

Εισαγωγή στον Προγραμματισμό

ΔΙΑΔΙΚΑΣΤΙΚΑ

Δεν παραδίδονται σετ ασκήσεων τις 3 πρώτες εβδομάδες διδασκαλίας. Κατόπιν ισχύει ότι σας πει ο αντίστοιχος διδάσκων.

Στις εξετάσεις υπάρχει 1 θέμα στη θεωρία από τις 3 πρώτες εβδομάδες

Σημείωση: Η αντιγραφή ασκήσεων/θεμάτων μηδενίζει όλα τα αντίγραφα και το πρωτότυπο.

Στις παραδόσεις παίρνουμε παρουσίες. Η 6^η απουσία συνεπάγεται υποχρέωση επανάληψης παρακολούθησης.

Παναγιώτης Τζουνάκης
Άνοιξη 2024



Εισαγωγή στον Προγραμματισμό ΛΕΙΤΟΥΡΓΙΚΑ

Απαραίτητος ο ιδρυματικός λογαριασμός ΑΠΘ (βλ. <https://it.auth.gr/el/account/univID>)



Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Άνοιξη 2024



Εισαγωγή στον Προγραμματισμό

ΛΕΙΤΟΥΡΓΙΚΑ

- 36 Η/Υ με Λειτουργικό Σύστημα MS Windows 10
- Δεν γίνεται backup στα αρχεία χρηστών. Οι φοιτητές διατηρούν τα αρχεία τους σε δικό τους αποθηκευτικό μέσο (usb disk, <https://it.auth.gr/el/cloudServices/myfiles>, Google drive, κλπ., κλπ.)
- Στο τέλος του μαθήματος κάνουμε Shutdown τα MS Windows

Παναγιώτης Τζουνάκης

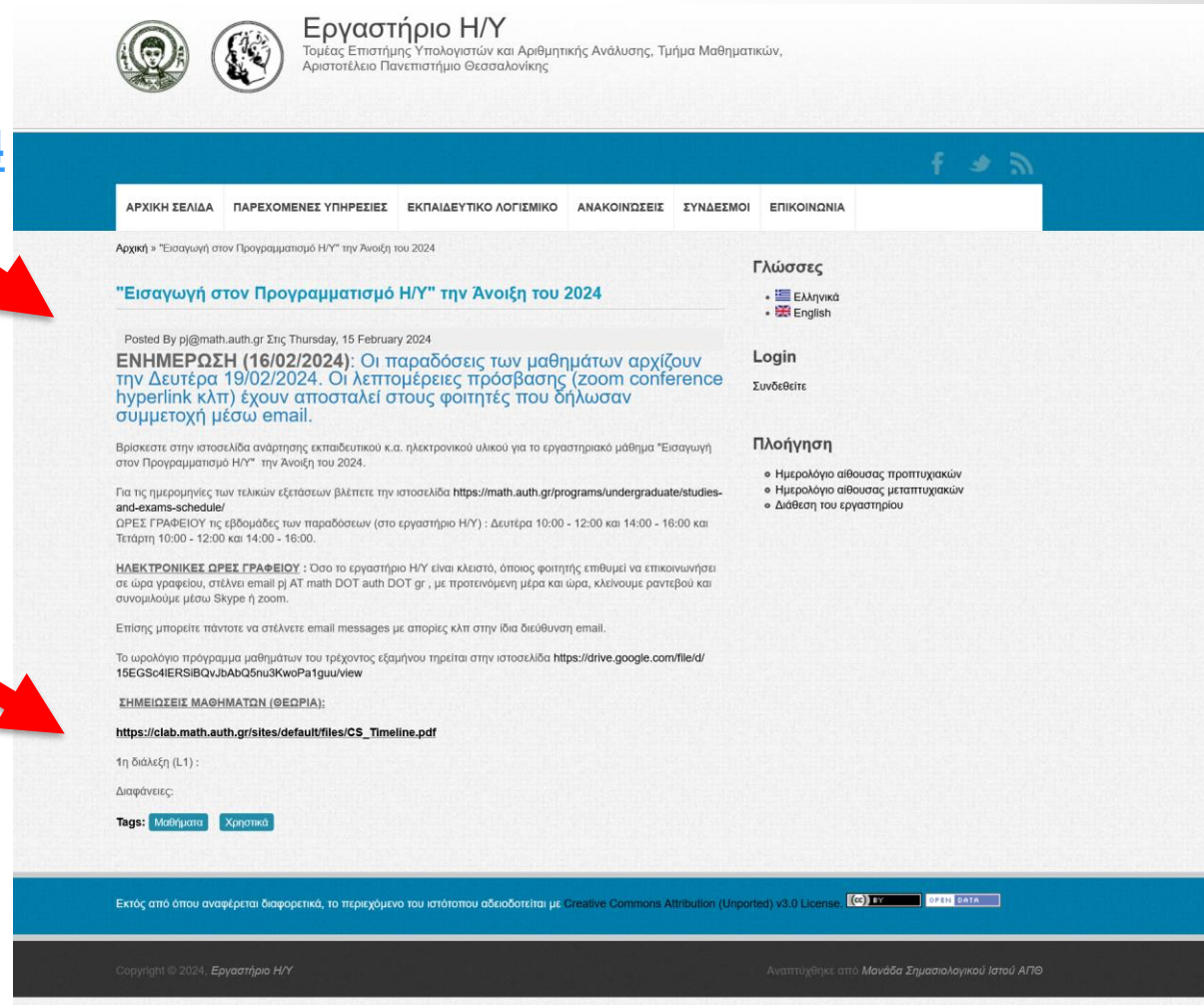


Εισαγωγή στον Προγραμματισμό

ΛΕΙΤΟΥΡΓΙΚΑ

Ανάρτηση σημειώσεων στον ιστοχώρο του εργαστηρίου Η/Υ :

<https://clab.math.auth.gr/el/progspr2024>



Εργαστήριο Η/Υ
Τόμας Επιστήμης Υπολογιστών και Αριθμητικής Ανάλυσης, Τμήμα Μαθηματικών,
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

ΑΡΧΙΚΗ ΣΕΛΙΔΑ ΠΑΡΕΧΟΜΕΝΕΣ ΥΠΗΡΕΣΙΕΣ ΕΚΠΑΙΔΕΥΤΙΚΟ ΛΟΓΙΣΜΙΚΟ ΑΝΑΚΟΙΝΩΣΕΙΣ ΣΥΝΔΕΣΜΟΙ ΕΠΙΚΟΙΝΩΝΙΑ

Αρχική » "Εισαγωγή στον Προγραμματισμό Η/Υ" την Άνοιξη του 2024

"Εισαγωγή στον Προγραμματισμό Η/Υ" την Άνοιξη του 2024

Posted By pj@math.auth.gr Στις Thursday, 15 February 2024

ΕΝΗΜΕΡΩΣΗ (16/02/2024): Οι παραδόσεις των μαθημάτων αρχίζουν την Δευτέρα 19/02/2024. Οι λεπτομέρειες πρόσβασης (zoom conference hyperlink κλπ) έχουν αποσταλεί στους φοιτητές που δήλωσαν συμμετοχή μέσω email.

Βρίσκεστε στην ιστοσελίδα ανάρτησης εκπαιδευτικού κ.α. ηλεκτρονικού υλικού για το εργαστηριακό μάθημα "Εισαγωγή στον Προγραμματισμό Η/Υ" την Άνοιξη του 2024.

Για τις ημερομηνίες των τελικών εξετάσεων βλέπετε την ιστοσελίδα <https://math.auth.gr/programs/undergraduate/studies-and-exams-schedule/>

ΟΡΕΣ ΓΡΑΦΕΙΟΥ τις εβδομάδες των παραδόσεων (στο εργαστήριο Η/Υ) : Δευτέρα 10:00 - 12:00 και 14:00 - 16:00 και Τετάρτη 10:00 - 12:00 και 14:00 - 16:00.

ΗΛΕΚΤΡΟΝΙΚΕΣ ΟΡΕΣ ΓΡΑΦΕΙΟΥ : Όσο το εργαστήριο Η/Υ είναι κλειστό, όποιος φοιτητής επιθυμεί να επικοινωνήσει σε ώρα γραφείου, στέλνει email pj@math.auth.gr, με προτεινόμενη μέρα και ώρα, κλείνουμε ραντεβού και συνομιλούμε μέσω Skype ή zoom.

Επίσης μπορείτε πάντοτε να στέλνετε email messages με απορίες κλπ στην ίδια διεύθυνση email.

Το ωρολόγιο πρόγραμμα μαθημάτων του τρέχοντος εξαμήνου τηρείται στην ιστοσελίδα <https://drive.google.com/file/d/15EGSc4IERSIBQvJbAbQ5nu3KwoPa1guu/view>

ΣΗΜΕΙΩΣΕΙΣ ΜΑΘΗΜΑΤΩΝ (@EQPIA):
https://clab.math.auth.gr/sites/default/files/CS_Timeline.pdf

1η διάλεξη (L1) :
Διαφάνειες:
Tags: [Μαθήματα](#) [Χρηστικά](#)

Εκτός από όπου αναφέρεται διαφορετικά, το περιεχόμενο του ιστότοπου αδειοδοτείται με: [Creative Commons Attribution \(Unported\) v3.0 License](#) [CC BY](#) [Open Data](#)

Copyright © 2024, Εργαστήριο Η/Υ Αναπτύχθηκε από: Μονάδα Σημαιολογικού Ιστού ΑΠΘ



Εισαγωγή στον Προγραμματισμό

Εισαγωγική ενότητα (25%) :
Βασικές έννοιες και τομείς
της Επιστήμης Υπολογιστών
1^η ομιλία

Παναγιώτης Τζουνάκης

Άνοιξη 2024



Περιεχόμενα

1. Εισαγωγή και επισκόπηση

Πληροφορία, Ιστορικά στοιχεία, αριθμητικά συστήματα (2/8/16-ικό), άλγεβρα Boole, αρχιτεκτονική Η/Υ, hard/firm/soft-ware, επίπεδα γλωσσών (μηχανή/assembly), μεταγλώτιση/διερμηνεία

2. Ανάλυση προβλημάτων, Αλγόριθμοι, Οργάνωση και δομές δεδομένων

3. Γλώσσες και μέθοδοι προγραμματισμού

Στάδια, κύκλος ζωής λογισμικού



Περιεχόμενα

4. Λειτουργικά Συστήματα, Πληροφοριακά Συστήματα

Εξοικείωση με DevC++

5. Τηλεπικοινωνίες και δίκτυα υπολογιστών

Δομή, τοπολογία, πρωτόκολλα, πρότυπα, οργάνωση διαχείριση και λειτουργία

6. Το Διαδίκτυο (Internet) και ο παγκόσμιος ιστός (World Wide Web)

Web1.0/2.0/3.0 , υπηρεσίες και εφαρμογές υπολογιστικού νέφους (cloud)



Στόχος η ανάπτυξη:

1. αναλυτικής σκέψης και ικανότητα σύνθεσης
2. νοοτροπίας σαφήνειας και αυστηρής (τυπικής) διατύπωσης
3. δημιουργικότητας, σχεδιαστικής φαντασίας
4. μεθοδικότητας στην εργασία
5. δεξιοτήτων αλγοριθμικής σκέψης-προσέγγισης
6. ικανότητας (αποδοτικής) επίλυσης προβλημάτων με ευθύ (απλό) τρόπο και ελαχιστοποίηση πόρων





Déu Janus, escultura al Vaticà (Catalan)

Όψεις της Πληροφορίας

- Λεξικό Τριανταφυλλίδη

πληροφορία η [pliroforía] O25 : 1. στοιχείο, μήνυμα (είδηση, ανακοίνωση, δήλωση, αναφορά κτλ.) που περιέχει και μεταδίδει μια γνώση για κτ. ή για κτ.

- Στην καθομιλουμένη

«πληροφορία» είναι ένα αφηρημένο ουσιαστικό που σημαίνει οποιαδήποτε ποσότητα δεδομένων, προγραμμάτων Η/Υ ή κειμένων που αποθηκεύονται, μεταφέρονται ή υφίστανται επεξεργασία σε κάποιο (φυσικό) μέσο. (πηγή: <https://plato.stanford.edu/entries/information/>)

- Διαισθητικά

Αποσαφήνιση πραγμάτων, καταστάσεων, εννοιών (π.χ. ανακοίνωση βαθμών εξετάσεων)

Άρση αβεβαιότητας και σκόπιμη θέση αντί τυχαιότητας (π.χ. αποτελέσματα κλήρωσης λαχείου)

Προσθήκη νέας (ενδεχομένως απρόσμενης) γνώσης, καινοτομία (π.χ. καταγραφή σωματιδίου σε πείραμα φυσικής)

Το ταίριασμα (συγκρότηση) της ποικιλομορφίας των μερών για τη σύνθεση του όλου σε δομή (π.χ. DNA)

Εμπορεύσιμο είδος, πηγή κέρδους, πόρος (π.χ. πρόβλεψη τιμής μετοχής στο χρηματιστήριο)

Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης





Déu Janus, escultura al Vaticà (Catalan)

Όψεις της Πληροφορίας

Μερικές επιστημονικές προσεγγίσεις

- Η πληροφορία ως εύρος (range) ενδεχομένων, επιλογών ή διαφορετικών διαμορφώσεων του συστήματος
Ενημερωμένη πληροφορία (update) μειώνει τις επιτρεπτές τιμές και συνεπώς την αβεβαιότητα της κατάστασης. Όσο λιγότερο πιθανό είναι το στοιχείο που τελικά προκύπτει, τόσο μεγαλύτερη είναι η πληροφορία της αντίστοιχης κατάστασης.
- Η πληροφορία ως συσχέτιση (correlation) μερών ενός δομημένου συστήματος
Η δομή καθορίζει κανάλια ανάμεσα σε μέρη όπου ρέει η πληροφορία. Συμπεραίνουμε πληροφορία σε ένα άκρο του καναλιού (μέρος του συστήματος) από την πληροφορία στο άλλο άκρο (άλλο μέρος).
- Η πληροφορία ως πρόγραμμα υπολογιστή (πολυπλοκότητα Kolmogorov)
Η πληροφορία σε μια ακολουθία δυαδικών ψηφίων (bits) είναι το μικρότερο πρόγραμμα το οποίο μπορεί να παράγει την ακολουθία αυτή, τρέχοντας σε μια μηχανή Turing.

Πηγές: <https://plato.stanford.edu/entries/information/> , <https://plato.stanford.edu/entries/logic-information/>



Είδη της Επεξεργασίας



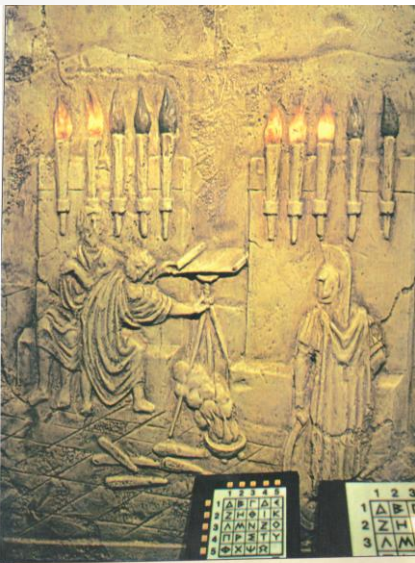
Ο Ήφαιστος κατασκευάζει την ασπίδα του Αχιλλέα

Υπολογισμοί

1. Εισάγουμε τα δεδομένα εισόδου στον υπολογιστή.
2. Η εξελικτική διαδικασία του υπολογισμού διαμορφώνει επιτόπου τα δεδομένα εξόδου από τα δεδομένα εισόδου μετά από κάποιο χρόνο.
3. Εξάγουμε τα δεδομένα εξόδου από τον ίδιο υπολογιστή.

Επικοινωνία

1. Εισάγουμε τα δεδομένα εισόδου στον υπολογιστή.
2. Ο υπολογιστής μεταφέρει αυτούσια τα δεδομένα εισόδου σε ένα δίκτυο τηλεπικοινωνιών. Το δίκτυο παραδίδει τα δεδομένα αυτά σε απομακρυσμένο (σε άλλο τόπο) υπολογιστή μετά από κάποιο χρόνο.
3. Εξάγουμε τα δεδομένα εξόδου (ιδανικά ίδια με τα δεδομένα εισόδου) από τον απομακρυσμένο υπολογιστή.



Σχματό με την πυραία ή τον οπτικό τηλεγράφο των Κλεοξένη και Δημοκλεΐτου [150 π.Χ.]
αναδημοσίευση από περιοδικό "Μουσείο Τηλεπικοινωνιών" Ενημερωτική έκδοση ΟΤΕ.



