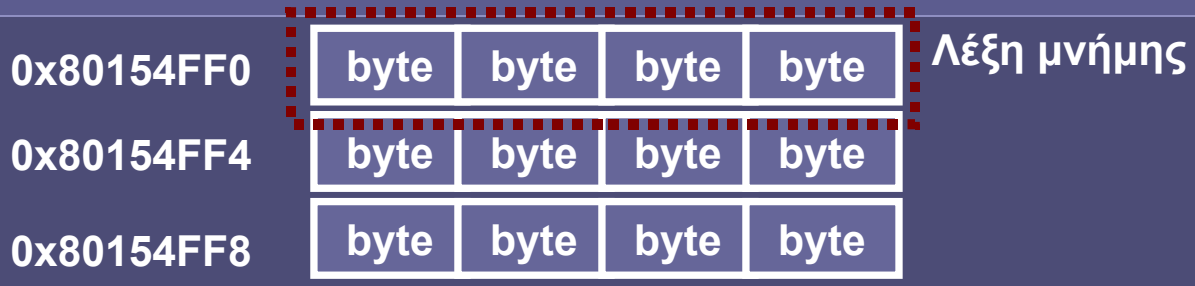
- Random Access Memory (RAM)
  - Λέξη μνήμης (**word**) με εύρος **M** bits
  - Διεύθυνση (**address**) επιλογής λέξης, **N** bits
  - Μέγεθος (χωρητικότητα) μνήμης 2^Nx**M** bits

Διευθυνσιοδότηση μνήμης RAM

- Κύρια Μνήμη
- RAM

;

Πώς προσπελαύνω bytes όταν η λέξη μνήμης έχει πολλαπλάσιο εύρος;

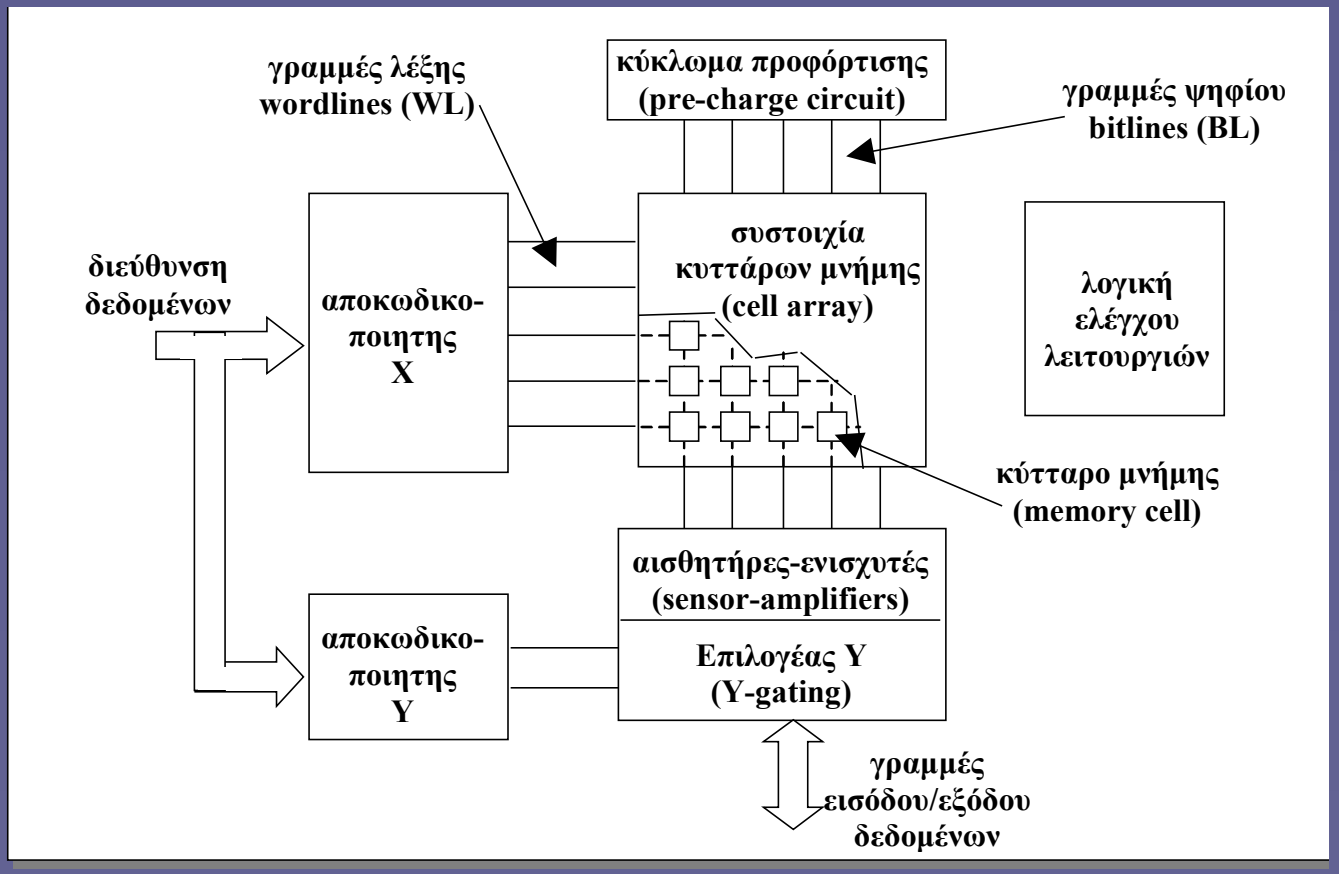


- Byte addressing
  - Οι διευθύνσεις μνήμης που παράγει ο επεξεργαστής αυξάνονται ανά byte
    - **Ακόμα κι όταν η λέξη μνήμης έχει πολλαπλάσιο εύρος!**
    - **Μετάφραση από ελεγκτή μνήμης**
  - Ευθυγραμμισμένη προσπέλαση:
    - **Αντικείμενο των s bytes στη διεύθυνση A ⇒ *A mod s = 0***
- Εναλλακτικά: word addressing
  - Υπερυπολογιστές – εδώ η προσπέλαση ανά byte είναι σπάνια

Οργάνωση Μνήμης Τυχαίας Προσπέλασης

- Κύρια Μνήμη
- RAM

Τυπική εσωτερική οργάνωση μνήμης



Οι μεγαλύτερες μνήμες RAM διαθέτουν πολλαπλές συστοιχίες κυττάρων μνήμης

Ταχύτητα Προσπέλασης RAM

- Κύρια Μνήμη
- RAM

- Access Time (χρόνος προσπέλασης)
  - Ο απαιτούμενος χρόνος για την ολοκλήρωση μιας αίτησης προς τη μνήμη RAM
    - Ανάγνωση - Εγγραφή
- Cycle Time (χρόνος κύκλου προσπέλασης)
  - Ο ελάχιστος απαιτούμενος χρόνος μεταξύ διαδοχικών αιτήσεων προς τη μνήμη RAM
    - Πρόβλεψη ενδιάμεσων λειτουργιών
- Ταχύτητα διαφορετική για ανάγνωση - εγγραφή!

Τύποι Μνήμης Τυχαίας Προσπέλασης

- Κύρια Μνήμη
- RAM
- SRAM

Ο χρόνος προσπέλασης μιας μνήμης SRAM βρίσκεται μεταξύ 0,5 και 5 ns

Το κόστος ενός GB SRAM είναι $4000-$10.000 [Patterson, Hennessy]

- **Στατική Μνήμη RAM (SRAM)**
  - Κάθε bit αποθηκεύεται σε κύτταρο 6 τρανζίστορ
    - **Ανάλογο ενός flip-flop**
  - Διατήρηση όσο υπάρχει τροφοδοσία της μνήμης
  - **Η προσπέλαση είναι γρήγορη**
  - Μεγαλύτερο κόστος
  - Πολυπλοκότερο κύκλωμα
    - **Δεν επιτρέπει μεγάλη ολοκλήρωση**
    - **Μικρής χωρητικότητας**
  - Μεγαλύτερη κατανάλωση ενέργειας
    - **Αλλά: ελάχιστη κατανάλωση σε standby mode**
  - **Χρησιμοποιείται στις κρυφές μνήμες (caches)**

Τύποι Μνήμης Τυχαίας Προσπέλασης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM

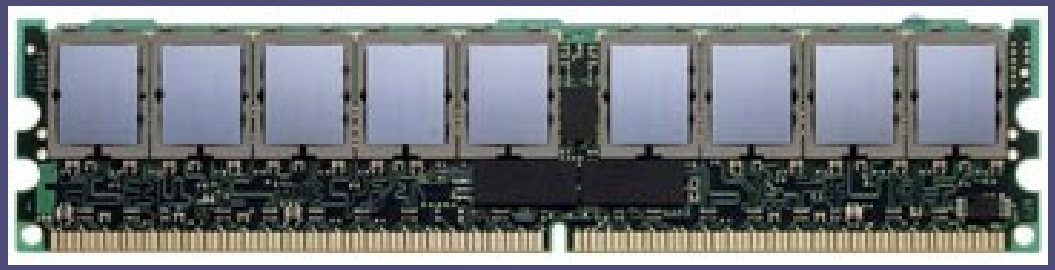
Ο χρόνος προσπέλασης μιας μνήμης DRAM βρίσκεται μεταξύ 50 και 70 ns

Το κόστος ενός GB DRAM είναι $100-$200 [Patterson, Hennessy]

- Δυναμική Μνήμη RAM (DRAM)
  - Κάθε bit αποθηκεύεται ως φορτίο
    - **Ανάλογο ενός πυκνωτή**
  - Διατήρηση μόνο με συχνή **ανανέωση** του φορτίου
    - Κάθε 16 έως 128 ms (5% συνολικού χρόνου)
  - Απλούστερο κύκλωμα – μεγάλη ολοκλήρωση
    - **Πολύ μεγάλες χωρητικότητες (1Gbit/ολοκληρωμένο κυκλ.)**
  - Η προσπέλαση είναι αργή
    - **Αρχιτεκτονικές βελτιώσεις για αύξηση ρυθμού μεταφοράς δεδομένων**
  - Χρησιμοποιείται για τη συγκρότηση της κύριας μνήμης όλων των σύγχρονων υπολογιστικών συστημάτων

Τμήματα μνήμης DRAM

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM



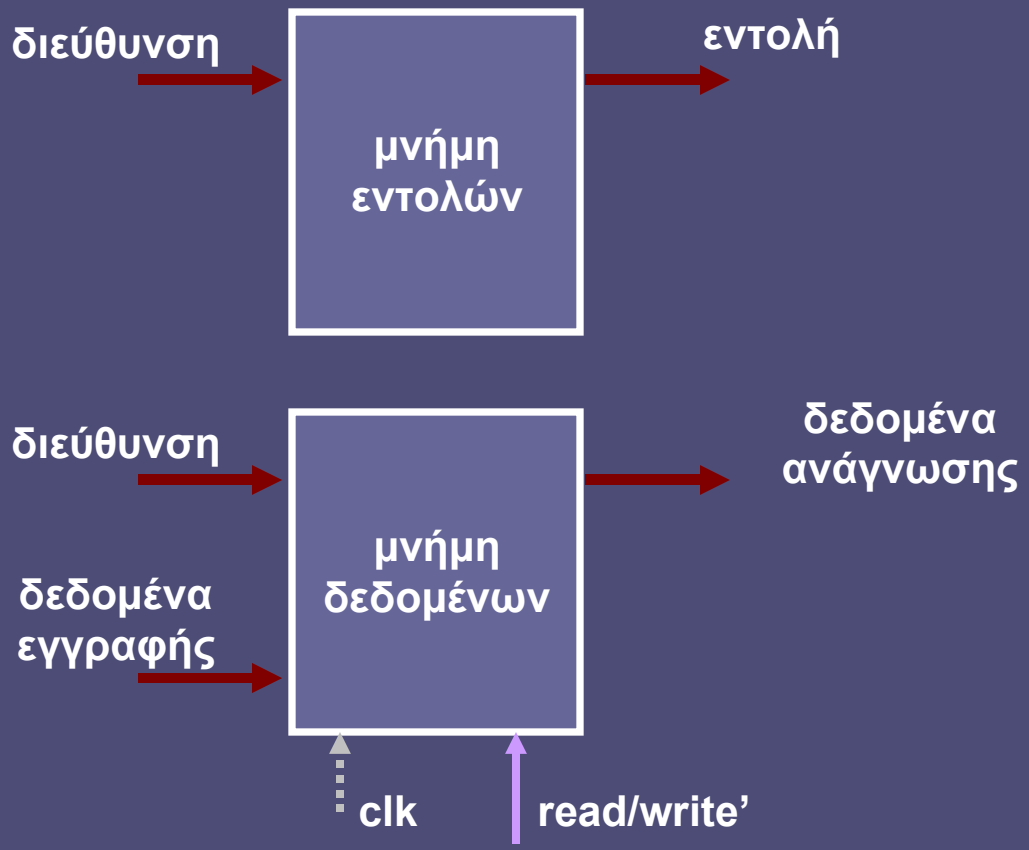
- 64 έως 72 bits δεδομένων (έως 4GB)
- Σύγχρονες μεταφορές στις 2 ακμές ρολογιού
- Σήματα ανίχνευσης και αναγνώρισης
- Διατάξεις βελτίωσης ηλεκτρικών χαρακτηριστικών σημάτων
- Ρυθμός μεταφοράς > 3.2GB/s

Η "ιδανική μνήμη"

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

;

Πόσο απέχει η ιδανική εικόνα από την πραγματικότητα;



- **Ολοκλήρωση ανάγνωσης-εγγραφής σε έναν κύκλο ρολογιού...**

Η πραγματική εικόνα

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

;

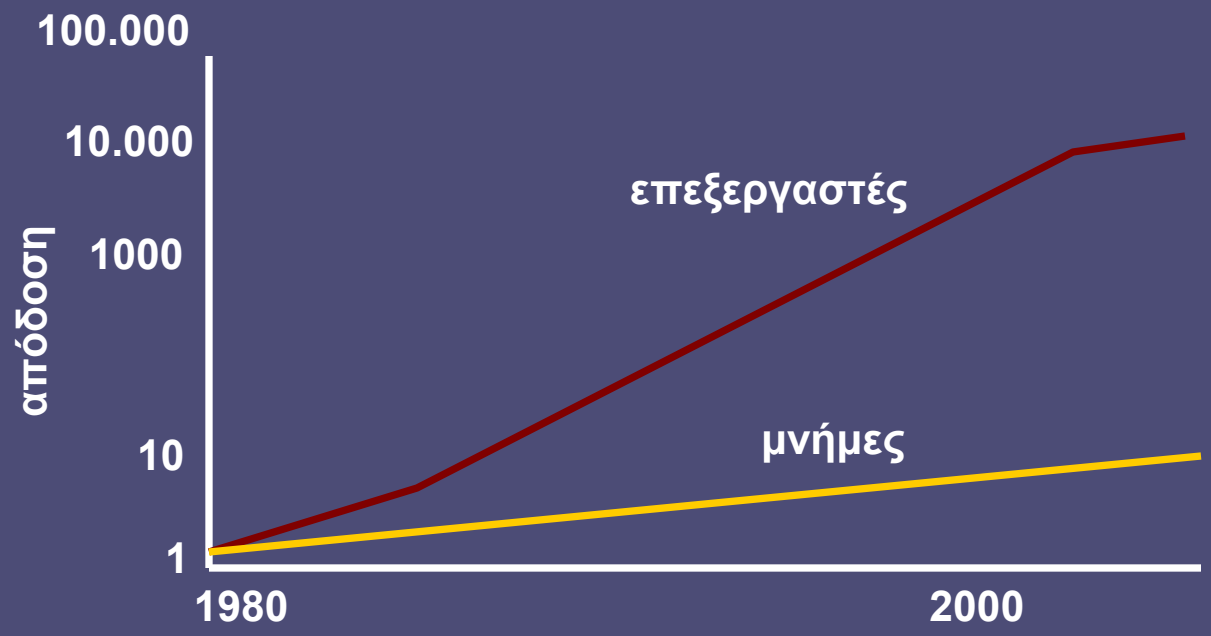
Η ιδανική μνήμη είναι πρακτικά αδύνατο να υλοποιηθεί. Ποια η πιθανή λύση;

- Ένας σύγχρονος superscalar επεξεργαστής
  - με ρολόι 3 GHz
  - και έναρξη εκτέλεσης έως και 8 εντολών ανά κύκλο
  - απαιτεί από τη μνήμη 24G εντολές/sec!
- Η "ιδανική μνήμη" θα έπρεπε να είναι
  - Πολύ γρήγορη
  - Πολύ φθηνή
  - Με πολύ μεγάλη χωρητικότητα

Το χάσμα απόδοσης μεταξύ επεξεργαστή-μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

Οι μνήμες ακολουθούν τον νόμο του Moore στην αύξηση της χωρητικότητάς τους, όχι όμως και στην απόδοση



- Επεξεργαστές: αύξηση απόδοσης 35%-55% /έτος
- Μνήμες: αύξηση απόδοσης 7% /έτος

[Patterson-Hennessy]

Η αρχή της τοπικότητας

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

"ένα πρόγραμμα εκτελεί το 90% των εντολών του μέσα στο 10% του κώδικά του"

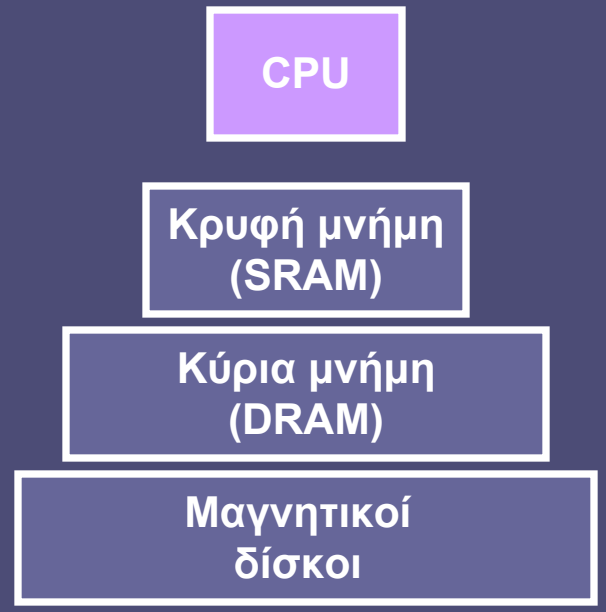
- **Χρονική Τοπικότητα**
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστεί ξανά στο άμεσο μέλλον
  - Π.χ. για εντολές ενός βρόχου (loop)
- **Χωρική Τοπικότητα**
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστούν και οι γειτονικές θέσεις στο άμεσο μέλλον
  - Εντολές προγραμμάτων
  - Δεδομένα σε πίνακες κλπ

Ιεραρχίες Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

Και οι δικτυακές τοποθεσίες μπορούν να θεωρηθούν μέρος της ιεραρχίας μνήμης (το χαμηλότερο)

- **Πολλαπλά επίπεδα μνήμης**
  - Διαφορετικής τεχνολογίας
  - Με διαφορετική ταχύτητα και μέγεθος
  - Γρηγορότερη μνήμη κοντά στον επεξεργαστή

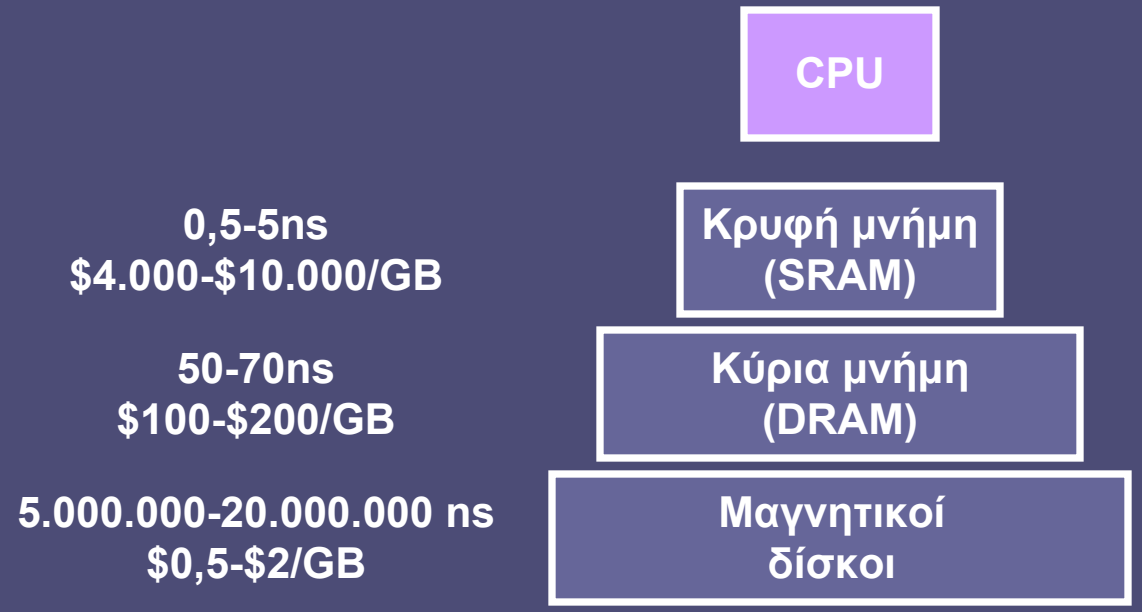


Σκοπός της Ιεραρχίας Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

Για να επιτύχει τον σκοπό της η ιεραρχία μνήμης εκμεταλλεύεται την αρχή της **τοπικότητας**

- **Προσέγγιση της ιδανικής μνήμης**
  - Ο επεξεργαστής να βλέπει "μνήμη"
  - Με την ταχύτητα του υψηλότερου επιπέδου
  - Και το μέγεθος του χαμηλότερου



Αποθήκευση δεδομένων στην Ιεραρχία Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

- **Αποθήκευση δεδομένων**
  - Τα υψηλότερα επίπεδα είναι υποσύνολα των χαμηλότερων
  - Όλα τα δεδομένα αποθηκεύονται τελικά στο χαμηλότερο επίπεδο
- **Μεταφορά δεδομένων**
  - Αντιγραφή από επίπεδο σε επίπεδο
  - Το ελάχιστο σύνολο δεδομένων που μεταφέρεται μεταξύ δύο επιπέδων ονομάζεται **μπλοκ**
    - Πολλαπλά bytes

Αναζήτηση δεδομένων στην Ιεραρχία Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

- **Αναζήτηση δεδομένων**
  - Ο επεξεργαστής ζητά **πάντοτε** τα δεδομένα από το κοντινότερο σε αυτόν επίπεδο
  - Τα δεδομένα υπάρχουν στο επίπεδο αυτό: **hit**
  - Τα δεδομένα δεν βρίσκονται στο επίπεδο αυτό: **miss**
    - Η αίτηση προωθείται στο επόμενο (χαμηλότερο) επίπεδο
    - Και το μπλοκ που περιέχει τα δεδομένα αντιγράφεται στο ανώτερο επίπεδο

Μετρήσεις απόδοσης στην Ιεραρχία Μνήμης

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης

- **Hit Rate**
  - Ποσοστό προσπελάσεων μνήμης, όπου τα δεδομένα βρίσκονται στο ανώτερο επίπεδο
- **Miss Rate**
  - Ποσοστό προσπελάσεων μνήμης, όπου τα δεδομένα δεν βρίσκονται στο ανώτερο επίπεδο
    - (1-hit rate)
- **Hit Time**
  - Ο χρόνος για την προσπέλαση δεδομένων σε hit
- **Miss Penalty**
  - Ο χρόνος για την προσπέλαση, μεταφορά και τοποθέτηση των δεδομένων miss από το χαμηλότερο στο ανώτερο επίπεδο

Εισαγωγή στις κρυφές μνήμες (caches)

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης
- Κρυφές Μνήμες

- **Κρυφή μνήμη**
  - Μεταξύ του επεξεργαστή και της κύριας μνήμης
  - Εμφάνιση στη δεκαετία του 60
  - Σήμερα δεν υπάρχει υπολογιστικό σύστημα χωρίς κρυφή μνήμη
- **Αποθήκευση δεδομένων στην κρυφή μνήμη**
  - Όχι ανά λέξη μνήμης ή ανά byte…
  - …αλλά ανά **μπλοκ** (64-512bits)
  - Μεταφορά δεδομένων από την κύρια προς την κρυφή μνήμη σε **ριπές (bursts)**
  - Το σύστημα κύριας μνήμης έχει βελτιστοποιηθεί αρχιτεκτονικά για αυτού του τύπου τις μεταφορές

Θέματα κρυφών μνημών

- Κύρια Μνήμη
- RAM
- SRAM
- DRAM
- Ιεραρχίες Μνήμης
- Κρυφές Μνήμες

- Πού αποθηκεύεται ένα μπλοκ στην κρυφή μνήμη;
- Πώς εντοπίζεται ένα μπλοκ στην κρυφή μνήμη;
- Ποιο μπλοκ θα αντικατασταθεί όταν χρειαστεί;
- Τι συμβαίνει στην εγγραφή νέων δεδομένων;
- Πώς υπολογίζεται η απόδοση της ιεραρχίας μνήμης;

(στο επόμενο μάθημα..)

