

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Αρχιτεκτονική Υπολογιστών
2020-21

Αρχιτεκτονικές Συνόλου Εντολών

(Instruction Set Architectures - ISA)

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

Μ.Στεφανιδάκης

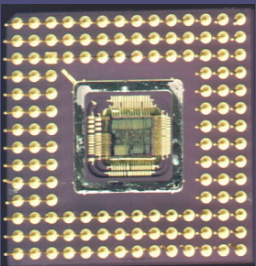


Ο (μικρο)επεξεργαστής

- Ο επεξεργαστής



Τι περιέχεται στη συσκευασία ενός μικροεπεξεργαστή σήμερα;

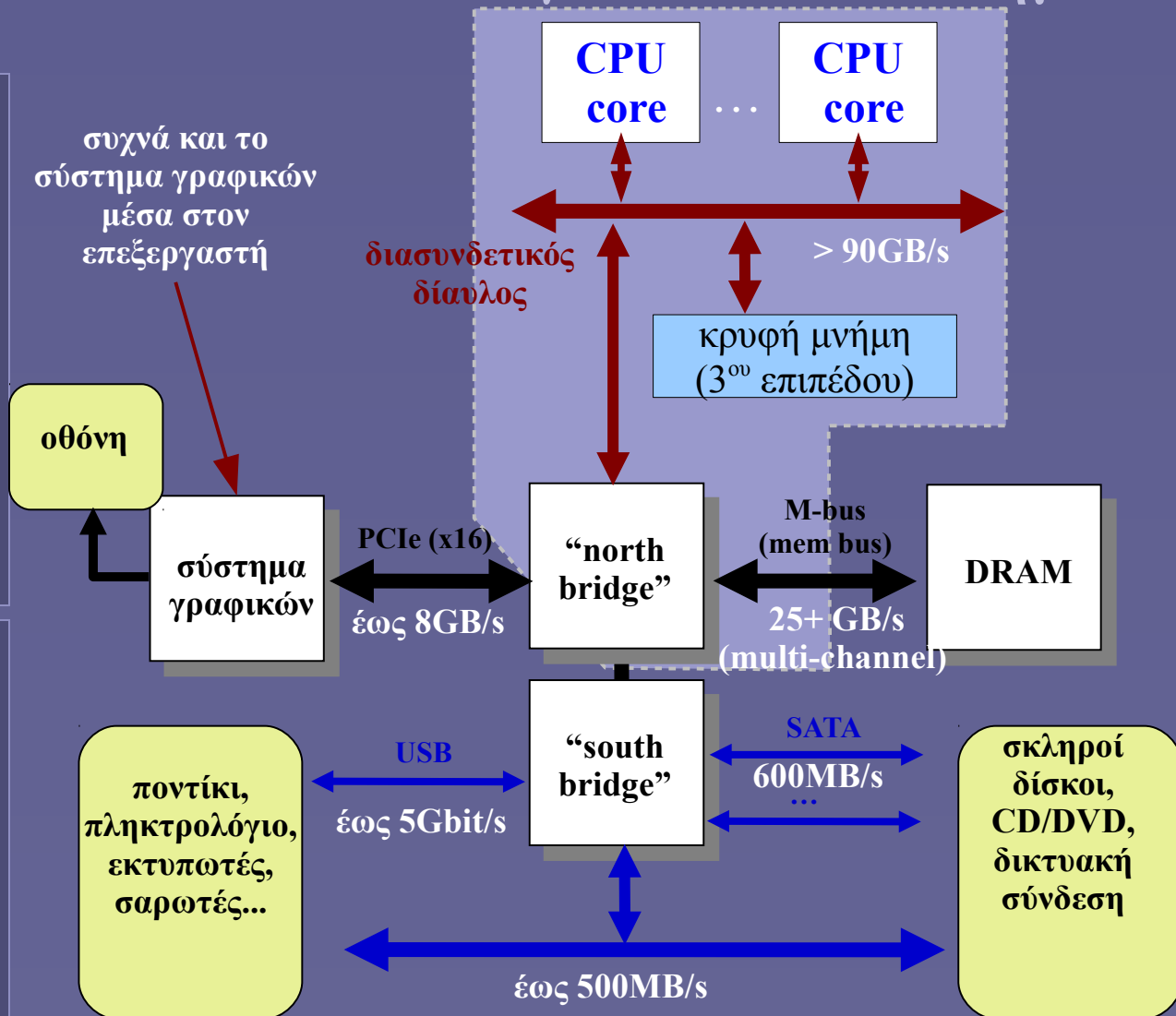


- (Micro)processor
 - Μέρος ενός ευρύτερου υπολογιστικού συστήματος (ή «υπολογιστή»)
 - Αρχικά: μόνο η Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)
 - Central Processing Unit (CPU)
 - Περιέχει σήμερα πολλαπλές υπομονάδες επεξεργασίας
