242 - Εισαγωγή στους Η/Υ

Τμήμα Μαθηματικών, Πανεπιστήμιο Ιωαννίνων

Ακαδημαϊκό Έτος 2018-2019

Άρτια Α.Μ. (0-2-4-6-8)
Νικόλαος Γλυνός
207α, Β' όροφος
Τηλ: 8251
e-mail: nglinos@uoi.gr

Διδασκαλία και Τρόπος Εξέτασης

- Κάθε εβδομάδα
  - 3ωρη διάλεξη: Θεωρία + κατανόηση εννοιών
  - 2ωρο εργαστήριο: πρακτική εξάσκηση μέσα από συγκεκριμένα προβλήματα
- Εργαστήρια:
  - 8~ περίπου εβδομαδιαία εργαστήρια (εξέταση εργαστηριακών ασκήσεων)
  - Σκοπός των εργαστηρίων είναι η εμπέδωση της ύλης, η εξάσκηση και η λύση αποριών
  - Η παρακολούθηση είναι υποχρεωτική και είναι το *διαβατήριο* για να λάβετε μέρος στις τελικές εξετάσεις
  - Μπορείτε να λείψετε σε 2 **το πολύ** εργαστήρια για σοβαρό λόγο

Διδασκαλία και Τρόπος Εξέτασης

- Εργαστήριο Η/Υ: 1ο όροφο δίπλα από το «Αναγνωστήριο»,

- Στα Εργαστήρια θα πρέπει:
  - Να κατανοείτε το πρόβλημα και να προσπαθείτε να σχεδιάσετε την λύση **πριν έλθετε στο εργαστήριο**.
  - Να συμμετέχετε ενεργά και να είστε προετοιμασμένοι να απαντάτε σε ερωτήσεις σχετικά με το μάθημα

- Για να είναι <u>επιτυχής η παρακολούθηση</u> των εργαστηρίων θα πρέπει:
  1. να μην έχετε παραπάνω από δύο απουσίες
  2. να έχετε <u>προσπαθήσει</u> ΟΛΕΣ τις εργασίες που θα σας ζητούνται στα εργαστήρια

Χρήσιμο Υλικό

- Διδακτικό υλικό, Βιβλία:
- C Προγραμματισμός, 7η Έκδοση, Abbey Deitel,Harvey Deitel.
- C++ ΓΙΑ ΜΑΘΗΜΑΤΙΚΟΥΣ: ΜΙΑ ΕΙΣΑΓΩΓΗ ΓΙΑ ΣΠΟΥΔΑΣΤΕΣ ΚΑΙ ΚΑΘΗΓΗΤΕΣ, EDWARD SCHEINERMAN.
- ΜΑΘΕΤΕ ΤΗ C++ ΑΠΟ ΤΟ ΜΗΔΕΝ, HERBERT SCHILDT.
- C++ Προγραμματισμός, 9η Εκδ., Paul J. Deitel, Harvey M. Deitel.
- C++, 9η Έκδοση, Savitch Walter.
- Πλήρης C++, Savitch Walter.
- Διαφάνειες και εργαστηριακές ασκήσεις:
  ιστότοπος μαθήματος → users.uoi.gr/nglinos

Βαθμολογία

- 70% έχει βάρος η τελική εξέταση
- 30% έχει βάρος η βαθμολογία εργαστηρίων (test/εξέταση εργαστήριο)

- Προϋπόθεση: επιτυχή παρακολούθηση εργαστηρίων

- τελικός βαθμός = 0.7 (τ.ε) + 0.3 (εργ)

Τμήματα Εργαστηρίων

- Τα εργαστήρια θα αρχίσουν την

  <u>Τετάρτη 13/3/19</u>

  ****Μην ξεχάσετε να δηλώσετε το μάθημα!!!*

Τμήματα Εργαστηρίων (με βάση τον Α.Μ. - **ΑΡΤΙΟΙ**):

Γίνονται Τετάρτη και Παρασκευή. Ενημερωθείτε για τις ώρες από την σελίδα του μαθήματος users.uoi.gr/nglinos

- Αλλαγές ΔΕΝ επιτρέπονται.
- Δύο το πολύ απουσίες.
- Παλαιότεροι φοιτητές: Πρέπει να συμμετάσχουν στην εξέταση του εργαστηρίου για να έχουν και τον αντίστοιχο βαθμό.

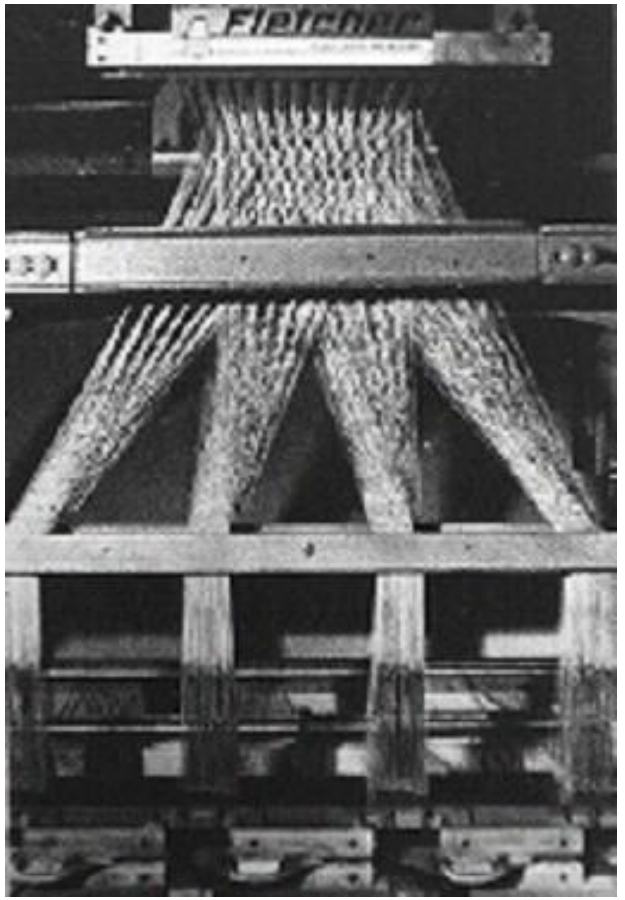
Δομή και οργάνωση Ηλεκτρονικού Υπολογιστή

Κατασκευή υπολογιστικών μηχανών

- Αρχαιότητα: υπολογιστικές μηχανές, μηχανισμός των Αντικυθήρων, κ.λπ.
  - είδος αστρολάβου που κατασκευάσθηκε περίπου το 65 π.Χ.
  - ο «μεσολάβος» (Ερατοσθένης, 210 π.Χ.)
  - η «διόπτρα» (Ήρων ο Αλεξανδρεύς, 100 π.Χ.)
- 17ος αιώνας, Pascal και Leibniz, μηχανικές υπολογιστικές αριθμομηχανές ⇒ στοιχειώδεις αριθμητικές πράξεις
  - βασισμένη σε μετρητές-τροχούς

Κατασκευή υπολογιστικών μηχανών

- 1804, Η μηχανή του Jacquard
- Αυτόματος αργαλειός που προγραμματιζόταν για την ύφανση πολύπλοκων σχεδίων μέσω διάτρητων χαρτοταινιών ή καρτελών
- Καθώς περιστρεφόταν μια ζώνη με τρύπες, κινούνταν βελόνες
- Το αν υπήρχε μία τρύπα ή όχι στην ανάλογη θέση μιας κάρτας καθόριζε τι θα υφανθεί



Κατασκευή υπολογιστικών μηχανών

- 1830–1840, Babbage, “διαφορική μηχανή” “αναλυτική μηχανή” ⇒ πολυωνυμικές συναρτήσεις, λογάριθμοι, τριγωνομετρικές συναρτήσεις
  - Πολύπλοκες μηχανικές κατασκευές που δεν ολοκληρώθηκαν
  - Η «αναλυτική μηχανή» θα μπορούσε να προγραμματιστεί χρησιμοποιώντας Διάτρητες κάρτες