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Κρυφές Μνήμες

(οργάνωση, λειτουργία και απόδοση)

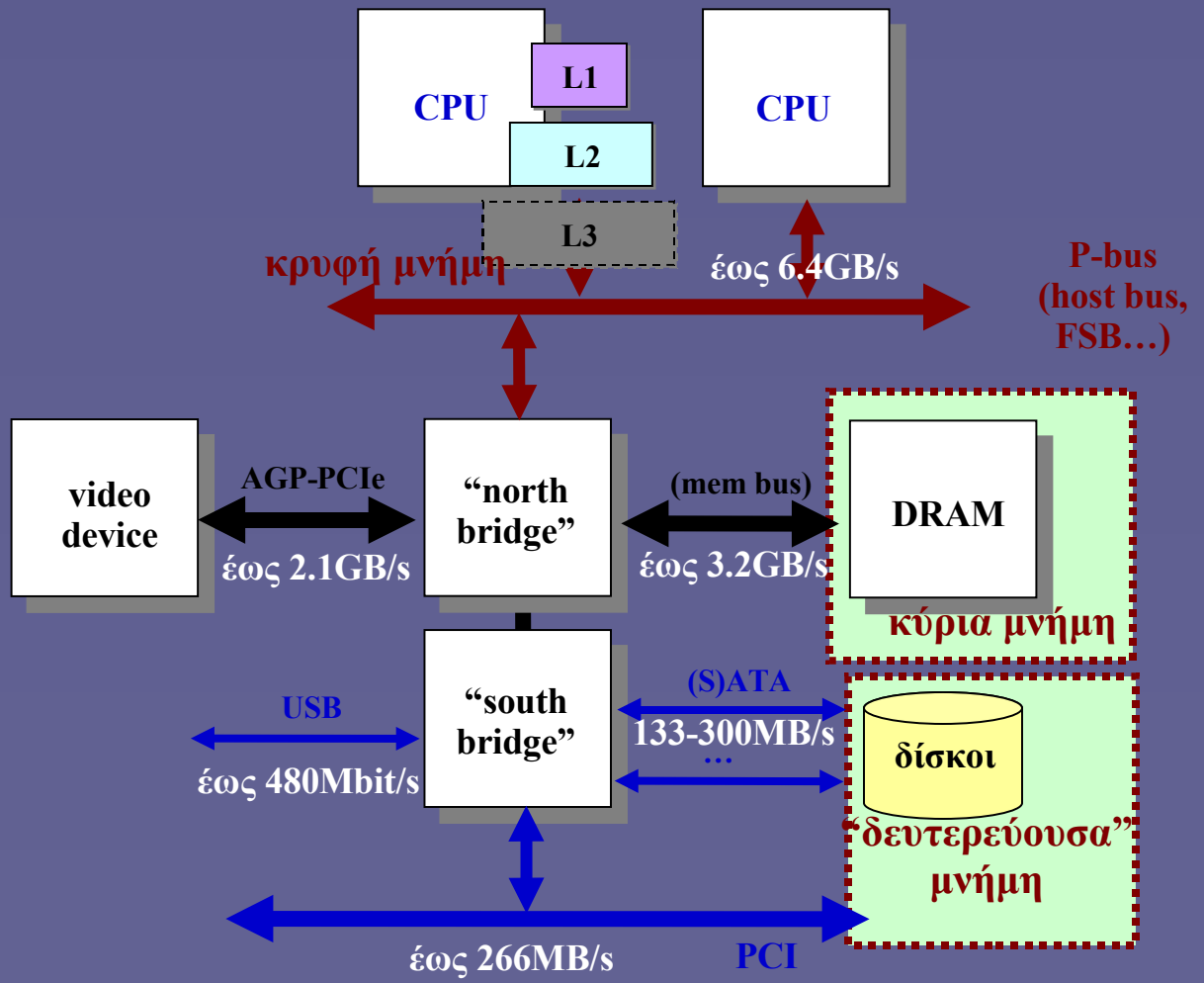
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης

Για βελτίωση της απόδοσης του συστήματος μνήμης (και του συνολικού υπολογιστικού συστήματος)

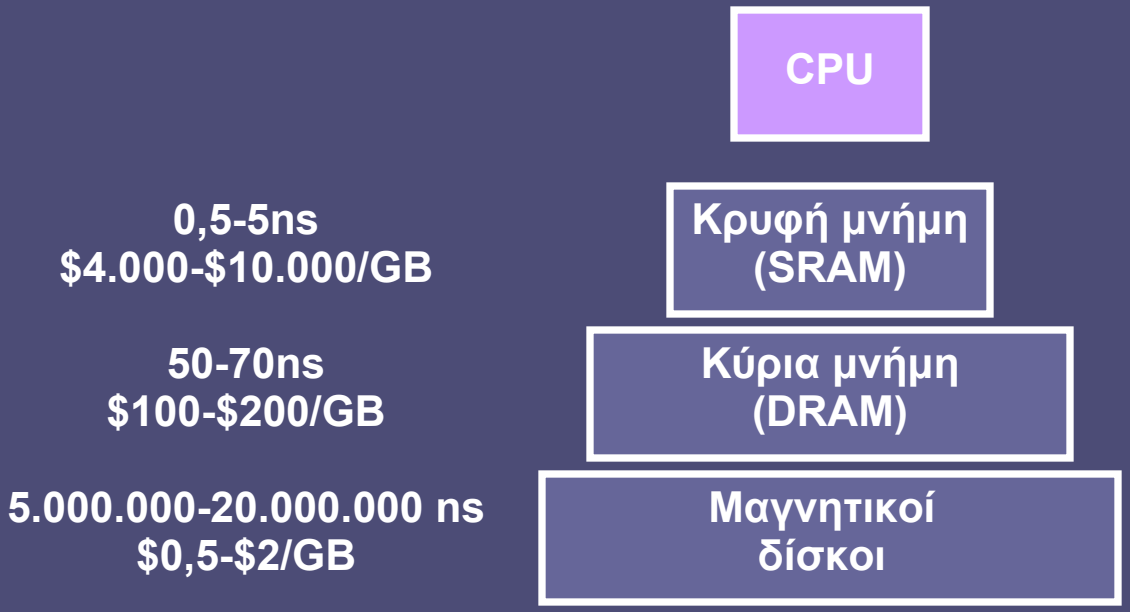


Σκοπός της Ιεραρχίας Μνήμης

- Ιεραρχία Μνήμης

Για να επιτύχει τον σκοπό της η ιεραρχία μνήμης εκμεταλλεύεται την αρχή της **τοπικότητας**

- Προσέγγιση της ιδανικής μνήμης
  - Ο επεξεργαστής να βλέπει "μνήμη"
  - Με την ταχύτητα του υψηλότερου επιπέδου
  - Και το μέγεθος του χαμηλότερου επιπέδου



Ιεραρχία μνήμης και τοπικότητα

- Ιεραρχία Μνήμης

"ένα πρόγραμμα εκτελεί το 90% των εντολών του μέσα στο 10% του κώδικά του"

- Χρονική Τοπικότητα
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστεί ξανά στο άμεσο μέλλον
    - Παράδειγμα: οι εντολές ενός βρόχου (loop)
- Εφαρμογή:
  - Δεδομένα – εντολές που βρίσκονται ήδη κοντύτερα στον επεξεργαστή (π.χ. στην κρυφή μνήμη) θα προσπελαστούν πολύ γρηγορότερα

Ιεραρχία μνήμης και τοπικότητα

- Ιεραρχία Μνήμης

"ένα πρόγραμμα εκτελεί το 90% των εντολών του μέσα στο 10% του κώδικά του"

- Χωρική Τοπικότητα
  - Εάν προσπελαστεί μια θέση μνήμης, είναι πολύ πιθανό να προσπελαστούν και οι γειτονικές θέσεις στο άμεσο μέλλον
    - Εντολές προγραμμάτων
    - Δεδομένα σε πίνακες κλπ
- Εφαρμογή:
  - Εάν προσπελαστεί μια θέση μνήμης, μεταφέρονται και οι διπλανές της λέξεις στη μνήμη του υψηλότερου επιπέδου
    - Μεταφορά σε μπλοκ (πολλαπλές λέξεις μνήμης)
  - Γρηγορότερη προσπέλαση

Κρυφές μνήμες

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

- Σημαντικό τμήμα στην ιεραρχία μνήμης
- Εξέλιξη συστημάτων κρυφής μνήμης
  - 1962: οι πρώτες ιεραρχίες μνήμης (Atlas)
    - Όχι όμως κρυφή μνήμη
  - 1965: η πρώτη περιγραφή κρυφής μνήμης (Wilkes)
    - Ο πρώτος υπολογιστής με κρυφή μνήμη (IBM 360/85)
  - 1968: η πρώτη χρησιμοποίηση του όρου "cache memory"
  - Μελέτες απόδοσης κρυφής μνήμης
  - Πολλαπλά επίπεδα κρυφής μνήμης (L1, L2, L3)
  - Βελτιωμένες αρχιτεκτονικές κρυφής μνήμης

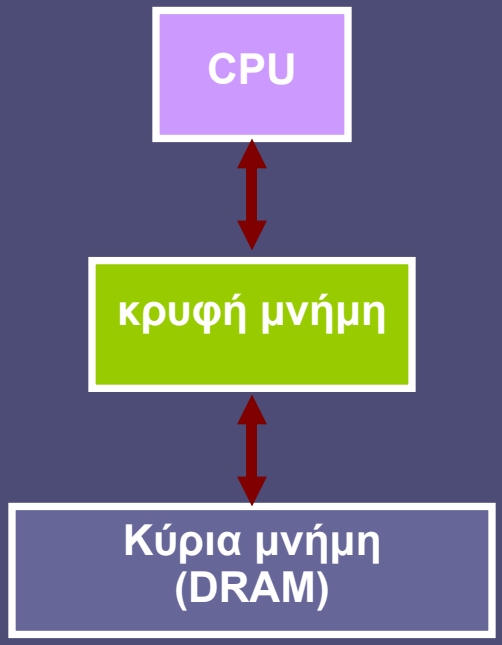
Απλό μοντέλο ιεραρχίας μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Οι αρχές λειτουργίας της απλής ιεραρχίας μπορούν να επεκταθούν σε πολλαπλά επίπεδα

Η διαχείριση της κρυφής μνήμης γίνεται από το υλικό

διαφανώς προς τις εφαρμογές



- Τα δεδομένα βρίσκονται αρχικά στην κύρια μνήμη
- Η κρυφή μνήμη περιέχει υποσύνολο των δεδομένων
- Μεταφορά μεταξύ επιπέδων μνήμης σε μπλοκ λέξεων

Αποθήκευση δεδομένων στην Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

- **Αποθήκευση δεδομένων**
  - Τα υψηλότερα επίπεδα είναι υποσύνολα των χαμηλότερων
  - Όλα τα δεδομένα αποθηκεύονται τελικά στο χαμηλότερο επίπεδο
- **Μεταφορά δεδομένων**
  - Αντιγραφή από επίπεδο σε επίπεδο
  - Το ελάχιστο σύνολο δεδομένων που μεταφέρεται μεταξύ δύο επιπέδων ονομάζεται **μπλοκ**
    - Πολλαπλά bytes

Αναζήτηση δεδομένων στην Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

- **Αναζήτηση δεδομένων**
  - Ο επεξεργαστής ζητά **πάντοτε** τα δεδομένα από το κοντινότερο σε αυτόν επίπεδο
  - Τα δεδομένα υπάρχουν στο επίπεδο αυτό: **hit**
  - Τα δεδομένα δεν βρίσκονται στο επίπεδο αυτό: **miss**
    - Η αίτηση προωθείται στο επόμενο (χαμηλότερο) επίπεδο
    - Όταν βρεθεί, το μπλοκ που περιέχει τα δεδομένα αντιγράφεται στο ανώτερο επίπεδο

Μπλοκ (γραμμές) κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

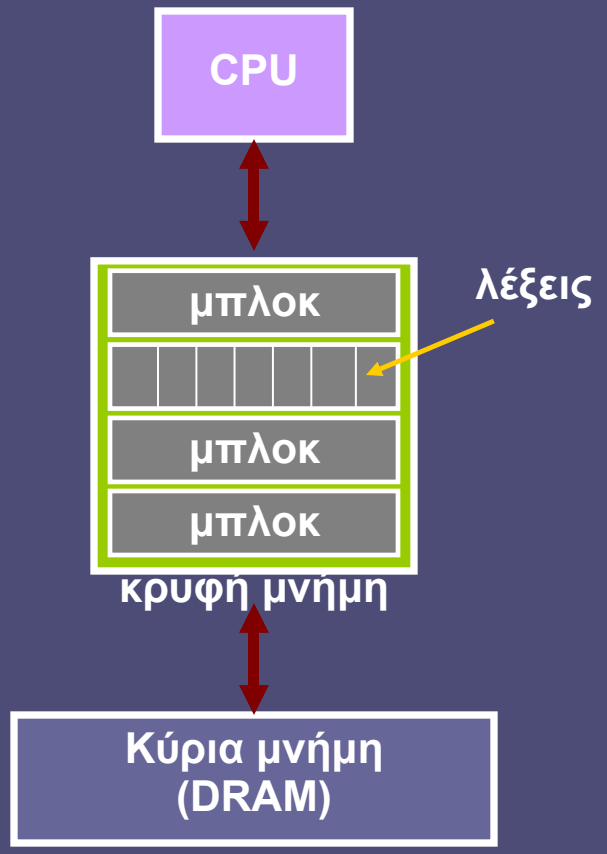
Οι σύγχρονοι επεξεργαστές διαθέτουν κρυφές μνήμες (L1) με μέγεθος μπλοκ έως 64 bytes

•Για την εκμετάλλευση της τοπικότητας

•Όταν πρέπει να μεταφερθεί μια λέξη, μεταφέρεται το μπλοκ που την περιέχει

•Το μέγεθος του μπλοκ είναι καθοριστικό για την απόδοση της ιεραρχίας μνήμης

•Το σύστημα κύριας μνήμης έχει βελτιστοποιηθεί αρχιτεκτονικά για μεταφορές μπλοκ



Τοποθέτηση ενός μπλοκ

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

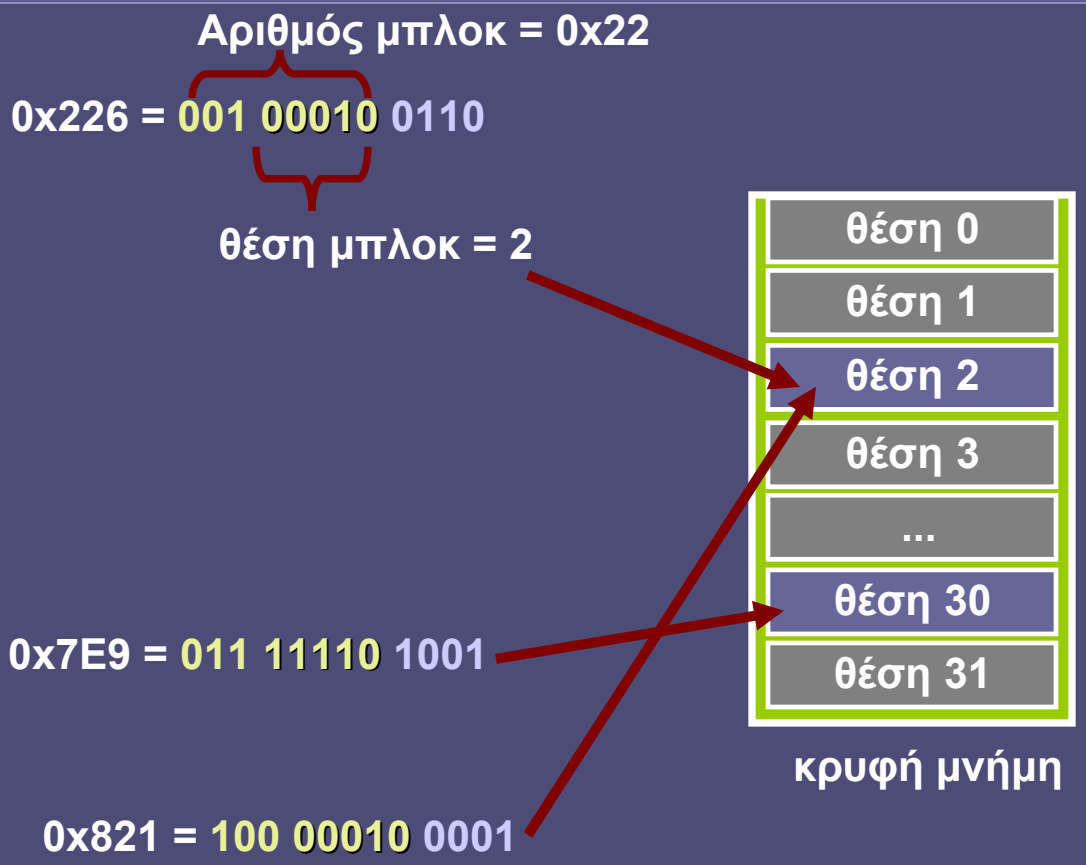
- Η κύρια μνήμη περιέχει πολύ περισσότερα μπλοκ από όσα χωρούν στην κρυφή μνήμη
  - Συνεπώς, στην ίδια θέση της κρυφής μνήμης πρέπει να τοποθετηθούν περισσότερα από ένα μπλοκ
    - Σύγκρουση μπλοκ;
- Πώς αποφασίζεται η θέση ενός μπλοκ στην κρυφή μνήμη;
  - Η απλή λύση: άμεση απεικόνιση (direct mapped caches)
  - Κάθε μπλοκ πηγαίνει σε μία μόνο θέση

  (αριθμός μπλοκ) mod (θέσεις στην κρυφή μνήμη)
    - Υπολογίζεται πολύ εύκολα αν οι θέσεις είναι δύναμη του 2

Άμεση απεικόνιση θέσης μπλοκ

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Παράδειγμα:
Block = 16 bytes
Cache = 32 blocks

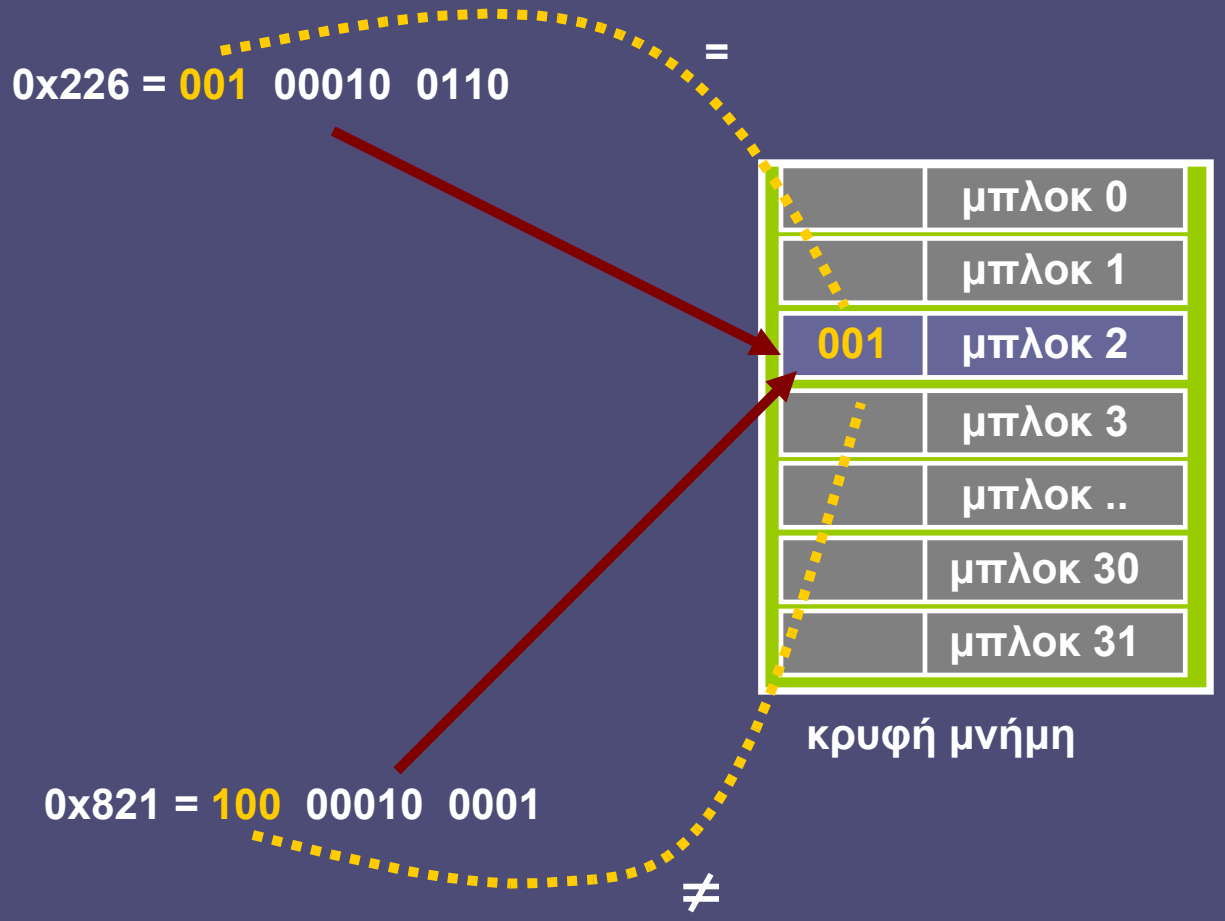


Ποιο μπλοκ βρίσκεται σε κάθε θέση;

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Είναι η θέση κατειλημμένη από κάποιο μπλοκ;