 - Τμήμα της ιεραρχίας μνήμης
 - Τμήμα των μονάδων ελέγχου συσκευών εισόδου - εξόδου

Ένα τυπικό υπολογιστικό σύστημα

- Ο επεξεργαστής

συχνά και το
σύστημα γραφικών
μέσα στον
επεξεργαστή



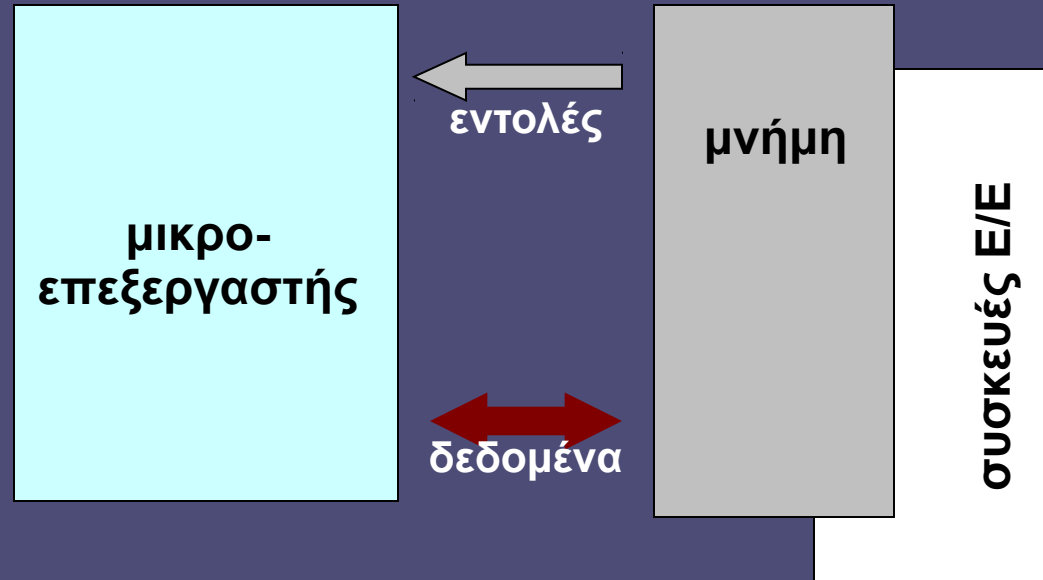
- Επεξεργαστής
- Μνήμη
- Συσκευές E/E

Είδη μικροεπεξεργαστών

- Ο επεξεργαστής
 - Γενικού σκοπού
 - Οι περισσότεροι γνωστοί επεξεργαστές
 - Συνεπεξεργαστές (co-processors)
 - Ειδικές λειτουργίες
 - Μικροελεγκτές (microcontrollers)
 - Συστήματα ελέγχου
 - Μέσα σε Systems-on-Chip
 - Μαζί με κυκλώματα σχετικά με τη λειτουργία του συστήματος
 - π.χ. με τον «ασύρματο» ενός κινητού