Θεωρία Πληροφορίας



ΑΡΙΣΤΟΤΕΛΕΙΟ

ΠΑΝΕΠΙΣΤΗΜΙΟ

ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ

ΜΑΘΗΜΑΤΙΚΩΝ

Οδηγός σπουδών 2020-21

[https://math.auth.gr/wp-](https://math.auth.gr/wp-content/uploads/2021/02/study_guide_2020-2021-3.pdf)

[content/uploads/2021/02/study_guide_2020-2021-3.pdf](https://math.auth.gr/wp-content/uploads/2021/02/study_guide_2020-2021-3.pdf)

Φωτογραφία εξωφύλλου: Claude Elwood Shannon (30 Απριλίου 1916 – 24 Φεβρουαρίου 2001)

Η φωτογραφία λήφθηκε από την ιστοσελίδα:

<https://history-computer.com/ModernComputer/thinkers/Shannon.html>

Στη φωτογραφία του εξωφύλλου εικονίζεται ο Claude Elwood Shannon, γνωστός ως «πατέρας της Θεωρίας Πληροφορίας και της Μαθηματικής Κρυπτογραφίας». Το 1936,



Ακρογωνιαίοι λίθοι στην κατασκευή υπολογιστών

αναλυτικά βλ.

https://en.wikipedia.org/wiki/History_of_computing_hardware

- 1642 Pascal +,-
- 1671 Leibniz +,-,*,/
- 1827 Babbage: Polynomial
Difference engine evaluation
(method of finite
differences)



- 1834 Babbage: Analytical Engine (Never completed) General purpose computation
- 1941 Zuse: Z3 General purpose computation
- 1944 Aiken: Harvard Mark I General purpose computation



- 1943-1946 ENIAC (Electronic Numerical Integrator and Calculator)
University of Pennsylvania
John Mauchly, Presper Eckert
(30 tons, 18000 vacuum tubes!)
- Μηχανισμός των Αντικυθήρων (ανακαλύφτηκε το 1900)
1971 Derek Price: Φορητός ημερολογιακός (ηλιακός-σεληνιακός) υπολογιστής
IEEE Micro (1984): Ο πρώτος φορητός υπολογιστής!!



Ιστορικά στοιχεία

Ο μηχανισμός των Αντικυθήρων



Ιστορικά στοιχεία

Ο μηχανισμός των Αντικυθήρων

Κατασκευή 150 – 100 π.Χ.

Διαστάσεις 30x20x10 cm



© Rien van de Wijngaert



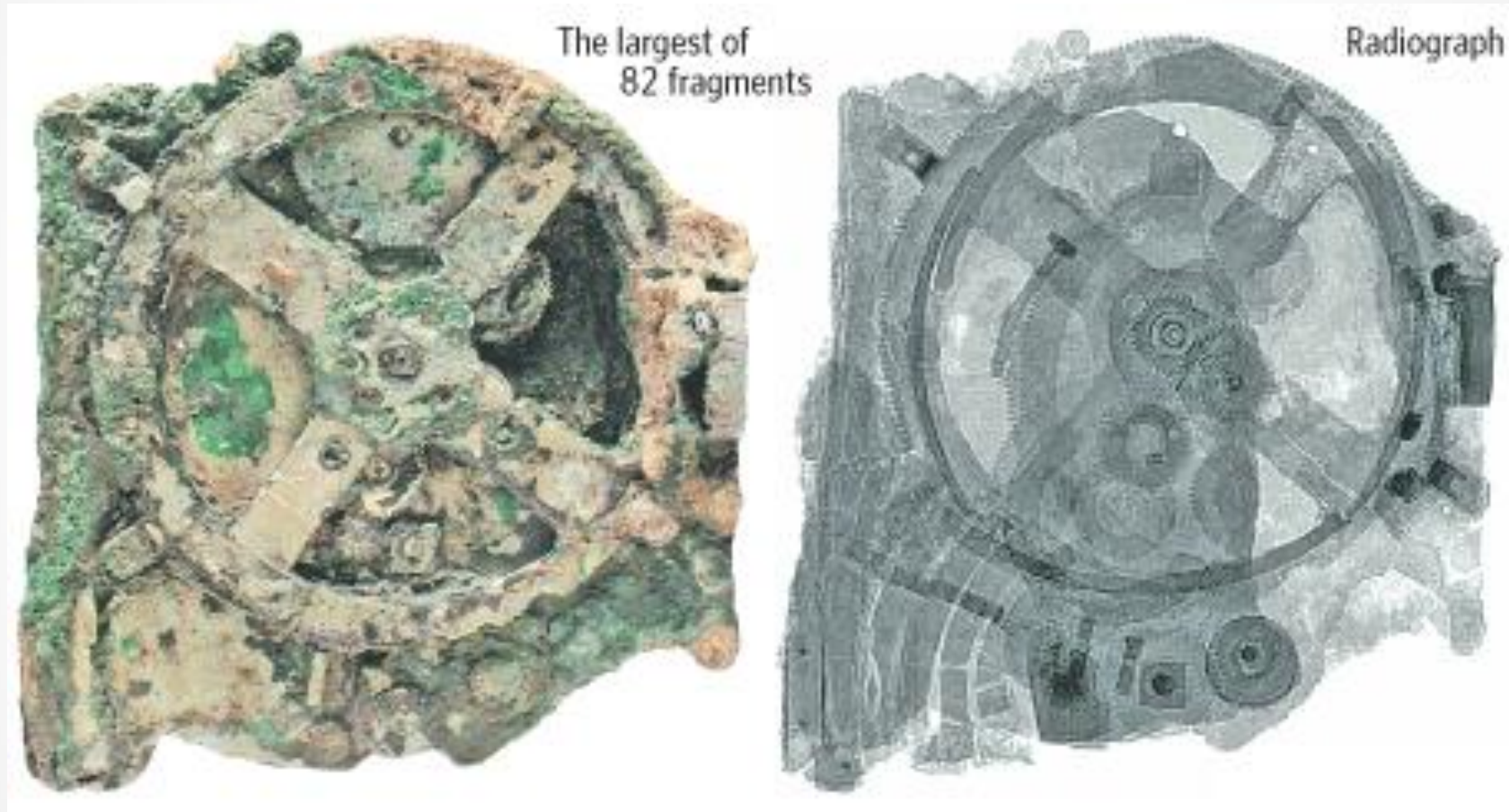
Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



Ιστορικά στοιχεία

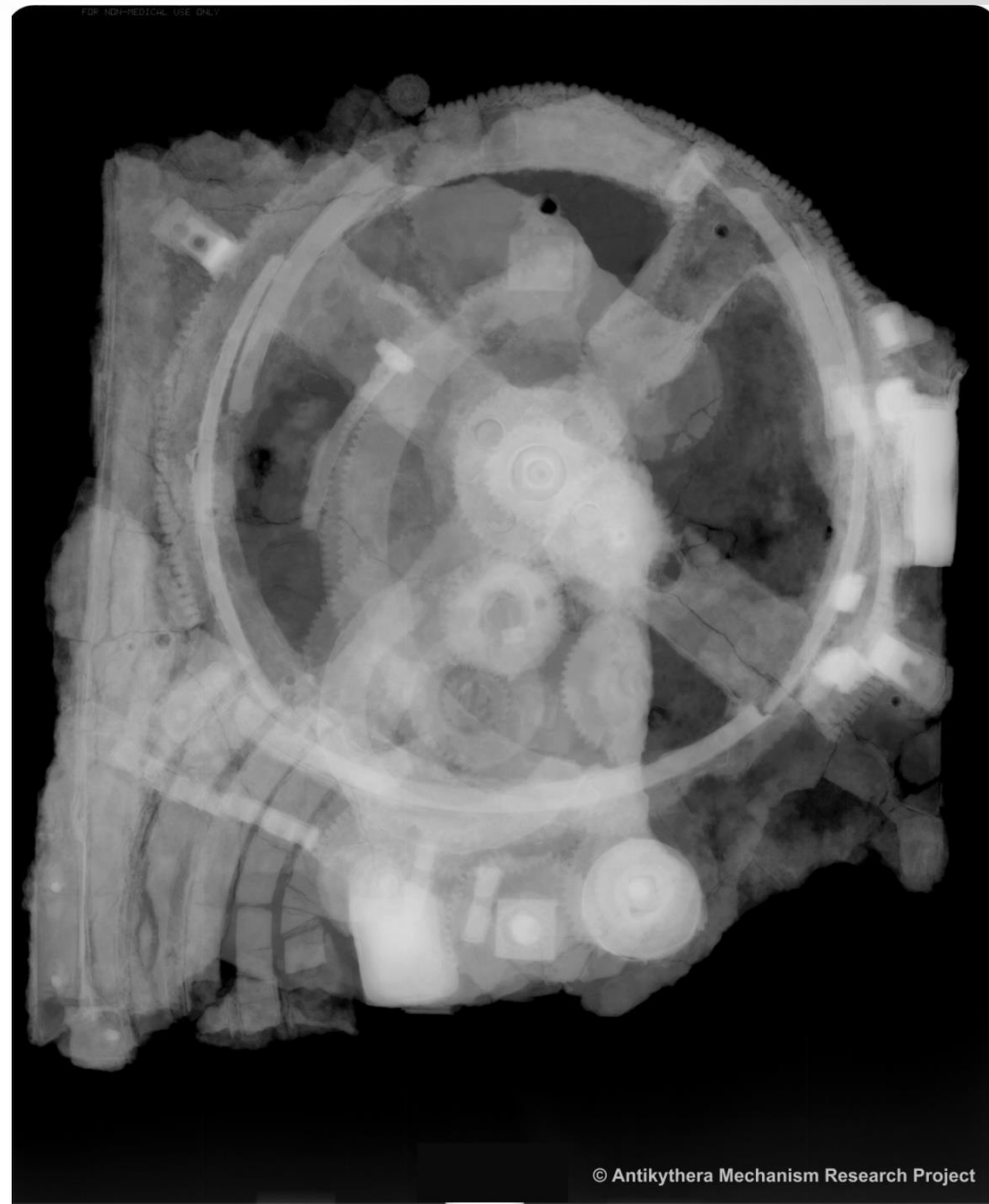
Ο μηχανισμός των Αντικυθήρων



Ιστορικά στοιχεία

Ο μηχανισμός των

Αντικυθήρων



© Antikythera Mechanism Research Project



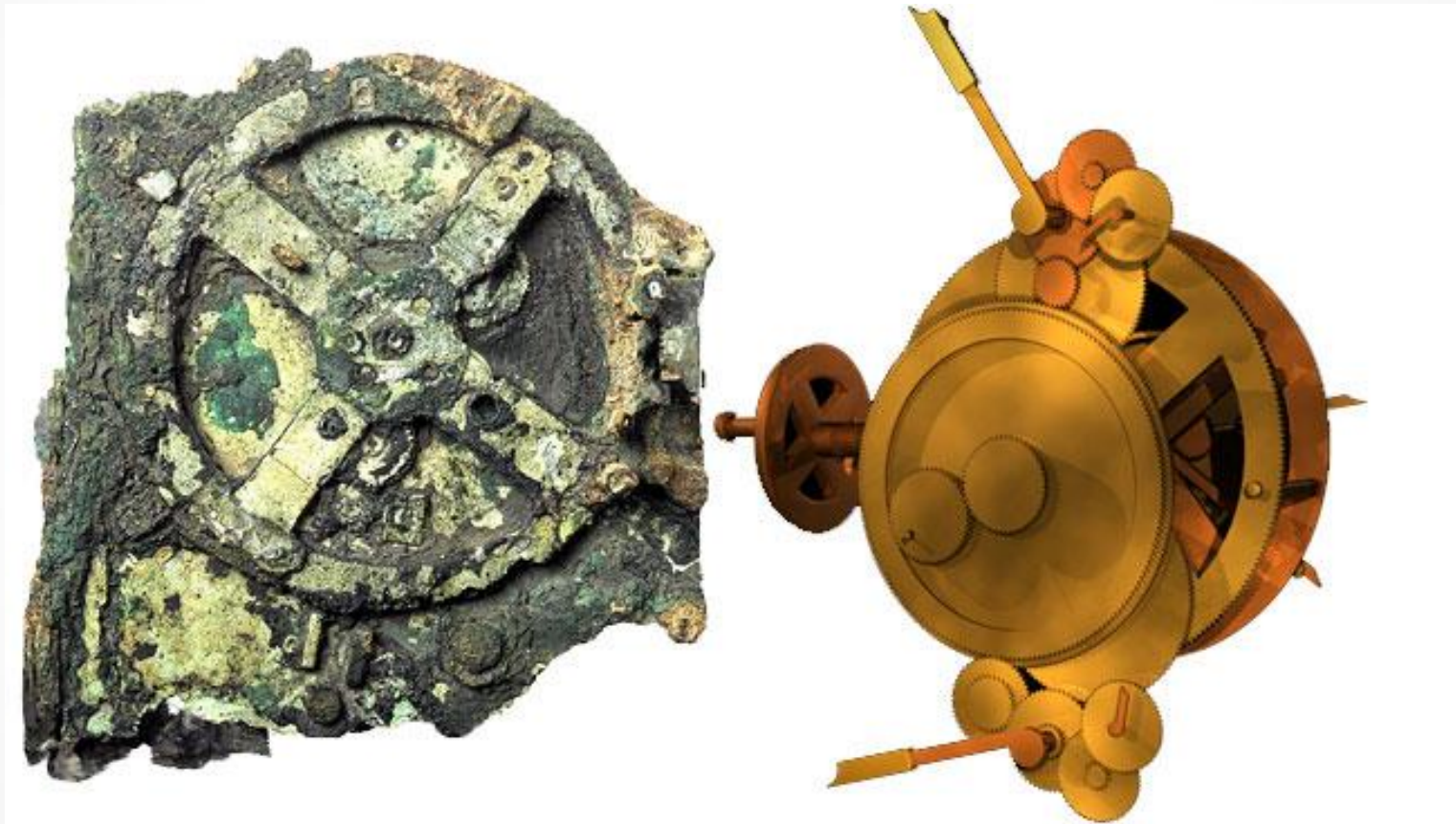
Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



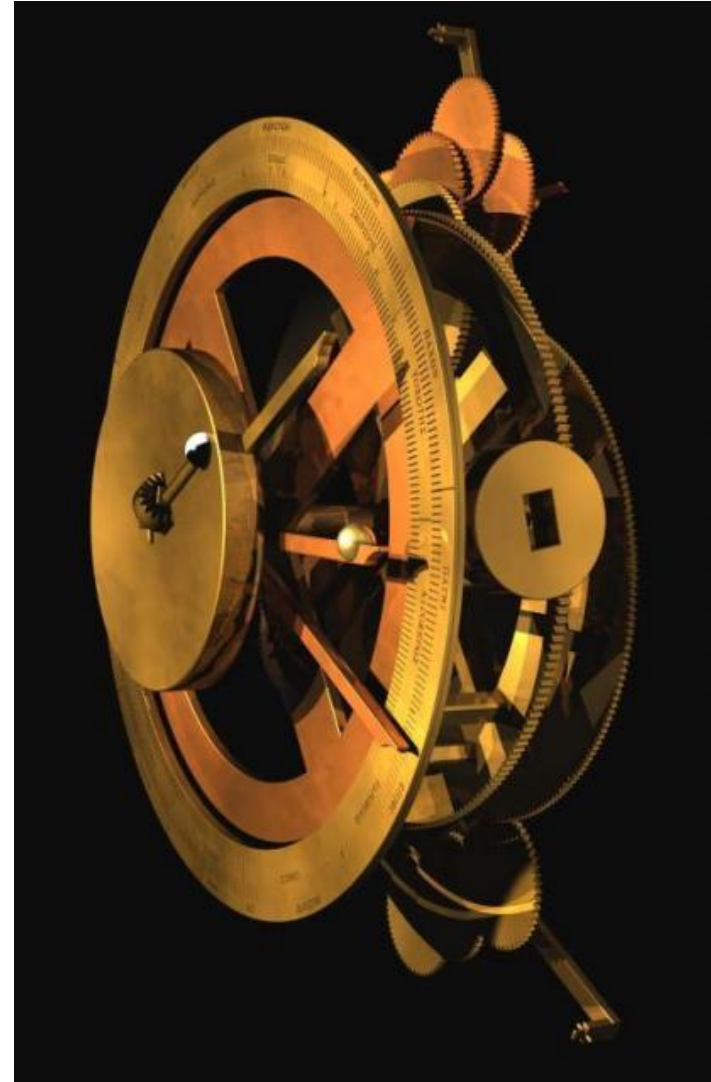
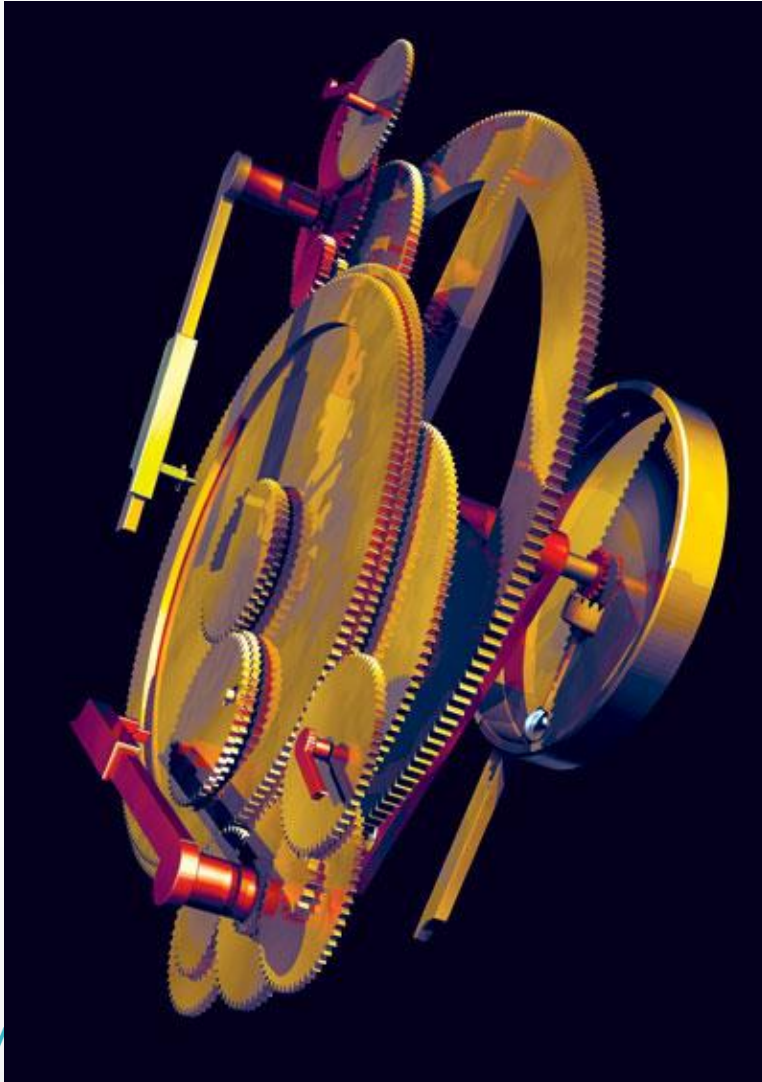
Ιστορικά στοιχεία