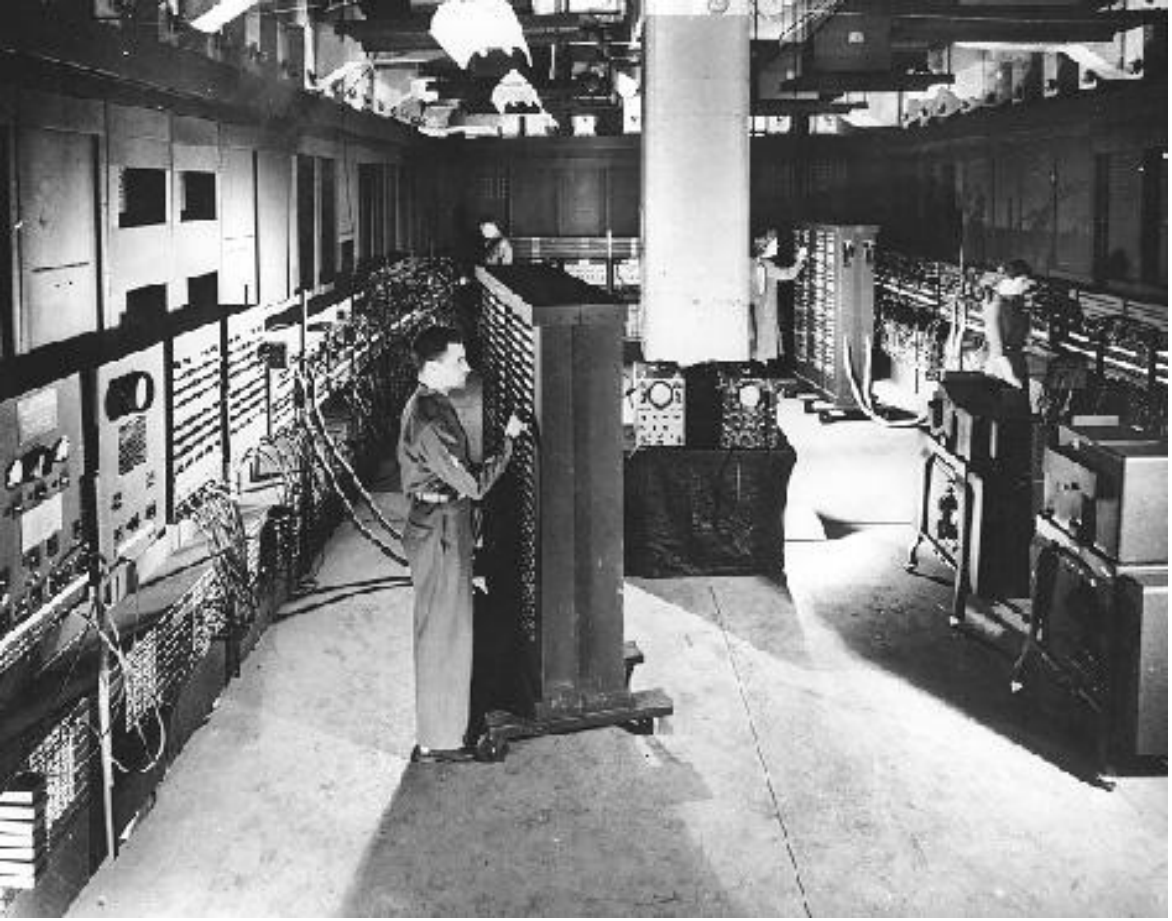
Κατασκευή υπολογιστικών μηχανών

- 1880–1890, Hollerith, μηχανή με διάτρητες κάρτες για την αυτοματοποίηση των εκλογών
- 1920 -1930, Bush, ηλεκτρική (αναλογική) υπολογιστική μηχανή
  ⇒ διαφορικές εξισώσεις

- 1940, Zuse, ηλεκτρονική (ψηφιακή) υπολογιστική μηχανή
  ⇒ πρόγραμμα και δεδομένα, χωριστά

- 1945–1950, μοντέλο von Neumann
  ⇒ πρόγραμμα και δεδομένα, από κοινού

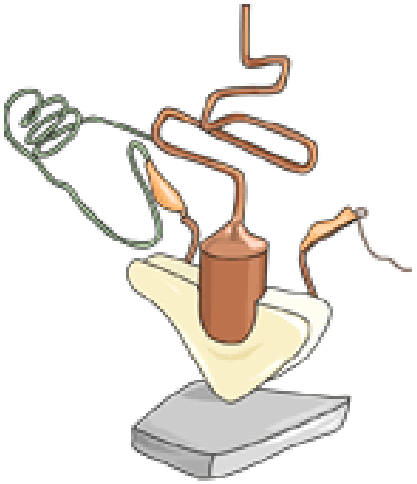
- 1950–σήμερα, ραγδαία ανάπτυξη της τεχνολογίας των ηλεκτρονικών υπολογιστών



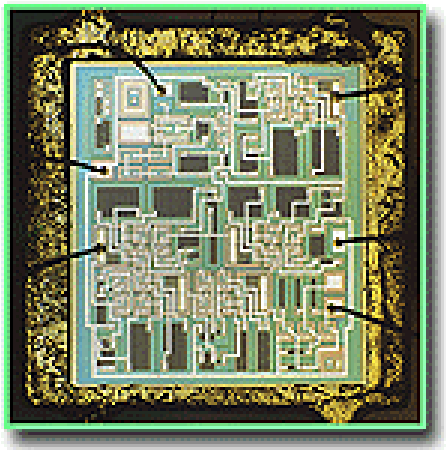
ENIAC (1940’s)

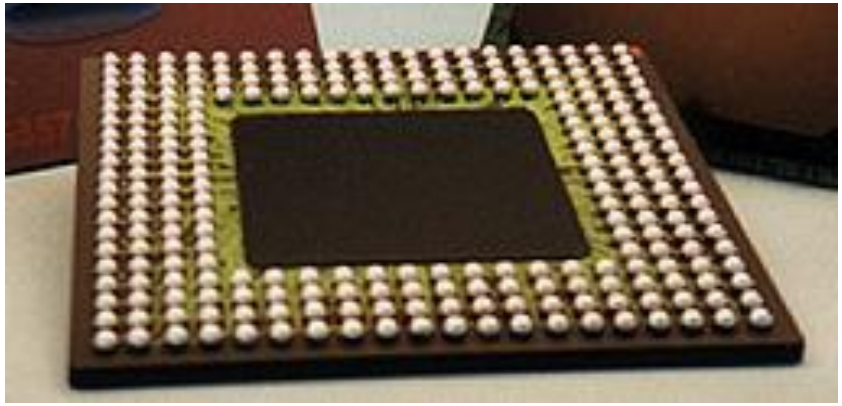
- Ο πρώτος Η/Υ!
- University of Pennsylvania
- Μέγεθος; Βλέπετε και μόνοι σας!
- 18,000 λυχνίες κενού
- 1500 ρελέ
- Βάρος; 30 τόνοι!
- Σχεδιαστές:
  - John Mauchly
  - J. Presper Eckert

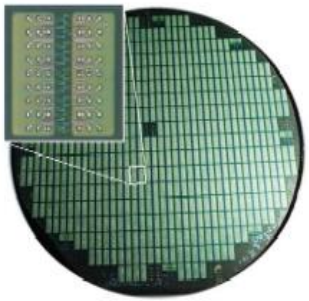
DEC LSI-11, Δεκαετία του 1980’s



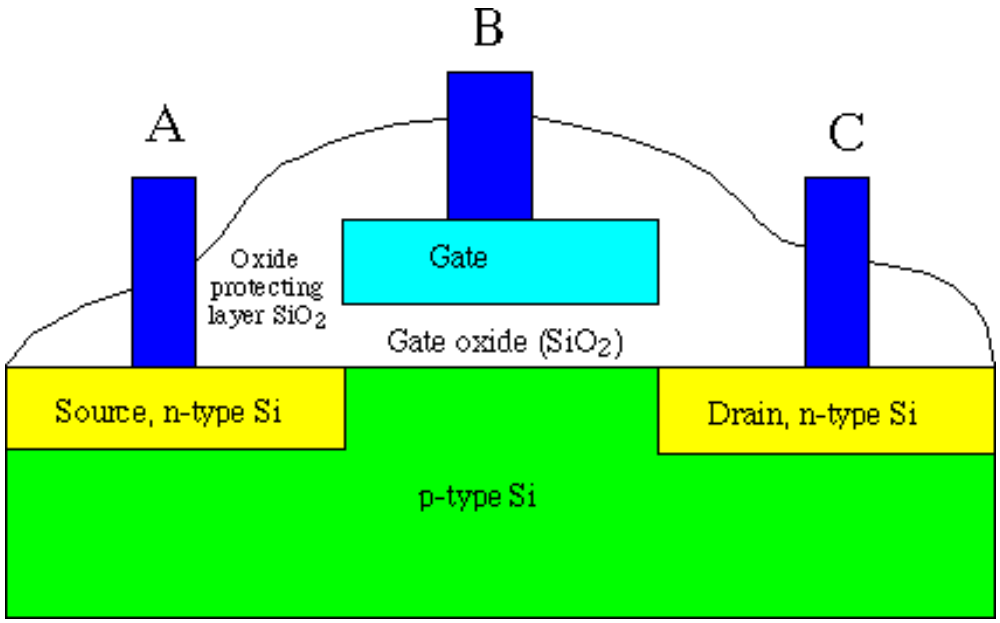

Η μεγάλη εφεύρεση!