Το μοντέλο von Neumann

- Ο επεξεργαστής



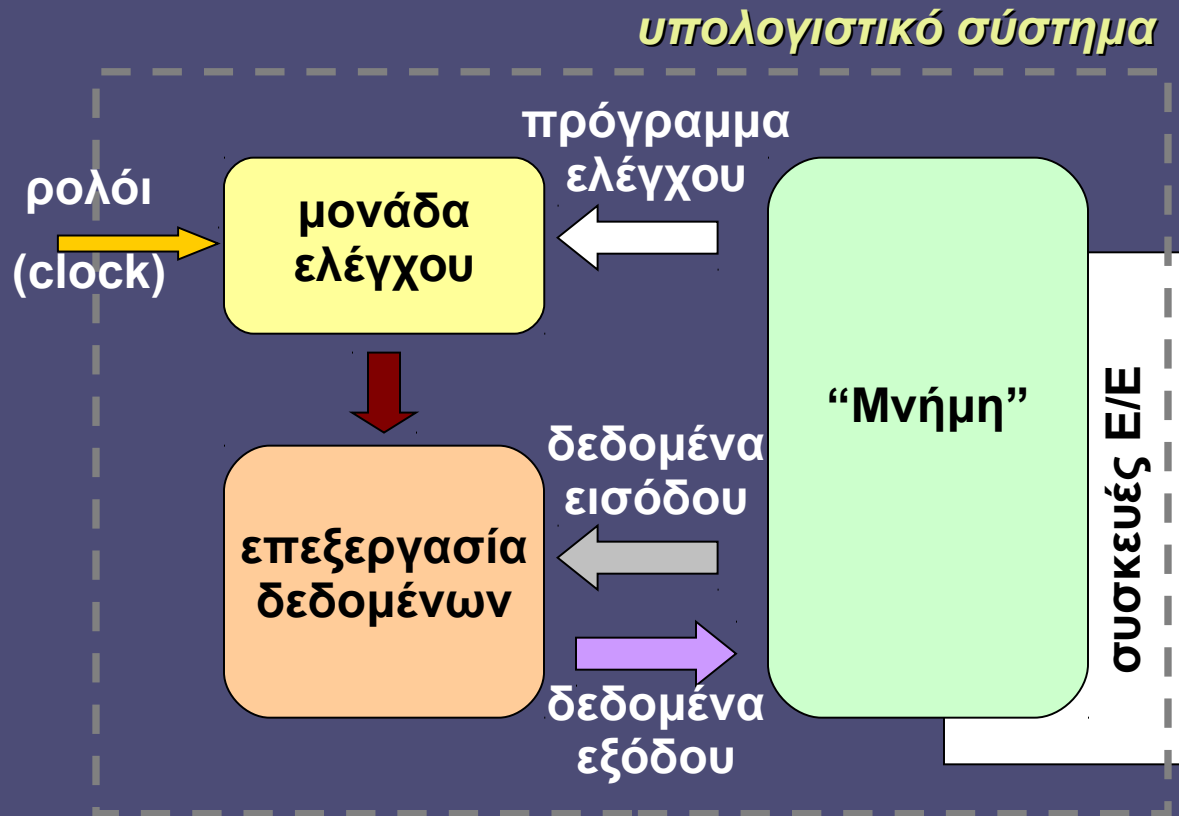
- Το **πρόγραμμα εκτέλεσης**, όπως και τα **δεδομένα**, αποθηκεύονται στη **μνήμη** του υπολογιστή
 - “Stored-program computer”

;

Σε ποια μορφή αποθηκεύονται οι εντολές;

Εκτέλεση ακολουθίας λειτουργιών

- Ο επεξεργαστής



;

Ποιος ο ρόλος του
σήματος ρολογιού;