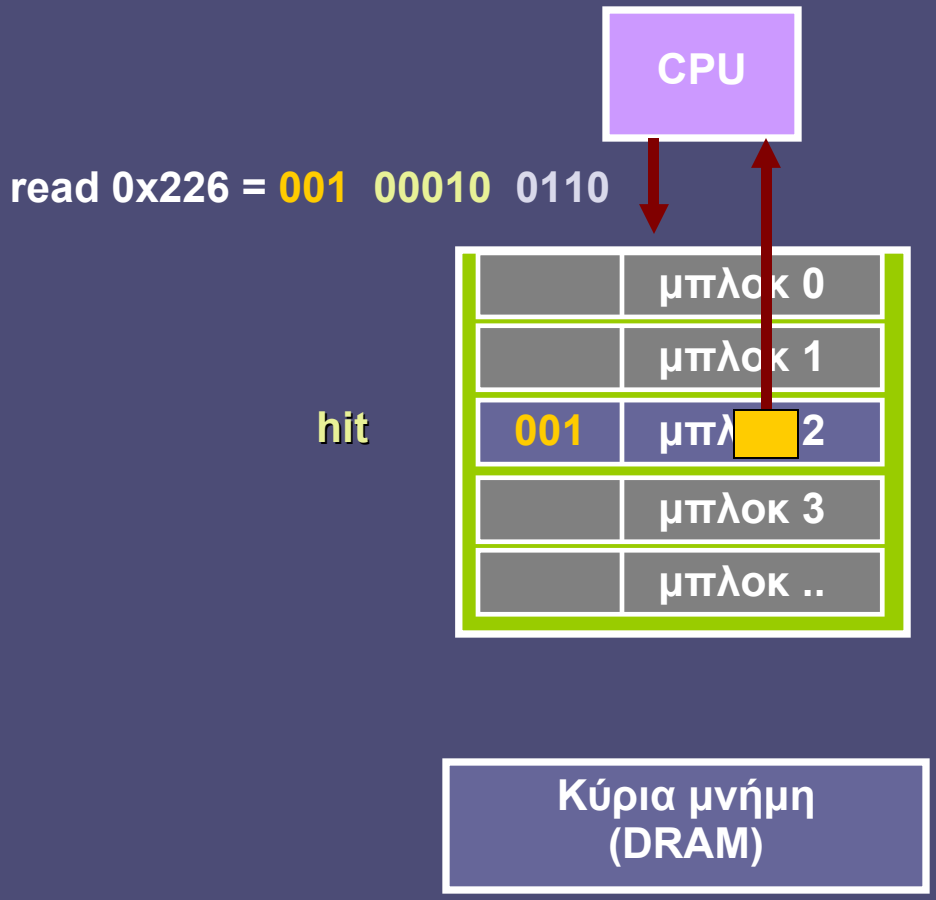
valid bit (V)



Ανάγνωση: Cache Hit

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Σε περίπτωση εύρεσης των δεδομένων στην κρυφή μνήμη, η ΚΜΕ μπορεί να τα λάβει ακόμα και σε 1 κύκλο

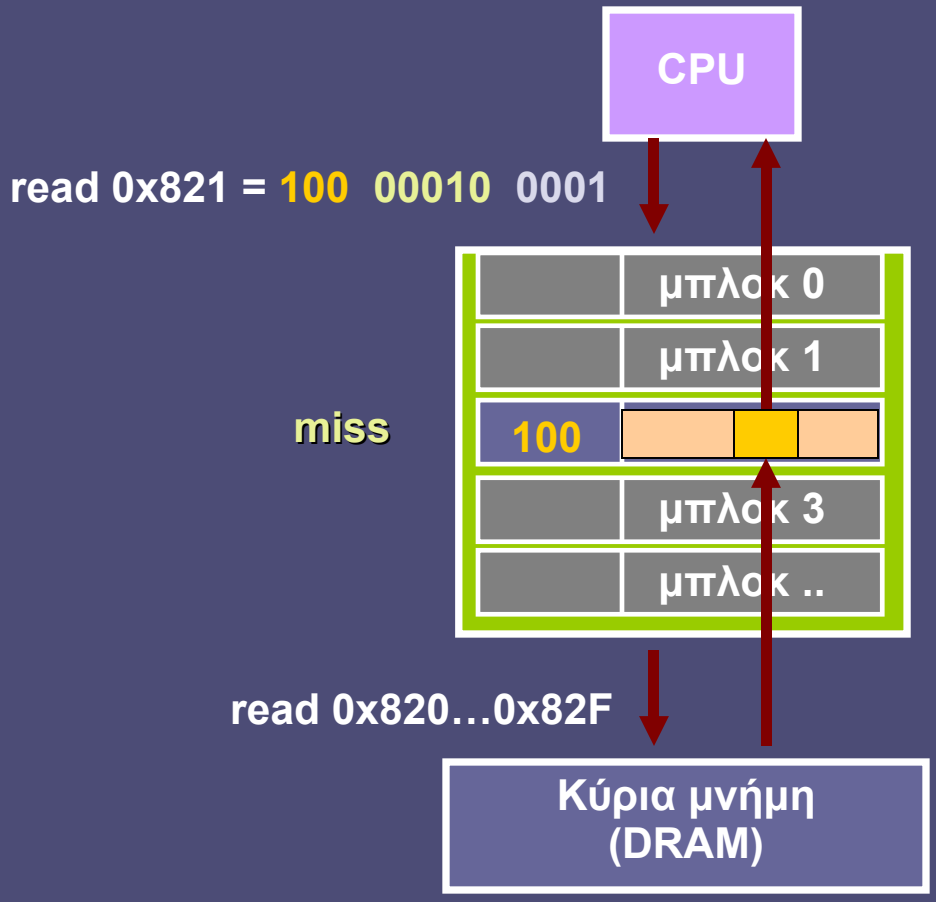


•Αιτήσεις για ανάγνωση: εντολές και δεδομένα

Ανάγνωση: Cache Miss

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Miss penalty:
ο χρόνος για την μεταφορά του μπλοκ από κύρια μνήμη και επιστροφή δεδομένων στον επεξεργαστή



•Αιτήσεις για ανάγνωση: εντολές και δεδομένα

Εγγραφή στην κρυφή μνήμη

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Συνοχή δεδομένων:
Πώς επηρεάζουν οι κρυφές μνήμες τη σχεδίαση πολυ-επεξεργαστικών συστημάτων;

- Μόνο για δεδομένα
- Write Hit – Ενημέρωση κρυφής μνήμης
  - Στη συνέχεια:
  - Είτε ενημερώνω αμέσως την κύρια μνήμη (write-through)
    - Επιβάρυνση σε κάθε εγγραφή (χρήση buffers)
  - Είτε μόνο όταν το μπλοκ πρέπει να φύγει από την κρυφή μνήμη (write-back)
    - Απαιτείται πολύπλοκη λογική για τον έλεγχο της σωστής αποθήκευσης των δεδομένων
- Write Miss
  - Πρέπει το μπλοκ να έρθει πρώτα στην κρυφή μνήμη

Τι δημιουργεί cache misses;

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Η πρώτη φορά προσπέλασης ενός μπλοκ
  - Μπλοκ που δεν βρέθηκαν ποτέ μέχρι τώρα στην κρυφή μνήμη
- Λόγω χωρητικότητας της κρυφής μνήμης
  - Η κρυφή μνήμη δεν χωράει όλα τα μπλοκ (ταυτόχρονα)
- Λόγω σύγκρουσης μπλοκ στην ίδια θέση
  - Μπλοκ που τοποθετούνται στην ίδια θέση στην κρυφή μνήμη, συναγωνίζονται για τη θέση αυτή
    - ανάλογα με τη μέθοδο τοποθέτησης
    - ακόμα κι αν μέρος της κρυφής μνήμης είναι ελεύθερο...

Χαρακτηριστικά απόδοσης κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- **Hit Rate**
  - Ποσοστό προσπελάσεων μνήμης, όπου τα δεδομένα βρίσκονται στην κρυφή μνήμη
- **Miss Rate**
  - Ποσοστό προσπελάσεων μνήμης, όπου τα δεδομένα δεν βρίσκονται στην κρυφή μνήμη
    - (1-hit rate)
- **Hit Time**
  - Ο χρόνος για την προσπέλαση δεδομένων σε hit
- **Miss Penalty**
  - Ο χρόνος για την προσπέλαση, μεταφορά και τοποθέτηση των δεδομένων miss από την κύρια στην κρυφή μνήμη και στον επεξεργαστή

Το κόστος των cache misses

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Χαμένοι κύκλοι ρολογιού
  - Σε αναμονή για προσπέλαση κύριας μνήμης

  Κύκλοι Αναμονής =

  Προσπελάσεις μνήμης * Miss Rate * Miss Penalty
- Απλουστευμένο μοντέλο
  - Miss Rate ανά κατηγορίες εντολών
  - Διαφορετική συμπεριφορά για ανάγνωση-εγγραφή
  - Σύνθετη ανάλυση για εκτέλεση εκτός σειράς
- Βελτίωση της απόδοσης
  - Μείωση του miss rate
  - Μείωση του miss penalty

Τεχνικές μείωσης miss rate

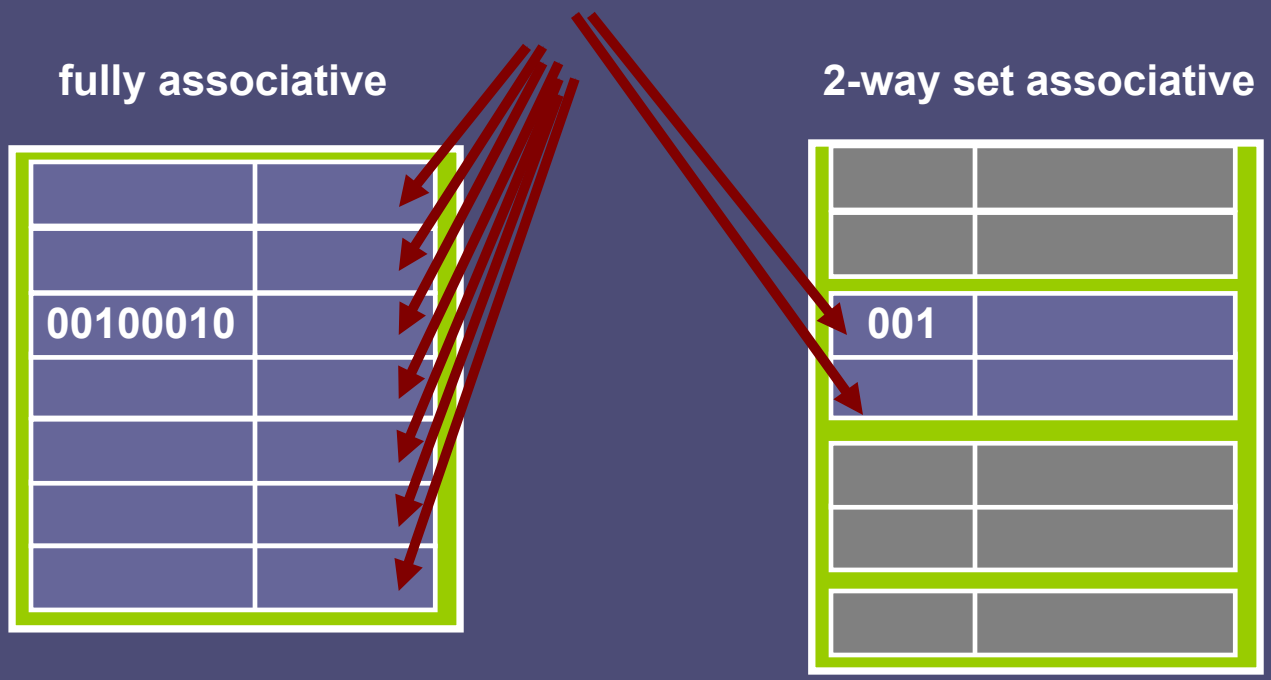
- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Αντιμετώπιση αιτιών που προκαλούν misses
- Αύξηση χωρητικότητας κρυφής μνήμης
  - Αλλά: μια μεγάλη κρυφή μνήμη μπορεί να είναι πιο αργή! (αύξηση hit time)
- Αύξηση του μεγέθους του μπλοκ
  - Προσπάθεια εκμετάλλευσης της τοπικότητας
  - Αλλά: αυξάνει το miss penalty
  - Πιθανόν να αυξάνει τελικά το miss rate, λόγω λιγότερων μπλοκ στην κρυφή μνήμη
- Ευέλικτες τεχνικές τοποθέτησης των μπλοκ
  - Ώστε να παραμένουν περισσότερο στην κρυφή μνήμη

Ευέλικτες τεχνικές τοποθέτησης μπλοκ

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη

Πιθανή η αύξηση του hit time λόγω πολύπλοκου κυκλώματος



•Αντικατάσταση: least recently used (LRU)

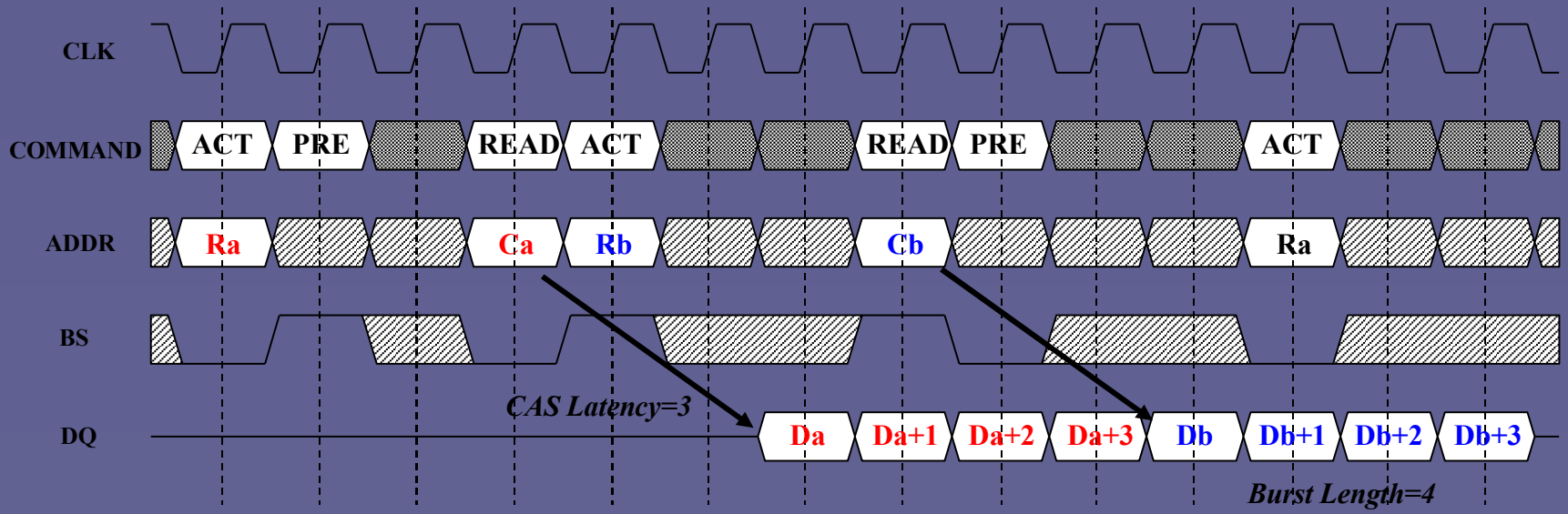
Τεχνικές μείωσης miss penalty

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

Οι σύγχρονοι επεξεργαστές έχουν τουλάχιστον L1 και L2 cache μέσα στο ίδιο το chip τους

- Μείωση των χρόνων μεταφοράς μπλοκ
- Βελτιστοποιήσεις στην επικοινωνία με την κύρια μνήμη
  - Έτσι ώστε ένα ολόκληρο μπλοκ να μεταφέρεται με τη μικρότερη δυνατή καθυστέρηση (bursts)
- Πολυεπίπεδες ιεραρχίες κρυφής μνήμης
  - Μείωση miss penalty πρώτου επιπέδου (L1)
  - L1: μικρότερο μέγεθος, μεγαλύτερη ταχύτητα
    - Μεγαλύτερο miss rate αλλά miss penalty μικρότερο
  - L2: μεγαλύτερο μέγεθος, μικρότερη ταχύτητα
    - Αργότερη αλλά δεν επηρεάζει hit time επεξεργαστή

Ανάγνωση από σύγχρονη κύρια μνήμη



Πολυεπίπεδη οργάνωση κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

Οι σύγχρονοι επεξεργαστές έχουν ξεχωριστή κρυφή μνήμη L1 για εντολές και δεδομένα. Ποια τα πλεονεκτήματα-μειονεκτήματα;

Τύπος	Μέγεθος	Χρόνος προσπέλασης	Ρυθμός μεταφοράς
L1	έως 64KB	4ns	50GB/s
L2	έως 8MB	10ns	25GB/s
L3	έως 64MB	20ns	10GB/s

- Παράδειγμα: Pentium4
  - L1 cache: 4 κύκλοι ρολογιού (pipelined: 1)
  - L2 cache: 20 κύκλοι ρολογιού
  - Προσπέλαση στη μνήμη: >100 κύκλοι ρολογιού

Intel “Montecito”: Επίπεδα κρυφής μνήμης

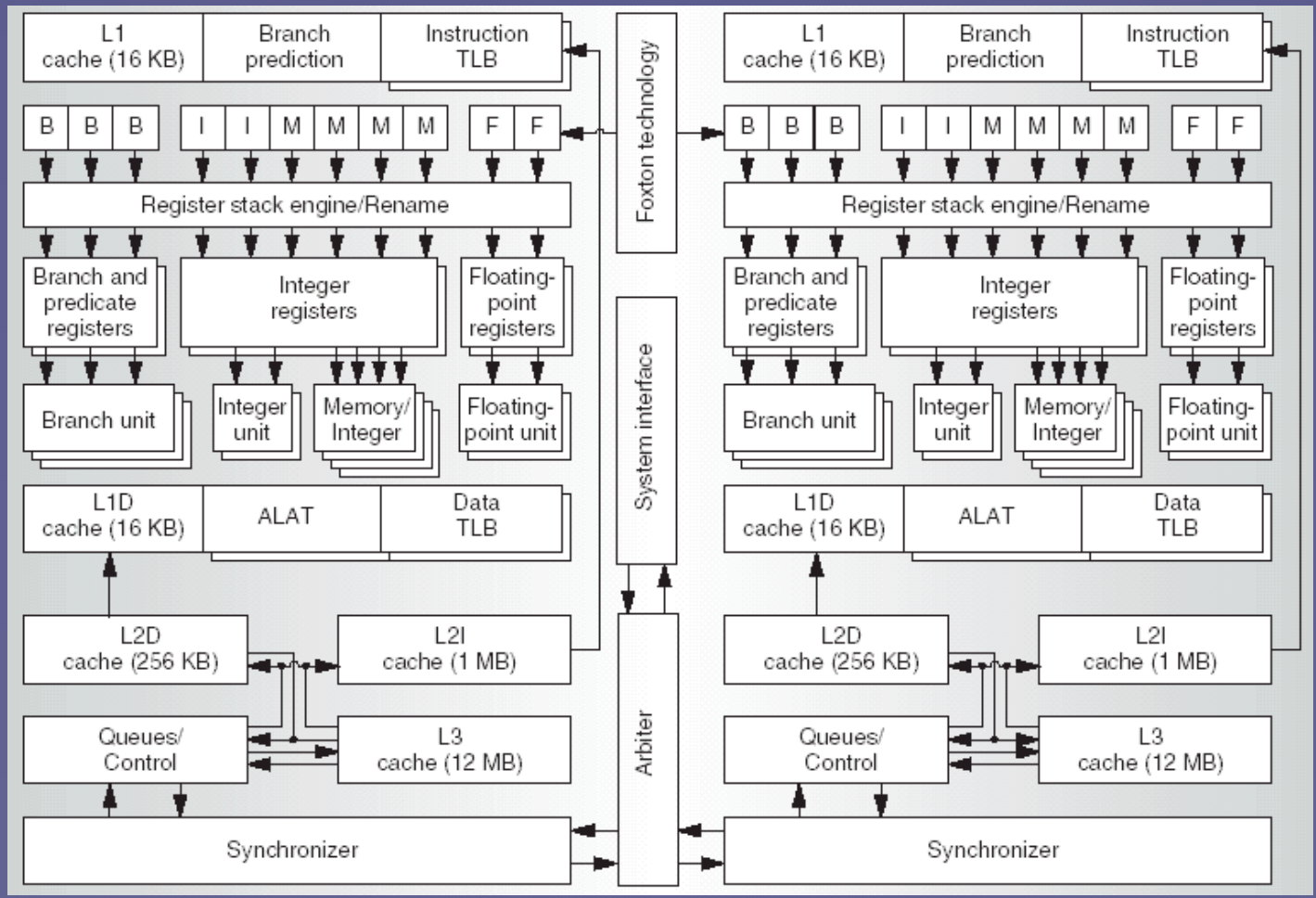
- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

Intel Montecito:

1,72 δις τρανζίστορ

2 επεξεργαστές Itanium2

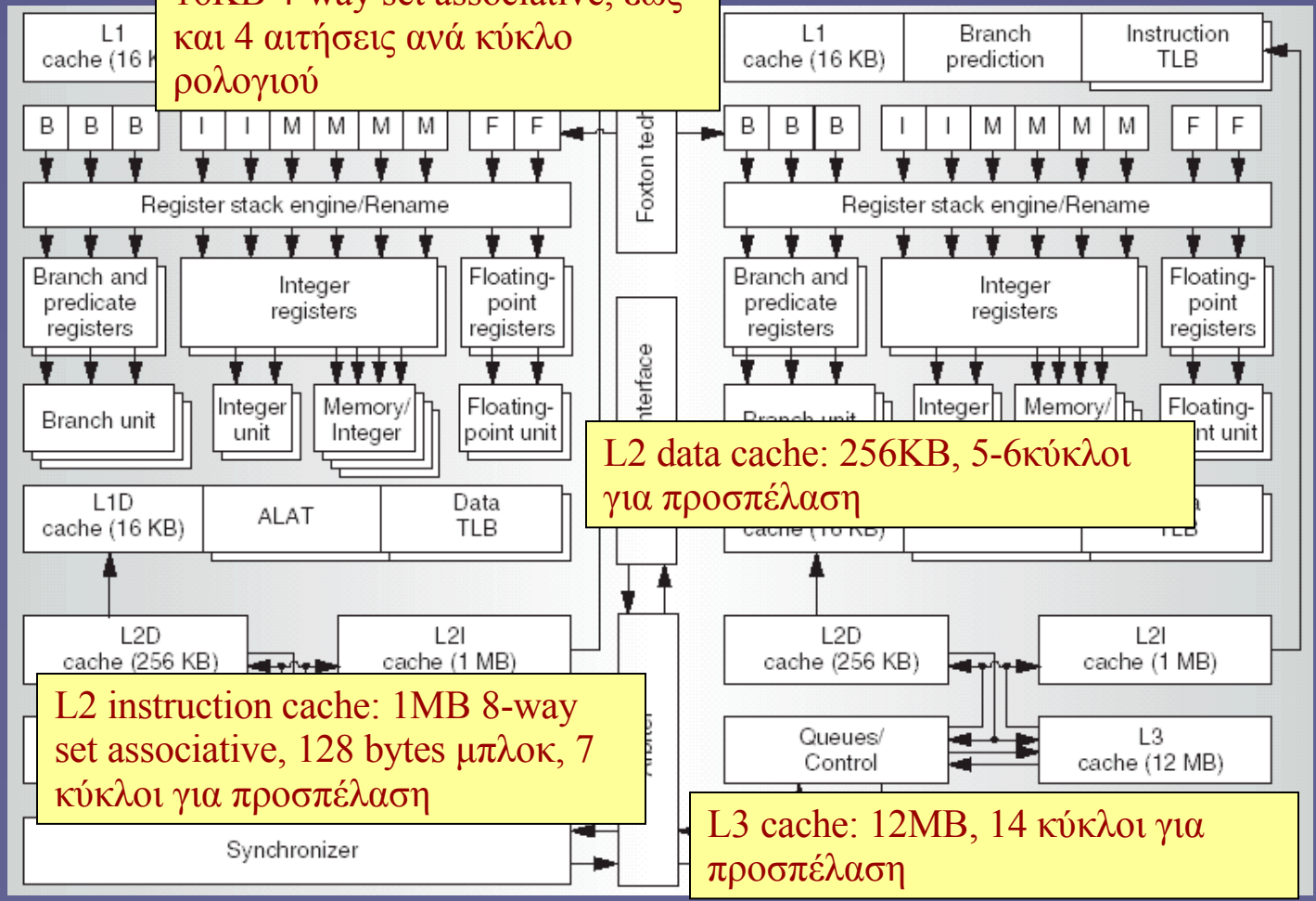
1.8GHz @ 100W



Intel “Montecito”: Επίπεδα κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

Intel Montecito:
συνολικά 27MB κρυφή μνήμη μέσα στο chip



Βελτιστοποίηση απόδοσης κρυφής μνήμης

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Αρχιτεκτονικές βελτιώσεις
  - Pipelining
  - Non-blocking – εξυπηρέτηση πολλαπλών αιτήσεων
  - Πολλαπλά επίπεδα κρυφής μνήμης στο chip του επεξεργαστή
- Ο ρόλος του λογισμικού (μεταγλωττιστές)
  - Αναδιοργάνωση προγραμμάτων για αύξηση της τοπικότητας (κυρίως στους βρόχους επανάληψης)
  - Prefetching: μετακίνηση δεδομένων στην κρυφή μνήμη πριν αυτά χρειαστούν στον επεξεργαστή!