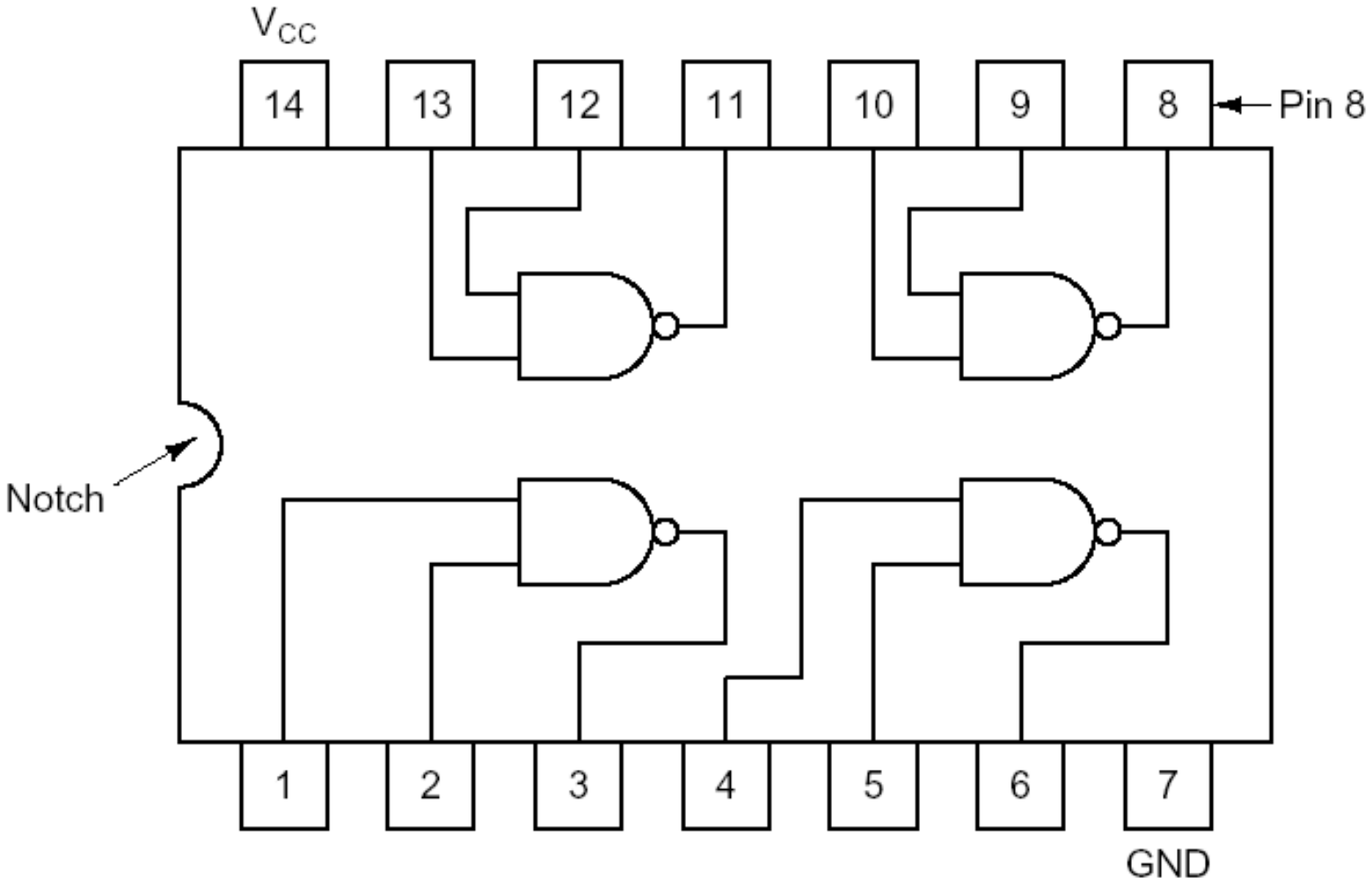




Σε επίπεδο ολοκληρωμένου κυκλώματος: ⇛



SSI- μικρής κλίμακας ολοκληρωμένο κύκλωμα



An SSI chip containing four gates.

- Οι μικρουπολογιστές περιέχουν ένα τεράστιο αριθμό τρανζίστορ:
  - **(ENIAC – χωρίς τρανζίστορ!):** 19,500 vacuum tubes and relays
  - **Intel 8088 processor (1st PC):** 29,000 transistors
  - **Intel Pentium II processor:** 7 million transistors
  - **Intel Pentium III processor:** 28 million transistors
  - **Intel Pentium 4 processor:** 42 million transistors
- Κάθε τρανζίστορ δεν είναι παρά ένας απλός διακόπτης on-off!
- Τα τρανζίστορ συνδυάζονται ώστε να υλοποιούν τις βασικές συναρτήσεις – πύλες (gates)
  - AND, OR, NOT
- Οι πύλες με τη σειρά τους συνδυάζονται για να διατελέσουν πιο πολύπλοκες λειτουργίες
  - adder, multiplexor, decoder, register, …

Άμεσα αποτελέσματα ανάπτυξης της τεχνολογίας των υπολογιστών

- Μείωση κατανάλωσης και όγκου μηχανών
- Αύξηση της ταχύτητας επεξεργασίας
- Αύξηση του μεγέθους της μνήμης
- Αύξηση της αποθηκευτικής ικανότητας
- Βελτίωση της πιστότητας φύλαξης δεδομένων

Έμμεσα Αποτελέσματα

- Αναπτύχθηκαν ορισμένες Επιστήμες

  (Αριθμητική Ανάλυση, Πληροφορική, Ρομποτική, …)

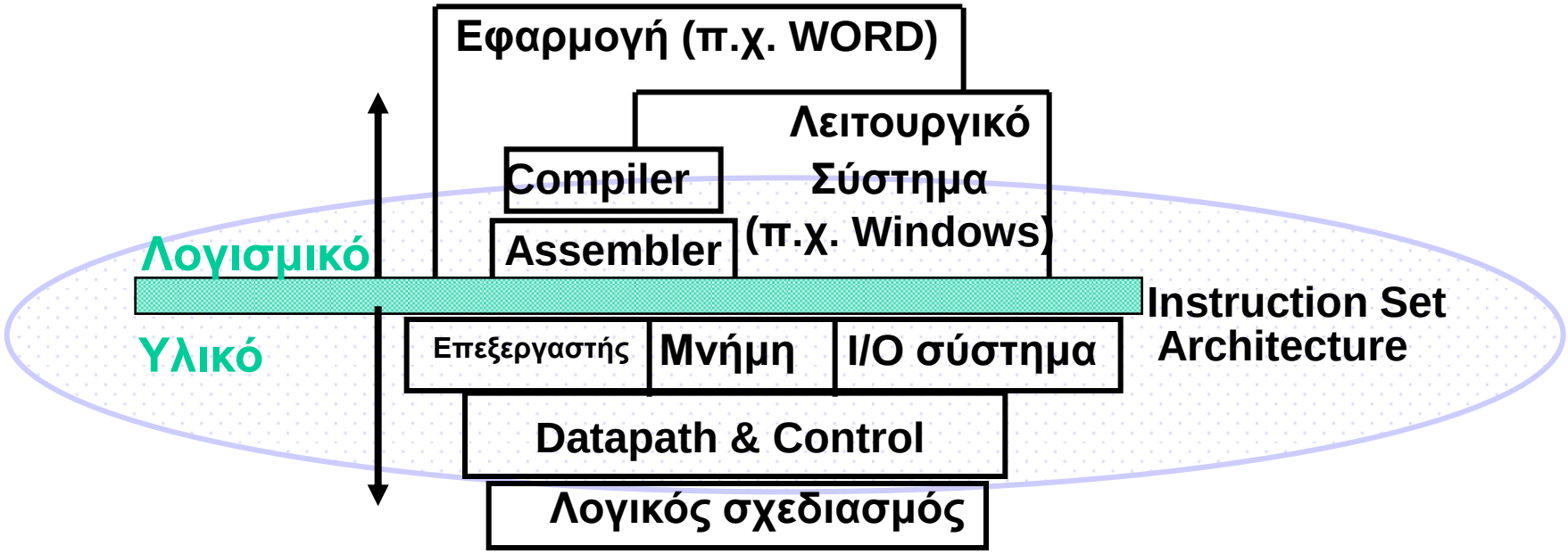
- Νέες Τεχνικές

  (Προσομοίωση, Monte Carlo, Μοριακή Δυναμική, …)