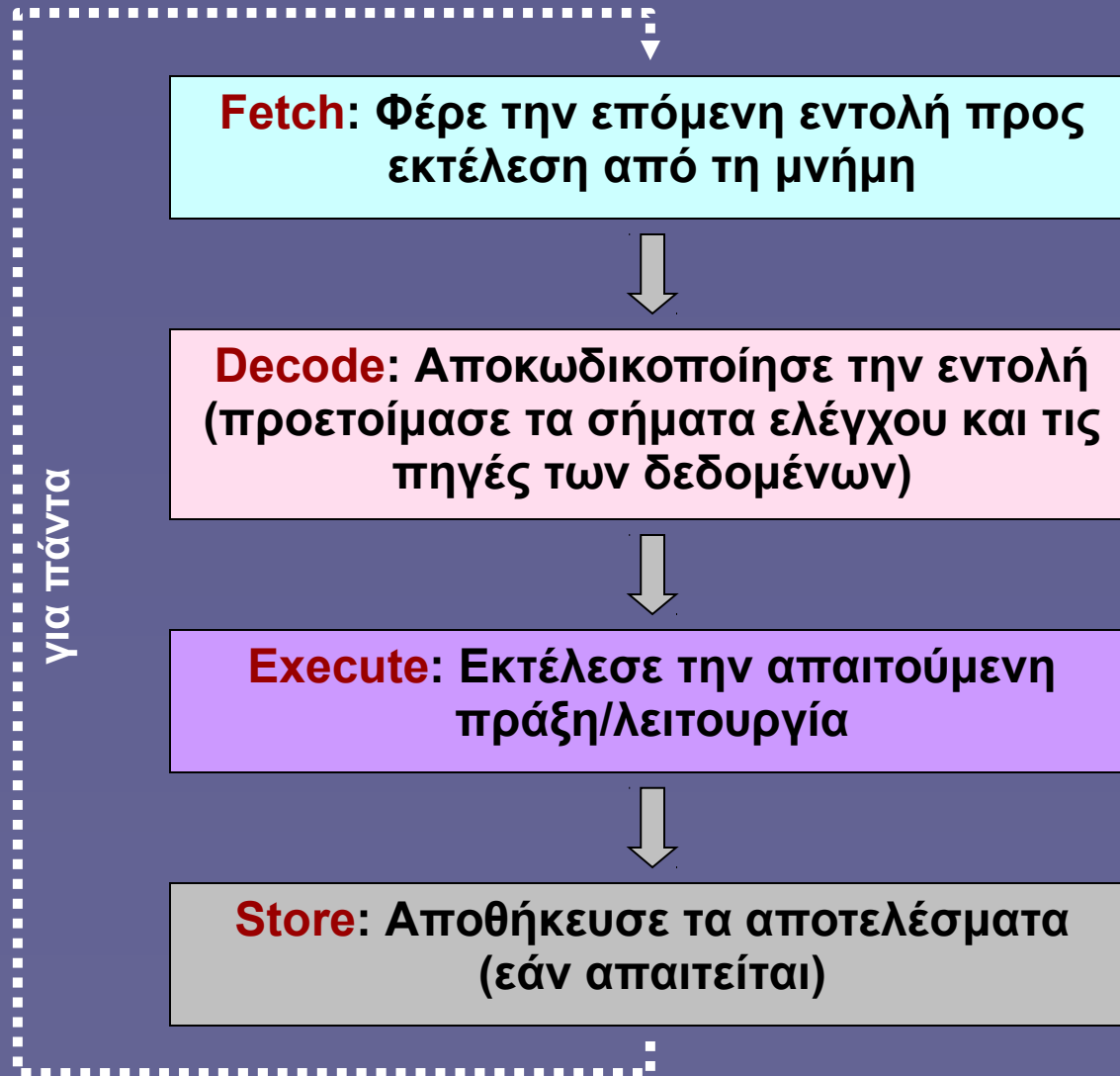
Εκτέλεση εντολών: ο κύκλος μηχανής

- Ο επεξεργαστής

;

Τι συμβαίνει με τις εξωτερικές διακοπές (interrupts);

Ο χρόνος εκτέλεσης είναι ο ίδιος για όλες τις εντολές;



Εκτέλεση εντολών

- Ο επεξεργαστής

- Επόμενη εντολή προς εκτέλεση
 - **Program Counter (PC)**: καταχωρητής που περιέχει τη διεύθυνση της θέσης μνήμης όπου βρίσκεται 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
- Τερματισμός επανάληψης κύκλου μηχανής
 - Συμβατικά, ποτέ