Η απόδοση της κρυφής μνήμης συνοπτικά

- Ιεραρχία Μνήμης
- Κρυφή Μνήμη
- Απόδοση κρυφής μνήμης

- Καθοριστική για τα σύγχρονα υπολογιστικά συστήματα
- Μείωση του miss rate ή του miss penalty
  - Όμως: η συμπεριφορά της ιεραρχίας μνήμης επηρεάζεται από πολλούς παράγοντες!
- Η πραγματική συμπεριφορά
  - Είναι σύνθετη – απαιτούνται εξομοιώσεις πριν τη σχεδίαση νέων συστημάτων
  - Είναι διαφορετική ανά εφαρμογή – δεν υπάρχει ένα μόνο αντιπροσωπευτικό πρόγραμμα!
  - Είναι διαφορετική ανά υπολογιστικό σύστημα – desktop, server ή embedded

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Εικονική Μνήμη

(και ο ρόλος της στην ιεραρχία μνήμης)

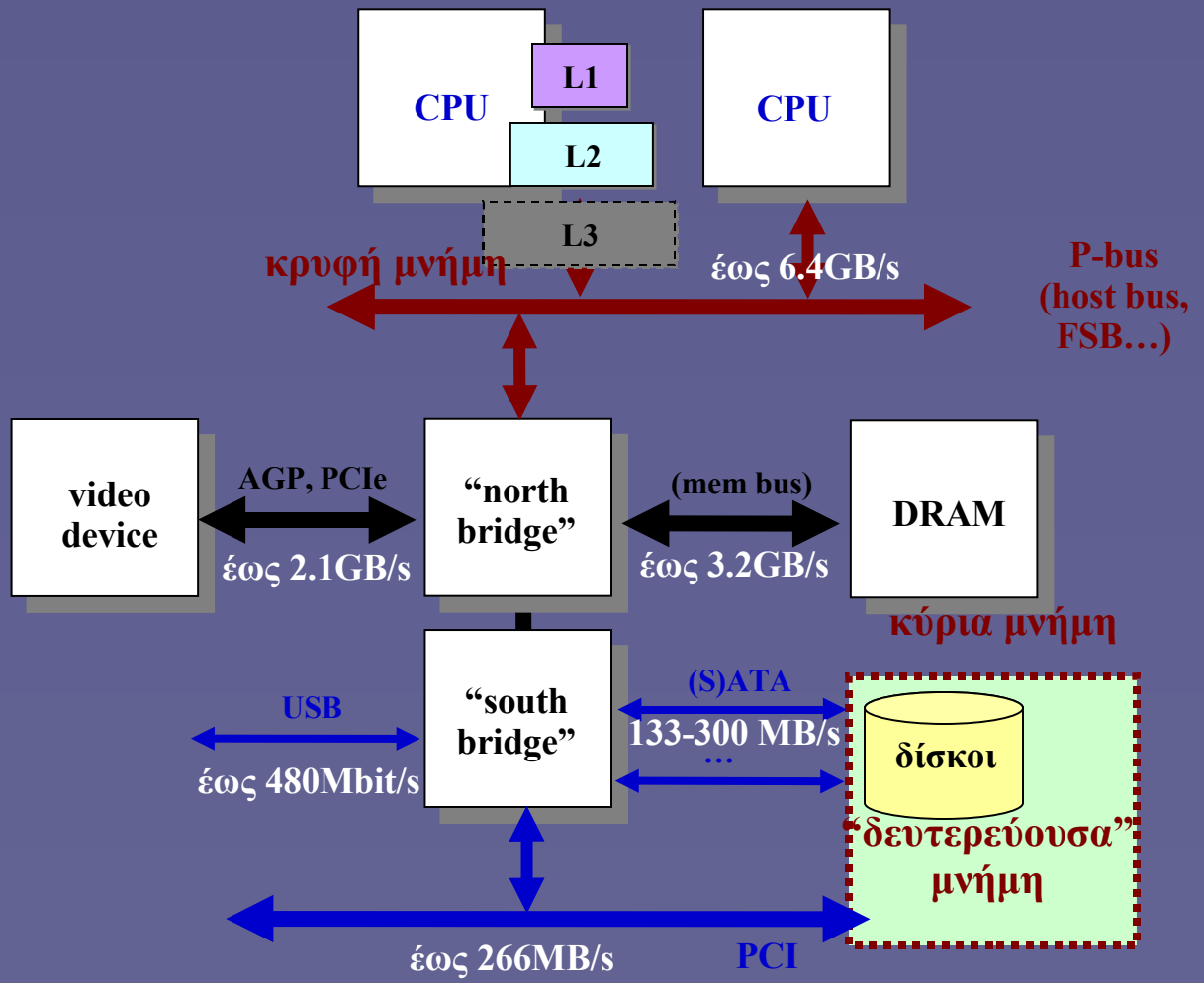
http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης

"Δευτερεύουσα μνήμη": Το τελευταίο επίπεδο στην ιεραρχία εντός του υπολογιστικού συστήματος

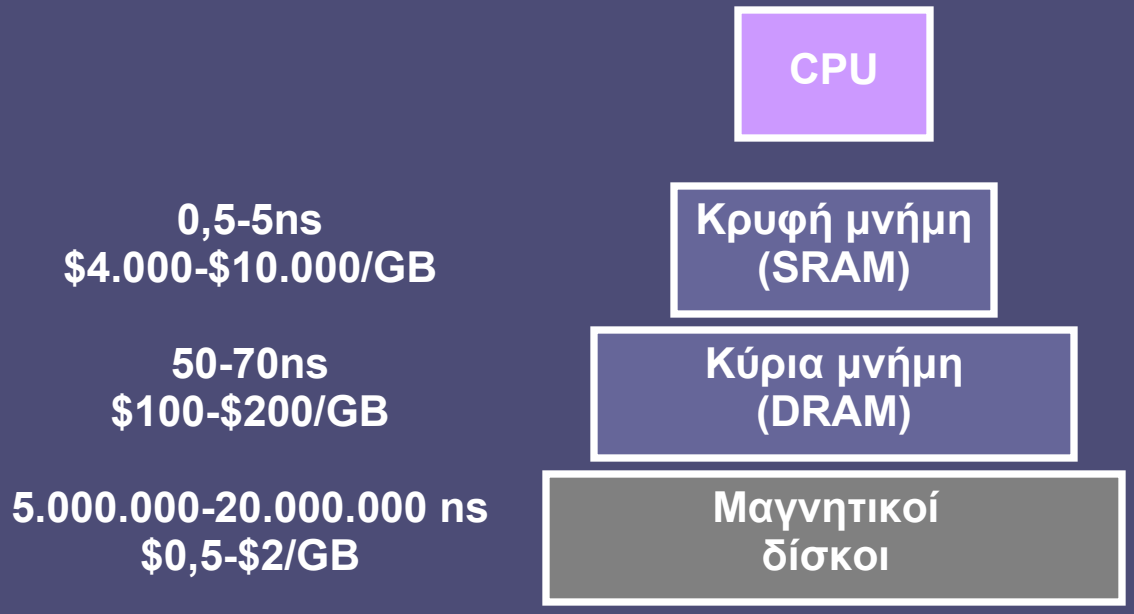


Επεκτείνοντας την Ιεραρχία Μνήμης

- Ιεραρχία Μνήμης

Επέκταση ιεραρχίας εκτός του συστήματος: δικτυακές θέσεις αποθήκευσης

- Εννοιολογικά:
  - Κάθε υψηλότερο επίπεδο δρα ως "κρυφή μνήμη" για το αμέσως χαμηλότερο
    - Η **κύρια μνήμη** λειτουργεί ως "κρυφή μνήμη" των **μαγνητικών δίσκων**
    - Τα περιεχόμενα στην **κύρια μνήμη** είναι υποσύνολο εκείνων στους **δίσκους** του συστήματος



Εικονική μνήμη (virtual memory)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

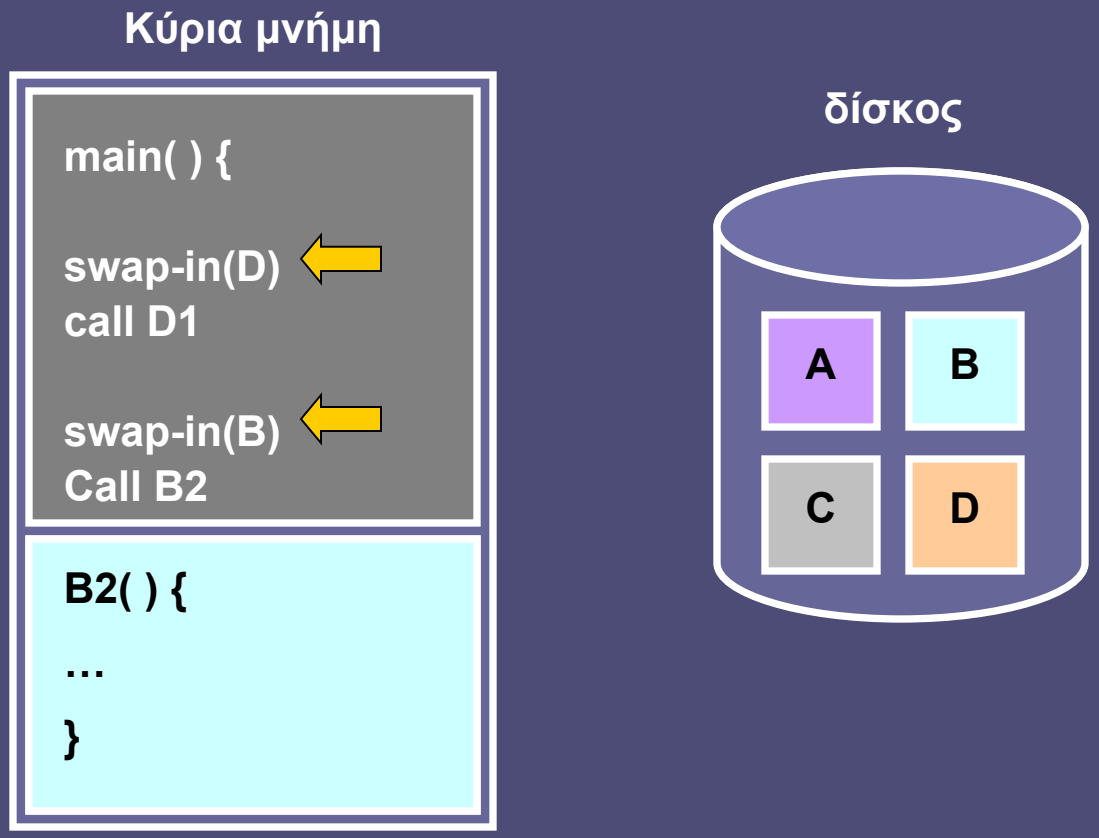
Το πρώτο σύστημα εικονικής μνήμης παρουσιάστηκε το 1962 (Atlas computer)

- Για ποιον λόγο εμφανίστηκε;
  - Στους πρώτους υπολογιστές το μέγεθος της κύριας μνήμης ήταν περιορισμένο
  - Ακόμα και στην περίπτωση του μονοπρογραμματισμού η κύρια μνήμη ήταν ανεπαρκής
    - Τεχνικές με overlays
  - Εμφάνιση ΛΣ με υποστήριξη πολυπρογραμματισμού: αδυναμία ταυτόχρονης διατήρησης πολλών προγραμμάτων στην κύρια μνήμη
  - Η λύση: εικονική μνήμη
    - Μέρος των δεδομένων βρίσκεται στους δίσκους του συστήματος
    - Μεταφορά στην κύρια μνήμη όταν χρειαστεί
    - Πιθανότατα αντικαθιστώντας άλλα τμήματα δεδομένων
    - Τα τελευταία μεταφέρονται πίσω στους δίσκους

Πριν την εικονική μνήμη: overlays

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η τεχνική των overlays απαιτούσε από τον προγραμματιστή να χειρίζεται τις λεπτομέρειες (π.χ διευθύνσεις!) φόρτωσης και κλήσης των υπο-προγραμμάτων!



- Καταλληλότερο για στατικά δεδομένα
  - υποπρογράμματα

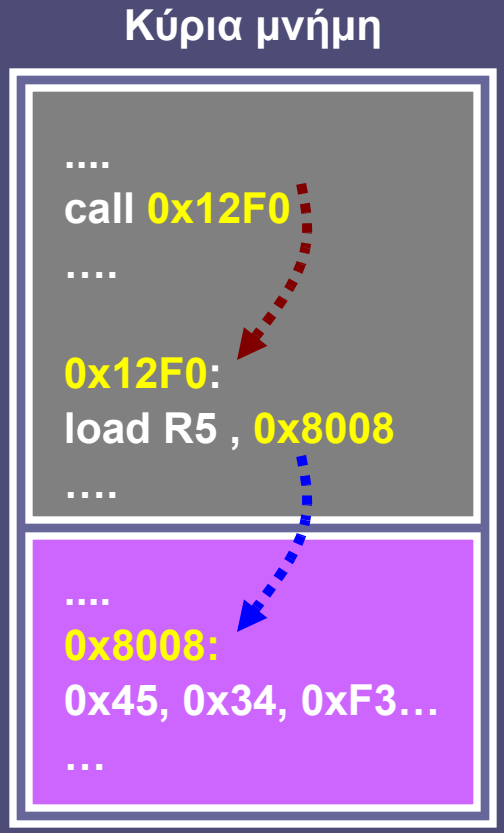
Χώρος διευθύνσεων προγράμματος

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Ποιες οι διευθύνσεις που δημιουργούσε ο μεταγλωττιστής;

Τι συνέβαινε αν το πρόγραμμα δεν φορτωνόταν πάντοτε στον ίδιο χώρο μνήμης;

- Address Space
  - Εκτελούμενο πρόγραμμα στη μνήμη:
  - Διευθύνσεις κώδικα
    - Εντολές διακλάδωσης
  - Διευθύνσεις δεδομένων
    - Εντολές load-store
- Πριν την εικονική μνήμη:
  - Φυσικές διευθύνσεις
  - Άμεση αντιστοιχία με διευθύνσεις κύριας μνήμης

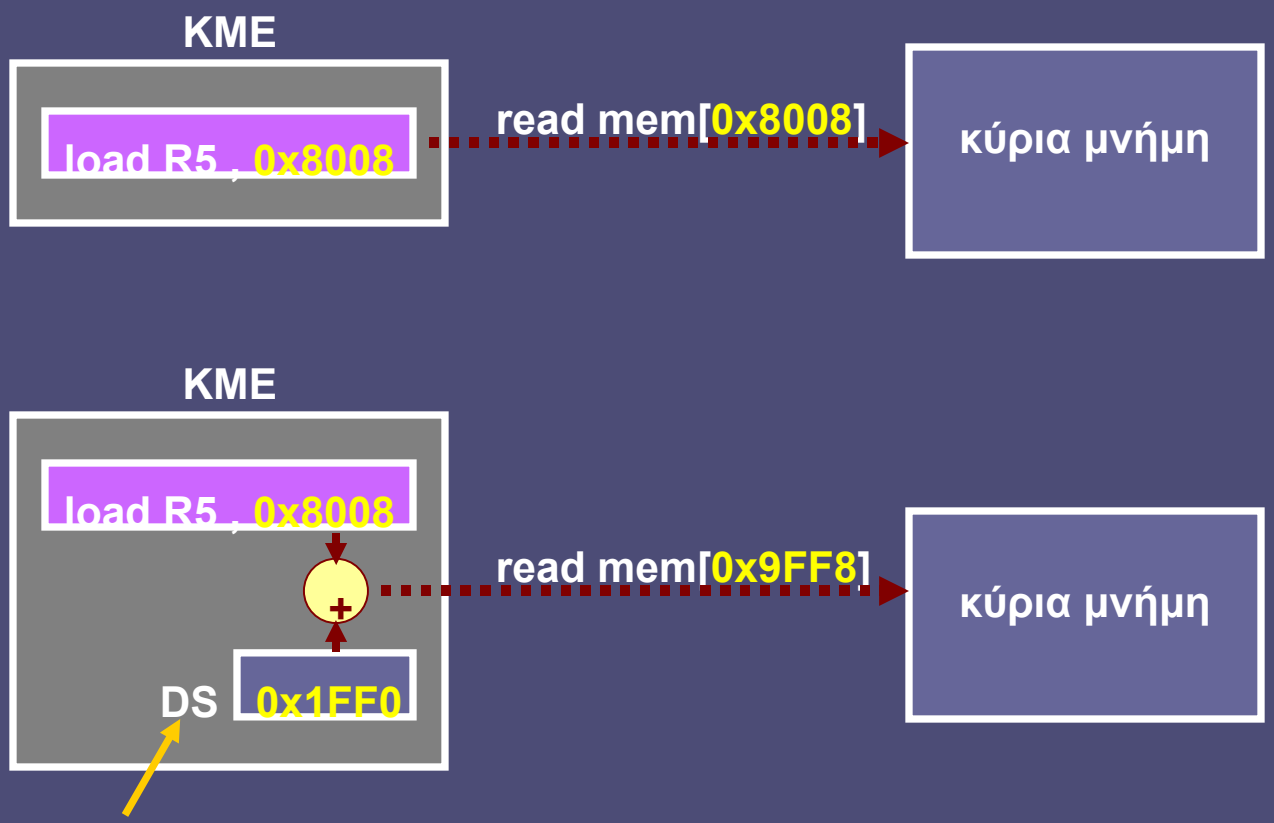


Πολυπρογραμματισμός πριν την εικονική μνήμη

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Χρήση καταχωρητών βάσης (χωρίς άλλη υποστήριξη) σε χαμηλού κόστους επεξεργαστές (π.χ. Intel 8086).

Διευκόλυνση τοποθέτησης προγραμμάτων οπουδήποτε στη μνήμη.



- Καταχωρητές βάσης κώδικα, δεδομένων
- Αλλαγή τιμής καταχωρητών βάσης ανά πρόγραμμα

Μεταβαίνοντας σε εικονικές διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

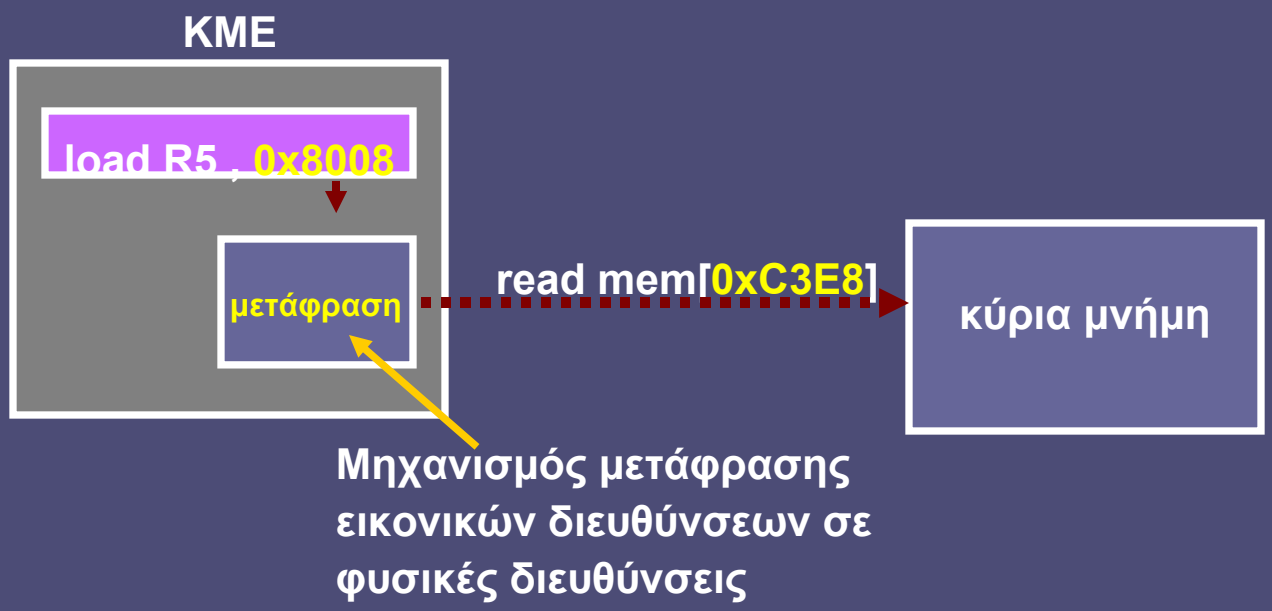
- Το προηγούμενο σχήμα
  - Εισήγαγε την αποσύνδεση των λογικών διευθύνσεων των προγραμμάτων από τις φυσικές διευθύνσεις κύριας μνήμης
  - Με απλή αντιστοιχία:

    φυσική διεύθυνση = λογική διεύθυνση + καταχωρητής βάσης
  - Απαιτείται υποστήριξη από το υλικό (ΚΜΕ)
  - Το πρόγραμμα μπορεί να φορτωθεί σε οποιαδήποτε θέση μνήμης (relocation)
    - Δεν περιέχει αναφορές σε φυσικές διευθύνσεις
  - Εισάγεται η έννοια των ξεχωριστών χώρων διευθύνσεων (κώδικα, δεδομένων...) ανά πρόγραμμα
    - χωρίς περαιτέρω υποστήριξη όμως!

Εικονικές Διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Ο κύριος ρόλος της εικονικής μνήμης στα σημερινά υπολογιστικά συστήματα, με την άφθονη κύρια μνήμη, είναι η υποστήριξη και προστασία των εικονικών χώρων διευθύνσεων ανά πρόγραμμα



- Μετάφραση εικονικών διευθύνσεων προγραμμάτων
- Προστασία χώρων διευθύνσεων
  - Προνόμια προσπέλασης χώρων μνήμης
- Απαιτείται συνδυασμένη υποστήριξη από ΚΜΕ, κρυφή μνήμη και λειτουργικό σύστημα

Η εικονική μνήμη (ξανά)

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η διαχείριση της εικονικής μνήμης έχει ομοιότητες με τη διαχείριση κρυφής-κύριας μνήμης. Τα δύο επίπεδα εδώ όμως (μνήμη-δίσκοι) έχουν σημαντικές διαφορές στα χαρακτηριστικά τους!