- Αναπτύχθηκε πλήθος εφαρμογών

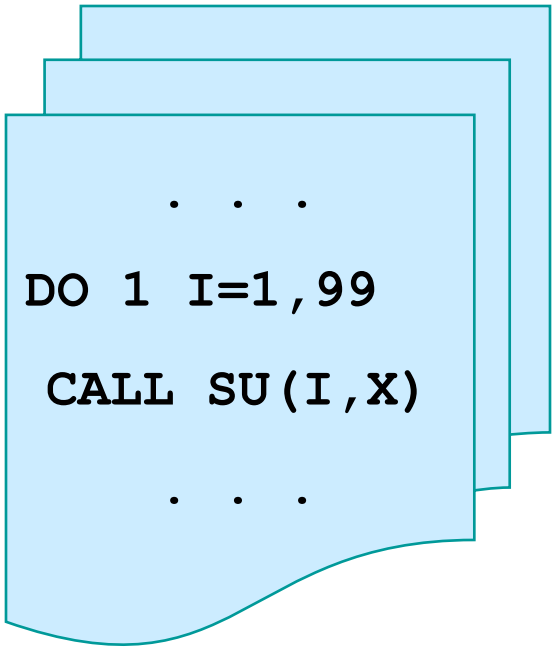
  (Τυπογραφία, Επικοινωνίες, Έλεγχος, Σχεδίαση, Διαχείριση πόρων, Εμπόριο, Κρυπτογραφία, Πρόγνωση καιρού, κλπ)

Οργάνωση ενός υπολογιστικού συστήματος



- Συντονισμός μεταξύ πολλών *αφαιρετικών επιπέδων*

ΛΟΓΙΣΜΙΚΟ

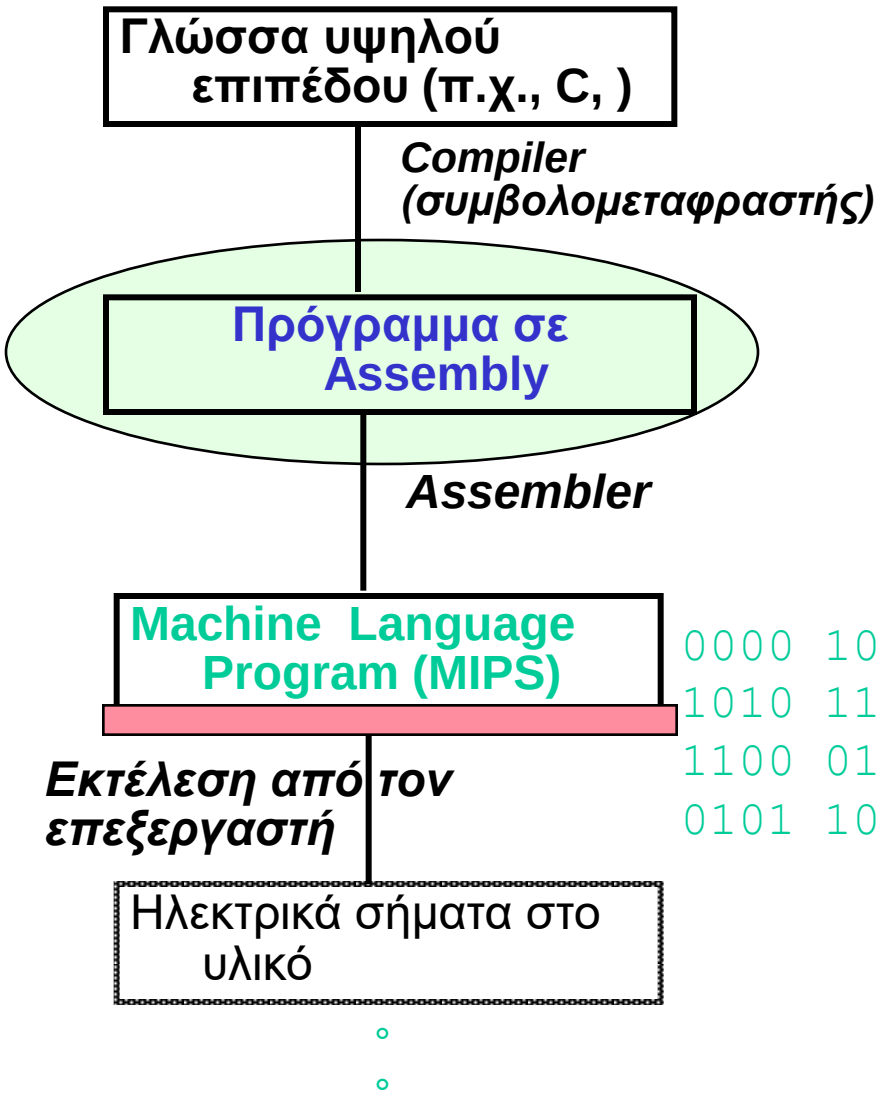


- Λειτουργικά συστήματα
- Γραφικοί μεσολαβητές χρήσης
- **Γλώσσες Προγραμματισμού**
- Μεταφραστές και διερμηνευτές
- Προγράμματα εφαρμογών
- Μεθοδολογία ανάπτυξης λογισμικού
- Λογισμικό αξιολόγησης του *"Υλικού"*

Γλώσσες Προγραμματισμού

- Διευκολύνουν την διατύπωση των διεργασιών που πρέπει να εκτελεστούν.
- Υπάρχουν πολλές γλώσσες
- Ορισμένες γλώσσες είναι καταλληλότερες από άλλες για συγκεκριμένο τύπο προβλήματος.
  - **FORTRAN** (*Επιστημονικοί Υπολογισμοί*)
  - **C** (*Προγραμματισμός Λειτουργικών*)
  - **PROLOG** (*Τεχνητή Νοημοσύνη*)

Επίπεδα προγραμματισμού



temp = v[k];

v[k] = v[k+1];

v[k+1] = temp;

```
lw      $to,  0($2)
lw      $t1,  4($2)
sw      $t1,  0($2)
sw      $t0,  4($2)
```