Ο μηχανισμός των Αντικυθήρων



Ιστορικά στοιχεία

Ο μηχανισμός των Αντικυθήρων



F. Charette

High tech from Ancient Greece

Nature, vol 444, 30 November 2006, 551-552

J. Seiradakis, et al.

Decoding the ancient Greek astronomical
calculator known as the Antikythera

Mechanism

Nature, vol. 444, 30 November 2006, 587-591

T. Freeth, et al.

Calendars with Olympiad display and eclipse prediction on
the Antikythera Mechanism

Nature, vol. 454, 31 July 2008, 614-617

<http://www.antikythera-mechanism.gr>

<http://www.antikythera-mechanism.gr/media/movies>



Ιστορικά στοιχεία

<https://ethw.org/w/images/1/19/Timeline.pdf>



https://en.wikipedia.org/wiki/History_of_computer_science

https://en.wikipedia.org/wiki/Timeline_of_computing

https://en.wikipedia.org/wiki/List_of_pioneers_in_computer_science



Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



Ιστορικά στοιχεία

Η Επιστήμη Υπολογιστών σήμερα:

- Θεωρητική
 - Ανάλυση Αλγορίθμων
 - Θεωρία Υπολογισιμότητας (Computability)
 - Θεωρία Πολυπλοκότητας (Complexity)
 - Θεωρία Γλωσσών
 - Formal Methods (Specification & Verification)
 - Θεωρία Πληροφορίας, **κρυπτογραφία**
 - ...



Ιστορικά στοιχεία

Η Επιστήμη Υπολογιστών σήμερα:

- Εφαρμοσμένη

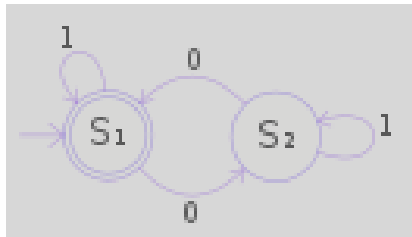
- Υλικό (hardware) : ψηφιακή σχεδίαση, αρχιτεκτονική Η/Υ
- Λογισμικό : Εφαρμογές, Λειτουργικά Συστήματα, ...
- Τεχνητή Νοημοσύνη
- Δίκτυα Δεδομένων
- Πληροφοριακά Συστήματα, Βάσεις δεδομένων/γνώσης
- Γραφικά, διεπαφή (interface) χρήστη,
- Ασφάλεια, **κρυπτογραφία**
- ...



Η Επιστήμη Υπολογιστών με εικόνες

βλ. https://en.wikipedia.org/wiki/Computer_science

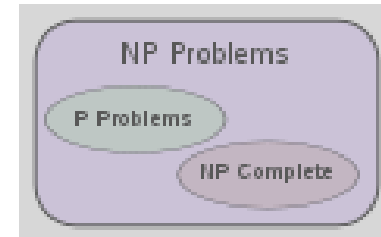
Theory of computation



Automata theory

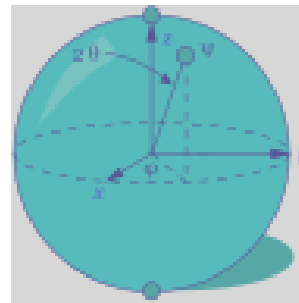
$$M = \{X : X \notin X\}$$

Computability theory



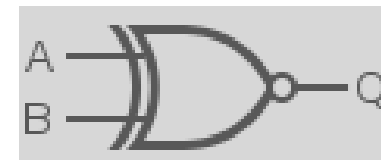
Computational complexity theory

GNITIRW-TERCES



Cryptography

Quantum computing theory



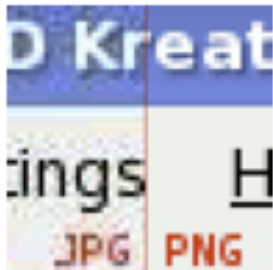
Digital circuit theory



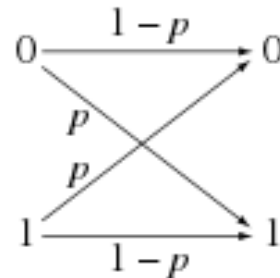
Η Επιστήμη Υπολογιστών με εικόνες

βλ. https://en.wikipedia.org/wiki/Computer_science

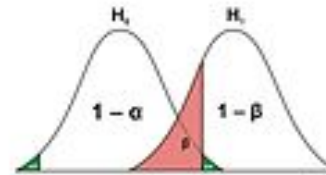
Information and coding theory



Source coding



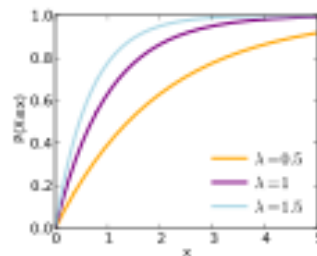
Channel capacity



Signal detection theory

```
gcc -O3 foo.c
```

Algorithmic information theory



Information geometry

$$\hat{\theta} = \operatorname{argmax}_{\theta} P(x|\theta)$$

Estimation theory

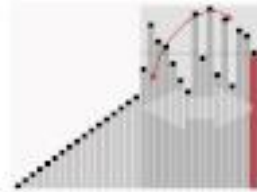


Η Επιστήμη Υπολογιστών με εικόνες

βλ. https://en.wikipedia.org/wiki/Computer_science

Data structures and algorithms

$O(n^2)$



Analysis of algorithms

Algorithms

Data structures

Combinatorial optimization

Computational geometry



Τμήμα Μαθηματικών

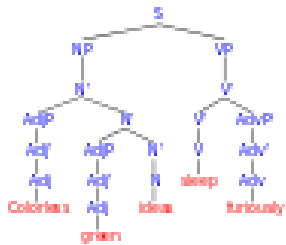
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



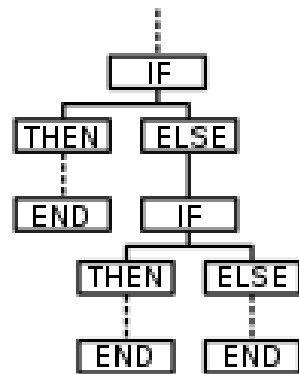
Η Επιστήμη Υπολογιστών με εικόνες

βλ. https://en.wikipedia.org/wiki/Computer_science

Programming language theory



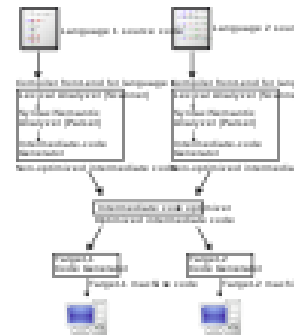
Formal languages



Formal semantics

$\Gamma \vdash x : \text{Int}$

Type theory



Compiler design

```
def add(x, y):  
    return x + y  
  
def multiply(x, y):  
    return x * y  
  
def main():  
    x = 10  
    y = 5  
    z = add(x, y)  
    print(z)  
    w = multiply(x, y)  
    print(w)  
  
if __name__ == '__main__':  
    main()
```

Programming languages



Η Επιστήμη Υπολογιστών με εικόνες

βλ. https://en.wikipedia.org/wiki/Computer_science

Formal methods

$(a \vee \neg b) \wedge b$



$\neg a$

```

% Formal verification
% Example: (a ∨ ¬b) ∧ b ⇒ ¬a
% Proof:
% 1. Assume (a ∨ ¬b) ∧ b.
% 2. Assume a.
% 3. Assume ¬a.
% 4. From 2 and 3, we have a ∧ ¬a, which is a contradiction.
% 5. Therefore, ¬a.
% 6. From 1 and 5, we have (a ∨ ¬b) ∧ b ∧ ¬a.
% 7. From 1, we have a ∨ ¬b.
% 8. From 7 and 5, we have ¬b.
% 9. From 1 and 8, we have ¬b ∧ b, which is a contradiction.
% 10. Therefore, ¬((a ∨ ¬b) ∧ b).
% 11. From 1 and 10, we have ¬((a ∨ ¬b) ∧ b) ∧ ((a ∨ ¬b) ∧ b).
% 12. From 11, we have a contradiction.
% 13. Therefore, ¬a.

```

Formal verification

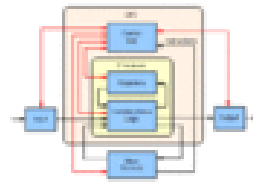
Automated theorem proving



Η Επιστήμη Υπολογιστών με εικόνες

βλ. https://en.wikipedia.org/wiki/Computer_science

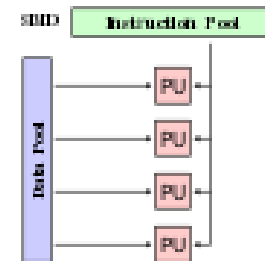
Computer
architecture
& engineering



Processor



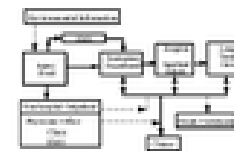
Microarchitecture



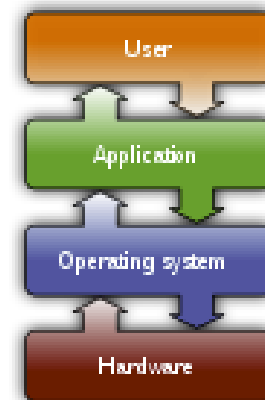
Multiprocessing



Ubiquitous computing



Systems architecture



Operating systems



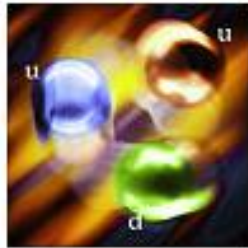
Η Επιστήμη Υπολογιστών με εικόνες

βλ. https://en.wikipedia.org/wiki/Computer_science

Scientific computing and simulation



Numerical analysis



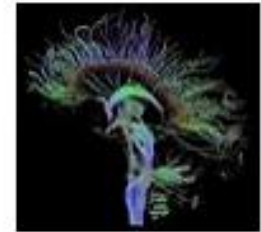
Computational physics



Computational chemistry



Bioinformatics



Computational neuroscience



Η Επιστήμη Υπολογιστών με εικόνες

βλ. https://en.wikipedia.org/wiki/Computer_science

General
Artificial Intelligence



Machine learning



Computer vision

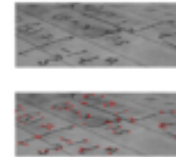
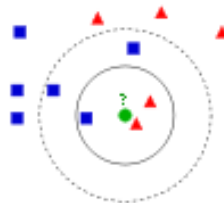
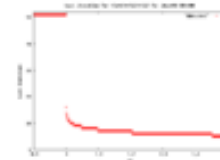


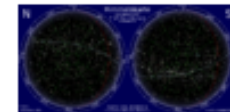
Image processing



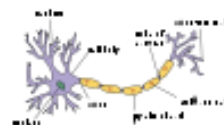
Pattern recognition



Data mining



Evolutionary computation



Knowledge representation and reasoning

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Natural language processing



Robotics



Η Επιστήμη Υπολογιστών σήμερα:

Οι υπολογιστές είναι ο καταλύτης για την 4^η Βιομηχανική Επανάσταση (4IR)

https://en.wikipedia.org/wiki/Fourth_Industrial_Revolution

<https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>

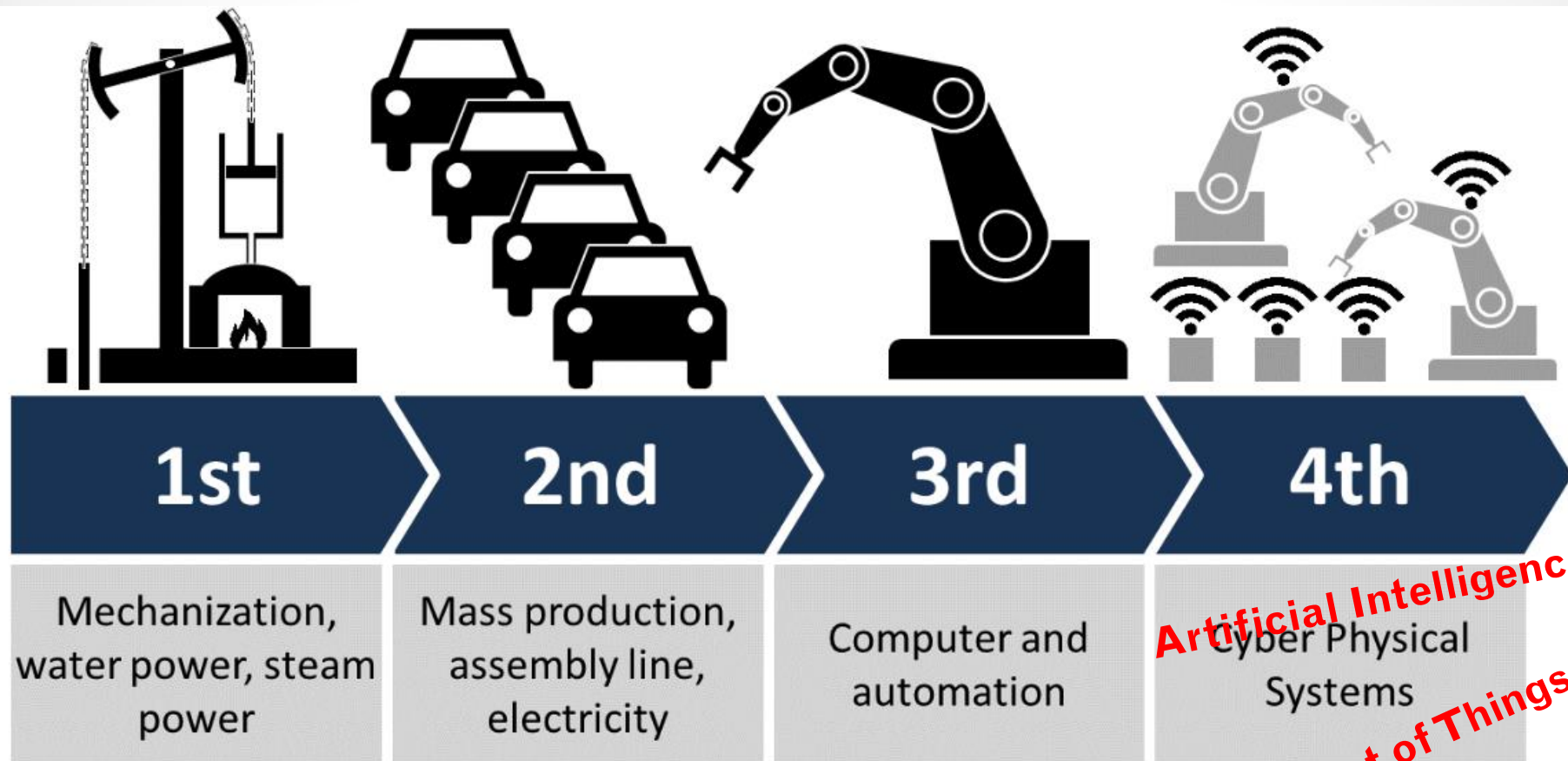


Image Credit: "Christoph Roser at AllAboutLean.com"



Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



Αριθμητικά συστήματα

Δεκαδικό σύστημα (βάση είναι το 10, δεκαδικά ψηφία) :

$$(d_n \dots d_2 d_1 d_0)_{10} \stackrel{\text{def}}{=} d_n \times 10^n + \dots + d_2 \times 10^2 + d_1 \times 10^1 + d_0 \times 10^0$$

όπου $d_0, d_1, d_2, \dots, d_n \in \{0, 1, \dots, 9\}$

$$(532)_{10} \stackrel{\text{def}}{=} 5 \times 100 + 3 \times 10 + 2 \times 1 \stackrel{\text{def}}{=} 5 \times 10^2 + 3 \times 10^1 + 2 \times 10^0$$

Δυαδικό σύστημα (βάση είναι το 2, δυαδικά ψηφία – binary digits (bits)) :

$$(d_n \dots d_2 d_1 d_0)_2 \stackrel{\text{def}}{=} d_n \times 2^n + \dots + d_2 \times 2^2 + d_1 \times 2^1 + d_0 \times 2^0$$

όπου $d_0, d_1, d_2, \dots, d_n \in \{0, 1\}$

$$(532)_{10} \stackrel{\text{def}}{=} (1000010100)_2 \stackrel{\text{def}}{=} 1 \times 512 + \dots + 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1$$
$$= 1 \times 2^9 + 1 \times 2^4 + 1 \times 2^2$$



Αριθμητικά συστήματα

Οκταδικό σύστημα (βάση είναι το 8, οκταδικά ψηφία - oct) :

$$(d_n \dots d_2 d_1 d_0)_8 \stackrel{\text{def}}{=} d_n \times 8^n + \dots + d_2 \times 8^2 + d_1 \times 8^1 + d_0 \times 8^0$$

όπου $d_0, d_1, d_2, \dots, d_n \in \{0, 1, 2, 3, 4, 5, 6, 7\}$

$$(532)_8 \stackrel{\text{def}}{=} 5 \times 64 + 3 \times 8 + 2 \times 1 \stackrel{\text{def}}{=} 5 \times 8^2 + 3 \times 8^1 + 2 \times 8^0$$

ΠΡΟΣΟΧΗ!!! : $(532)_8 \neq (532)_{10}$

binary 2 octal : 3 bits στη σειρά αντικαθίστανται από το αντίστοιχο οκταδικό ψηφίο, αρχίζοντας από το λιγότερο σημαντικό bit.