- Ποιος ο ρόλος της;
  1. Η εικονική χρήση μνήμης πέρα από όση είναι πραγματικά διαθέσιμη
  2. Η αντιστοίχιση εικονικών διευθύνσεων σε φυσικές και η διαχείριση των προνομίων προσπέλασης
     1. Σημαντικότερο σήμερα!
- Ποιος τη διαχειρίζεται;
  - Αυτόματη διαχείριση από το λειτουργικό σύστημα
  - Υποστήριξη από το υλικό (ΚΜΕ/κρυφή μνήμη)
- Πώς υλοποιείται;
  - Μετακίνηση τμημάτων μνήμης από/προς τους δίσκους
  - Εκμετάλλευση αρχής τοπικότητας
    - Μερικά μέρη μόνο των προγραμμάτων είναι "ενεργά" κάθε στιγμή

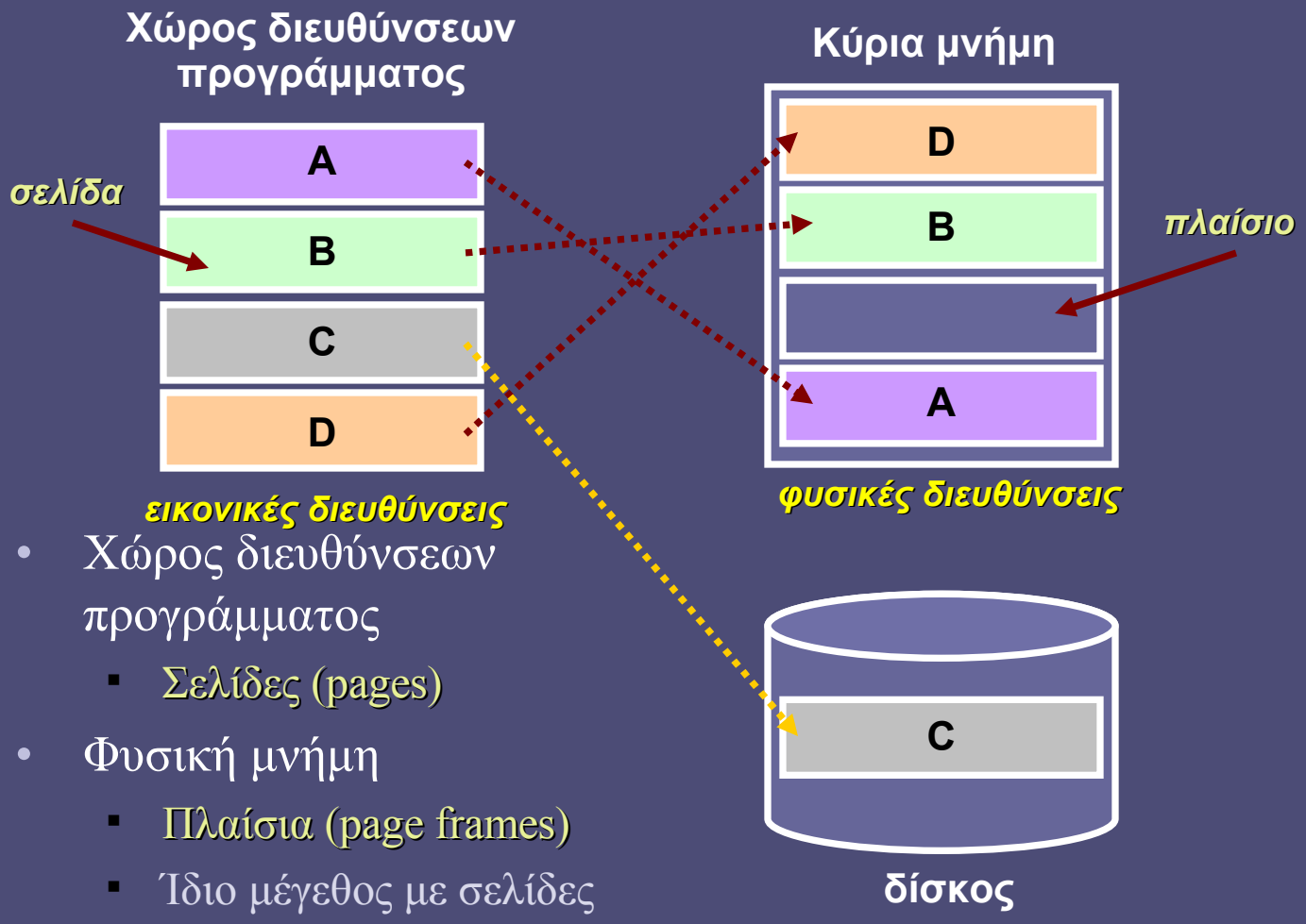
Βασικό σχήμα Εικονικής Μνήμης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Σελιδοποίηση (paging):

Το μέγεθος των σελίδων καθορίζεται από τους μηχανισμούς του υλικού.

Συνήθη μεγέθη 4KB έως 16KB



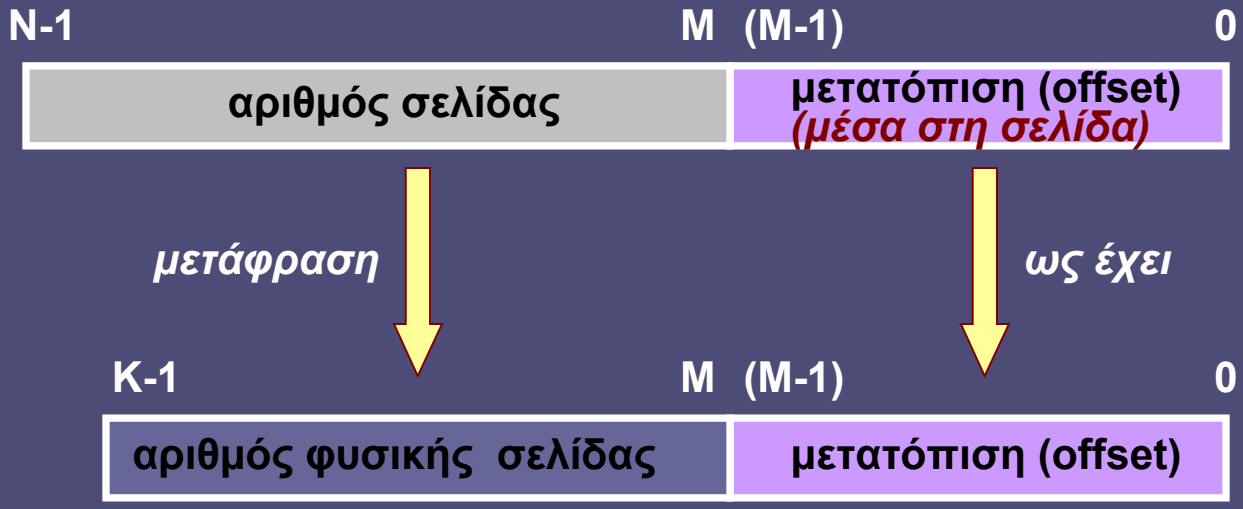
- Χώρος διευθύνσεων προγράμματος
  - Σελίδες (pages)
- Φυσική μνήμη
  - Πλαίσια (page frames)
  - Ίδιο μέγεθος με σελίδες

Σελίδες και εικονικές διευθύνσεις

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η μετάφραση γίνεται μέσα στην ΚΜΕ, άρα εκεί (δηλ. στο υλικό) καθορίζεται το μέγεθος της σελίδας

Ο επεξεργαστής παράγει εικονική διεύθυνση



Στη μνήμη στέλνεται φυσική διεύθυνση

- Ο επεξεργαστής μπορεί να παράγει έως και 2^N εικονικές διευθύνσεις
- Η φυσική μνήμη μπορεί να έχει έως 2^K διευθύνσεις
- Το μέγεθος σελίδας είναι 2^M bytes

“Σελίδες” εναντίον “Τμημάτων”

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η πολυπλοκότητα του μηχανισμού χρήσης τμημάτων (segmentation) είναι ο λόγος που η τεχνική αυτή σπάνια υποστηρίζεται ή χρησιμοποιείται σήμερα...

- Σελίδες (pages)
  - Σταθερό μέγεθος (4KB-16KB)
  - Μία λέξη εικονικής διεύθυνσης (page+offset)
  - Ευκολία τοποθέτησης και αντικατάστασης σελίδων στην κύρια μνήμη
  - Βέλτιστο μέγεθος για μεταφορά από-προς δίσκο
  - Ενδεχομένως μέρος σελίδας μένει αχρησιμοποίητο
- Τμήματα (segments)
  - Μεταβλητό μέγεθος (από 1 byte(!) έως 2^{32} bytes)
  - Δύο λέξεις ανά διεύθυνση (segment, offset)
  - Αναγκαίος ο έλεγχος ορίων
  - Δυσκολία τοποθέτησης στη μνήμη (εύρεση θέσης)
  - “ταιριάζει καλύτερα στα χαρακτηριστικά των εφαρμογών” (;)

Σελιδοποίηση κατ'απαίτηση

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Η μεγάλη επιβάρυνση για τη μετακίνηση των σελίδων από και προς τον δίσκο είναι καθοριστική για την απόδοση ενός συστήματος

- Οι σελίδες των προγραμμάτων (κώδικας-δεδομένα) βρίσκονται αρχικά μόνο στον δίσκο
- Το ΛΣ τις σημειώνει ως "απούσες" από τη μνήμη
- Όταν προσπελαστεί μια "απούσα" σελίδα, δημιουργείται ένα σφάλμα εκτέλεσης (page fault)...
- ...και το ΛΣ τη φορτώνει σε ένα πλαίσιο στη μνήμη
- Ενδεχομένως εκτοπίζοντας πίσω στον δίσκο μια άλλη σελίδα από τη μνήμη
  - Η τελευταία σημειώνεται ως "απούσα"
- Page faults: μεγάλο κόστος σε κύκλους αναμονής
  - 1-10Μκύκλοι

Κρίσιμα σημεία στη σχεδίαση εικονικής μνήμης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

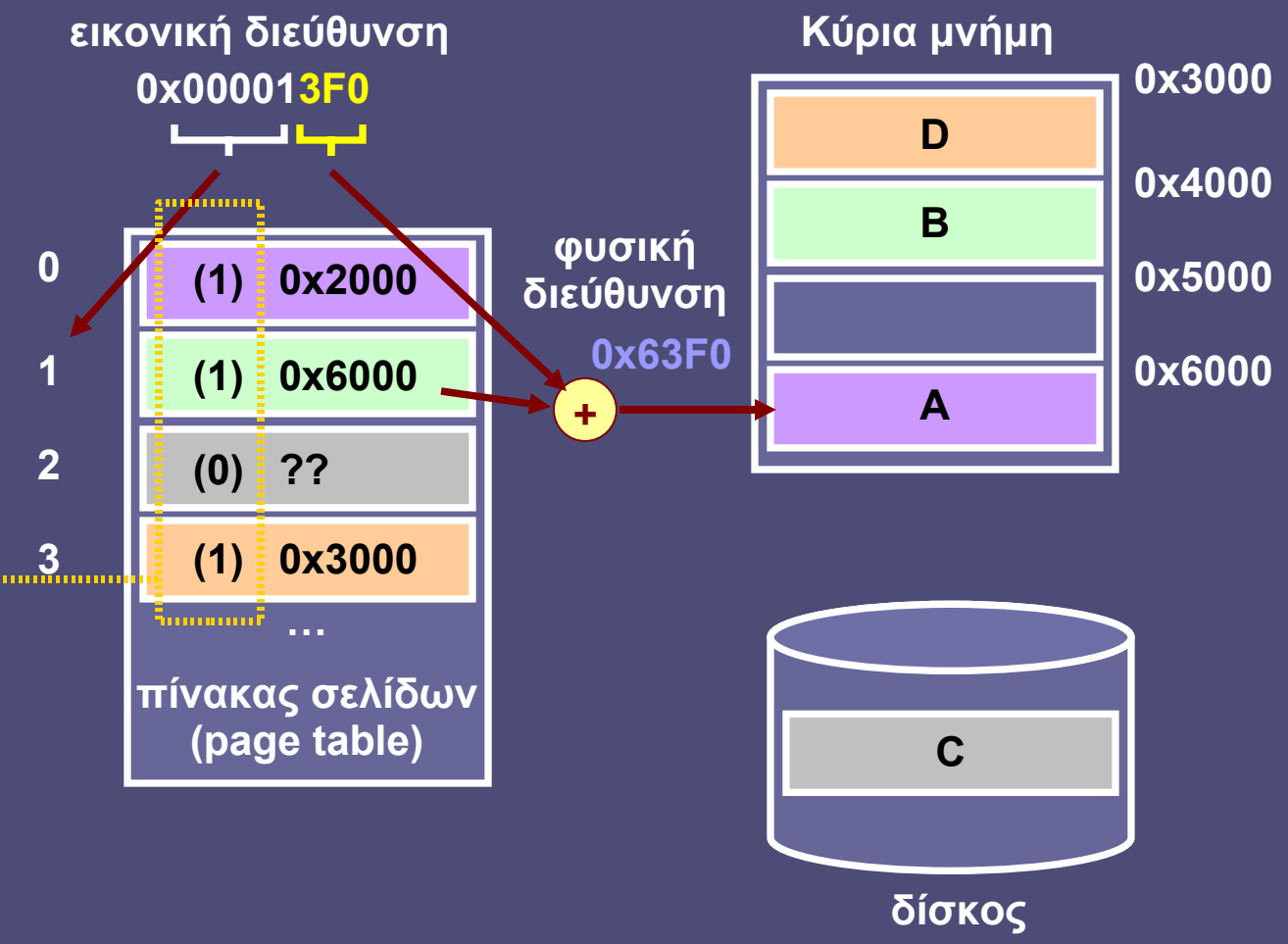
Ακόμα και μικρή μείωση στην εμφάνιση page faults μπορεί να έχει σημαντικό όφελος για την απόδοση του συστήματος

- Οι σελίδες πρέπει να έχουν ικανό μέγεθος
  - Για εξισορρόπηση του κόστους προσπέλασης του δίσκου
- Η μείωση των page faults είναι επιβεβλημένη
  - Οι σελίδες τοποθετούνται οπουδήποτε μέσα στη μνήμη
    - Σχήμα ανάλογο των **fully-associative** κρυφών μνημών
- Η διαχείριση της εικονικής μνήμης μπορεί να γίνεται από λογισμικό (ΛΣ)
  - Μικρή επιβάρυνση συγκρινόμενη με χρόνο μετακίνησης σελίδων στους δίσκους
  - Δυνατότητα χρήσης πολυπλοκότερων αλγορίθμων για τοποθέτηση-αντικατάσταση σελίδων στη μνήμη
- Δεν είναι δυνατή η ενημέρωση στο δίσκο με κάθε εγγραφή νέων δεδομένων στη σελίδα

Μετάφραση εικονικών διευθύνσεων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Presence bit: βρίσκεται η σελίδα στην κύρια μνήμη;



Πίνακας σελίδων

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Τι συμβαίνει στην περίπτωση πολλαπλών προγραμμάτων;

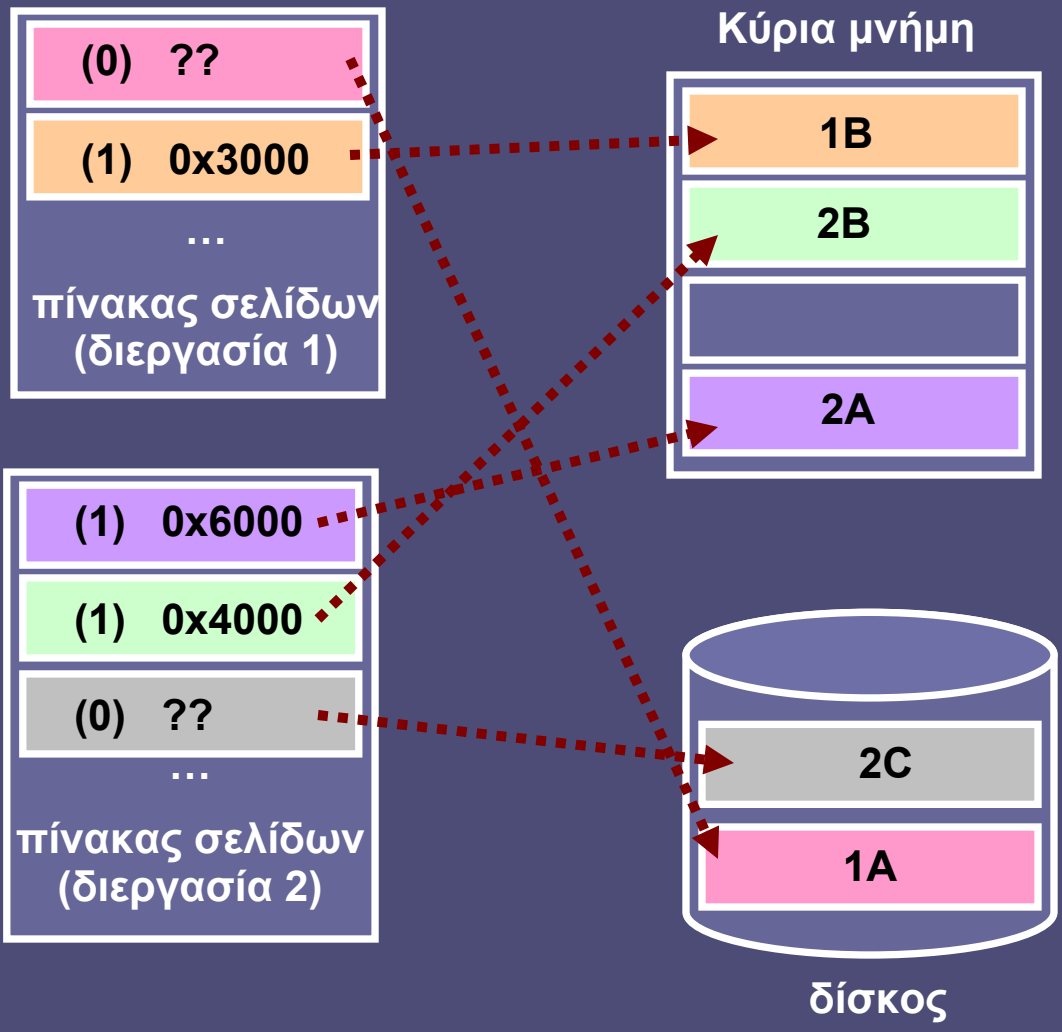
- Πού βρίσκεται;
  - Στην κύρια μνήμη!
- Πόσες θέσεις διαθέτει;
  - Θεωρητικά: ίσες με τον μέγιστο αριθμό σελίδων
    - Για 2^{20} σελίδες με 4 bytes ανά γραμμή, απαιτούνται 4MB
  - Πρακτικά: πίνακες πολλαπλών επιπέδων
  - Πιθανόν: σελιδοποίηση πινάκων (!)
    - Αρκεί να υπάρχει πάντα στη μνήμη ένα μέρος του πίνακα
- Ποια πρόσθετη πληροφορία χρειάζεται;
  - Βρίσκεται η σελίδα στη μνήμη; Έχει αλλάξει;
  - Αν όχι, σε ποιο σημείο του δίσκου είναι;
    - Συχνά μέσα στον πίνακα σελίδων
  - Ποια σελίδα βρίσκεται σε κάθε πλαίσιο μνήμης;
  - Πόσο πρόσφατα χρησιμοποιήθηκε;
    - Δομές του ΛΣ για αλγόριθμο αντικατάστασης σελίδων

Πίνακες σελίδων και πολλαπλά προγράμματα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Κάθε διεργασία έχει τους δικούς του πίνακες σελίδων!

Κατά την εναλλαγή διεργασιών αλλάζει και ο καταχωρητής-δείκτης στους πίνακες



Translation-Lookaside Buffer

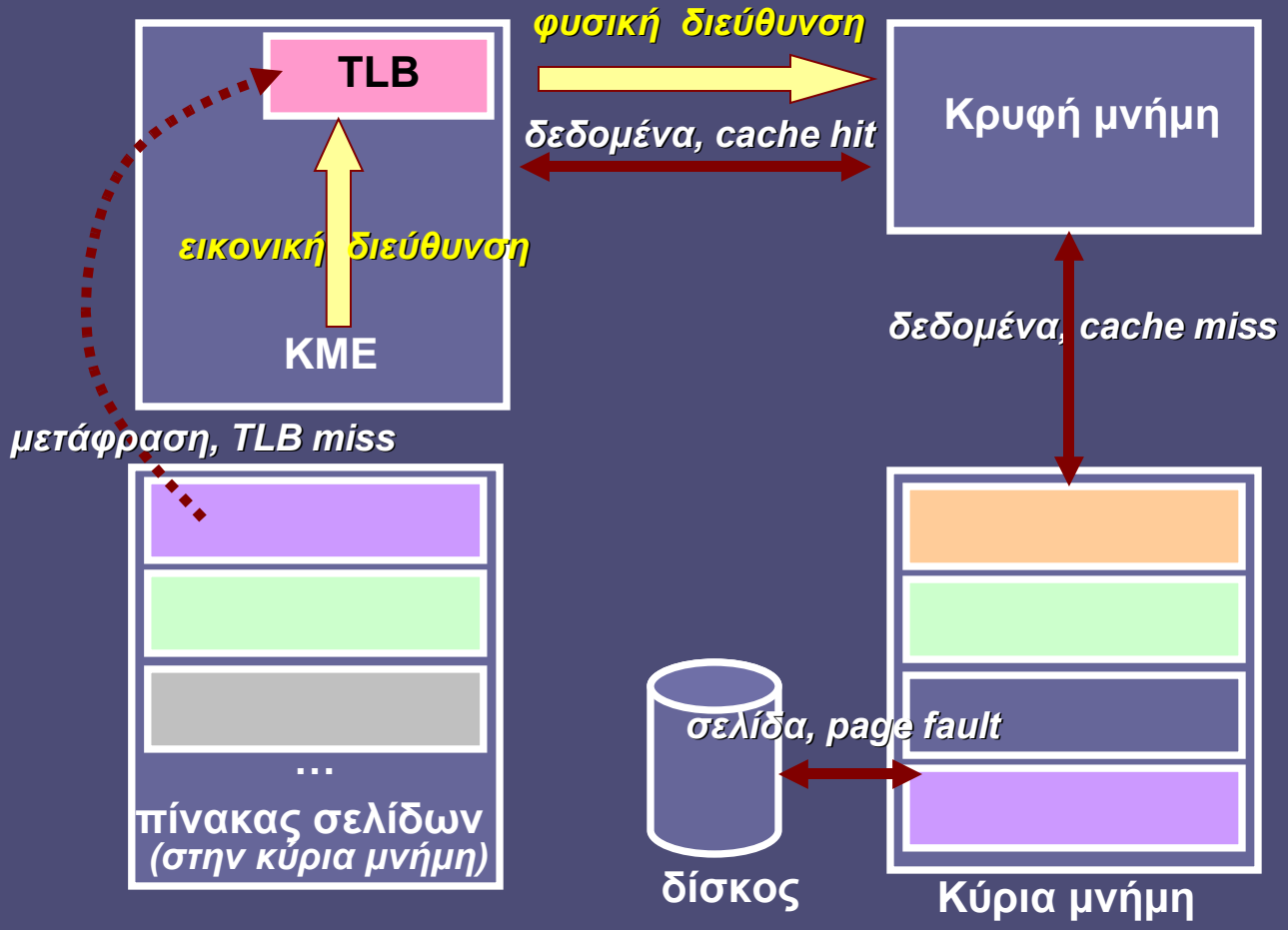
- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- **Το πρόβλημα με τους πίνακες σελίδων**
  - Βρίσκονται στην κύρια μνήμη
  - Για κάθε προσπέλαση μνήμης απαιτείται μια δεύτερη!
    - Για τον πίνακα σελίδων
    - Μη αποδεκτή χρονική επιβάρυνση!
- **Translation-Lookaside Buffer (TLB)**
  - Μικρή κρυφή μνήμη για πρόσφατες μεταφράσεις εικονικών διευθύνσεων
    - Μέσα στην ΚΜΕ
    - 16-512 θέσεις, 1-2 γραμμές του πίνακα σελίδων ανά θέση
    - Προσπέλαση < 1 κύκλο ρολογιού
    - Διάφορες οργανώσεις κρυφής μνήμης
    - Ενημέρωση από υλικό ή λογισμικό
    - Παρατηρούμενο Miss rate: 0.01% - 1%

Προσπέλαση μνήμης: η συνολική εικόνα

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

Στο σχήμα η κρυφή μνήμη δέχεται φυσικές διευθύνσεις. Για λόγους απόδοσης υπάρχουν αρχιτεκτονικές με εικονική κρυφή μνήμη ή συνδυασμό εικονικής-φυσικής κρυφής μνήμης



Εικονική Μνήμη και Προστασία Προσπέλασης

- Ιεραρχία Μνήμης
- Εικονική Μνήμη

- **Προστασία προσπέλασης σελίδων**
  - Με διαφορετικούς πίνακες σελίδων ανά διεργασία είναι αδύνατη η προσπέλαση "ξένων" σελίδων
  - Δικαιώματα προσπέλασης ανά σελίδα
  - User mode και Supervisor Mode
    - Σε user mode δεν είναι δυνατή η προσπέλαση του TLB, του πίνακα σελίδων και των αντίστοιχων καταχωρητών συστήματος
    - Υπάρχουν αρχιτεκτονικές με περισσότερα από 2 επίπεδα προνομίων
  - Ελεγχόμενη προσπέλαση συναρτήσεων ΛΣ
    - Call gates: ελεγχόμενη εκτέλεση συναρτήσεων ΛΣ από χρήστη, στο επίπεδο όμως των προνομίων του χρήστη → δεν είναι δυνατή η προσπέλαση "ξένων" δεδομένων!