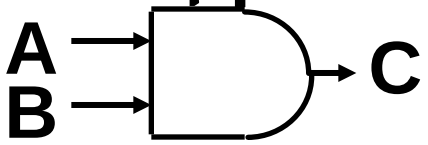
```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

Δομικά στοιχεία

Δομικά στοιχεία (για μια Αριθμητική Λογική Μονάδα – Arithmetic Logic Unit)

AND πύλη

Σύμβολο

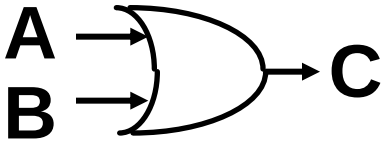


Ορισμός

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

OR πύλη

Σύμβολο

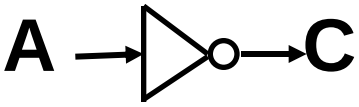


Ορισμός

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

Inverter

σύμβολο

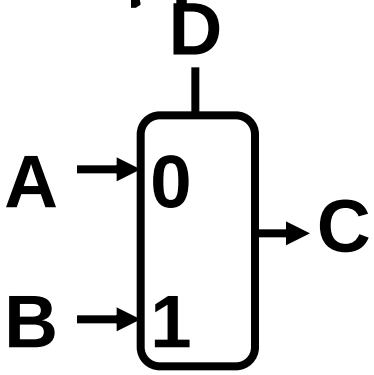


Ορισμός

A	C
0	1
1	0

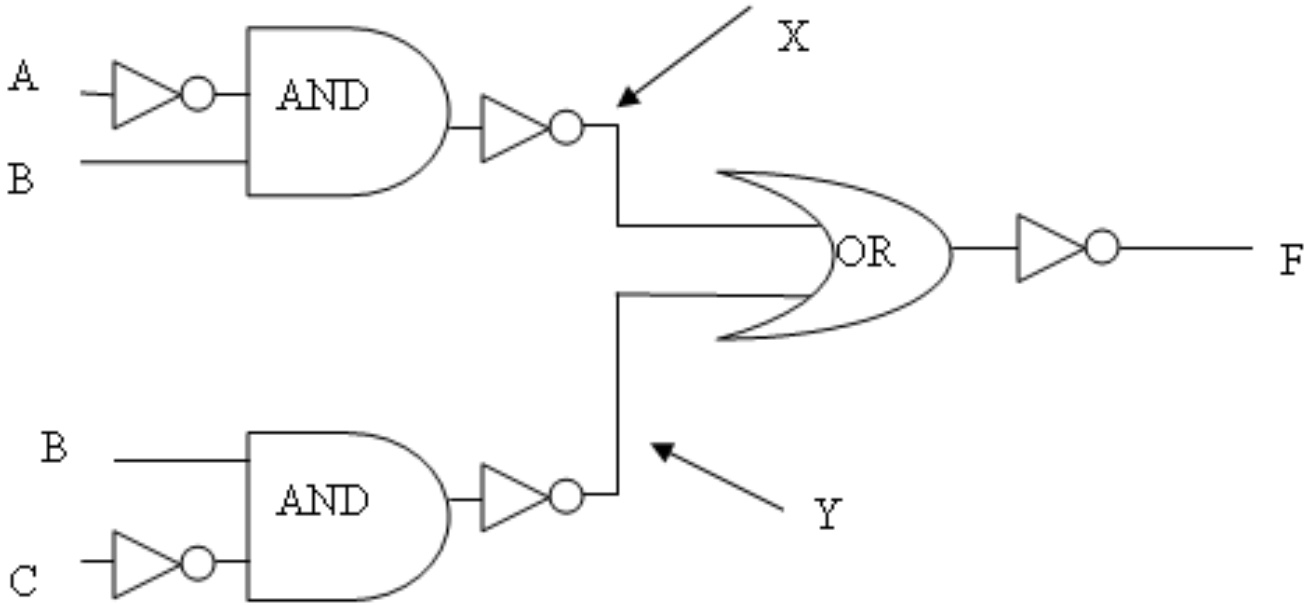
Multiplexer

Σύμβολο

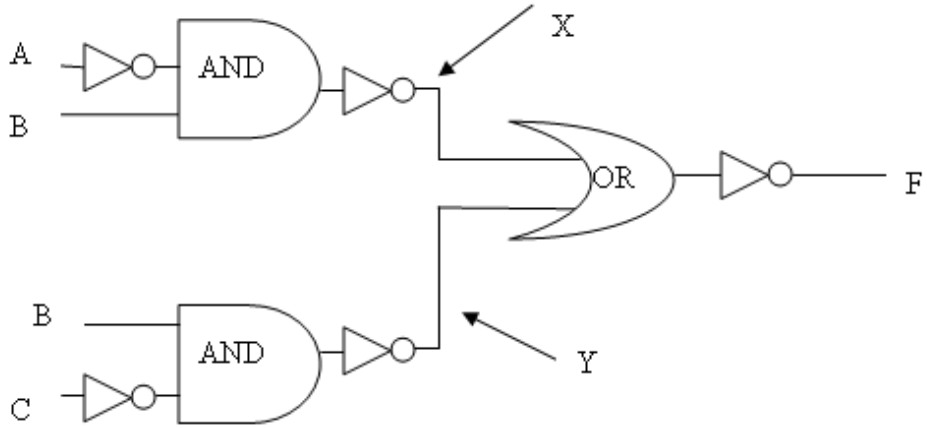


Ορισμός

D	C
0	A
1	B

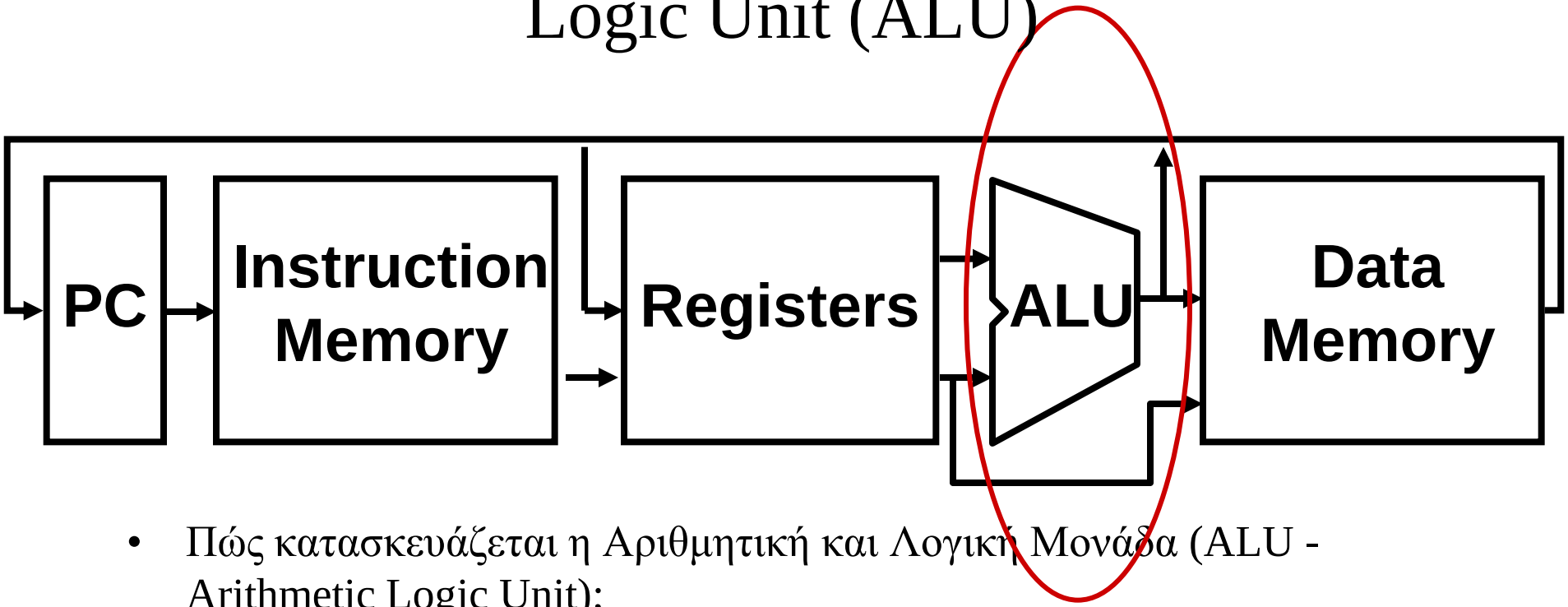


X = NOT ((NOT A) AND B) και
Y = NOT (B AND (NOT C))



A	**B**	**C**	NOT A	(NOT A) AND B	X	NOT C	B AND (NOT C)	Y	X OR Y	**F**
0	**0**	**0**	1	0	1	1	0	1	1	**0**
0	**0**	**1**	1	0	1	0	0	1	1	**0**
0	**1**	**0**	1	1	0	1	1	0	0	**1**
0	**1**	**1**	1	1	0	0	0	1	1	**0**
1	**0**	**0**	0	0	1	1	0	1	1	**0**
1	**0**	**1**	0	0	1	0	0	1	1	**0**
1	**1**	**0**	0	0	1	1	1	0	1	**0**
1	**1**	**1**	0	0	1	0	0	1	1	**0**

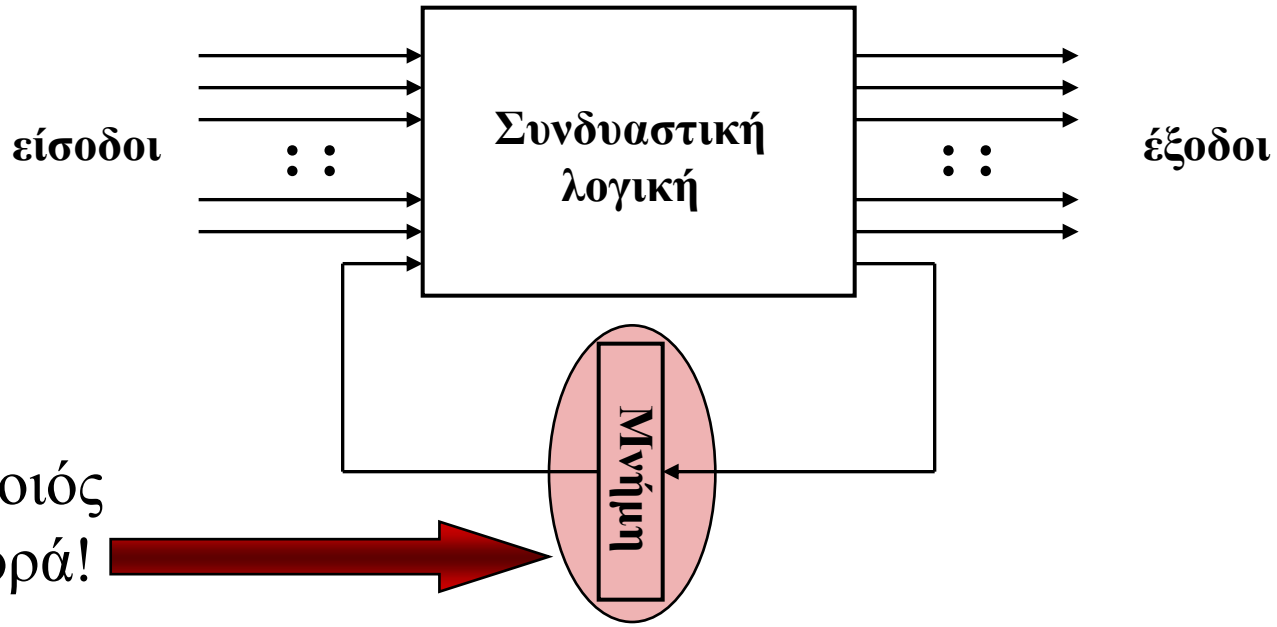
Αριθμητική και Λογική Μονάδα Arithmetic Logic Unit (ALU)



- Πώς κατασκευάζεται η Αριθμητική και Λογική Μονάδα (ALU - Arithmetic Logic Unit);
- PC – program counter - επόμενη εντολή για εκτέλεση
- Instruction memory – μνήμη με εντολές
- Registers - καταχωρητές
- Data memory – μνήμη δεδομένων

- Συνδυαστική λογική (combinational logic):

- Ακολουθιακή λογική (sequential logic):



Η ειδοποιός διαφορά!

Σχεδιαστικές τεχνικές

- Σχεδιαστικές τεχνικές για συνδυαστικά κυκλώματα:
  - Σε επίπεδο λογικών πυλών (π.χ. AND, OR, NOT, XOR, NAND)
  - Σε επίπεδο δομικών στοιχείων (building blocks)
- Και οι λογικές πύλες και τα δομικά στοιχεία είναι διαθέσιμα σε ολοκληρωμένα κυκλώματα (Integrated Circuits ή chips)

Δομικά στοιχεία: Ολοκληρωμένα κυκλώματα

- Είδη ολοκληρωμένων με βάση την πυκνότητα ολοκλήρωσης (ισοδύναμος αριθμός λογικών πυλών):
  - Small-scale integration (SSI): μέχρι 12 πύλες
  - Medium-scale integration (MSI): 12-99 πύλες
  - Large-scale integration (LSI): 100-9999 πύλες
  - Very large-scale integration (VLSI): 10,000-99,999 πύλες
  - Ultra large-scale integration (ULSI): > 100,000 πύλες
- Κύριοι στόχοι μιας ψηφιακής σχεδίασης:
  - (i) μείωση κόστους
    - Μείωση αριθμού πυλών (εάν χρησιμοποιούνται SSI ολοκληρωμένα)
    - Μείωση αριθμού ολοκληρωμένων (για πολύπλοκες σχεδιάσεις)
  - (ii) αύξηση ταχύτητας
  - (iii) απλότητα σχεδίασης (με επαναχρησιμοποίηση δομικών στοιχείων όπου αυτό είναι εφικτό)

Σχεδίαση σε επίπεδο πυλών: Ημιαθροιστής (Half Adder)

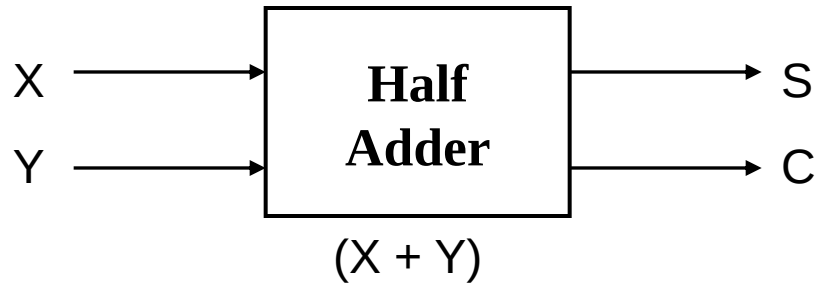
- Διαδικασία σχεδίασης:

  1) Ποιο είναι το πρόβλημα;

  Σχεδίαση Half Adder για πρόσθεση 2 bits

  2) Καθορισμός εισόδων, εξόδων και της σχέσης μεταξύ τους.

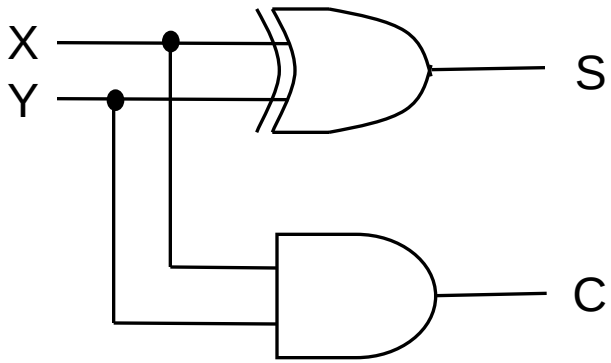
  Δύο είσοδοι και δύο έξοδοι (συναρτήσεις των εισόδων):



X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

C = XY

S = X'Y + XY' = X⊕Y



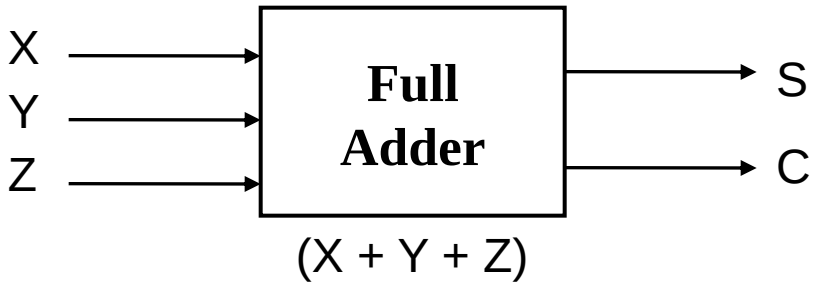
Half Adder

Σχεδίαση σε επίπεδο πυλών: Πλήρης Αθροιστής (Full Adder)

- Όμως! Ο ημιαθροιστής προσθέτει μόνο 2 bits!
- Για την πρόσθεση δύο δυαδικών αριθμών θα χρειαστούμε να προσθέσουμε 3 bits (το ένα από αυτά είναι το κρατούμενο)