Συμπληρώνουμε μηδενικά αριστερά του περισσότερο σημαντικού bit εφόσον χρειάζεται.

$$(1101001010)_2 = (1512)_8$$



Αριθμητικά συστήματα

Δεκαεξαδικό σύστημα (βάση είναι το 16, δεκαεξαδικά ψηφία - hex) :

$$(d_n \dots d_2 d_1 d_0)_2 \stackrel{\text{def}}{=} d_n \times 16^n + \dots + d_2 \times 16^2 + d_1 \times 16^1 + d_0 \times 16^0$$

όπου $d_0, d_1, d_2, \dots, d_n \in \{0, 1, \dots, 9, A, B, C, D, E, F\}$

$$(532)_{16} \stackrel{\text{def}}{=} 5 \times 256 + 3 \times 16 + 2 \times 1 \stackrel{\text{def}}{=} 5 \times 16^2 + 3 \times 16^1 + 2 \times 16^0$$

binary 2 hexadecimal : 4 bits στη σειρά αντικαθίστανται από το αντίστοιχο δεκαεξαδικό ψηφίο, αρχίζοντας από το λιγότερο σημαντικό bit. Συμπληρώνουμε μηδενικά αριστερά του περισσότερου σημαντικού bit εφόσον χρειάζεται.

$$(1101001010)_2 = (34A)_{16} = (1512)_8$$



Παρένθεση: Κλασσικά bits και ...πιο «εξωτικά» bits

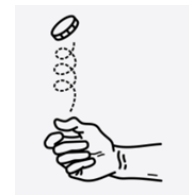


Bit
0



1

State: 0 or 1

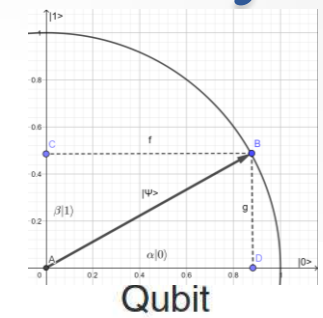


Pbit
0

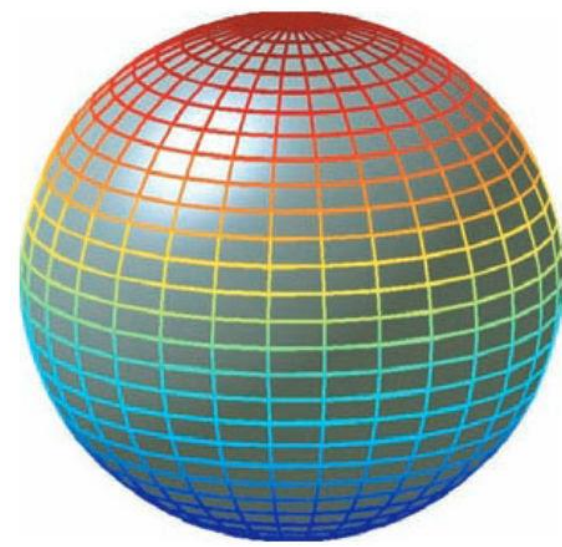


1

{p:0, (1 - p):1}



Qubit
|0>



|1>

$$\alpha|0\rangle + \beta|1\rangle$$

$$|\alpha|^2 + |\beta|^2 = 1$$



Πηγή: https://open.substack.com/pub/normalcomputing/p/thermodynamic-ai-intelligence-from?utm_campaign=post&utm_medium=web

Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



Αριθμητικά συστήματα

Αριθμητικές πράξεις : (business) as usual!

Πρόσθεση:

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1 \quad (\text{carried digits}) \\ 0\ 1\ 1\ 0\ 1 \\ +\ 1\ 0\ 1\ 1\ 1 \\ \hline = 1\ 0\ 0\ 1\ 0\ 0 = 36 \end{array}$$

Πολλαπλασιασμός:

$$\begin{array}{r} 1\ 0\ 1\ 1 \quad (A) \\ \times 1\ 0\ 1\ 0 \quad (B) \\ \hline 0\ 0\ 0\ 0 \quad \leftarrow \text{Corresponds to the rightmost 'zero' in } B \\ + 1\ 0\ 1\ 1 \quad \leftarrow \text{Corresponds to the next 'one' in } B \\ + 0\ 0\ 0\ 0 \\ + 1\ 0\ 1\ 1 \\ \hline = 1\ 1\ 0\ 1\ 1\ 1\ 0 \end{array}$$



Αριθμητικά συστήματα

Δυαδικοί αριθμοί με πρόσημο : Το συμπλήρωμα ως προς 2

Ορισμός: Για δυαδικό αριθμό x που αναπαριστάται με N το πλήθος bits, είναι ο δυαδικός αριθμός y , τέτοιος ώστε $x + y = 2^N$

Υπολογισμός: Αντιστρέφουμε όλα τα bits (σημ: αυτό ονομάζεται «συμπλήρωμα ως προς 1») και κατόπιν προσθέτουμε 1.

Three-bit signed integers

Decimal value	Two's-complement Representation
0	000
1	001
2	010
3	011
-4	100
-3	101
-2	110
-1	111



Αριθμητικά συστήματα

https://en.wikipedia.org/wiki/Two's_complement

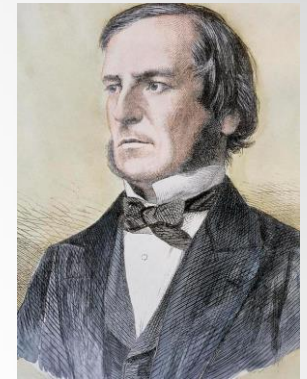
https://en.wikipedia.org/wiki/Ones'_complement

Eight-bit signed integers

Decimal value ↕	Two's-complement Representation ↕
0	0000 0000
1	0000 0001
2	0000 0010
126	0111 1110
127	0111 1111
-128	1000 0000
-127	1000 0001
-126	1000 0010
-2	1111 1110
-1	1111 1111



Άλγεβρα Boole



https://en.wikipedia.org/wiki/Boolean_algebra

Βασικές πράξεις:

ΚΑΙ – σύζευξη (AND – conjunction) : $x \wedge y$ (x AND y)

Η – διάζευξη (OR – disjunction) : $x \vee y$ (x OR y)

ΟΧΙ - άρνηση (NOT – negation) : $\neg x$ (NOT x, Nx ή !x)

Πίνακες αληθείας

x	y	$x \wedge y$	$x \vee y$	x	$\neg x$
0	0	0	0	0	1
1	0	0	1	1	0
0	1	0	1		
1	1	1	1		



Άλγεβρα Boole

https://en.wikipedia.org/wiki/Boolean_algebra

δευτερεύουσες πράξεις: Υλική συνεπαγωγή, Αποκλειστικό Η (Exclusive OR – XOR), Ισοδυναμία (equivalence)

$$x \rightarrow y = \neg x \vee y$$

$$x \oplus y = \neg(x \equiv y) = (x \vee y) \wedge (\neg x \vee \neg y) = (x \wedge \neg y) \vee (\neg x \wedge y)$$

$$x \equiv y = \neg(x \oplus y) = (x \wedge y) \vee (\neg x \wedge \neg y)$$

Πίνακες αληθείας

x	y	$x \rightarrow y$	$x \oplus y$	$x \equiv y$
0	0	1	0	1
1	0	0	1	0
0	1	1	1	0
1	1	1	0	1



Άλγεβρα Boole

Νόμοι (κατ' αναλογία με συνήθη άλγεβρα με + αντί \vee και * αντί \wedge)

Associativity of \vee :

$$x \vee (y \vee z) = (x \vee y) \vee z$$

Associativity of \wedge :

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z$$

Commutativity of \vee :

$$x \vee y = y \vee x$$

Commutativity of \wedge :

$$x \wedge y = y \wedge x$$

Distributivity of \wedge over \vee :

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

Identity for \vee :

$$x \vee 0 = x$$

Identity for \wedge :

$$x \wedge 1 = x$$

Annihilator for \wedge :

$$x \wedge 0 = 0$$



Άλγεβρα Boole

Νόμοι (δεν ισχύουν στη συνήθη άλγεβρα με + , *)

Annihilator for \vee : $x \vee 1 = 1$

Idempotence of \vee : $x \vee x = x$

Idempotence of \wedge : $x \wedge x = x$

Absorption 1: $x \wedge (x \vee y) = x$

Absorption 2: $x \vee (x \wedge y) = x$

Distributivity of \vee over \wedge : $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$



Άλγεβρα Boole

Κι άλλοι νόμοι!

Για το συμπλήρωμα :

Complementation 1

$$x \wedge \neg x = 0$$

Complementation 2

$$x \vee \neg x = 1$$

Διπλή άρνηση:

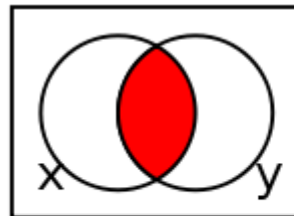
$$\neg(\neg x) = x$$

Νόμοι De Morgan:

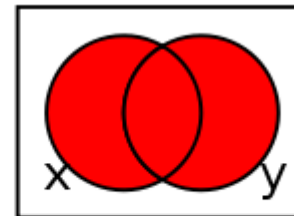
$$\neg x \wedge \neg y = \neg(x \vee y)$$

$$\neg x \vee \neg y = \neg(x \wedge y)$$

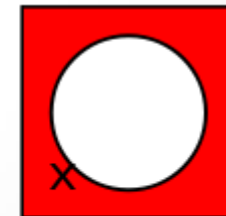
Διαγράμματα Venn για πράξεις στην άλγεβρα Boole :



$$x \wedge y$$



$$x \vee y$$



$$\neg x$$



Άλγεβρα Boole και ψηφιακές λογικές πύλες

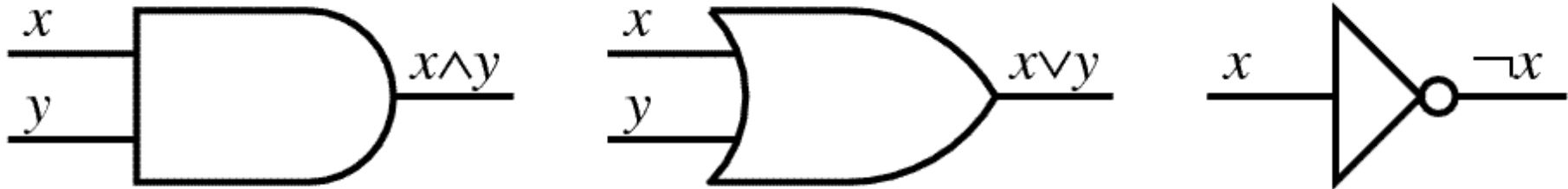


Figure 3. Logic gates

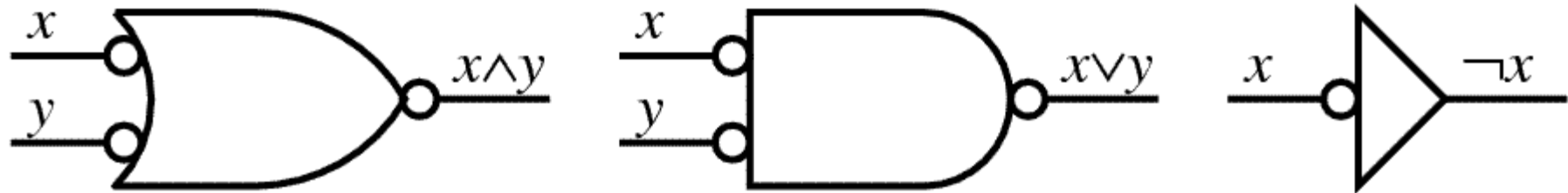


Figure 4. De Morgan equivalents

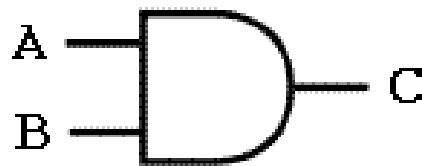


Αντιστρεψιμότητα (reversibility) υπολογισμών

- Φυσική και λογική αντιστρεψιμότητα

https://en.wikipedia.org/wiki/Reversible_computing

AND



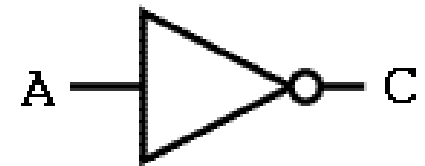
Inputs		Output
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1 ✓

OR



Inputs		Output
A	B	C
0	0	0 ✓
0	1	1
1	0	1
1	1	1

NOT



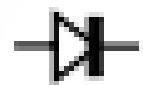
Input	Output
A	C
0	1 ✓
1	0 ✓



Δείτε την αρχή και το όριο του Landauer https://en.wikipedia.org/wiki/Landauer's_principle



Ψηφιακά κυκλώματα



Diode



Capacitor



Inductor



Resistor



DC voltage source



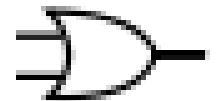
AC voltage source



And gate



Nand gate



Or gate



Nor gate



Xor gate

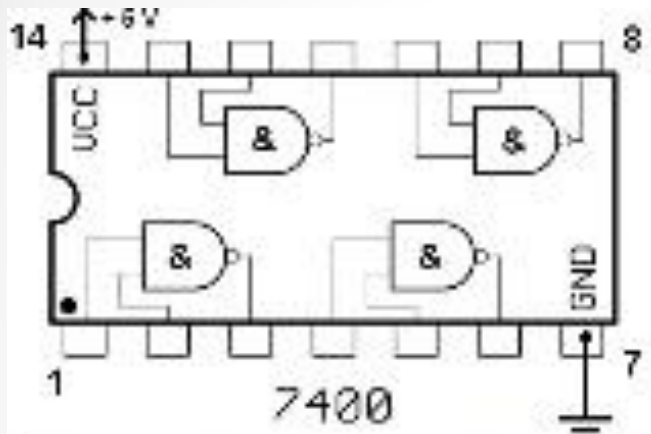


Inverter
(Not gate)

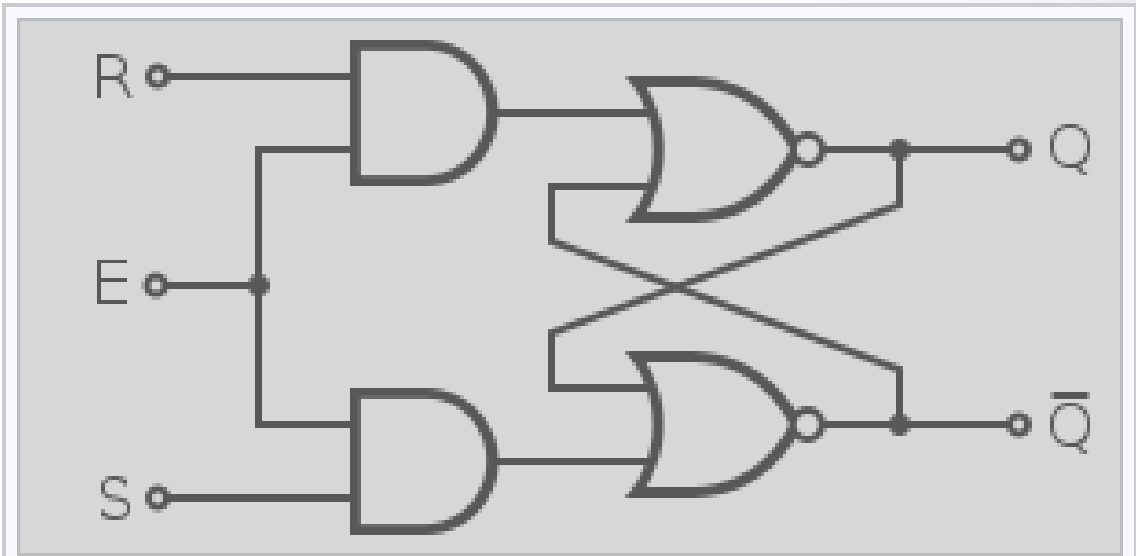
Συνήθη σύμβολα σε σχηματικά διαγράμματα κυκλωμάτων



Ψηφιακά κυκλώματα



Ακολουθιακό κύκλωμα



A gated SR latch circuit diagram constructed from AND gates (on left) and NOR gates (on right).