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Διασύνδεση Εισόδου-Εξόδου (Ι)

(δίαυλοι και μέθοδοι επικοινωνίας)

http://www.ionio.gr/~mistral/tp/comparch/

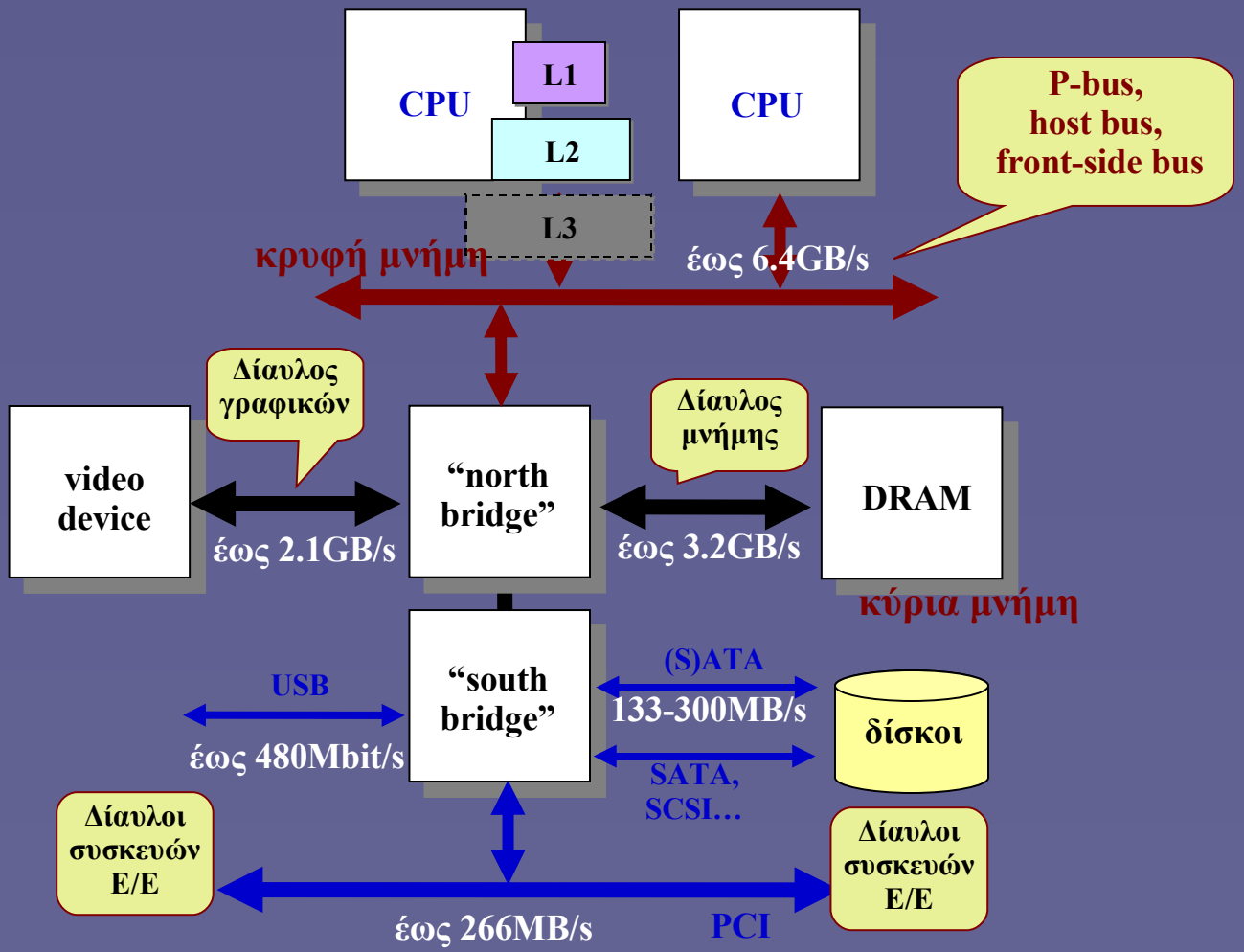
Μ.Στεφανιδάκης

Διασυνδετικοί Δίαυλοι

- Διασυνδετικοί δίαυλοι

Διασυνδετικοί δίαυλοι:

Αναλαμβάνουν την μεταφορά των δεδομένων σε ένα υπολογιστικό σύστημα



Μεταφορά δεδομένων

- Διασυνδετικοί δίαυλοι

Ποιο θα μπορούσε να είναι το ζητούμενο από έναν διασυνδετικό δίαυλο εκτός της απόδοσης;

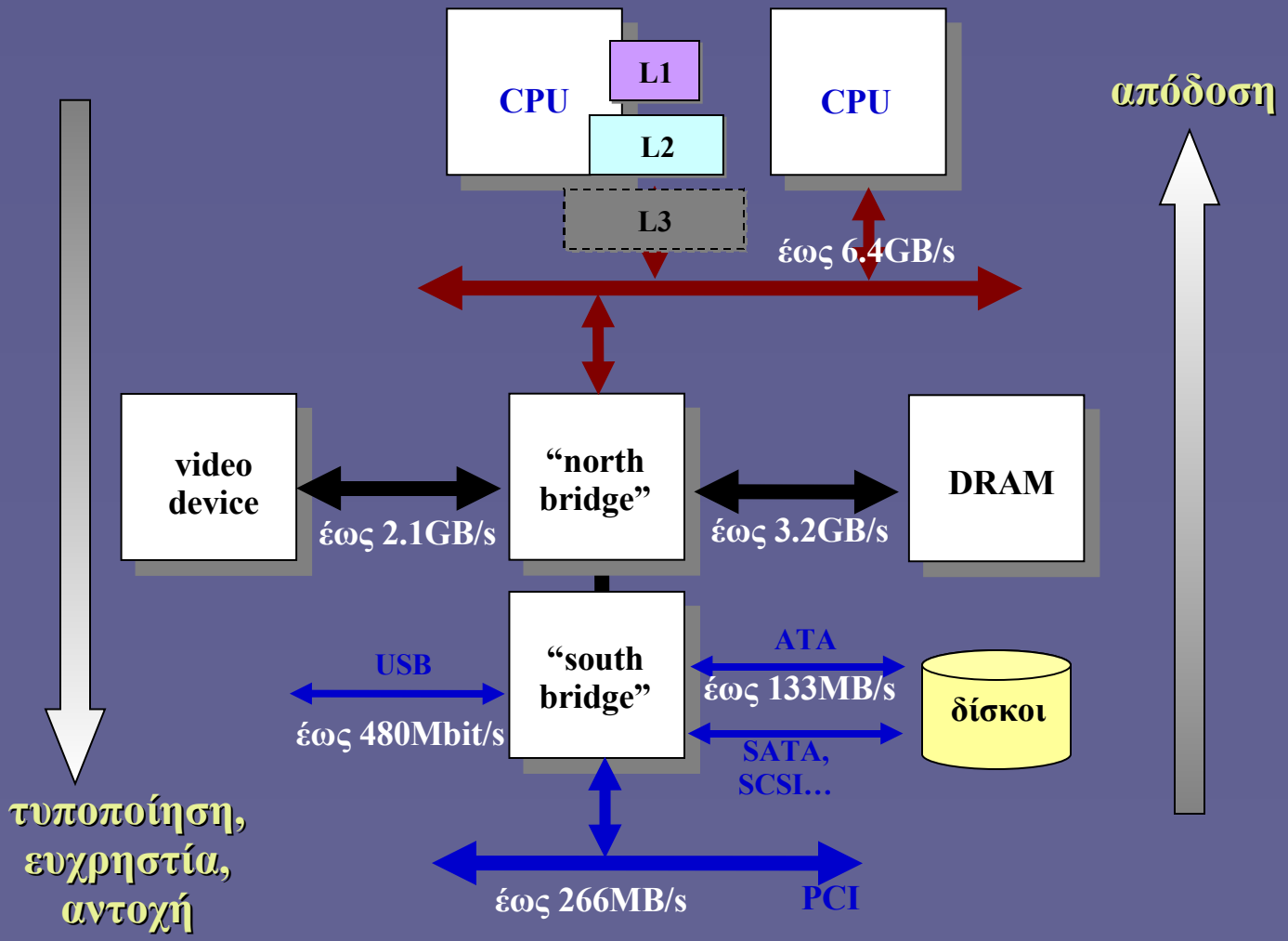
- Σε ένα υπολογιστικό σύστημα
  - Μεταφορά δεδομένων επεξεργασίας
    - Μεταξύ ΚΜΕ, κύριας μνήμης και συσκευών Ε/Ε
    - Τον ρόλο αναλαμβάνουν οι διασυνδετικοί δίαυλοι
  - Απόδοση των "συστημάτων μεταφοράς"
    - Σημαντική παράμετρος της συνολικής απόδοσης
    - Αλλά όχι το μόνο ζητούμενο!
- Χαρακτηριστικά διαύλων
  - Υπάρχει μεγάλη ποικιλία διαύλων
    - Με διαφορετικά λειτουργικά χαρακτηριστικά
  - Ανάλογα με τον ρόλο του καθενός διαύλου
    - Σε ένα υπολογιστικό σύστημα τα διασυνδεόμενα μέρη (και ιδίως οι συσκευές Ε/Ε) έχουν τελείως διαφορετικές ανάγκες μεταφοράς δεδομένων

Ιεραρχία διασυνδετικών διαύλων

- Διασυνδετικοί δίαυλοι

Το ζητούμενο από έναν διασυνδετικό δίαυλο εξαρτάται από τη θέση του στην ιεραρχία

Ο αριθμός των εναλλακτικών προϊόντων αυξάνεται στα χαμηλότερα επίπεδα της ιεραρχίας



Τι διασυνδέει ένας δίαυλος;

- Διασυνδετικοί δίαυλοι

Η μεταφορά των δεδομένων μέσα σε έναν υπολογιστή επιτυγχάνεται ηλεκτρικά μέσω αγωγών χαλκού. Νέες φωτονικές τεχνολογίες υπόσχονται πολύ μεγαλύτερες ταχύτητες (>10Gbps ανά αγωγό)

- Μέσα στο ολοκληρωμένο κύκλωμα
  - π.χ. ΚΜΕ-κρυφή μνήμη
    - Μέσα στο ολοκληρωμένο κύκλωμα επιτυγχάνεται η μέγιστη ταχύτητα μεταφοράς των δεδομένων, μέσω αγωγών μετάλλου "τυπωμένων" στην επιφάνεια πυριτίου
- Πάνω στο ίδιο τυπωμένο κύκλωμα
  - π.χ μεταξύ κύριας μνήμης και ελεγκτή μνήμης
    - Αγωγοί σχεδιασμένοι με ακρίβεια πάνω στο τυπωμένο κύκλωμα
- Μεταξύ τυπωμένων κυκλωμάτων
  - π.χ η διασύνδεση με τις μονάδες δίσκου
    - Διασύνδεση μέσω εύκαμπτων καλωδίων σε αποστάσεις της τάξης του ενός μέτρου
- Μεταξύ υπολογιστικών συστημάτων
  - π.χ η (τοπική) διασύνδεση της κάρτας δικτύου

Βασικά χαρακτηριστικά διαύλων

- Διασυνδετικοί δίαυλοι

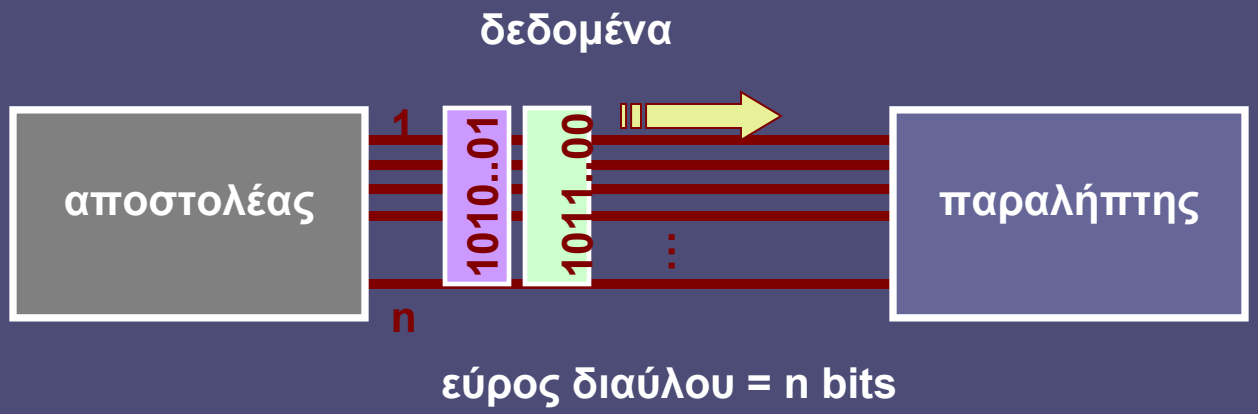
Μεταφορά δεδομένων: η γενική εικόνα

- Πόσα bits μεταφέρονται ταυτόχρονα;
- Πότε ο παραλήπτης θα διαβάσει την είσοδο;
- Ποια η τοπολογία του διαύλου;
- Πώς οργανώνεται η μεταφορά;

Παράλληλοι δίαυλοι

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου

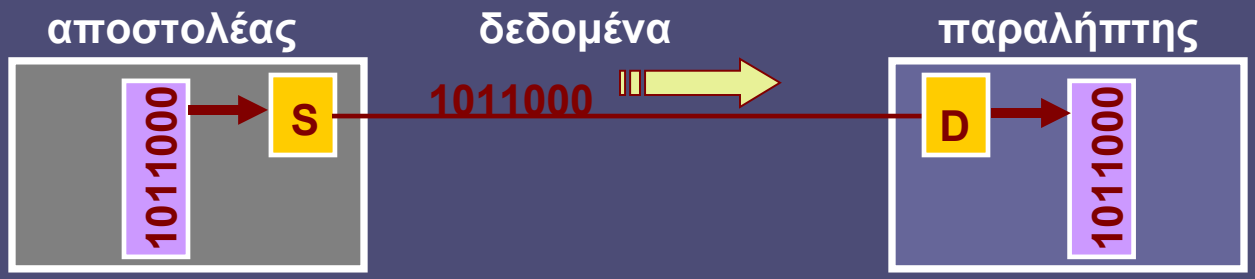
Ο ρυθμός μεταφοράς δεδομένων μετριέται πάντοτε σε μονάδες με βάση το 10: ρυθμός 1Mb/s = 10^6 b/s



- Πολλαπλασιασμός των bits που μεταφέρονται ταυτόχρονα
  - Συχνά ο ρυθμός μετριέται σε μεταφορές/s (T/s)
  - Π.χ. 10MT/s για δίαυλο εύρους 64 bits → 640Mb/s
- Πολύ υψηλοί ρυθμοί μεταφοράς
- Αλλά και μεγάλος ηλεκτρονικός "θόρυβος"
  - Απαιτείται ακριβής σχεδιασμός για να διατηρηθεί η ποιότητα του σήματος και ο συγχρονισμός στη μεταφορά δεδομένων
  - Χρησιμοποιείται στα υψηλότερα επίπεδα της ιεραρχίας διαύλων

Σειριακοί δίαυλοι

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου

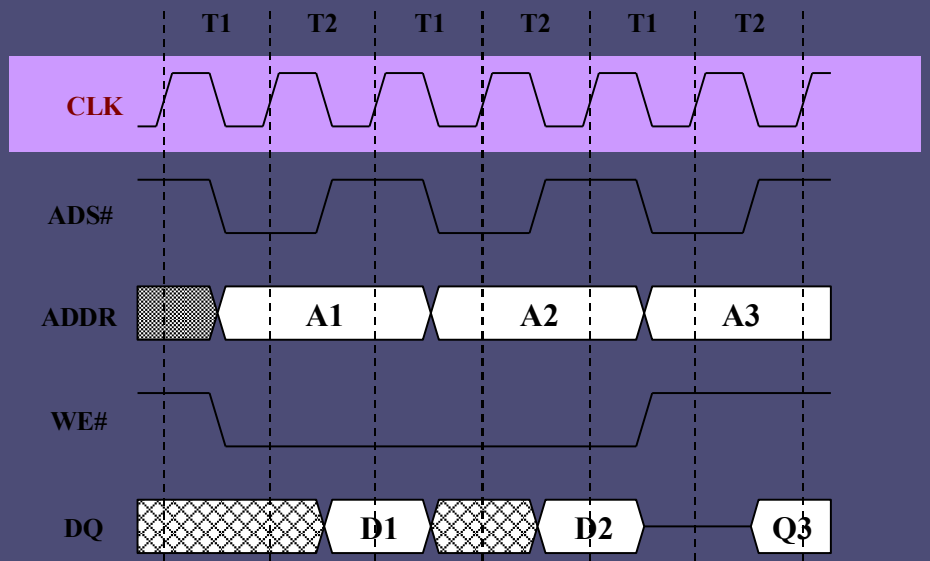


- Οι λέξεις (words) μετατρέπονται σε σειρά από bits πριν την αποστολή - και αντίστροφα μετά την παραλαβή
  - Serializer – deserializer (SerDes)
- Γιατί χρησιμοποιούνται;
  - Μεγαλύτερη ανοσία στον ηλεκτρονικό θόρυβο
  - Δυνατότητα μεταφοράς σε μεγαλύτερη απόσταση
  - Μικρότερες διαστάσεις αγωγού
  - Χρησιμοποιείται στα χαμηλότερα επίπεδα της ιεραρχίας διαύλων

Χρονισμός μεταφοράς

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός

Οι σύγχρονοι δίαυλοι χρησιμοποιούνται στα ανώτερα επίπεδα της ιεραρχίας: π.χ. στο σχήμα φαίνεται η λειτουργία ενός τυπικού διαύλου της ΚΜΕ (FSB)

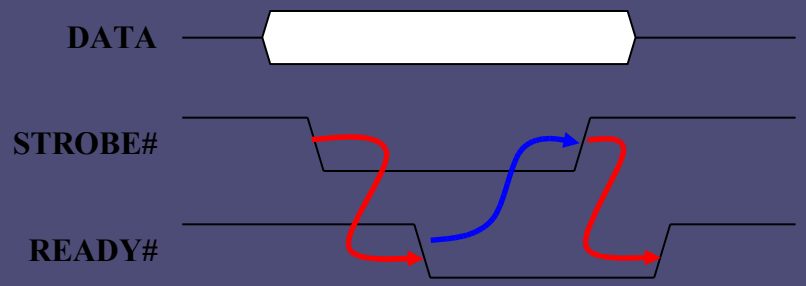


- Πότε ο αποστολέας θα στείλει τα δεδομένα στις εξόδους;
- Πότε ο παραλήπτης θα δειγματοληπτήσει τις εισόδους;
- Γραμμή ρολογιού (clock)
  - Στις ανερχόμενες ή/και κατερχόμενες ακμές
  - Σύγχρονοι δίαυλοι (synchronous)
  - Αλλά: ευαισθησία στις παραμορφώσεις του clock
  - Θόρυβος και κατανάλωση ενέργειας

Άλλα σχήματα χρονισμού

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός

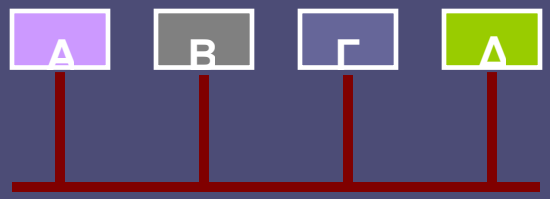
Σειριακοί δίαυλοι με ενσωμάτωση της χρονικής πληροφορίας στα δεδομένα χρησιμοποιούνται στα κατώτερα επίπεδα της ιεραρχίας (π.χ. ο δίαυλος USB)



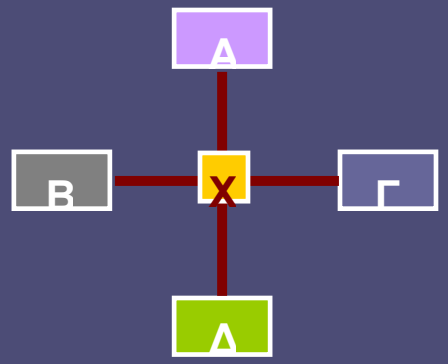
- Ασύγχρονοι δίαυλοι (χωρίς ρολόι)
  - Σήματα συγχρονισμού
  - Ανοσία σε παραμορφώσεις σημάτων
  - Αλλά: αργότερη μεταφορά – πολύπλοκη ανταλλαγή σημάτων
- Ενσωμάτωση ρολογιού στα δεδομένα
  - Δεν υπάρχει ξεχωριστή γραμμή clock
  - Αλλά ο παραλήπτης μπορεί να συμπεράνει για το ρολόι του αποστολέα αν το μεταδιδόμενο σήμα έχει ικανό αριθμό ακμών (εναλλαγές μεταξύ 0 και 1)

Τοπολογίες Διαύλων

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες



Μοιραζόμενος δίαυλος



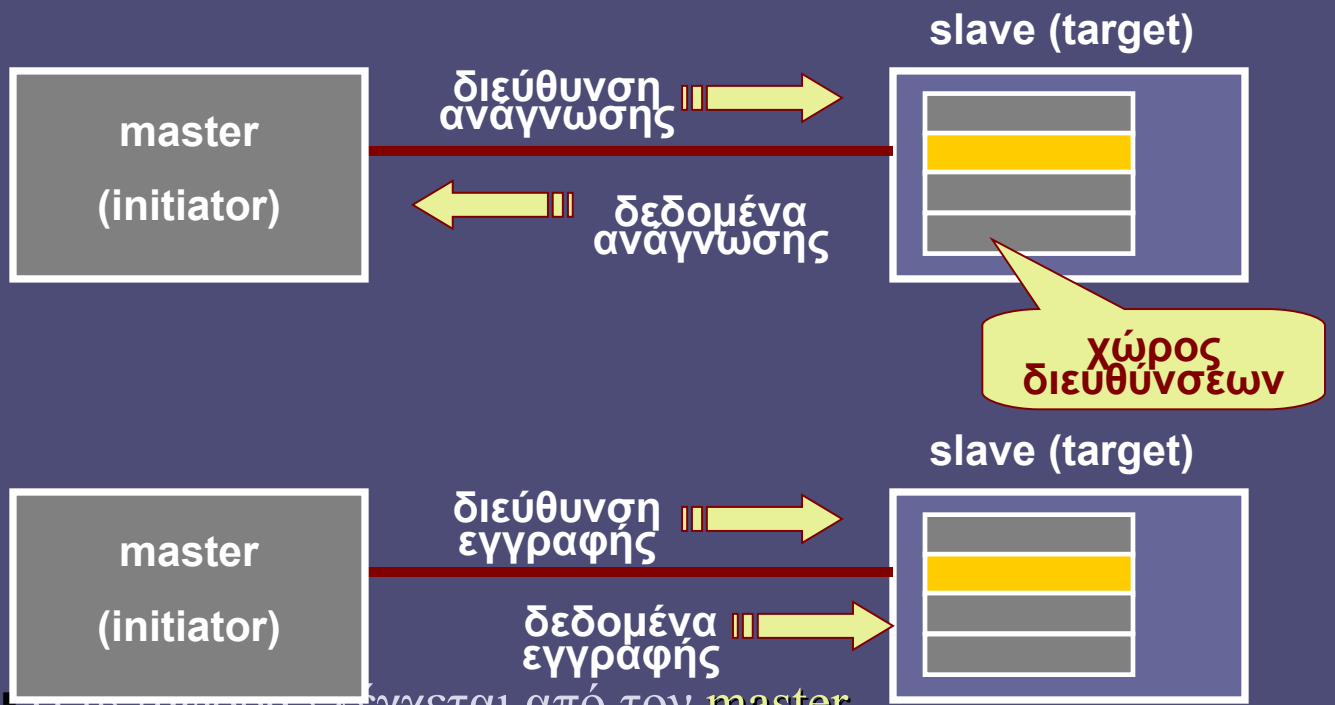
Point-to-point

- Μοιραζόμενος δίαυλος
  - Η παλαιότερη λύση
  - Αναγκαία η διαιτησία για την κατοχή του διαύλου
  - Κακή ποιότητα σημάτων → χαμηλότερη ταχύτητα
- Point-to-point
  - Διασύνδεση πάντα μεταξύ δύο σημείων
  - Τμήματα συγκέντρωσης-ανταλλαγής δεδομένων
    - Hubs ή switches
  - Νεώτερες λύσεις για υψηλότερη ταχύτητα μεταφοράς

Οργάνωση μεταφοράς

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς

Η φορά της "ανάγνωσης" και της "εγγραφής" ορίζεται κατά σύμβαση!