Αρχιτεκτονική Συνόλου Εντολών

- Ο επεξεργαστής
- ISA

- **Instruction Set Architecture (ISA)**
 - Το ορατό μέρος ενός υπολογιστικού συστήματος για τον προγραμματιστή (και τον 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
 - Operands
 - Μέθοδοι προσπέλασης μνήμης
 - Προέλευση των δεδομένων
 - Χώροι προσωρινής αποθήκευσης
 - Καταχωρητές
 - Διακοπές και καταστάσεις σφάλματος
 - Ποια η “αντίδραση” του επεξεργαστή

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 συναρτήσεων
- Άλλες κατηγορίες
 - Ειδικές εντολές συστήματος
 - ΛΣ, εικονική μνήμη
 - Απαιτούνται αυξημένα προνόμια εκτέλεσης
 - Επεξεργασία πολλαπλών δεδομένων ταυτόχρονα («παράλληλα»)
 - Χρήσιμο για γραφικά, σειρές χαρακτήρων, multimedia

Κωδικοποίηση Εντολών

- Ο επεξεργαστής
- ISA



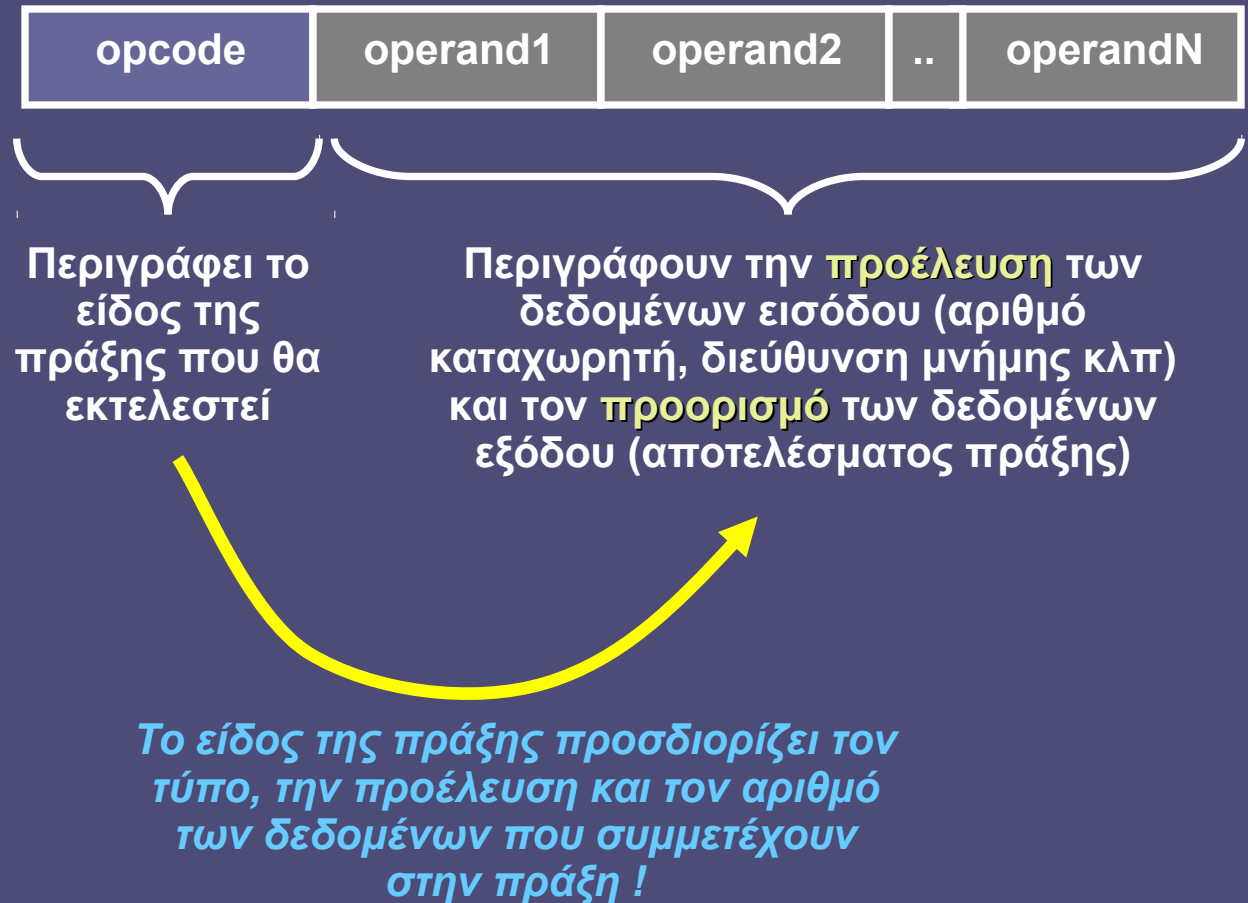
- Σειρά δυαδικών ψηφίων
 - Μεταβλητού μήκους
 - Περισσότερο συμπαγή προγράμματα
 - Πολυπλοκότερο υλικό!
 - Σταθερού μήκους
 - Απλούστερη και ταχύτερη λήψη-αποκωδικοποίηση
 - Μεγαλύτερα προγράμματα
 - Μέθοδοι συμπίεσης