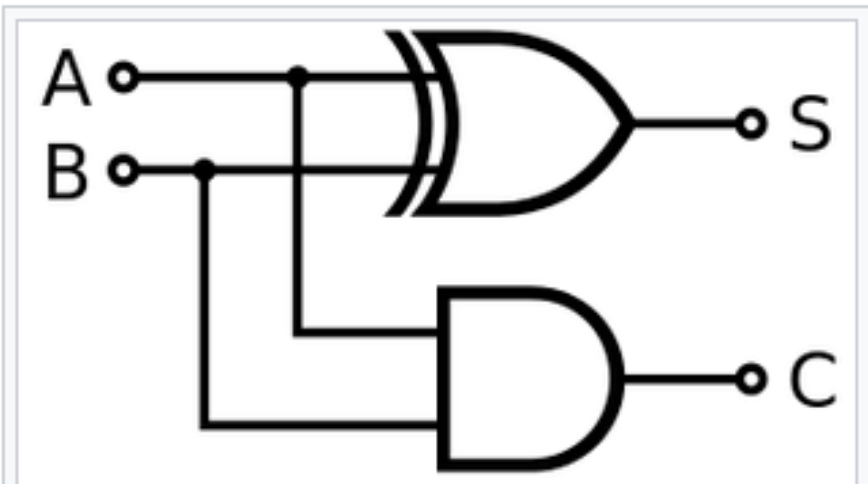
«τσιπάκι» (αριστερά) και μανταλωτής (δεξιά)

βλ. https://en.wikipedia.org/wiki/Logic_gate#Symbols και

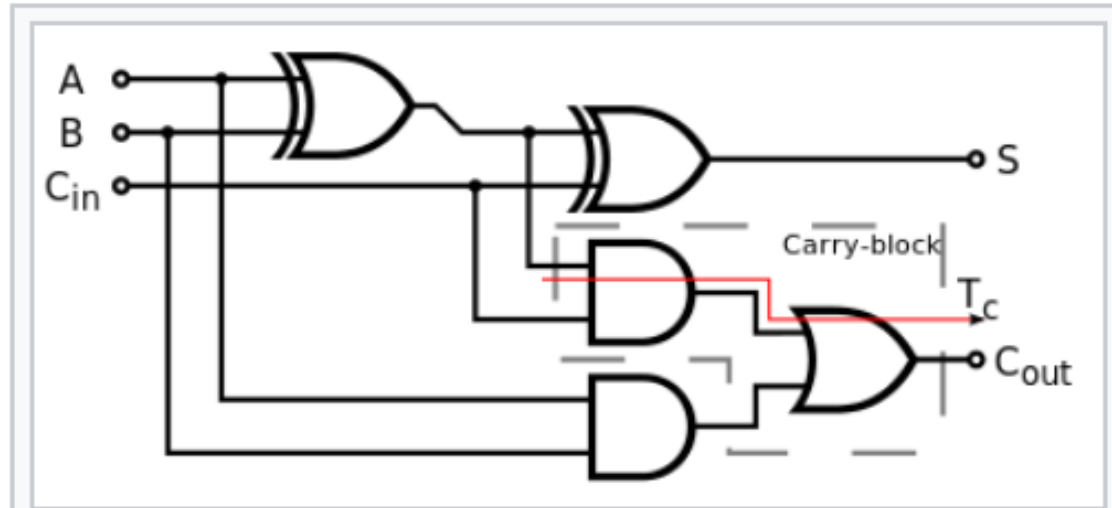
[https://en.wikipedia.org/wiki/Flip-flop_\(electronics\)](https://en.wikipedia.org/wiki/Flip-flop_(electronics)) αντίστοιχα



Αριθμητικές πράξεις με άλγεβρα Boole



Half adder logic diagram



Logic diagram for a full adder.

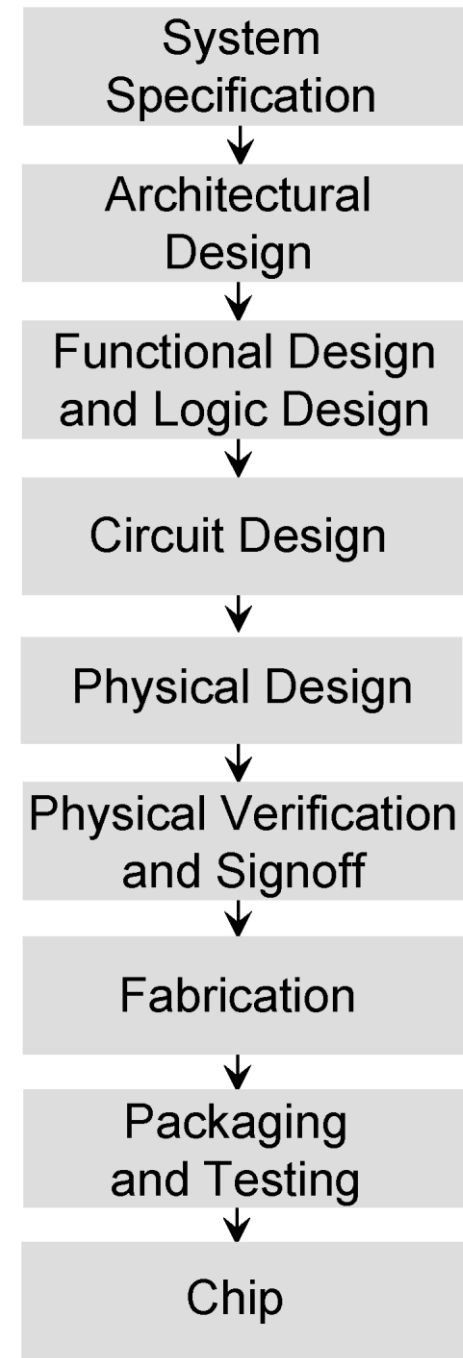
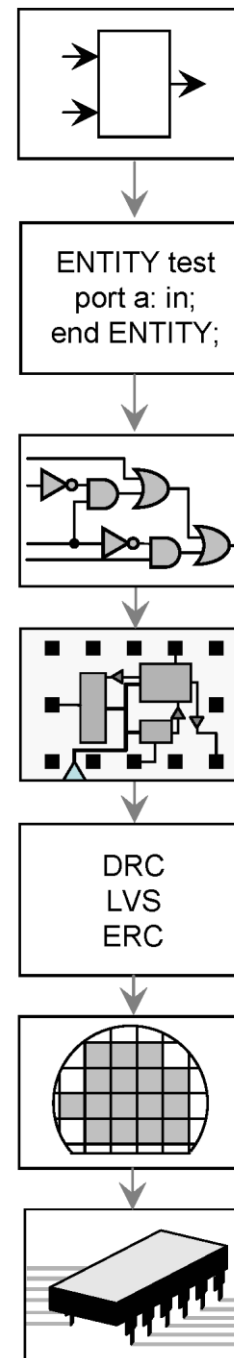
[https://en.wikipedia.org/wiki/Adder_\(electronics\)](https://en.wikipedia.org/wiki/Adder_(electronics))



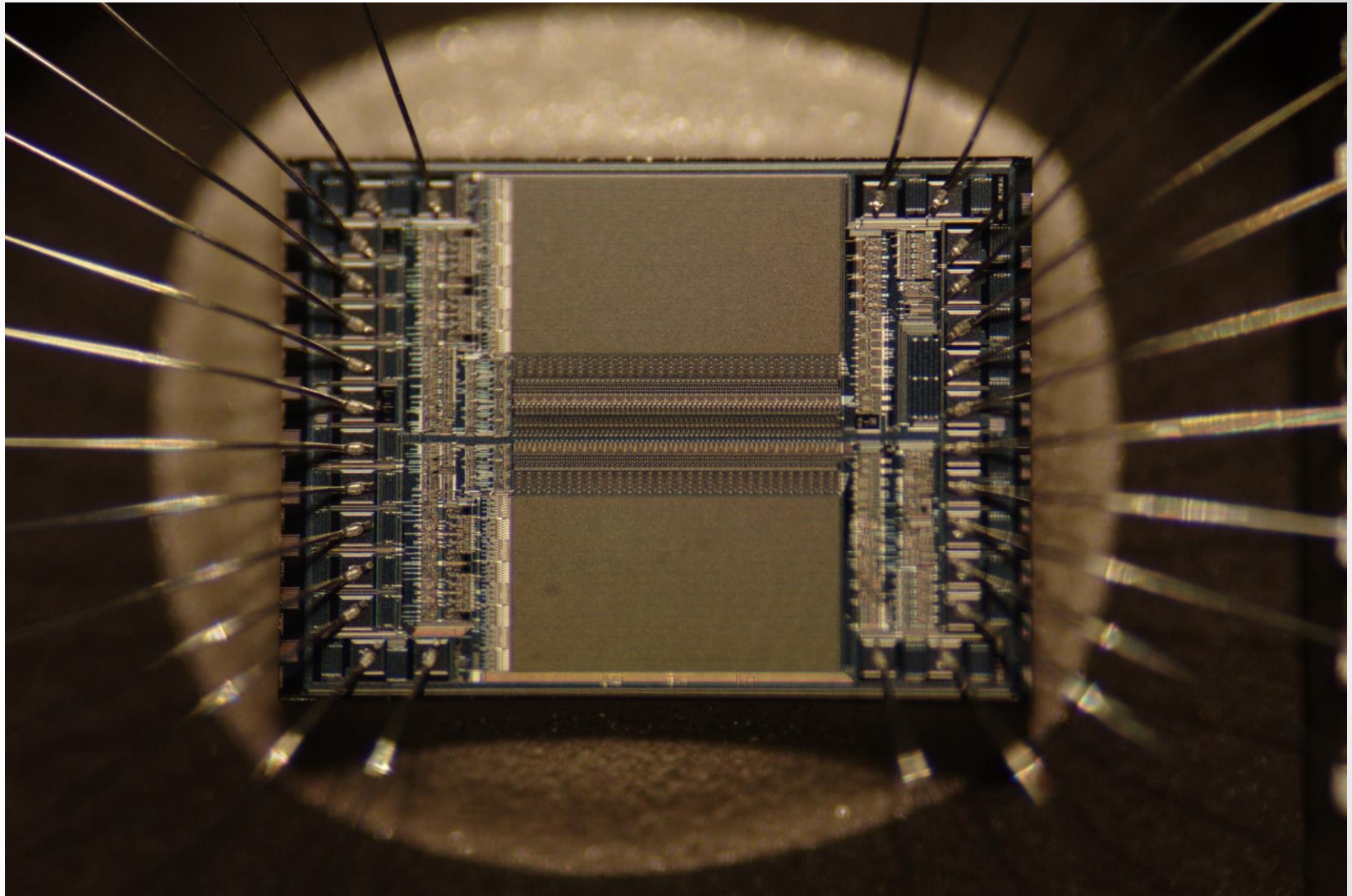
Ψηφιακή Σχεδίαση

Στάδια κατασκευής
ολοκληρωμένου κυκλώματος
(chip)

https://en.wikipedia.org/wiki/Integrated_circuit_design



Ψηφιακή σχεδίαση

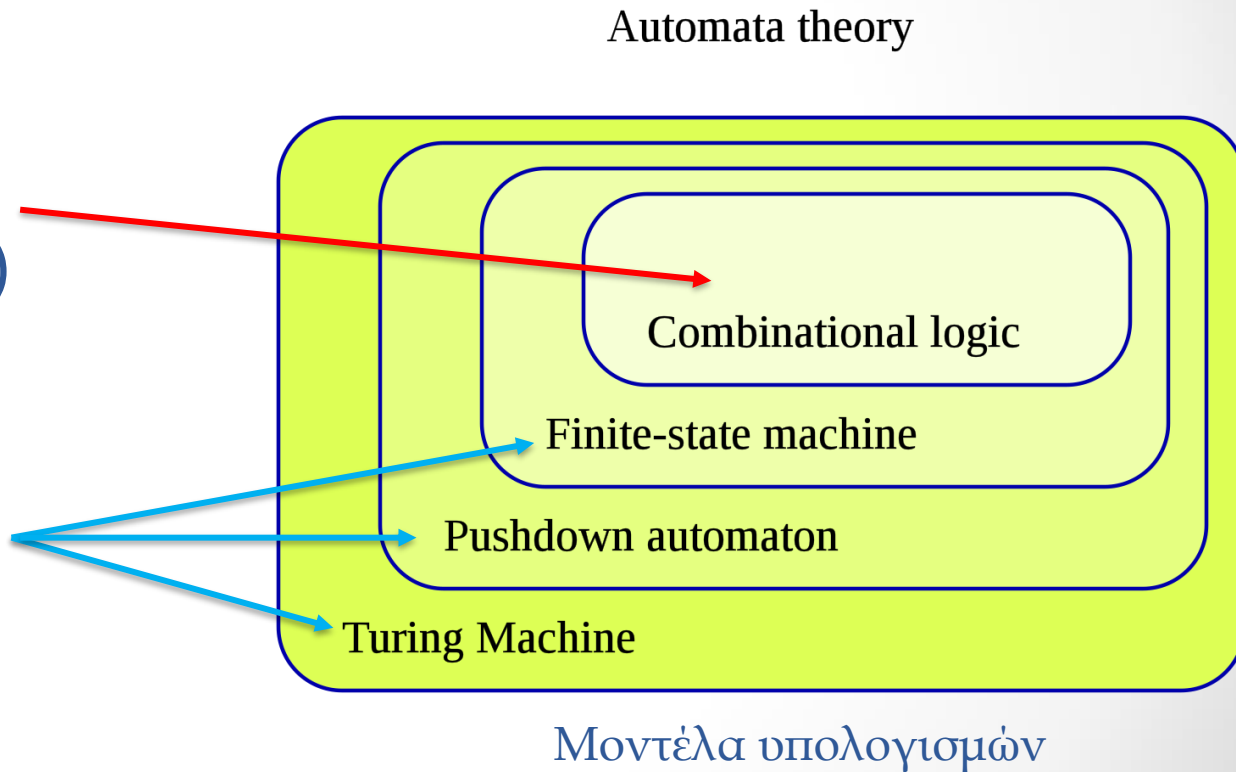


Integrated circuit from an EPROM memory microchip showing the memory blocks, the supporting circuitry and the fine silver wires which connect the integrated circuit die to the legs of the packaging

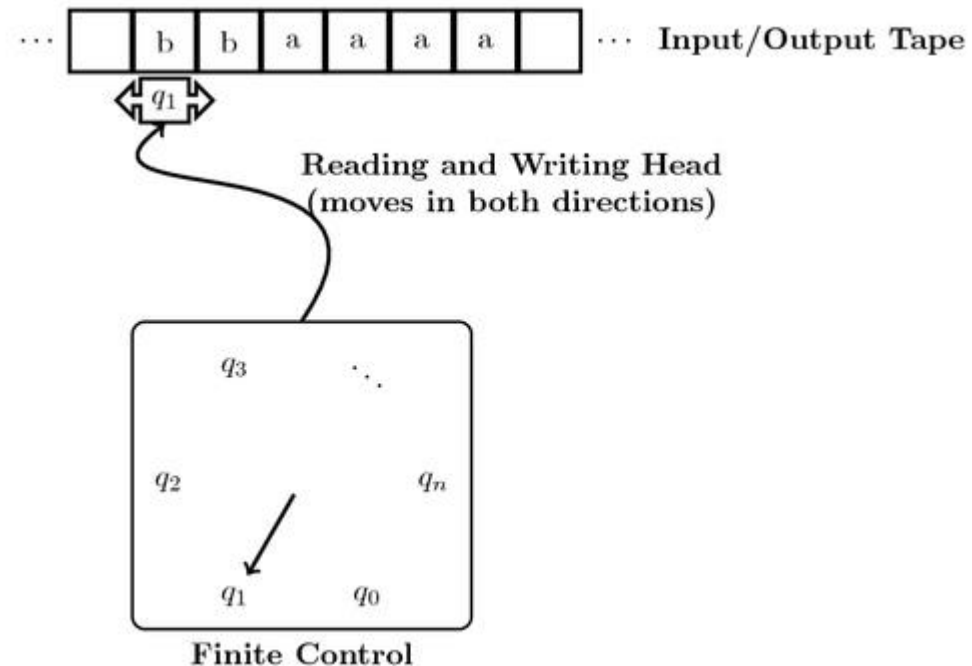
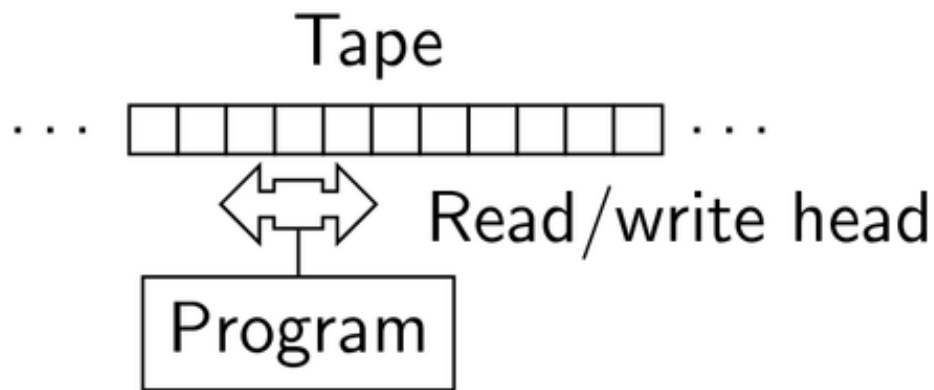
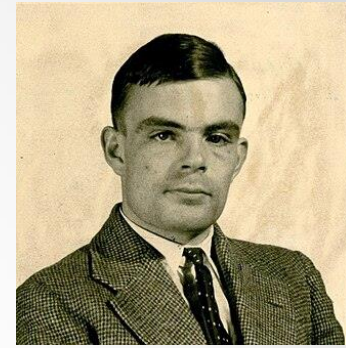


Από την ψηφιακή σχεδίαση στην αρχιτεκτονική Η/Υ

- Συνδυαστικά κυκλώματα (συνδυαστική λογική – combinational logic)
- Ακολουθιακά κυκλώματα (ακολουθιακή λογική – sequential logic) υλοποιεί μνήμη!