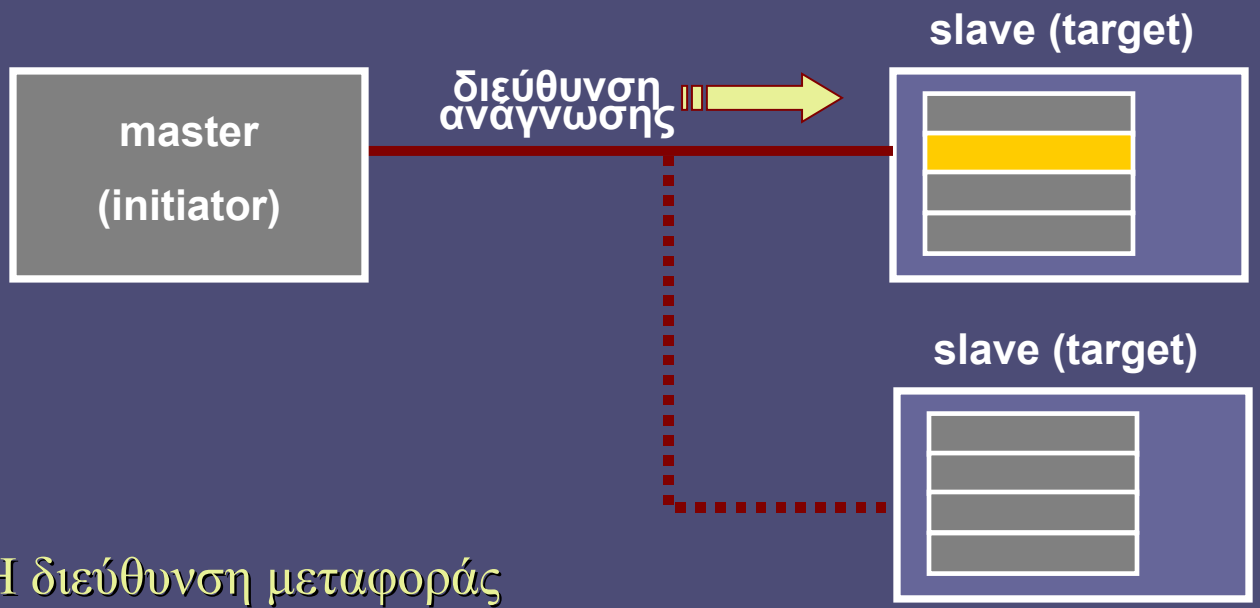


- Η μεταφορά ελέγχεται από τον master
- Ανάγνωση ή εγγραφή από/στον χώρο διευθύνσεων του slave
  - Σε ποιον slave και σε ποιες λέξεις του slave γίνεται η μεταφορά

Διεύθυνση μεταφοράς

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς

Το διπλανό σχήμα υλοποιείται φυσικά (σε μοιραζόμενους διαύλους) ή λογικά (όταν ο δίαυλος είναι point-to-point)



- Η διεύθυνση μεταφοράς
  - Επιλέγει slave
  - Επιλέγει λέξεις μέσα στον χώρο διευθύνσεων του slave
- Κατανομή διευθύνσεων
  - Είτε στατικά (στον σχεδιασμό)
  - Είτε δυναμικά (στην αρχικοποίηση του συστήματος)

Φάσεις μεταφοράς

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς

Λόγω των φάσεων μεταφοράς, η χρήσιμη πληροφορία είναι (max) το 70% της μεταδιδόμενης!

Σπάνια επιτυγχάνονται οι ρυθμοί μεταφοράς (peak rates) που αναφέρονται στις προδιαγραφές!

- Σε κάθε μεταφορά σε έναν δίαυλο
- Ορισμένες (ή όλες) από τις εξής φάσεις:
  - Διαιτησία (arbitration)
    - Σε μοιραζόμενους διαύλους, απόκτηση του διαύλου
  - Αίτηση (request)
    - Αποστολή διεύθυνσης, κατεύθυνσης μεταφοράς (read/write), μήκους μεταφερόμενων δεδομένων (σε bytes)
  - Απόκριση (response)
    - Κατάσταση μεταφοράς (ολοκλήρωση, αποτυχία, επανάληψη, αναμονή)
  - Δεδομένα (data)
    - Τα μεταφερόμενα δεδομένα (όταν υπάρχουν)

Πληροφορία μεταφοράς

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς

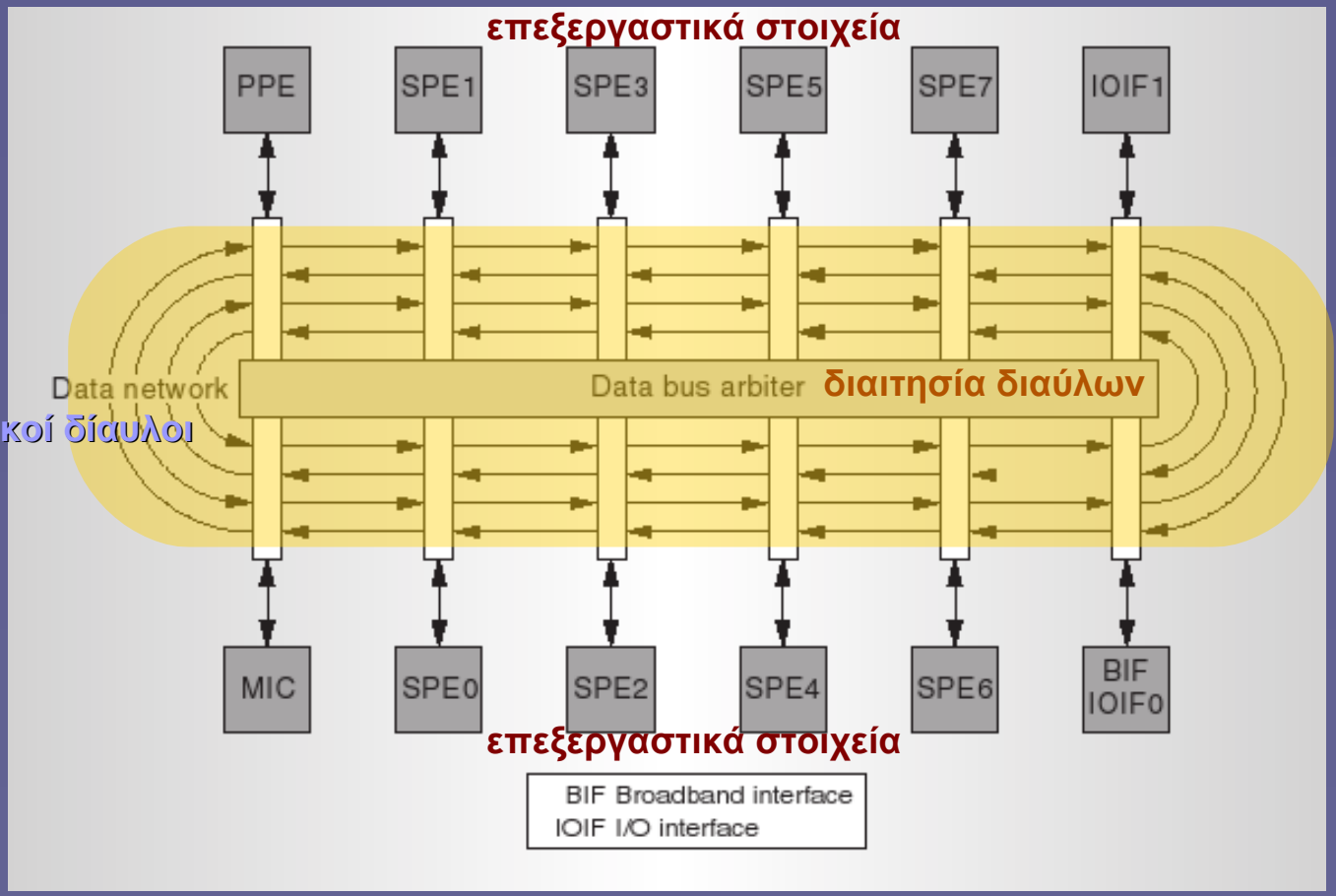
- Διεύθυνση – δεδομένα – εντολές/κατάσταση
- Παράλληλοι δίαυλοι
  - Συνήθως χρησιμοποιούνται ξεχωριστές γραμμές ανά είδος πληροφορίας
  - Συχνά κάποιες πληροφορίες είναι χρονικά πολυπλεγμένες για οικονομία στο πλήθος αγωγών
    - Π.χ. διεύθυνση-δεδομένα
- Σειριακοί δίαυλοι
  - Τα πακέτα δεδομένων που ανταλλάσσονται περιέχουν και τα 3 είδη πληροφορίας

Παράδειγμα διαύλων: Cell Multiprocessor

- Διασυνδετικοί δίαυλοι
- Εύρος Διαύλου
- Χρονισμός
- Τοπολογίες
- Οργάνωση μεταφοράς
- Παράδειγμα

διασυνδετικοί δίαυλοι

peak transfer rate >200GB/s



[IEEE Micro, May-June 2006]

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής

Αρχιτεκτονική Υπολογιστών

2007-08

Διασύνδεση Εισόδου-Εξόδου (ΙΙ)

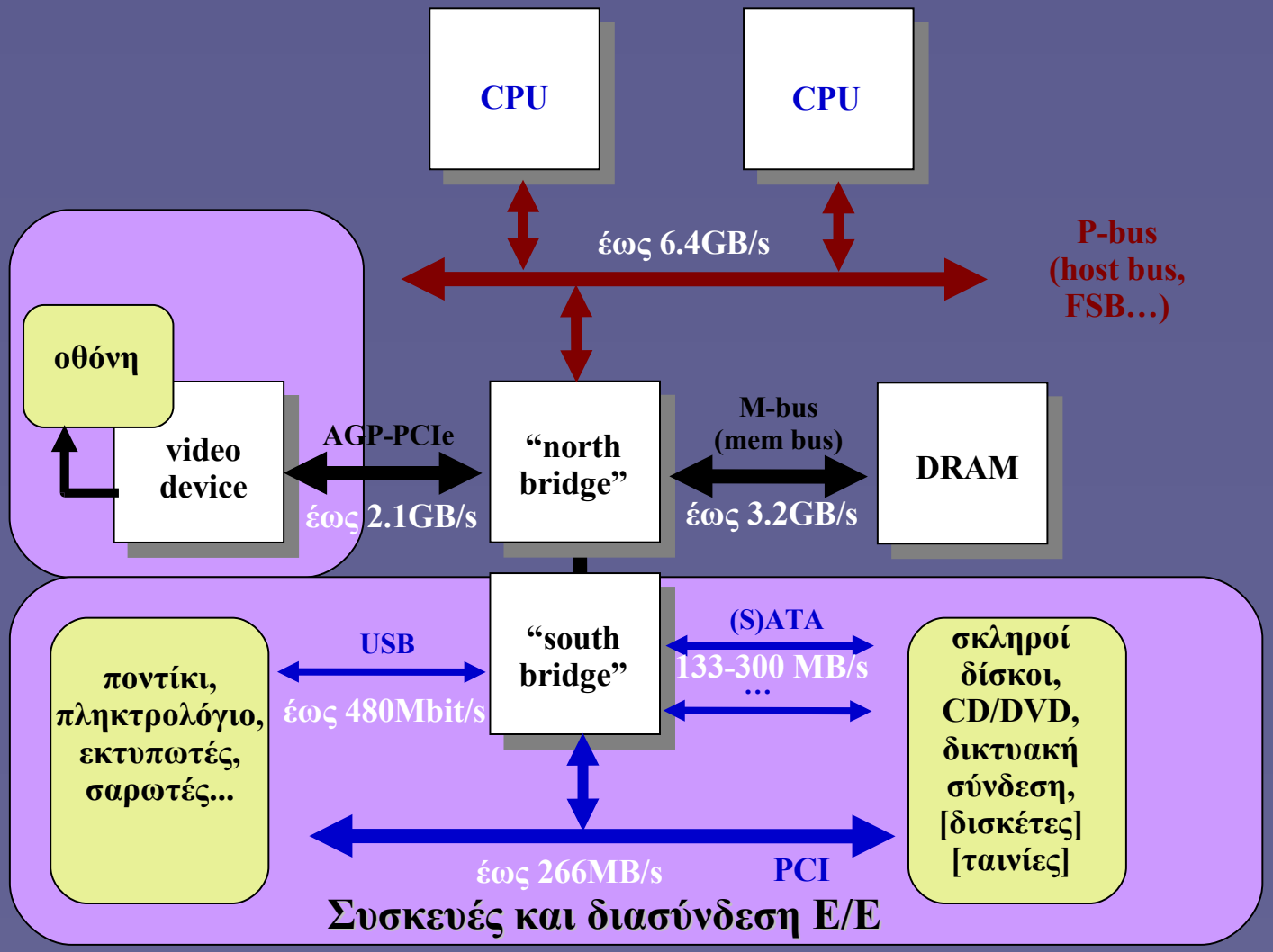
(συσκευές και δίαυλοι Ε/Ε)

http://www.ionio.gr/~mistral/tp/comparch/

Μ.Στεφανιδάκης

Διασύνδεση συσκευών εισόδου εξόδου

- Συσκευές Ε/Ε



Διασύνδεση συσκευών Ε/Ε

- Συσκευές Ε/Ε

- **Συσκευές εισόδου-εξόδου (Ε/Ε)**
  - Συμβατικά: οτιδήποτε εκτός ΚΜΕ και μνήμη
  - Συσκευές με **τελείως διαφορετικούς** μεταξύ τους ρόλους και χαρακτηριστικά
    - **Λειτουργία:** είσοδος, έξοδος ή και τα δύο (αποθήκευση)
    - **Συνεργάτης** (στην άλλη άκρη): άνθρωπος ή μηχανή
    - **Ρυθμός μεταφοράς** δεδομένων

Συσκευή	Λειτουργία	Ρυθμός (Mbps)
πληκτρολόγιο	είσοδος	0,0001
ποντίκι	είσοδος	0,0038
εκτυπωτής	έξοδος	3,2
Δίκτυο LAN	είσοδος/έξοδος	100-1000
μαγν. δίσκος	αποθήκευση	240-2565
οθόνη	έξοδος	800-8000

Διασύνδεση συσκευών Ε/Ε

- Συσκευές Ε/Ε

Λόγω της ποικιλότητας και του πλήθους των συσκευών Ε/Ε οι αντίστοιχοι δίαυλοι διασύνδεσης πρέπει να υπακούν σε πρότυπα – πολύ περισσότερο απ'ότι π.χ. το FSB ενός επεξεργαστή!

- **Απόδοση συστημάτων Ε/Ε**
  - Σύνθετο μέγεθος
  - Εξαρτάται από πολλούς παράγοντες
    - Χαρακτηριστικά συσκευών
    - Χαρακτηριστικά διαύλου διασύνδεσης
    - Ιεραρχία μνήμης
    - Λειτουργικό Σύστημα
  - Χρόνος απόκρισης – Ρυθμός ολοκλήρωσης
- **Σημαντικές παράμετροι (εκτός της απόδοσης)**
  - Αξιοπιστία
  - Μείωση κόστους
  - Επεκτασιμότητα και διαχείριση διαφορετικότητας συσκευών

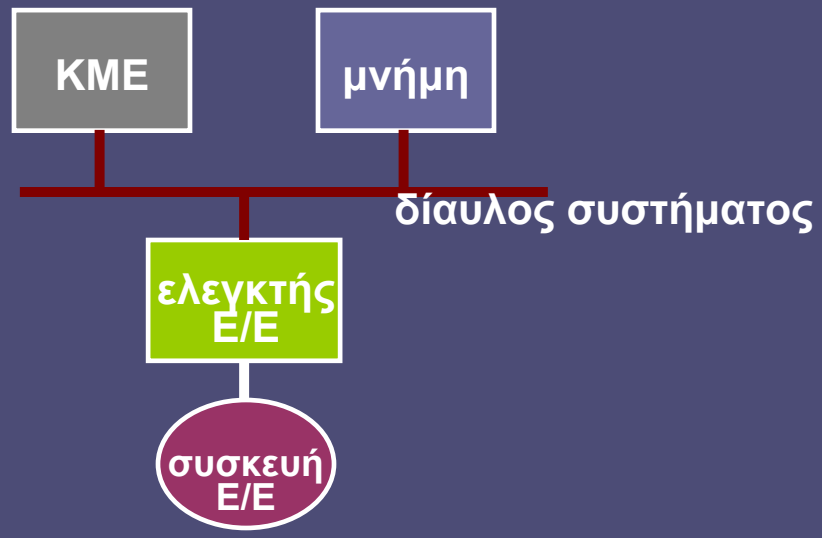
Διασύνδεση συσκευών Ε/Ε

- Συσκευές Ε/Ε

- Κατηγορίες συσκευών Ε/Ε ανάλογα με την διασύνδεσή τους στο σύστημα
  1. Καθορισμός από τη σχεδίαση του συστήματος
     - Βασικά μέρη συστήματος (στη μητρική πλακέτα)
     - Το BIOS γνωρίζει ήδη για τις συσκευές αυτές
  2. Προσθήκη κατά τη σύνθεση του συστήματος
     - Κάρτες επέκτασης
     - Αναγνώριση κατά την εκκίνηση του συστήματος
  3. Προσθήκη κατά τη λειτουργία του συστήματος
     - Συσκευές που συνδέονται εξωτερικά στο σύστημα
     - Αναγνώριση κατά τη λειτουργία του συστήματος (hot-plug)

Οι πρώτες μορφές διασύνδεσης

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε



- Διασύνδεση στον δίαυλο του συστήματος μέσω ενός ελεγκτή Ε/Ε (I/O controller ή adapter)
- Η ΚΜΕ "βλέπει" απευθείας τον ελεγκτή Ε/Ε
- Ο ελεγκτής αναλαμβάνει την επικοινωνία με την πολύ αργότερη συσκευή Ε/Ε

Ο ρόλος ενός ελεγκτή Ε/Ε

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε

Πώς επικοινωνεί η ΚΜΕ με τον ελεγκτή για την υλοποίηση των λειτουργιών αυτών;

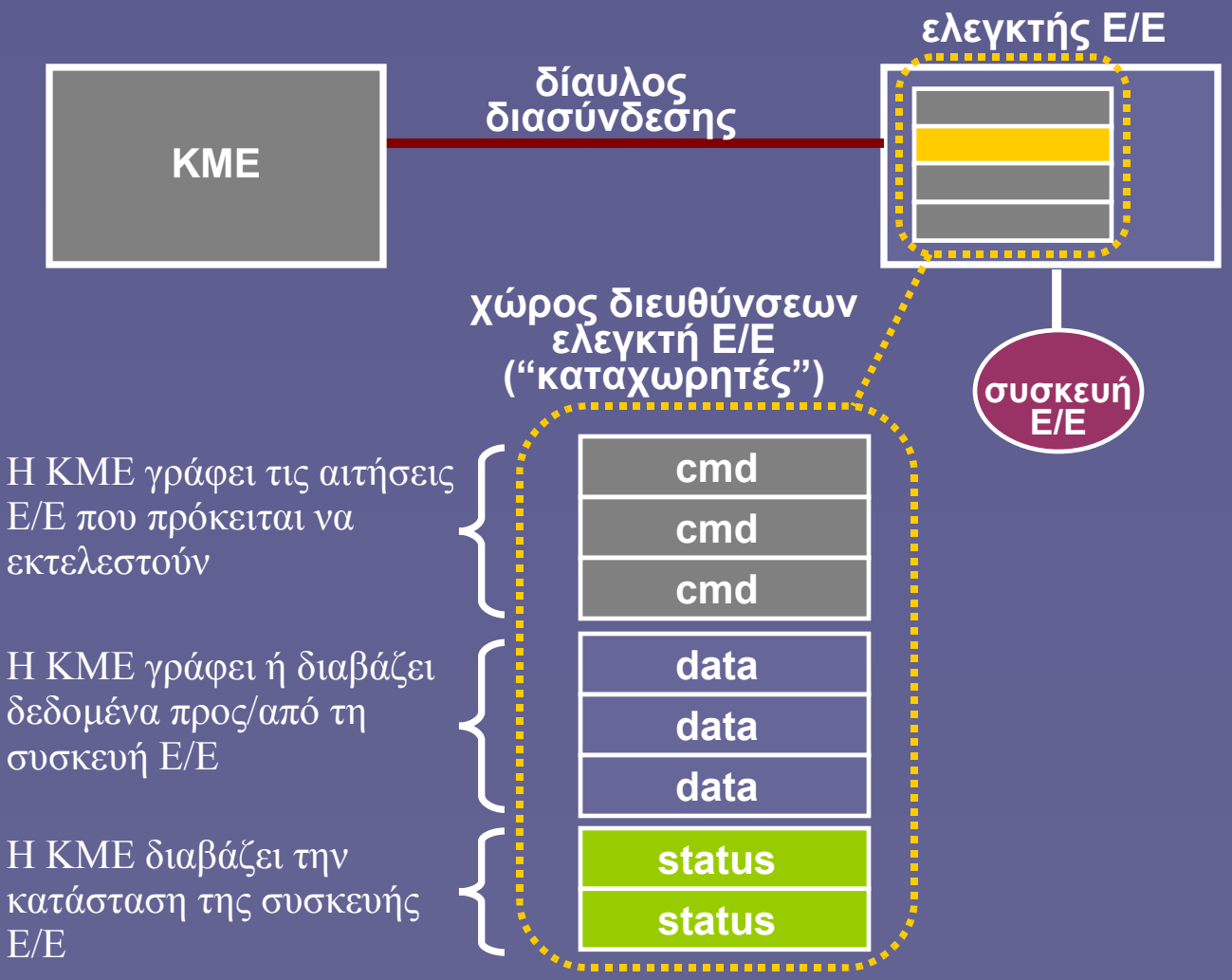
- Αποστολή εντολών προς τη συσκευή Ε/Ε
  - Μετάφραση των αιτήσεων Ε/Ε της ΚΜΕ
    - Μόνο εντολές load-store φτάνουν στον ελεγκτή από την ΚΜΕ!
- Λήψη ένδειξης κατάστασης συσκευής Ε/Ε
  - Δυνατότητα ελέγχου πορείας μιας αίτησης Ε/Ε
    - Οι αιτήσεις Ε/Ε ολοκληρώνονται πολύ αργά σε σχέση με την ταχύτητα επεξεργασίας της ΚΜΕ!
- Ειδοποίηση της ΚΜΕ
  - Μετά την ολοκλήρωση αίτησης Ε/Ε
    - Μηχανισμός interrupts
- Παροχή δεδομένων από συσκευή Ε/Ε
  - Προς την ΚΜΕ ή αυτόνομα προς τη μνήμη
    - Μηχανισμός DMA