;

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

Κωδικοποίηση Εντολών

- Ο επεξεργαστής
- ISA



Αριθμητικές/λογικές εντολές

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

;

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

- Αριθμητικές-λογικές πράξεις
 - Είδος πράξης
 - Είδος δεδομένων
 - Πηγές δεδομένων και προορισμός
 - Παράδειγμα:
 - `add R1, R2, R3 // R3 = R1+R2`

add	R1	R2	R3
-----	----	----	----

Εντολές μεταφοράς δεδομένων

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

;

Η απόλυτη διεύθυνση (π.χ. 0x7FF0) καλύπτει όλες τις επιθυμητές περιπτώσεις χρήσης;

- Μεταφορά δεδομένων

- Πηγή δεδομένων και προορισμός
- Μήκος μεταφερόμενης λέξης
- Παράδειγμα:
 - `load R1, 0x7FF0 // R1 = mem[0x7FF0]`



Εντολές διακλάδωσης

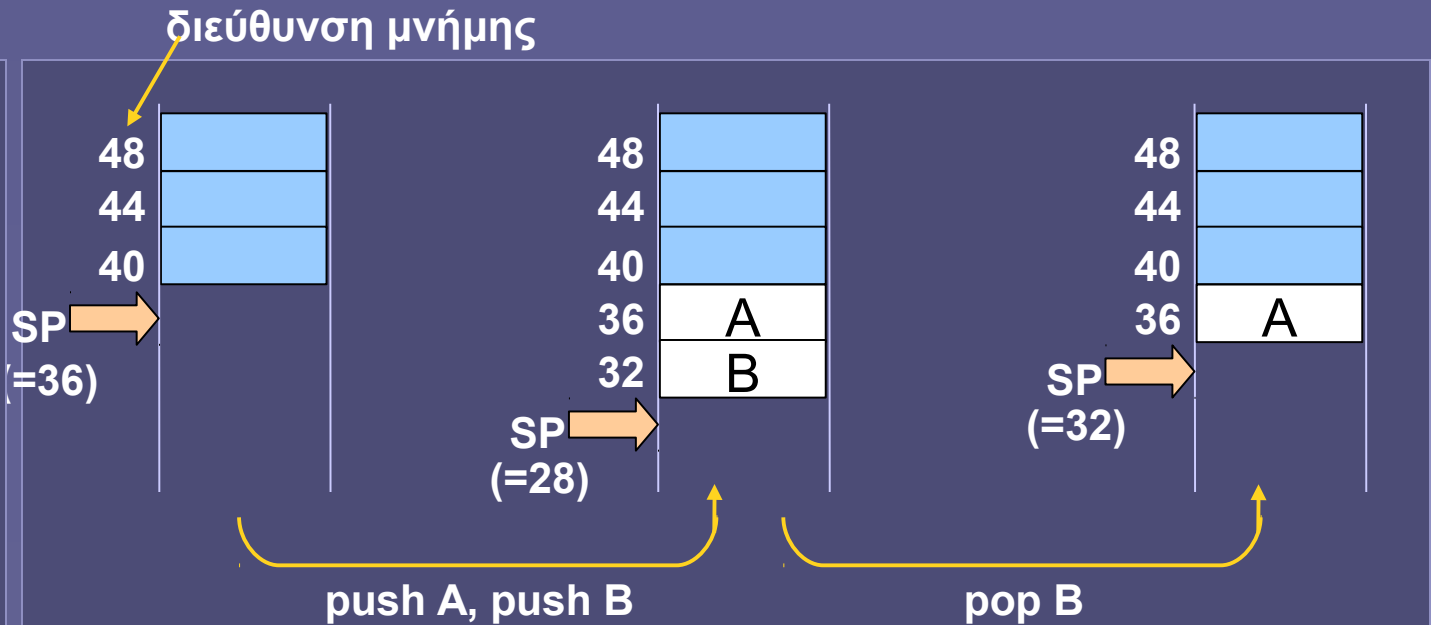
- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