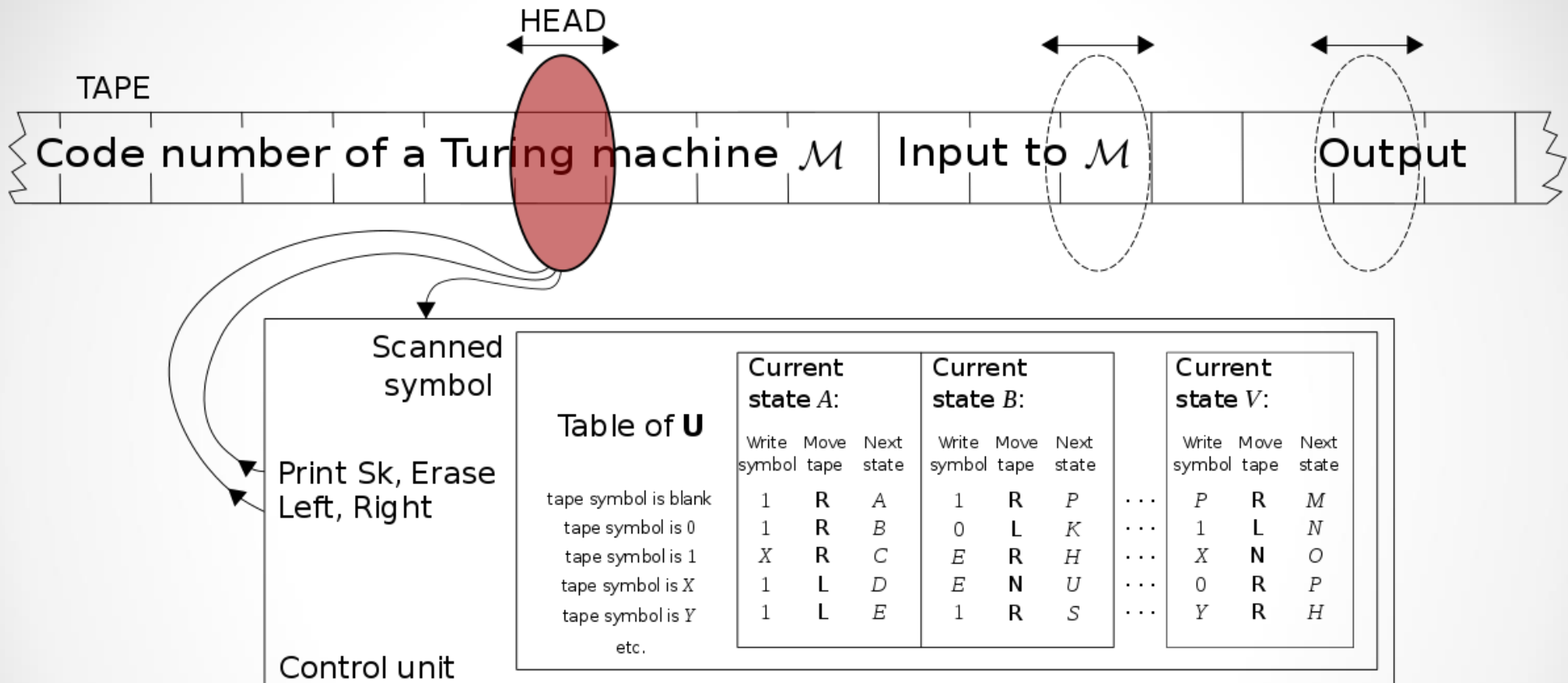
Η μηχανή του Turing



https://en.wikipedia.org/wiki/Turing_machine



Universal Turing Machine

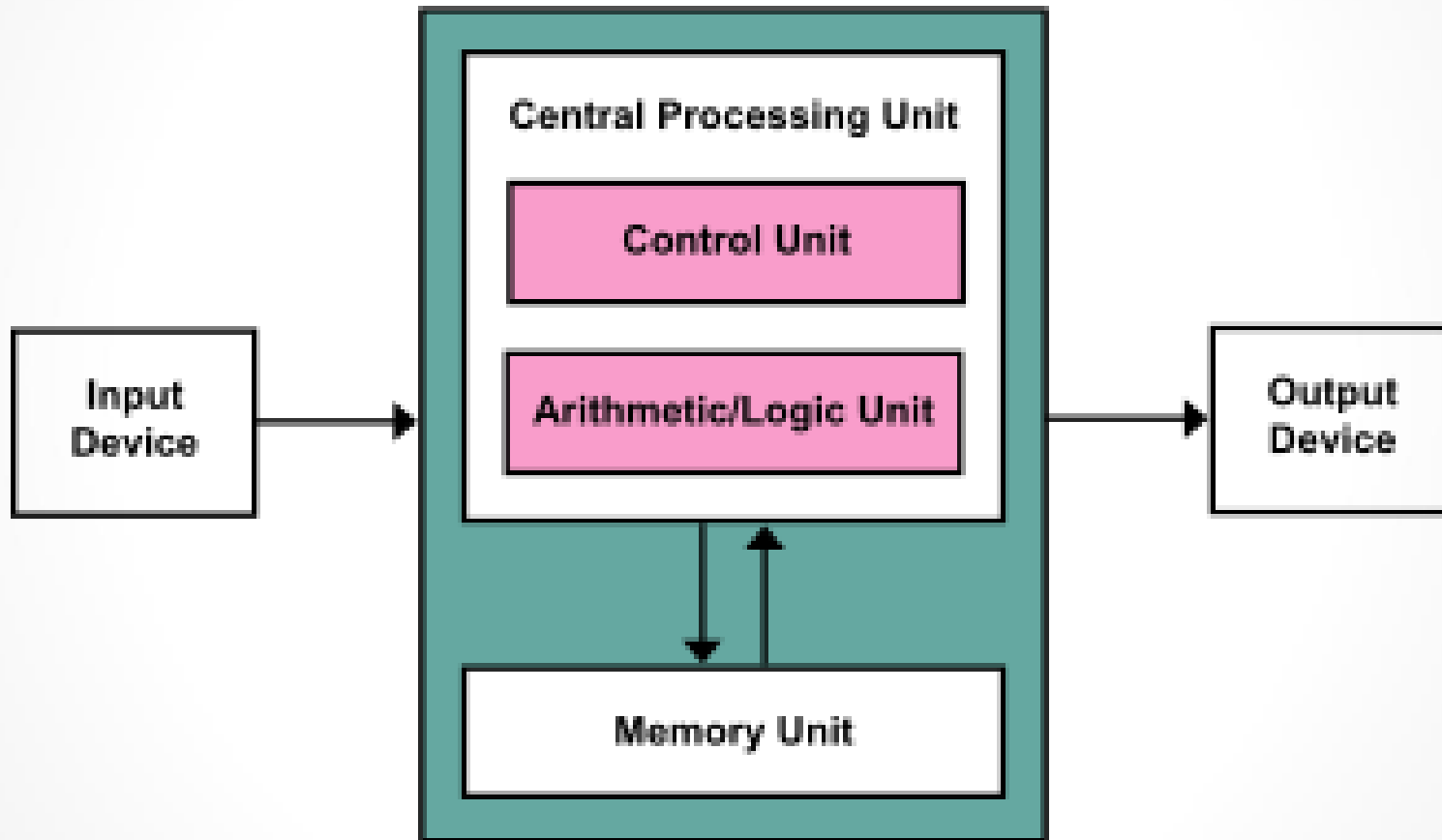


https://en.wikipedia.org/wiki/Universal_Turing_machine

“...the origin of the idea of a stored-program computer used by John von Neumann in 1946 for the "Electronic Computing Instrument" that now bears von Neumann's name: the von Neumann architecture.^[1]”



Αρχιτεκτονική Η/Υ



Αρχιτεκτονική **von Neumann**



ΟΔΗΓΟΣ ΣΠΟΥΔΩΝ
2021 - 2022

Αρχιτεκτονική von Neumann



Οδηγός σπουδών 2021-22
<https://drive.google.com/file/d/1MDznpDkSSWurOSABAUtrrUB8csyNgvEh/view>

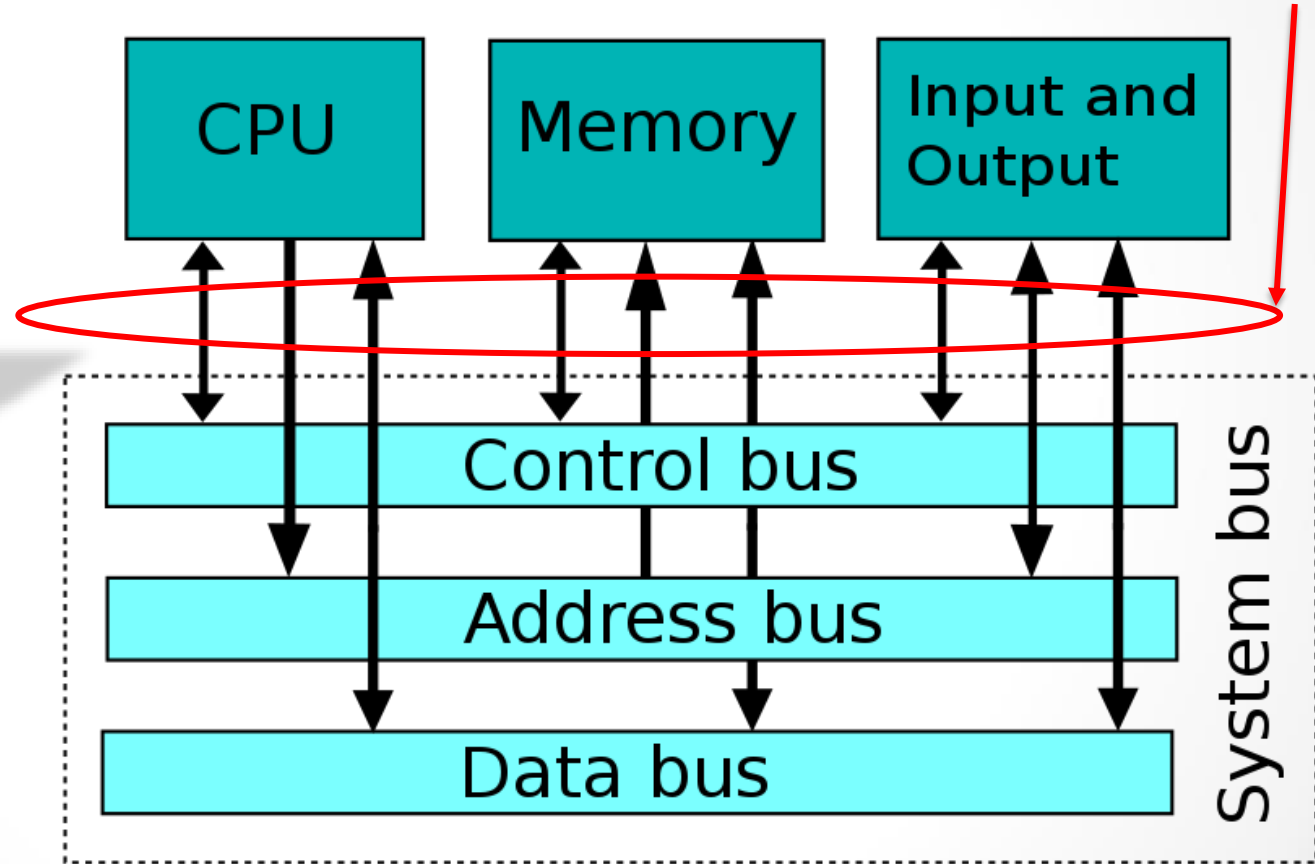


Αρχιτεκτονική Η/Υ

Μεταφορά bits
=>
von Neumann
bottleneck

- προσκόμιση (fetch)
- αποκωδικοποίηση (decode)
- εκτέλεση (execution)
- αποθήκευση (store)

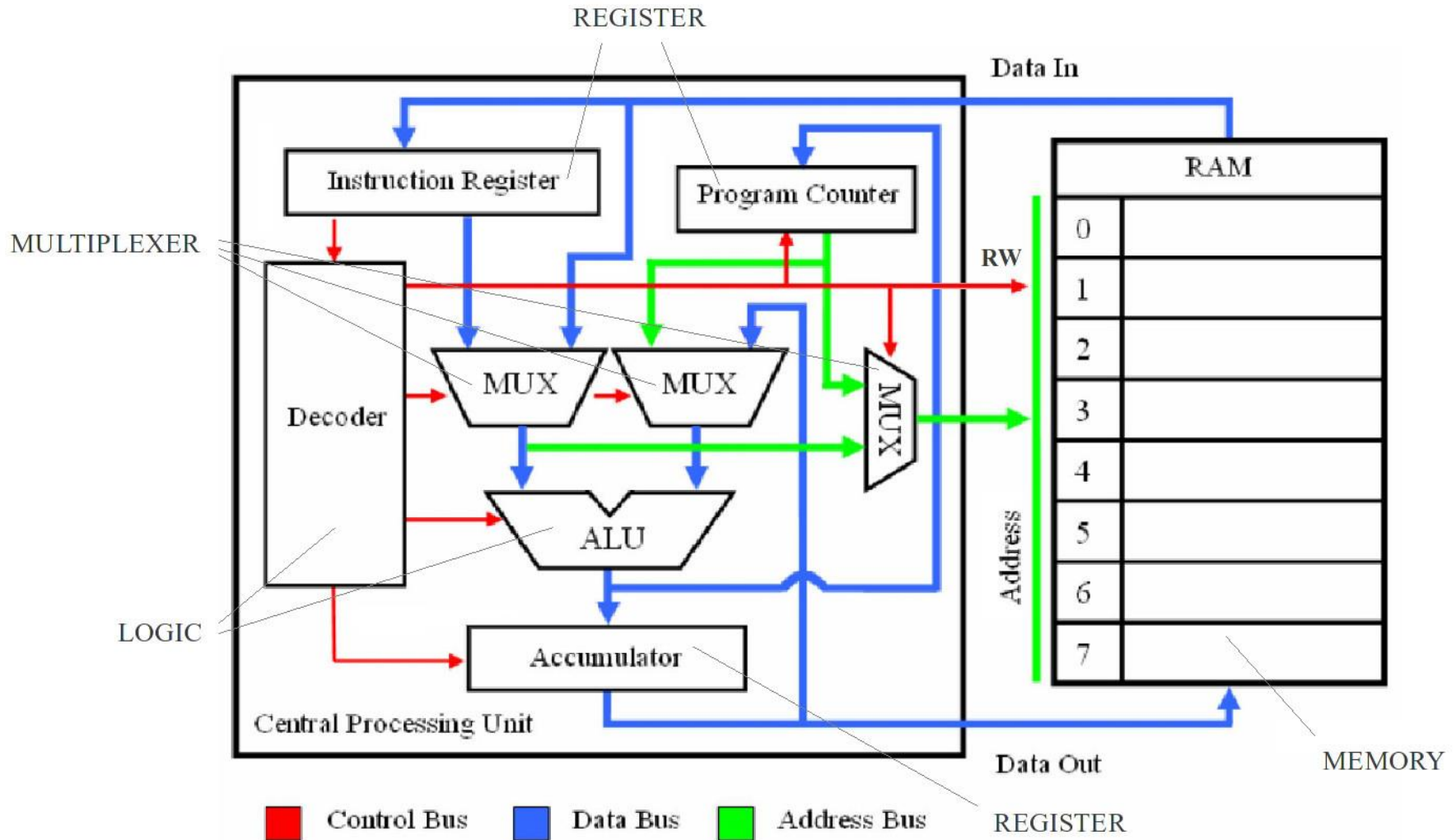
Απλουστευμένη παρουσίαση σταδίων εκτέλεσης εντολής μηχανής στη CPU