Πώς βλέπει η ΚΜΕ έναν ελεγκτή Ε/Ε

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε

"Μοντέλο καταχωρητών" του ελεγκτή Ε/Ε

Οι καταχωρητές βρίσκονται στον χώρο διευθύνσεων του συστήματος



Απεικόνιση διευθύνσεων Ε/Ε

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε

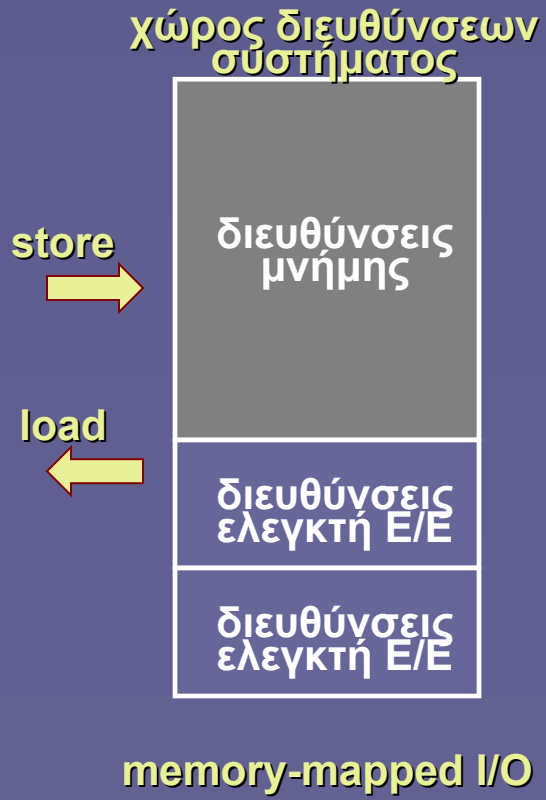
- Η τοποθέτηση των διευθύνσεων των ελεγκτών Ε/Ε στον χώρο διευθύνσεων του συστήματος
  - Στατικά
    - Η αρχιτεκτονική του συστήματος ορίζει τις διευθύνσεις
    - Όχι ευέλικτο σχήμα
  - Δυναμικά
    - Ανάθεση κατά την αρχικοποίηση του συστήματος
    - Απαιτείται συνεργασία υλικού και λογισμικού – το περίφημο "plug-n-play"
- Η ΚΜΕ υποστηρίζει ξεχωριστές εντολές Ε/Ε;
  - Ναι (isolated I/O): δύο ξεχωριστοί χώροι διευθύνσεων
    - in/out αντί load/store
  - Όχι (memory-mapped I/O): ενιαίος χώρος διευθύνσεων

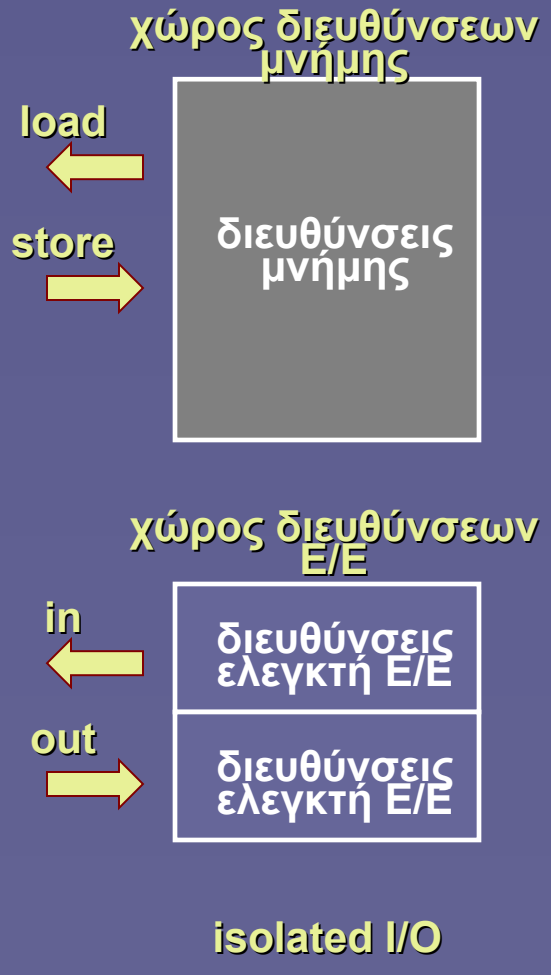
Χώροι διευθύνσεων για Ε/Ε

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε

Memory-mapped I/O: ενιαίος χώρος και διαχείριση των διευθύνσεων συστήματος

Isolated I/O: εύκολος διαχωρισμός εντολών αν πρέπει Ε/Ε να γίνεται μόνο από το ΛΣ

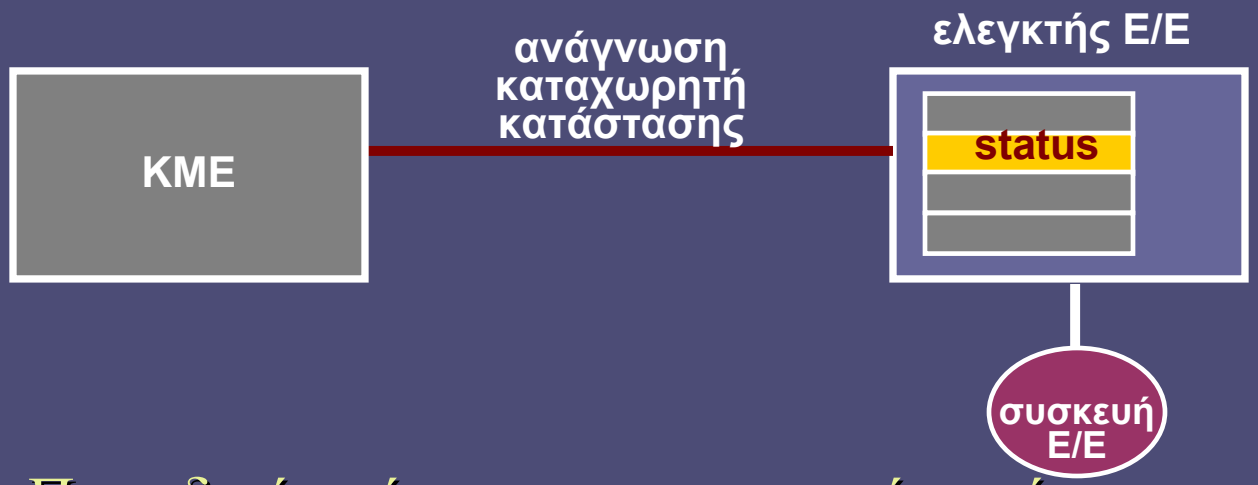




Ανίχνευση ολοκλήρωσης αίτησης Ε/Ε

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ

Το polling παρά την επιβάρυνση χρησιμοποιείται σε συστήματα ελέγχου, όπου ο έλεγχος πρέπει να γίνεται σε προκαθορισμένα διαστήματα

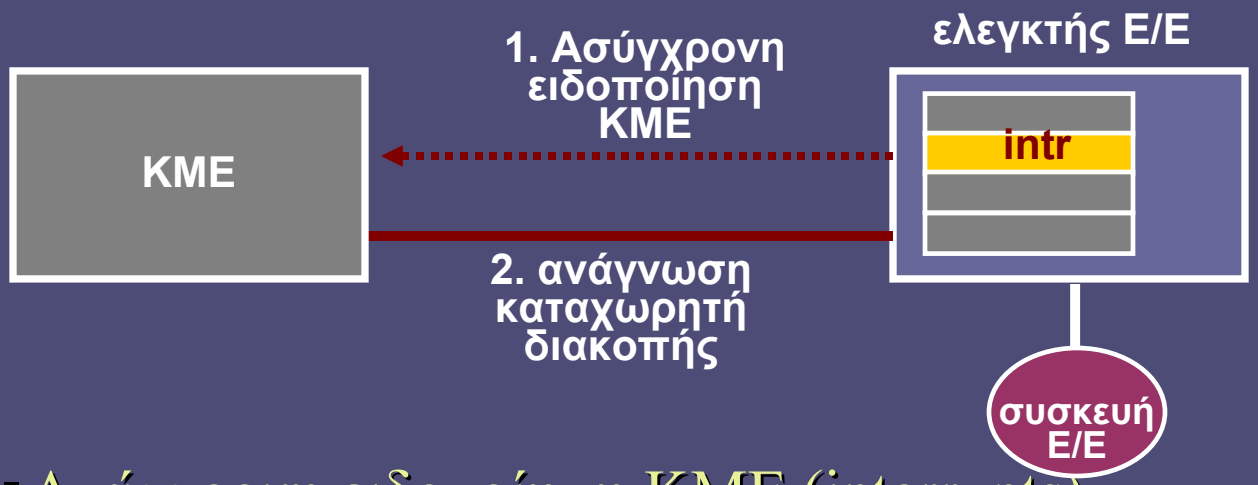


- Περιοδική ανάγνωση καταχωρητή κατάστασης
  - Ολοκληρώθηκε η τρέχουσα αίτηση;
  - Εμφανίστηκε κάποια αλλαγή κατάστασης στη συσκευή Ε/Ε;
  - Επαναληπτική διαδικασία polling
  - Επιβάρυνση ΚΜΕ με άσκοπους κύκλους εκτέλεσης – για μεγάλο διάστημα η κατάσταση θα παραμένει σταθερή περιμένοντας τη συσκευή Ε/Ε

Ανίχνευση ολοκλήρωσης αίτησης Ε/Ε

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ

Παλαιότερα ο ελεγκτής ειδοποιούσε την ΚΜΕ με ξεχωριστό σήμα διακοπής. Σήμερα αυτό υλοποιείται με ανταλλαγή μηνυμάτων μέσω του διαύλου διασύνδεσης

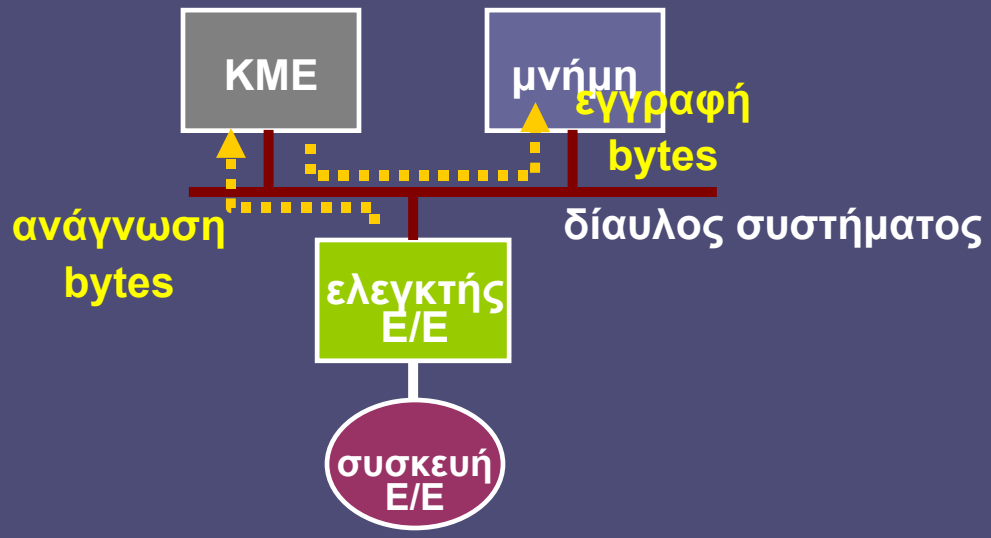


- Ασύγχρονη ειδοποίηση ΚΜΕ (interrupts)
  - Ο ελεγκτής ειδοποιεί την ΚΜΕ για αλλαγή κατάστασης της συσκευής Ε/Ε
  - Στο μεταξύ η ΚΜΕ μπορεί να εκτελεί άλλη διεργασία
  - Η διακοπή θα προκαλέσει την εκτέλεση ρουτίνας εξυπηρέτησης διακοπής (interrupt service routine – ISR)
  - Ενδεχομένως η ΚΜΕ θα χρειαστεί πρόσθετη πληροφορία από ελεγκτή σχετικά με την αιτία της διακοπής

Μετακίνηση δεδομένων από/προς μνήμη

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ

Εάν η συσκευή Ε/Ε είναι κάρτα δικτύου 100Mbps Ethernet και δημιουργεί διακοπή για κάθε πακέτο με μέσο μήκος 256 bytes, πόσες φορές/sec θα ζητηθεί εξυπηρέτηση από την ΚΜΕ;

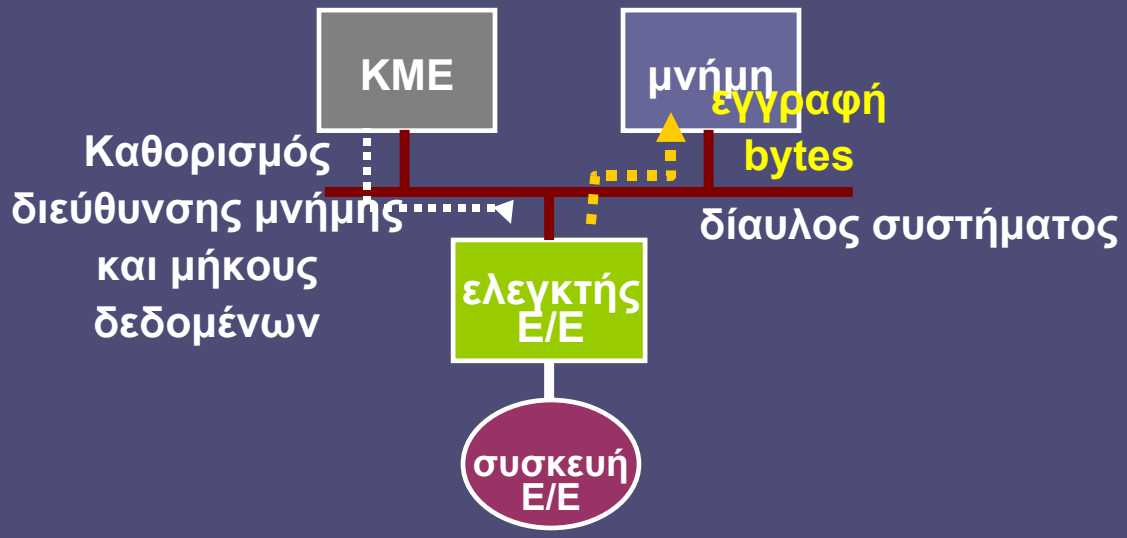


- Μετά από διακοπή ή polling
- Η ΚΜΕ επαναληπτικά διαβάζει δεδομένα από τον ελεγκτή Ε/Ε και τα γράφει στη μνήμη – και το αντίστροφο
- Σημαντική επιβάρυνση της ΚΜΕ όσο
  - το μέγεθος των δεδομένων κάθε αίτησης Ε/Ε αυξάνονται
  - ο ρυθμός ολοκλήρωσης αιτήσεων Ε/Ε αυξάνεται

Direct Memory Access (DMA)

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ

Όταν το σύστημα χρησιμοποιεί εικονική μνήμη, θα πρέπει να ληφθούν τα κατάλληλα μέτρα ώστε κατά τη μεταφορά DMA τα δεδομένα να τοποθετηθούν στη σωστή φυσική διεύθυνση

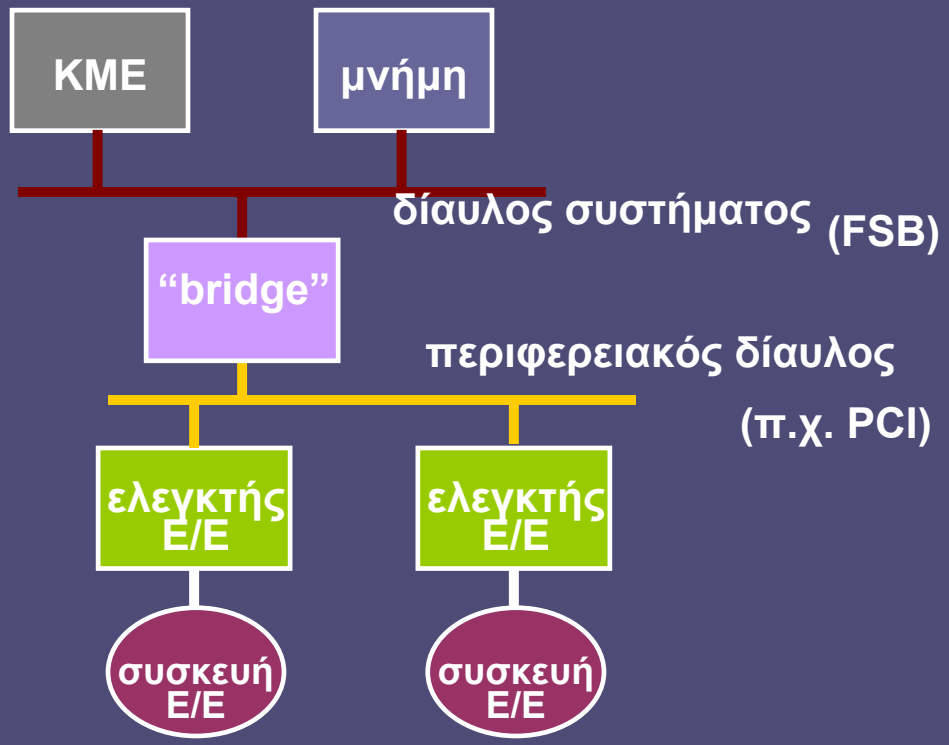


- Η ΚΜΕ απλά θέτει τις παραμέτρους της μεταφοράς
- Ο ελεγκτής διεκδικεί τον δίαυλο του συστήματος και μεταφέρει δεδομένα απευθείας προς/από τη μνήμη
  - Ή ξεχωριστός ελεγκτής DMA
  - Η ΚΜΕ μπορεί να εκτελεί άλλη διεργασία
  - Σύγκρουση στη μνήμη, αλλά ή ΚΜΕ έχει και την κρυφή μνήμη!

Η εμφάνιση του περιφερειακού διαύλου

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ
- Δίαυλοι Ε/Ε

Εμφάνιση της ιεραρχίας διασυνδετικών διαύλων, με διαφορετικούς ρυθμούς μεταφοράς

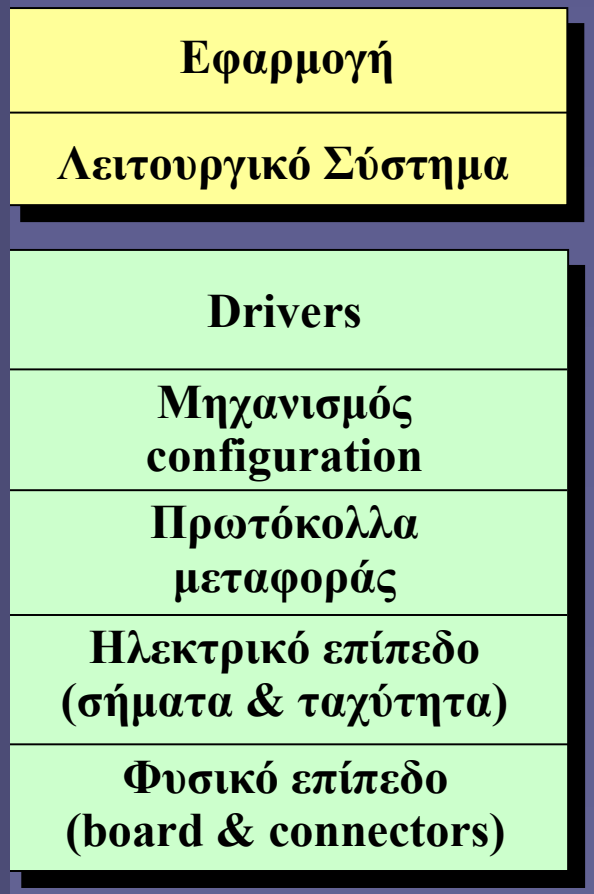


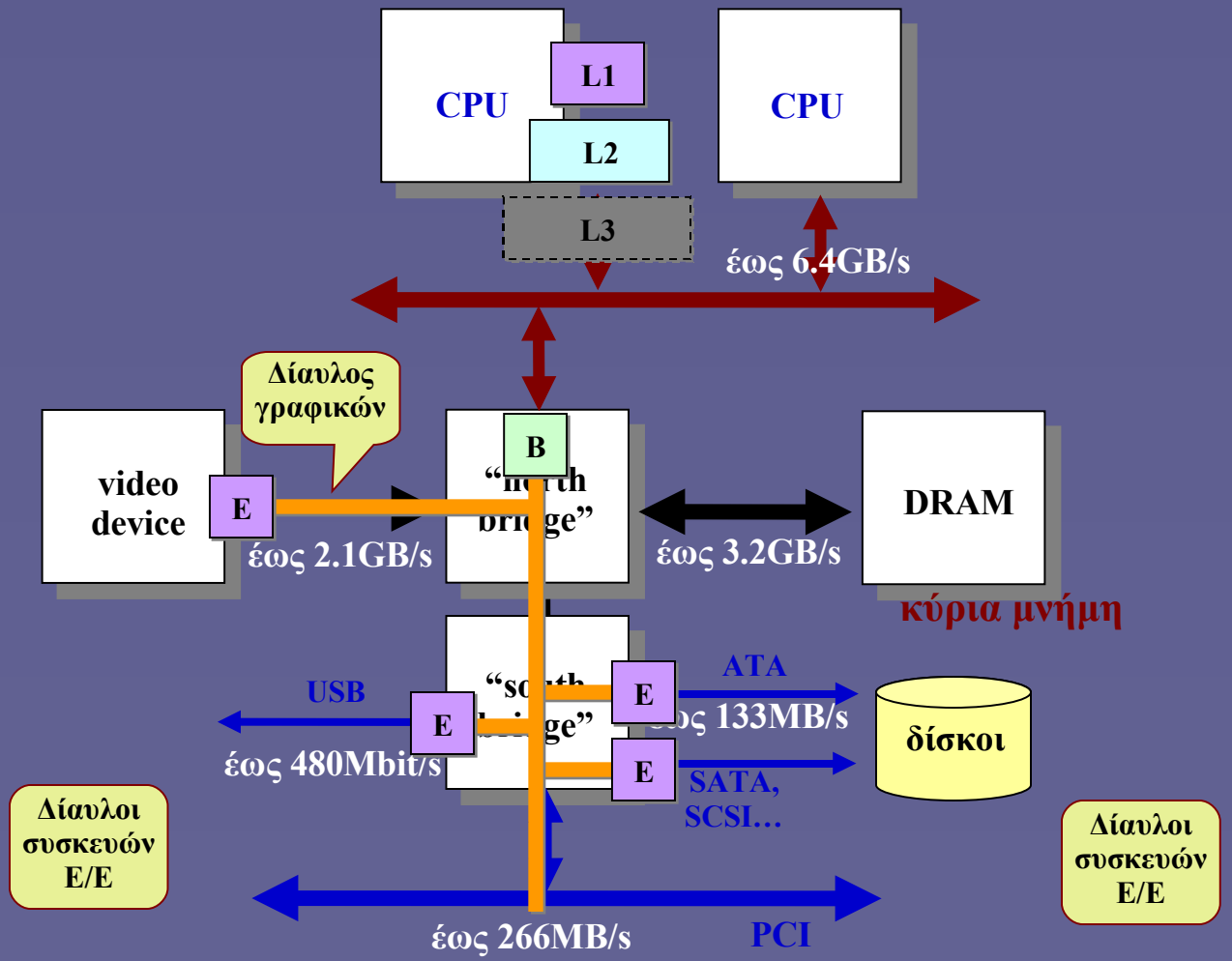
- Οι συσκευές Ε/Ε (περιφερειακές συσκευές) συνδέονται σε περιφερειακό δίαυλο
- Η ΚΜΕ "βλέπει" μόνο μια "συσκευή": peripheral bus bridge
  - Μέσω της οποίας γίνεται η προσπέλαση των συσκευών Ε/Ε

Περιφερειακοί δίαυλοι

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ
- Δίαυλοι Ε/Ε

- Ένας περιφερειακός δίαυλος ορίζεται από συνδυασμό επιπέδων υλικού-λογισμικού
- Δυναμική αρχικοποίηση συσκευών (configuration)
  - Ευέλικτη διασύνδεση
- Οδήγηση συσκευών (drivers)
  - Επεκτασιμότητα
- Ιεραρχική δομή
  - Υπο-δίαυλοι
  - Χρήση bridges



Η φυσική υλοποίηση σήμερα

- Συσκευές Ε/Ε
- Διασύνδεση Ε/Ε
- Εξυπηρέτηση Ε/Ε από ΚΜΕ
- Δίαυλοι Ε/Ε