- Παράδειγμα:

	1	1	1	← Μεταφερόμενο	Κρατούμενο
	0	0	1	1	X
+	0	1	1	1	Y
	1	0	1	0	S

Χρειαζόμαστε έναν Πλήρη Αθροιστή ή Full Adder (λέγεται έτσι γιατί μπορεί να κατασκευαστεί από 2 ημιαθροιστές):



Σχεδίαση σε επίπεδο πυλών: Πλήρης Αθροιστής (Full Adder)

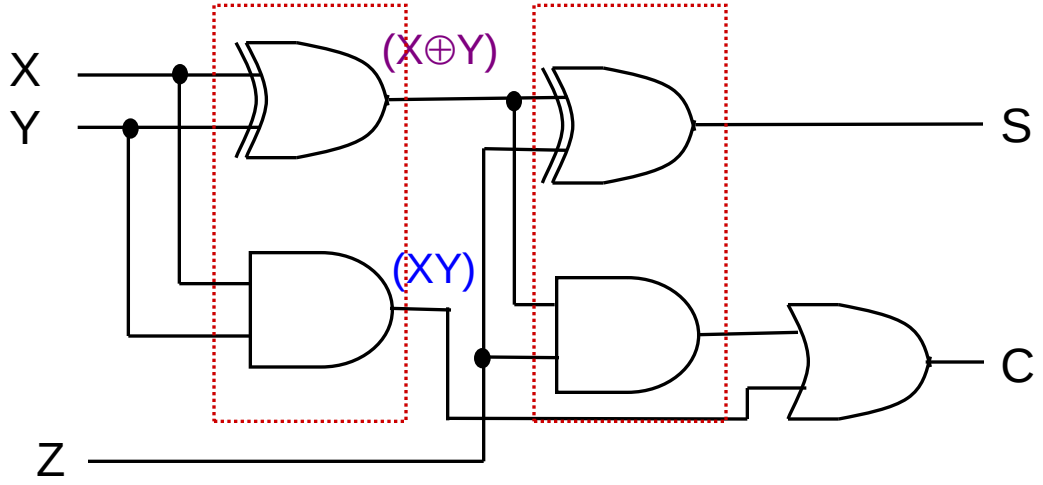
- Πίνακας αληθείας (πρέπει να τον βρείτε εσείς!):

X	Y	Z	C	S
0	0	0	;	;
0	0	1	;	;
0	1	0	;	;
0	1	1	;	;
1	0	0	;	;
1	0	1	;	;
1	1	0	;	;
1	1	1	;	;

Z – κρατούμενο από προηγούμενη άθροιση)
C – κρατούμενο προς επόμενη άθροιση

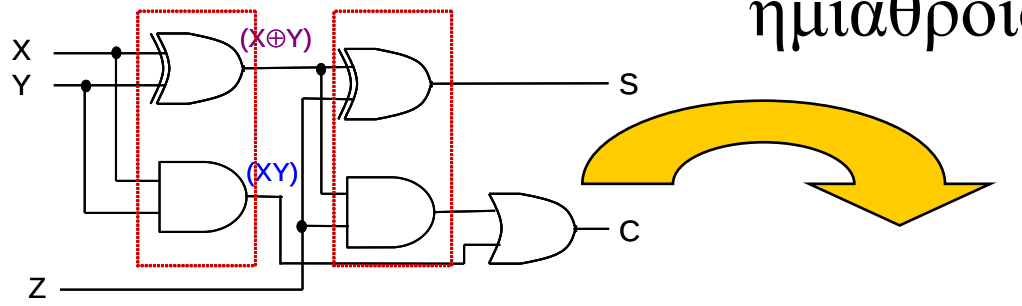
Βρείτε και τις λογικές εκφράσεις των C και S!

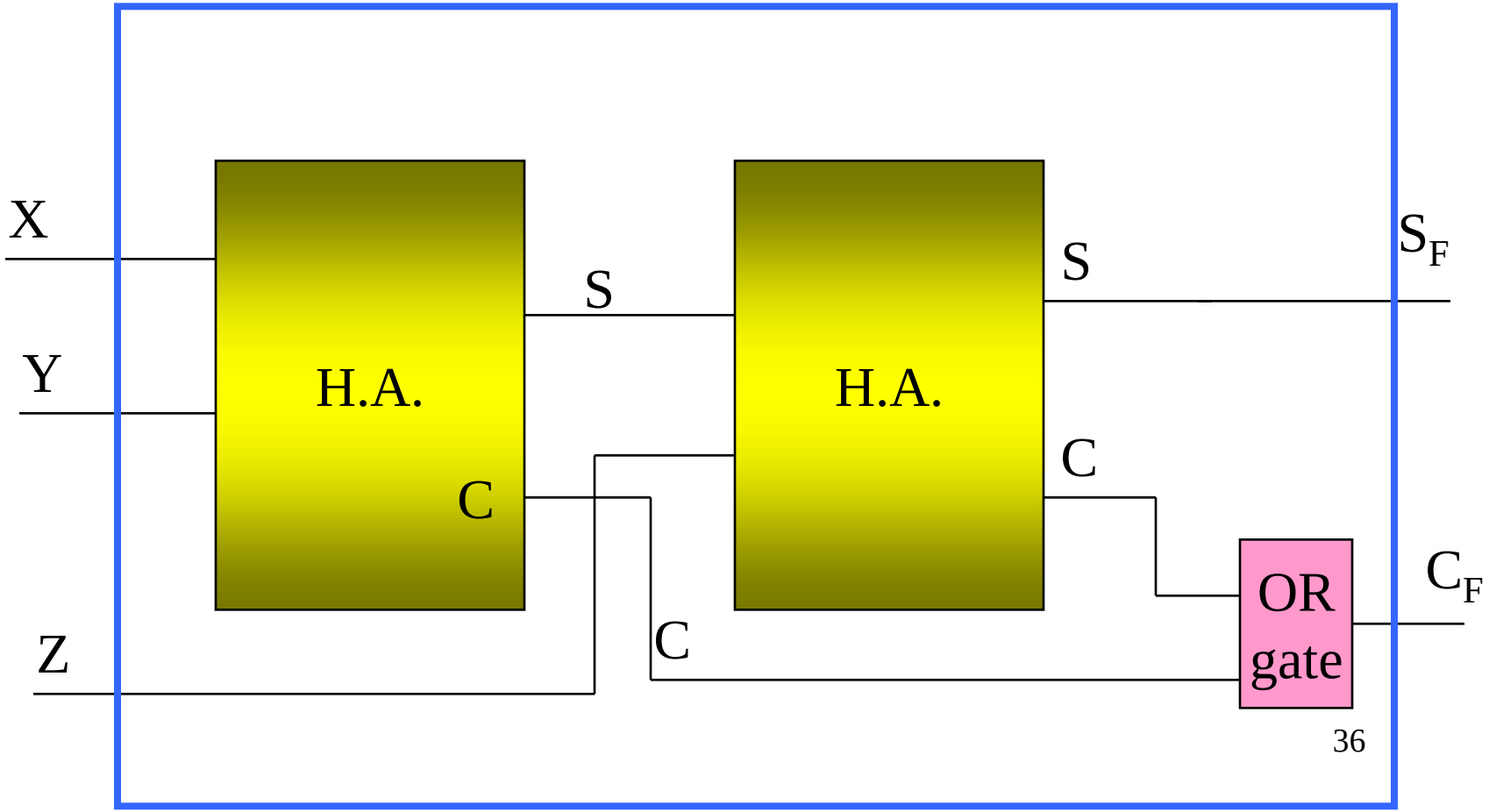
Ο πλήρης αθροιστής με λογικές πύλες:



Αποτελείται από δύο ημιαθροιστές (**H**alf-**A**dders – τα δύο κόκκινα πλαίσια) και μία πύλη OR

Ο πλήρης αθροιστής με άλλη απεικόνιση: blocks ημιαθροιστών



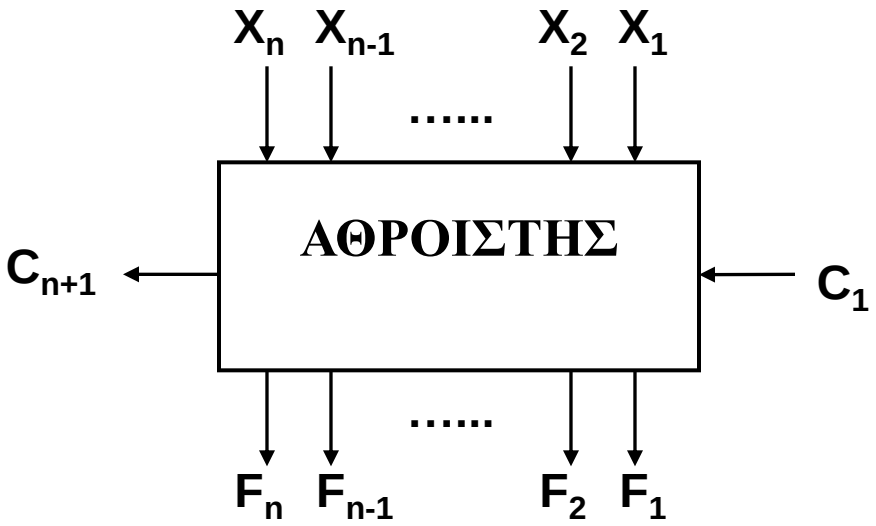


Σχεδίαση με χρήση δομικών στοιχείων

- Χρησιμοποιείται για τη σχεδίαση πολύπλοκων λογικών κυκλωμάτων – συναρτήσεων
- Γενικά, αυτά τα πολύπλοκα κυκλώματα δεν περιγράφονται απλά από έναν πίνακα αληθείας (λόγω τεράστιου αριθμού δυνατών εισόδων) αλλά από έναν αλγόριθμο, μια μέθοδο δηλαδή υπολογισμού της εξόδου με βάση τις εισόδους
- Οι αλγόριθμοι αυτοί προκύπτουν μέσα από την υποδιαίρεση του προβλήματος υπολογισμού σε υποπροβλήματα μέχρις ότου φτάσουμε σε λογικές συναρτήσεις που μπορούν να υπολογιστούν από ήδη διαθέσιμα λογικά στοιχεία).

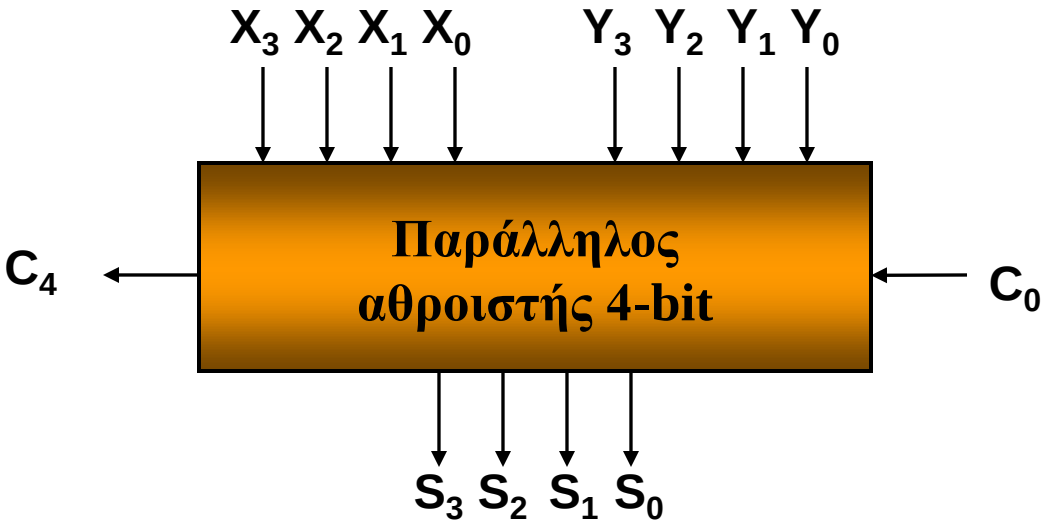
Μεθοδολογία σχεδίασης με δομικά στοιχεία

- Επαναληπτική μέθοδος
  - (i) Καθόρισε την επιθυμητή λειτουργία του λογικού κυκλώματος.