Αρχιτεκτονική von Neumann



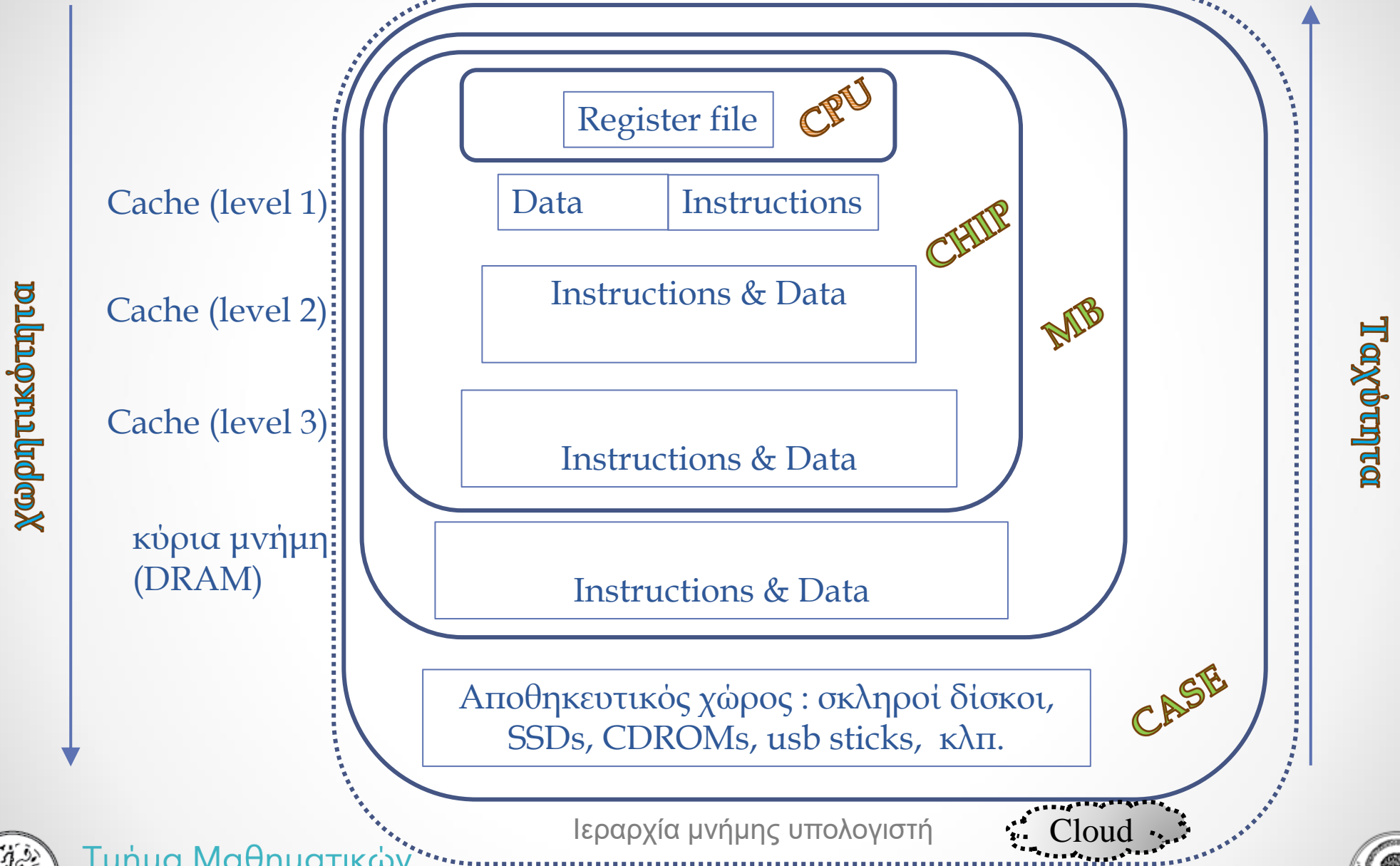
Αρχιτεκτονική Η/Υ



Πηγή : http://www.simplecpudesign.com/simple_cpu_v1/index.html



Αρχιτεκτονική Η/Υ



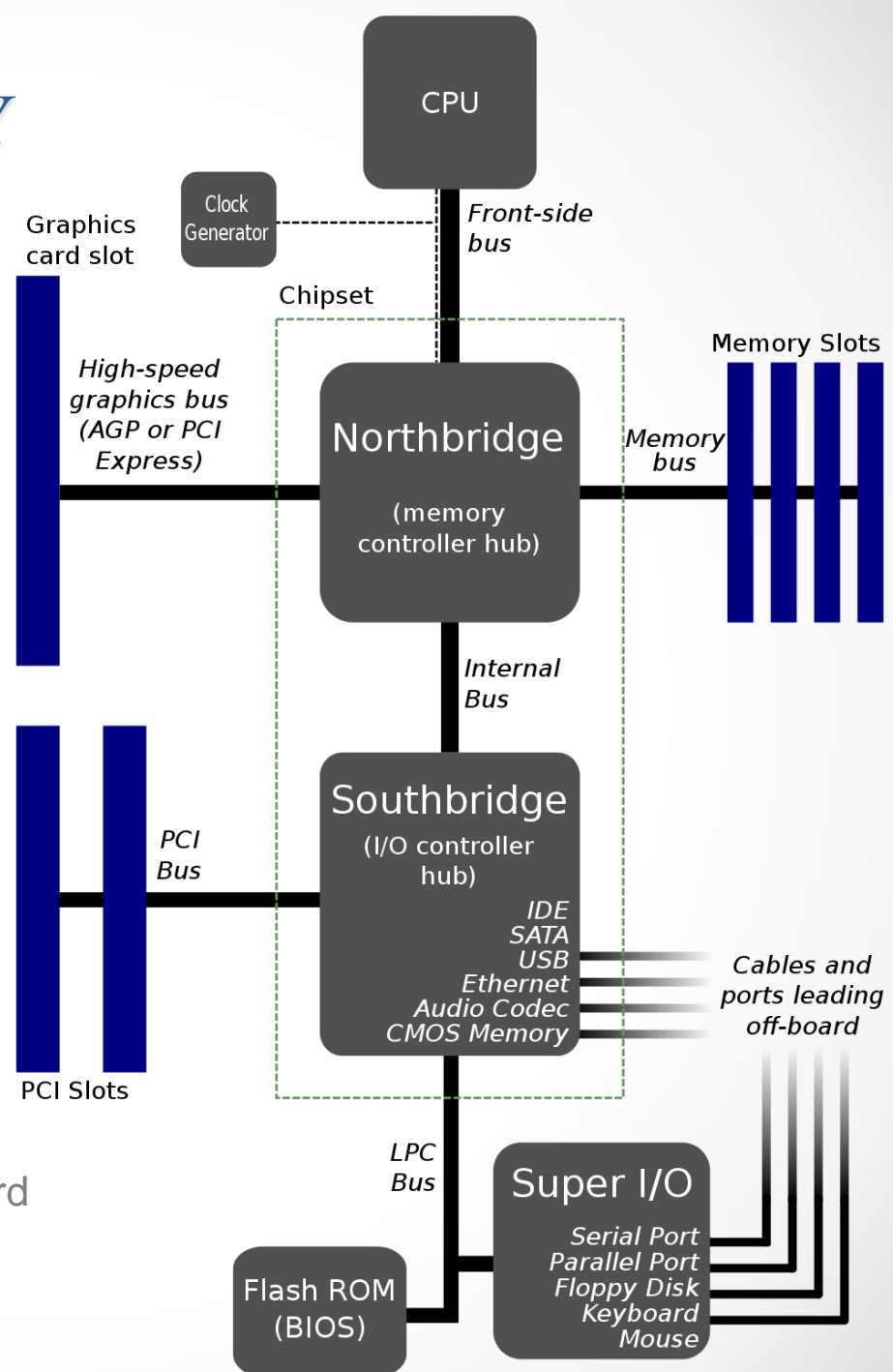
Ιεραρχία μνήμης υπολογιστή

Cloud



Αρχιτεκτονική Η/Υ

Σχηματικό διάγραμμα μητρικής κάρτας PC



Πηγή : <https://en.wikipedia.org/wiki/Motherboard>



Αρχιτεκτονική Η/Υ



Εμπορικά διαθέσιμη (2019) CPU της εταιρίας Intel για PC



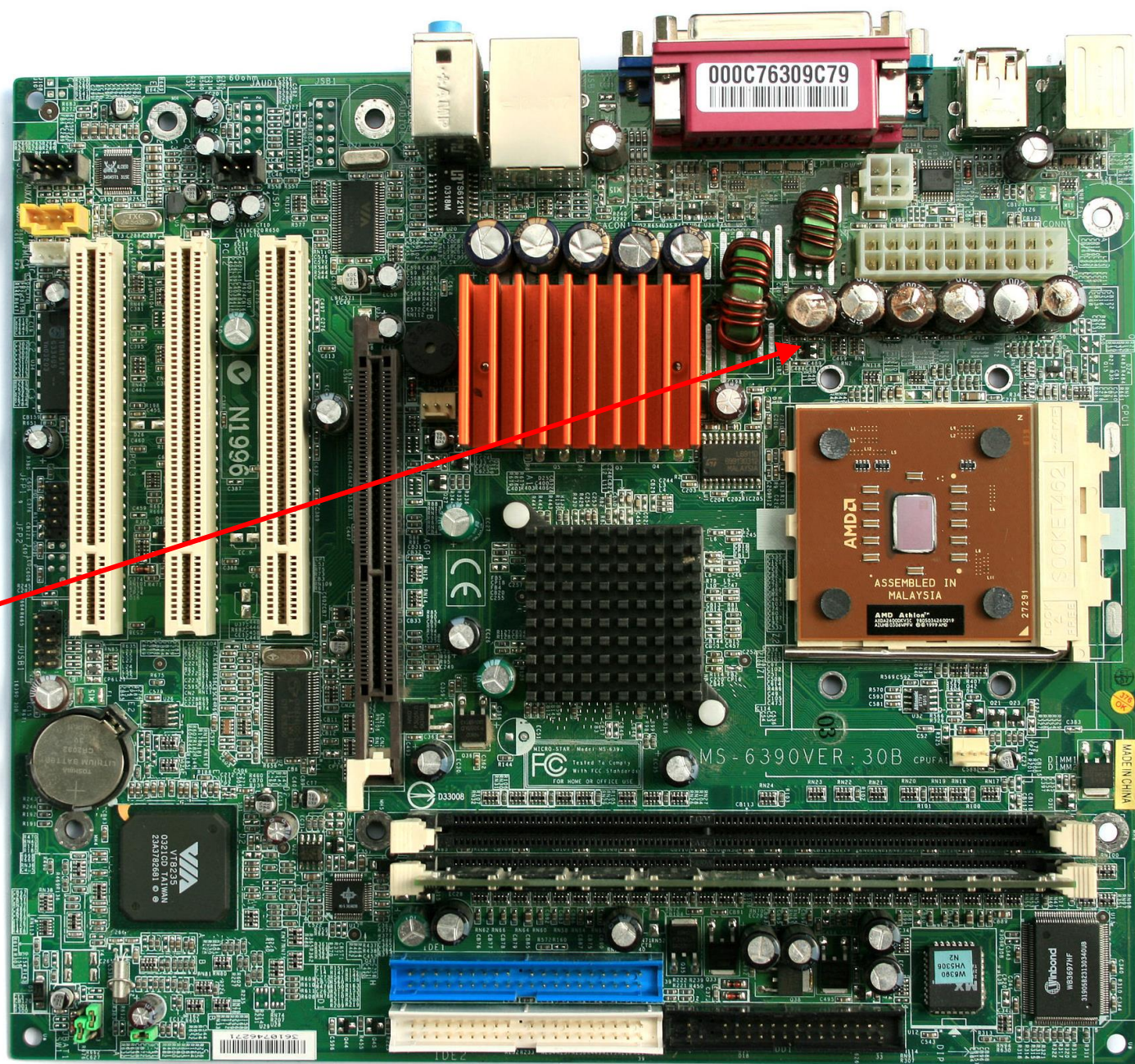
Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



Αρχιτεκτονική Η/Υ

μητρική κάρτα τύπου
microATX για PC
(παρατηρήστε τους
σκασμένους
πυκνωτές)



Πέρα από την αρχιτεκτονική von Neumann: Αντισυμβατική – πρωτότυπη υπολογιστική (Unconventional computing)

PT explain unconventional computing with images



I'm sorry, I am a text-based AI language model and do not have the ability to display images. However, I can describe unconventional computing and its different branches in detail.

Unconventional computing encompasses a wide range of computing paradigms and technologies that depart from the conventional Von Neumann architecture, which is based on a stored program and a central processing unit (CPU) executing instructions in a sequential manner. Some of the branches of unconventional computing include:

1. DNA computing - This branch of computing uses DNA molecules to store and manipulate data and perform computations. In this approach, chemical reactions are used to perform operations on the DNA.
2. Quantum computing - This branch of computing leverages the unique properties of quantum systems, such as superposition and entanglement, to perform computations. Quantum computers use qubits instead of classical bits to perform computations.
3. Optical computing - In this approach, information is processed using light instead of electricity. Optical computing has the potential to be faster and more energy-efficient than electronic computing.
4. Memristor computing - This type of computing uses memristors, which are circuit elements that can change their resistance based on the flow of current, to perform computations.
5. Molecular computing - This branch of computing uses molecular systems, such as molecular electronics and molecular biology, to perform computations.

These are just a few examples of unconventional computing. Each of these branches of unconventional computing offers unique benefits and challenges, and the field is rapidly evolving, with new technologies and approaches being developed all the time.

PT explain unconventional computing with public domain sources



Unconventional computing refers to computing paradigms and technologies that depart from the conventional Von Neumann architecture, which is based on a stored program and a central processing unit (CPU) executing instructions in a sequential manner.

Unconventional computing encompasses a wide range of computing approaches, including quantum computing, DNA computing, molecular computing, optical computing, memristor computing, chaos computing, and more.

One example of unconventional computing is DNA computing. DNA computing is a branch of computing that uses DNA, biochemistry, and molecular biology to solve computational problems. DNA molecules can store and manipulate data and perform computations by performing chemical reactions. In 1994, researchers demonstrated the use of DNA to solve a simple mathematical problem, and since then, DNA computing has been used to solve increasingly complex problems.

Another example of unconventional computing is quantum computing. Quantum computing leverages the unique properties of quantum systems, such as superposition and entanglement, to perform computations. Quantum computers use qubits (quantum bits) instead of classical bits, which enables them to perform certain types of computations much faster than classical computers. For example, quantum computers can efficiently solve problems that are intractable for classical computers, such as breaking cryptographic codes.

Optical computing is another example of unconventional computing, in which information is processed using light instead of electricity. Optical computing has the potential to be faster and more energy-efficient than electronic computing because light can travel faster and carry more information than electrical signals.

These are just a few examples of unconventional computing, and the field is rapidly evolving, with new technologies and approaches being developed all the time. If you want to learn more about unconventional computing, I recommend starting with the Wikipedia articles on the various types of unconventional computing and the general concept of unconventional computing.

Απαντήσεις του ChatGPT Jan 30, 23 Version.
<https://chat.openai.com/chat>



Πέρα από την αρχιτεκτονική von Neumann: Αντισυμβατική – πρωτότυπη υπολογιστική (Unconventional computing)

PT explain unconventional computing with images



I'm sorry, I am a text-based AI language model and do not have the ability to display images. However, I can describe unconventional computing and its different branches in detail.