- 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

bne	R1	R2	+8
-----	----	----	----

Η στοίβα (stack)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών



- Η ΚΜΕ παρέχει ειδικό καταχωρητή (stack pointer – SP) και εντολές **push** και **pop**
- Η στοίβα δεν είναι ειδική μνήμη αλλά ένα εναλλακτικό μοντέλο προσπέλασης της κανονικής μνήμης
 - Αποθήκευση διεύθυνσης επιστροφής από συναρτήσεις
 - Αποθήκευση τοπικών μεταβλητών δομημένων γλωσσών

Εντολές διακλάδωσης (2)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών

- Κλήση συνάρτησης (call)
 - Αποθήκευση της επόμενης διεύθυνσης εκτέλεσης (καταχωρητή PC) στη στοίβα (push)
 - Μετάβαση στη διεύθυνση της συνάρτησης
- και επιστροφή (return)
 - Χρήση αποθηκευμένης τιμής από στοίβα (pop)
 - Τοποθετείται στον καταχωρητή PC
 - Η εκτέλεση επιστρέφει στην επόμενη εντολή μετά το call

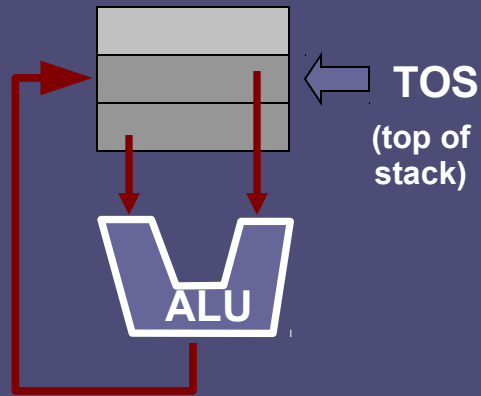
Προέλευση και αποθήκευση δεδομένων

- 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, συχνά ο μόνος καταχωρητής του συστήματος)
 - Μεταγενέστεροι υπολογιστές
 - Δεδομένα από καταχωρητές – μνήμη
 - Δεδομένα από καταχωρητές μόνο (αρχιτεκτονικές load-store)

Αρχιτεκτονική σωρού (stack)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων

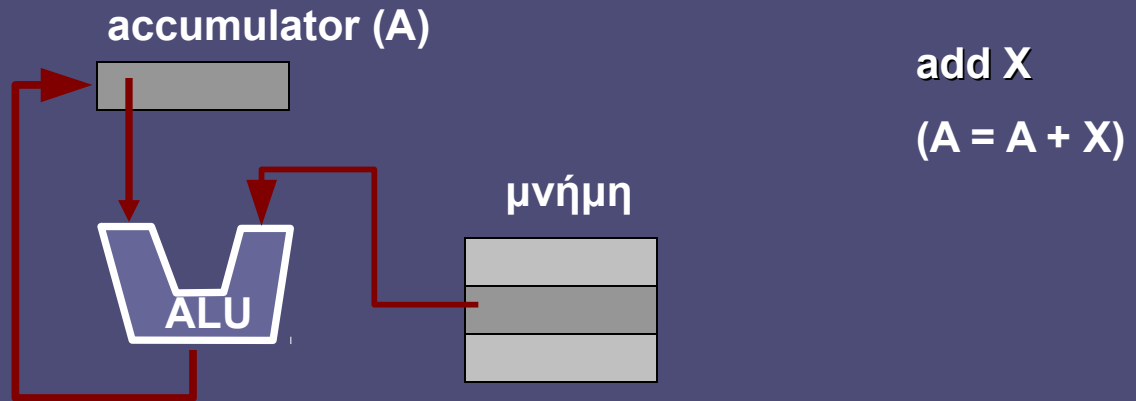


push A
push B
add
pop A

- Οι πηγές προσδιορίζονται έμμεσα
 - Δεν περιγράφονται στην εντολή!
 - 0-address architecture
 - Δημοφιλές σχήμα κατά τη δεκαετία του 60
 - Δύσκολη προσπέλαση σωρού, απαιτούνται πολλαπλές αντιμεταθέσεις και αντιγραφές για να έρθουν τα δεδομένα στη σωστή θέση

Αρχιτεκτονική συσσωρευτή (accumulator)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων



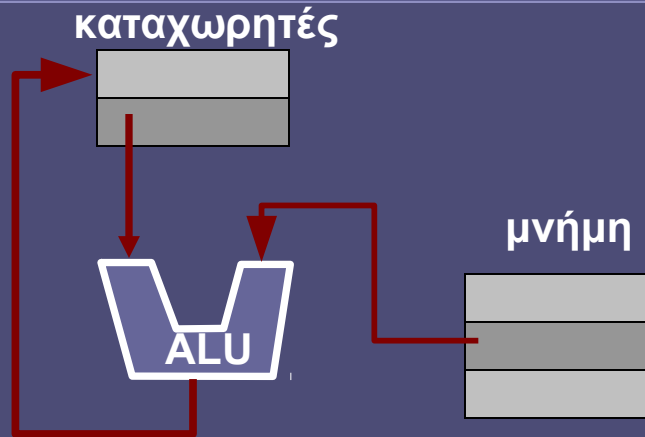
- Μια πηγή δεδομένων **και ταυτόχρονα** θέση αποθήκευσης του αποτελέσματος είναι πάντα ο συσσωρευτής
 - 1-address architecture
 - Αρχιτεκτονική των πρώτων υπολογιστών!

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**

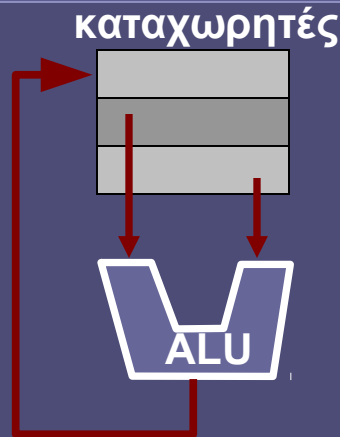
- Οποιαδήποτε εντολή μπορεί να προσπελάσει τη μνήμη

▪ Όμως:

- Πολλαπλές προσπελάσεις μνήμης
- Λήψη εντολής – Λήψη δεδομένων εντολής
- Πολύπλοκη εκτέλεση εντολής σε στάδια
- Συνωστισμός στον δίαυλο επικοινωνίας με μνήμη

Αρχιτεκτονικές με καταχωρητές (registers)

- Ο επεξεργαστής
- ISA
- Κατηγορίες εντολών
- Προέλευση δεδομένων



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

;

Είναι επιθυμητός ένας πολύ μεγάλος αριθμός καταχωρητών;

Μέθοδοι προσπέλασης μνήμης

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

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

Μέθοδοι προσπέλασης μνήμης

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

- Στο σχηματισμό της διεύθυνσης μνήμης μπορούν να συμμετέχουν:

- Απόλυτες τιμές διεύθυνσης
- Καταχωρητές
- Σταθερές τιμές μετατόπισης (offsets)

πιθανή
χρήση



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



Πώς
κωδικοποιούνται
οι μέθοδοι
προσπέλασης
μνήμης μέσα στην
εντολή;

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
 - Οι σημερινοί επεξεργαστές με εντολές CISC (π.χ. η αρχιτεκτονική x86), μεταφράζουν εσωτερικά σε εντολές RISC