  - (ii) Προσδιόρισε μικρά λειτουργικά στοιχεία που μπορούν, μαζί, να επιτελέσουν τη λειτουργία αυτή.
  - (iii) Συνέδεσε, κατάλληλα, τα μικρά αυτά στοιχεία.
- Παράδειγμα: Θέλουμε να κατασκευάσουμε έναν αθροιστή 2 αριθμών των *n* bits:



Πιο αναλυτικά!

- Ας περιοριστούμε στη σχεδίαση ενός αθροιστή για δύο αριθμούς των 4 bit, μαζί με κρατούμενο, που δίνει ως αποτέλεσμα 5-bits: άθροισμα και τελικό κρατούμενο:



5-bit αρκούν καθώς το αποτέλεσμα μπορεί να είναι το πολύ ίσο με

$$(1111)_2+(1111)_2+(1)_2 = (11111)_2$$

Παράλληλος αθροιστής

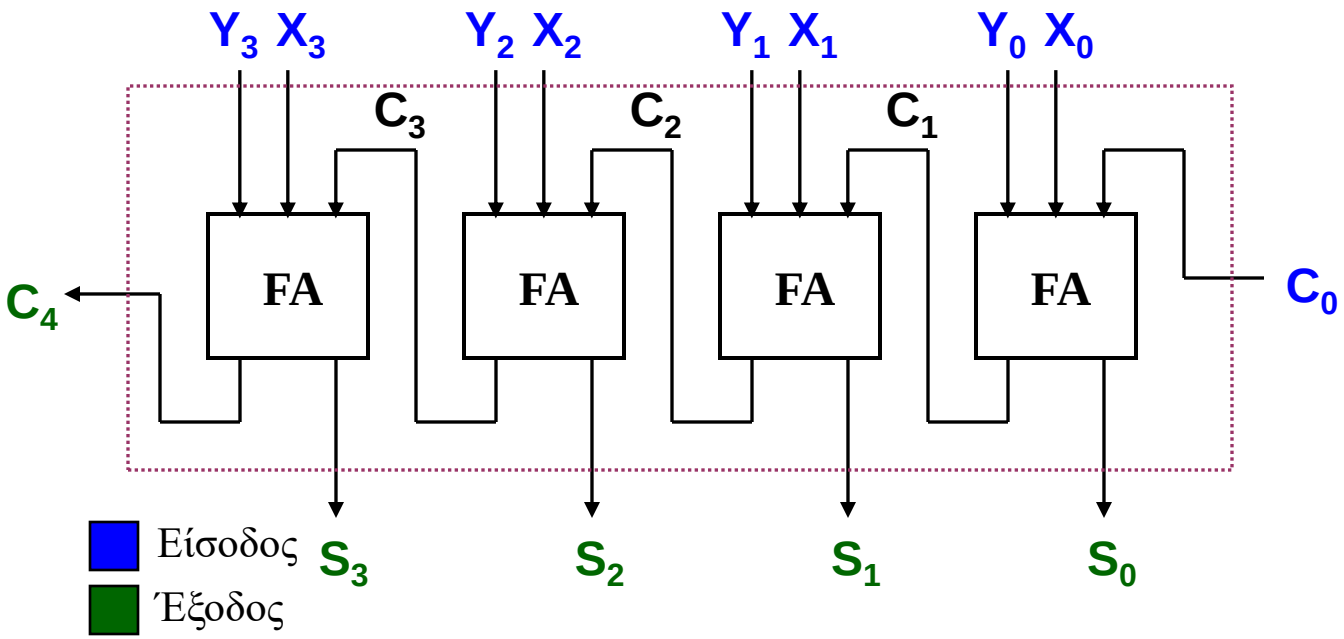
- Γράφουμε τον λογικό τύπο για το άθροισμα για κάθε ένα από τα ζευγάρια των bits εισόδου καθώς και για το κρατούμενο που διαδίδεται στο επόμενο άθροισμα:

$$C_{i+1} = X_i Y_i + (X_i \oplus Y_i) C_i$$
$$S_i = X_i \oplus Y_i \oplus C_i$$

- Παρατηρήστε ότι κάθε στάδιο άθροισης μπορεί να εκτελεστεί από έναν πλήρη αθροιστή, τον οποίο έχουμε ήδη σχεδιάσει και θα τον χρησιμοποιήσουμε ως βασικό δομικό στοιχείο!

Προχωράμε στη σχεδίαση!

- Συνδέοντας 4 πλήρεις αθροιστές ως εξής, παίρνουμε τον αθροιστή δύο αριθμών των 5 bits:



Μερικά σημεία προσοχής

- Προσέξτε ότι το κρατούμενο διαδίδεται από τον ένα πλήρη αθροιστή στον επόμενο με το να συνδέσουμε το κρατούμενο εξόδου του ενός στο κρατούμενο εισόδου του άλλου (cascading).
- Λέγεται Παράλληλος Αθροιστής (Parallel Adder) γιατί οι είσοδοι παρουσιάζονται ταυτόχρονα (παράλληλα) στις εισόδους των πλήρων αθροιστών. Καλείται, επίσης, και Ripple-Carry Adder καθώς το κρατούμενο διαδίδεται «γλιστρώντας» (ripple σημαίνει γλίστρυμα) από τον πρώτο αθροιστή έως τον τελευταίο.

- Όμως αυτά ανήκουν σε άλλη σειρά μαθημάτων.