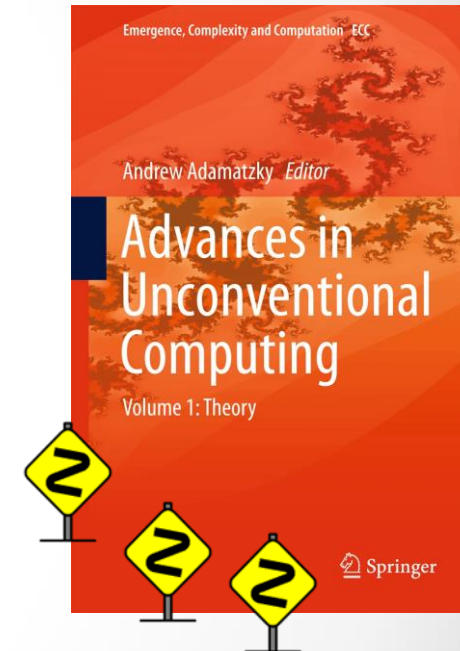
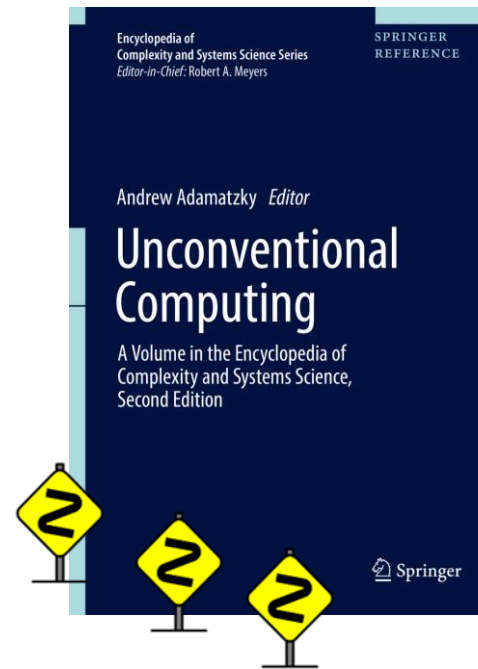
Unconventional computing encompasses a wide range of computing paradigms and technologies that depart from the conventional Von Neumann architecture, which is based on a stored program and a central processing unit (CPU) executing instructions in a sequential manner. Some of the branches of unconventional computing include:

1. DNA computing - This branch of computing uses DNA molecules to store and manipulate data and perform computations. In this approach, chemical reactions are used to perform operations on the DNA.
2. Quantum computing - This branch of computing leverages the unique properties of quantum systems, such as superposition and entanglement, to perform computations. Quantum computers use qubits instead of classical bits to perform computations.
3. Optical computing - In this approach, information is processed using light instead of electricity. Optical computing has the potential to be faster and more energy-efficient than electronic computing.
4. Memristor computing - This type of computing uses memristors, which are circuit elements that can change their resistance based on the flow of current, to perform computations.
5. Molecular computing - This branch of computing uses molecular systems, such as molecular electronics and molecular biology, to perform computations.

These are just a few examples of unconventional computing. Each of these branches of unconventional computing offers unique benefits and challenges, and the field is rapidly evolving, with new technologies and approaches being developed all the time.

Πιο ...συμβατική αναζήτηση:

1. https://en.wikipedia.org/wiki/Unconventional_computing
2. <https://link.springer.com/referencework/10.1007/978-1-4939-6883-1>
3. <https://link.springer.com/book/10.1007/978-3-319-33924-5>



Σε κάθε περίπτωση:
1. Έλεγχος πηγών, είναι αξιόπιστες;
2. Διασταύρωση στοιχείων
3. Ανάγνωση με κριτική σκέψη



*-ware

software

firmware

hardware

```
#include <stdio.h>
int main() {
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```



Περί γλωσσών

Η γλώσσα επικοινωνίας σταδιακά αλλάζει από επίπεδο σε επίπεδο. Διαδοχικά επίπεδα επικοινωνούν - «συνομιλούν» μέσω αμοιβαία κατανοητών στοιχείων τα οποία ορίζουν μια «διεπαφή» (interface) υποβολής και ανταπόκρισης αιτημάτων



Περί γλωσσών

Συμβολική γλώσσα
(assembly language)
για επεξεργαστή της
οικογένειας x86-64

```
_fib:
    movl $1, %eax
.fib_loop:
    cmpl $1, %edi
    jbe .fib_done
    movl %eax, %ecx
    addl %ebx, %eax
    movl %ecx, %ebx
    subl $1, %edi
    jmp .fib_loop
.fib_done:
    ret
```

```
8B542408 83FA0077 06B80000 0000C383
FA027706 B8010000 00C353BB 01000000
B9010000 008D0419 83FA0376 078BD989
C14AEBF1 5BC3
```

Γλώσσα μηχανής (Machine code)
για επεξεργαστή τύπου 32-bit x86

Κατώτερα επίπεδα γλωσσών προγραμματισμού

(https://en.wikipedia.org/wiki/Low-level_programming_language)



Περί γλωσσών

Συμβολικές γλώσσες Wolfram Mathematica,
προγραμματισμού Matlab, κλπ.



python™



Scripting languages

Παράδειγμα κώδικα
στη γλώσσα C

```
unsigned fib(unsigned n) {  
    if (!n)  
        return 0;  
    else if (n <= 2)  
        return 1;  
    else {  
        unsigned a, c;  
        for (a = c = 1; ; --n) {  
            c += a;  
            if (n <= 2) return c;  
            a = c - a;  
        }  
    }  
}
```

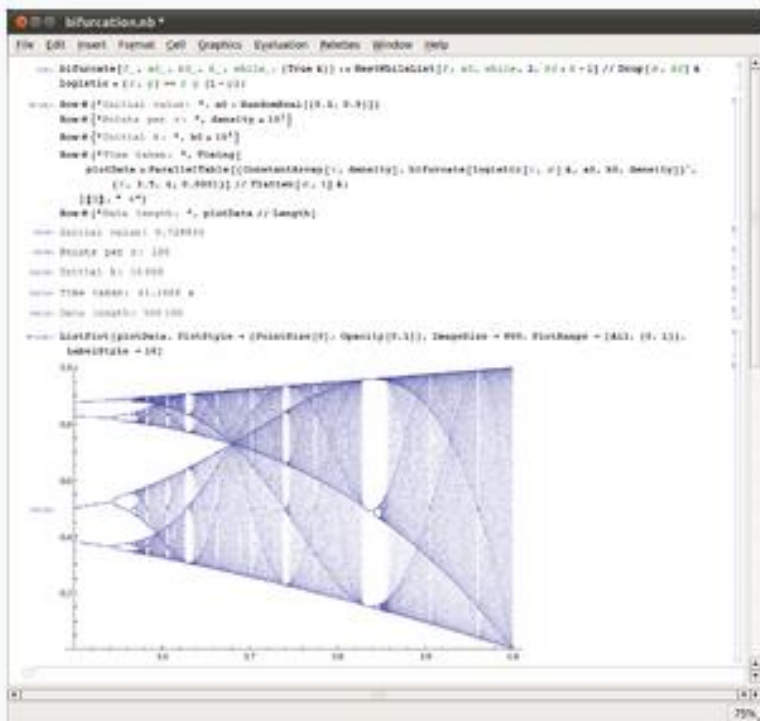
Ανώτερα επίπεδα γλωσσών προγραμματισμού

(https://en.wikipedia.org/wiki/High-level_programming_language)



Περί γλωσσών

Wolfram Mathematica



Mathematica 8.0.0 Linux frontend

The screenshot shows the Wolfram Documentation Center interface. The browser window title is "2-D and 3-D Plots — Examples". The page features a search bar and a "CONTENTS" sidebar on the left. The main content area is titled "2-D and 3-D Plots — Examples" and contains three example cards:

- Create 2-D Line Plot**: Includes a line plot of a sine wave and instructions to "Create a 2-D line plot and specify the line style, line color, and marker type." with an "Open Live Script" link.
- Create Common 2-D Plots**: Includes a scatter plot and instructions to "Create a variety of 2-D plots in MATLAB®." with an "Open Live Script" link.
- Overlay Bar Graphs**: Includes a bar chart with two series and instructions to "Overlay two bar graphs and specify the bar colors and widths. Then, it shows how to add a legend, display the grid lines, and specify the tick" with an "Open Live Script" link.

Συμβολικές γλώσσες προγραμματισμού

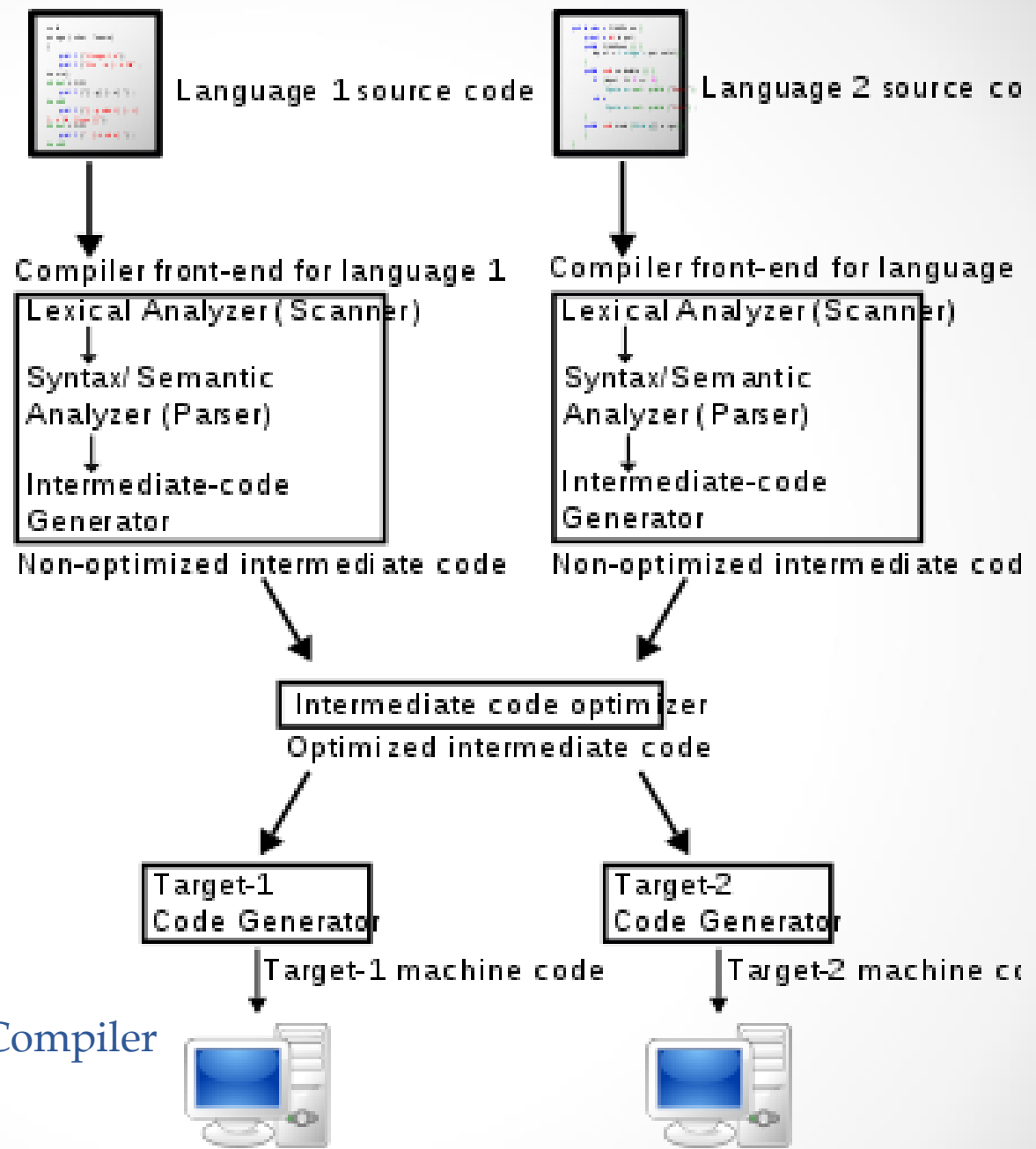


Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



Μεταφραστές



<https://en.wikipedia.org/wiki/Compiler>

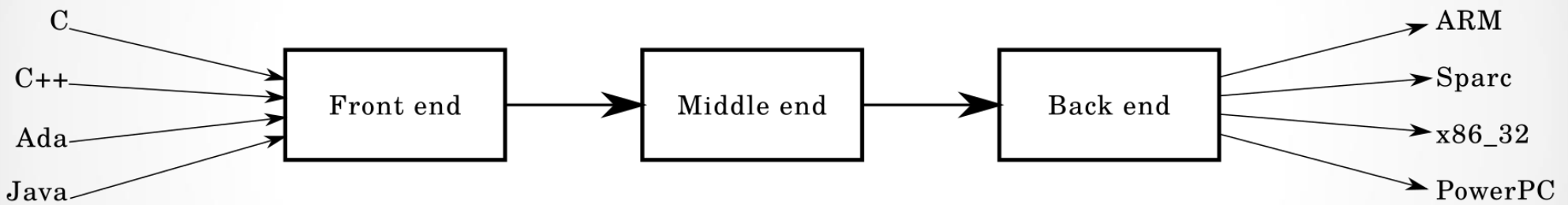


Τμήμα Μαθηματικών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης



Μεταφραστές



Μεταφραστές

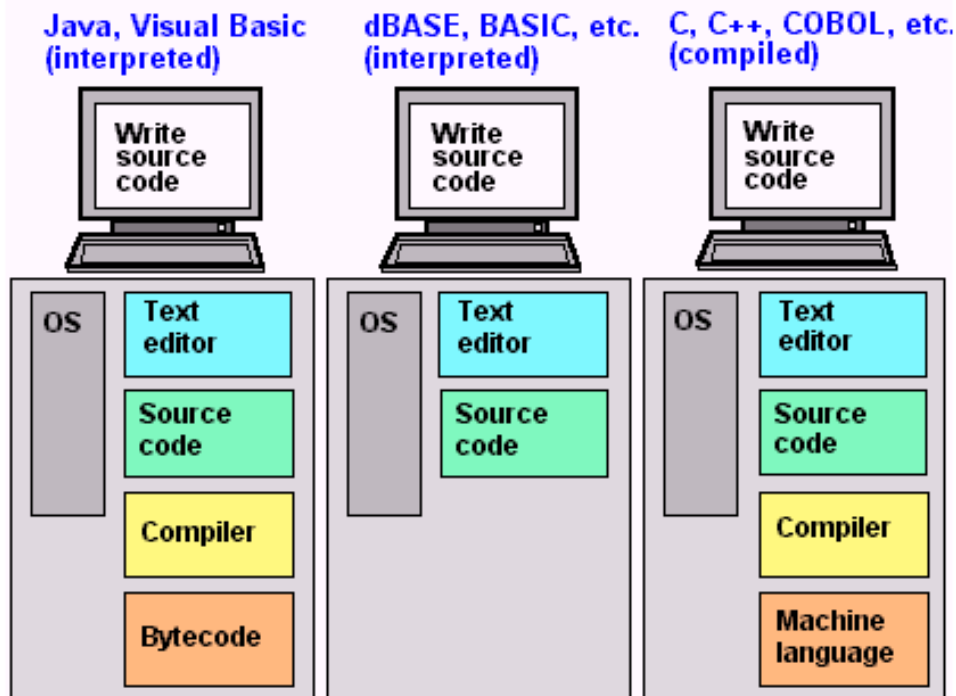
The screenshot shows the Dev-C++ 5.11 IDE interface. The main window displays the 'About Dev-C++ :)' dialog box, which provides detailed information about the software, including its version (5.11), build time (April 27 2015 - 17:11), and license (GNU General Public License, Version 2, June 1991). It also lists contact information, a 'Donate' button, and links to various resources like the repository, subreddits, and blogs. A second 'Information' dialog box is open, listing the authors and contributors: Johan Mes (Post-4.9.9.2 development), Colin Laplace, Mike Berg, Hongli Lai, Yiannis Mandravellos (Development), Peter Schraut, Marek Januszewski, Anonymous (Contributors), MinGW compiler system (Mumit Khan, J.J. van der Heijden, Colin Hendrix and GNU developers), Splash screen and association icons (Matthijs Crielaard), New Look theme (Gerard Caufield), Gnome icons (Gnome designers), and Blue theme (Thomas Thron). The 'OK' button is visible in the bottom right of the 'Information' dialog.



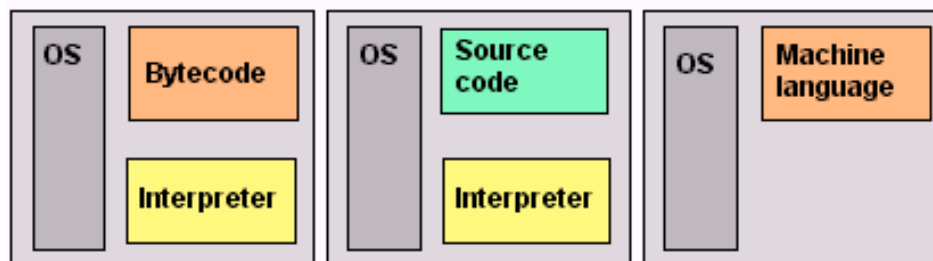
Μεταφραστές VS διερμηνευτές

From Computer Desktop Encyclopedia
© 2000 The Computer Language Co. Inc.

Create & Modify



Run



Ερωτήσεις & Απαντήσεις

