v.03_10_2024

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ | DEMOCRITUS UNIVERSITY OF THRACE

ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ
ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ (106ΕΥΥΚ)
ΧΕΙΜΕΡΙΝΟ 2024-2025

Διάλεξη Νο1:
Συστήματα Αριθμών

Δ. Καραμπατζάκης, Επίκουρος Καθηγητής

email. dkara@cs.duth.gr

Δήλωση προσβασιμότητας

Σε αυτό το μάθημα όλες/οι οι φοιτήτριες/τές απολαμβάνουν – και αντίστοιχα υποχρεούνται να σέβονται – το δικαίωμα της ίσης μεταχείρισης. Δεν είναι ανεκτή και αποδεκτή κανενός τύπου και μορφής διάκριση με κριτήρια την εθνικότητα, τη φυλή, την καταγωγή, τη γλώσσα, το φύλο, τη θρησκεία, την ηλικία, την υγεία, τη σωματική ικανότητα, την ιδιωτική ζωή, τον γενετήσιο προσανατολισμό, τη σωματική ικανότητα και την οικονομική και κοινωνική κατάσταση στην οποία αυτοί βρίσκονται.

Το Πανεπιστήμιο άγρυπνα μεριμνά για τη διασφάλιση της αρχής των ίσων ευκαιριών και της ίσης μεταχείρισης. Οι κοινωνικές προκαταλήψεις και οι ιδεολογικές παρωπίδες είναι έννοιες τελείως ξένες με την επιστημονική πρόοδο την οποία το Πανεπιστήμιο είναι ταγμένο να υπηρετεί.

Ο Διδάσκων

Πληροφορίες για το Μάθημα

Διδάσκων:

Δημήτρης Καραμπατζάκης, Επίκουρος Καθηγητής

Αναλογικά και Ψηφιακά Ηλεκτρονικά Συστήματα

Μέλος Εργαστηρίου Βιομηχανικών και Εκπαιδευτικών Ενσωματωμένων Συστημάτων

Επικοινωνία / πληροφορίες:

Email. dkara@cs.duth.gr

web. http://www.internetofthings.gr/

Ώρες Γραφείου:

Τετάρτη και Πέμπτη 10.00 π.μ. -12.00 μ.μ.,

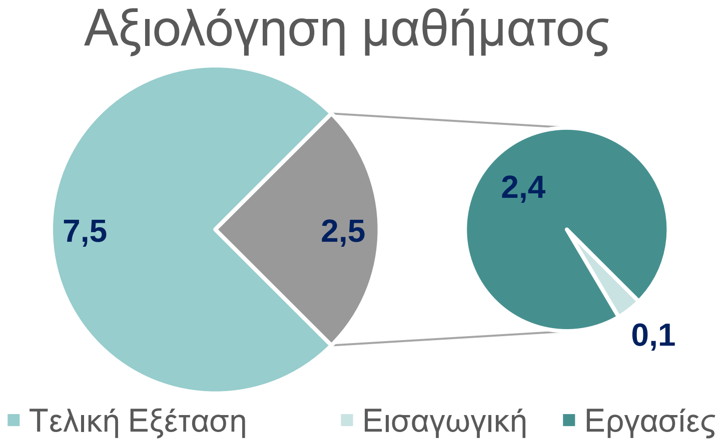
μετά από συνεννόηση με email στο ΦΕ 315 (πάνω από αιθ. Α1)

Πληροφορίες για το Μάθημα (Γενικές)

- **Κάθε Τρίτη 10.00 π.μ. - 12.00 μ.μ. και Πέμπτη 13.00 μ.μ. - 15.00 μ.μ. μάθημα θεωρίας στο Μεγάλο Αμφιθέατρο (μπορεί να αλλάζει με ανακοινώσεις).**
- **Η διαχείριση του μαθήματος θα γίνει με χρήση της υπηρεσίας https://courses.cs.duth.gr**
- **Όλοι οι φοιτητές πρέπει να έχουν λογαριασμό στο uregister.**
- **Η ιστοσελίδα με τις πληροφορίες του μαθήματος: http://iees.cs.ihu.gr/?page_id=3096**
- **Υλικό του μαθήματος στο moodle: https://moodle.cs.duth.gr/**

Πληροφορίες για το Μάθημα (Αξιολόγηση)

- **Η βαθμολογία είναι 75% από την τελική εξέταση και 25% από τις ατομικές εργασίες (1+1 σετ ασκήσεων) που θα δοθούν για το σπίτι.**
- **Η τελική εξέταση είναι με ανοιχτό το κύριο σύγγραμμα του μαθήματος.**
- **Ο βαθμός του μαθήματος (ΒΜ = ΓΕ*0,75 + ΣΑ*0,25) πρέπει να είναι τουλάχιστον πέντε (5).**



Πληροφορίες για το Μάθημα (Μονάδες)

- **Κωδικός Μαθήματος:** 106ΕΥΥΚ
- **Εξάμηνο:** 1ο
- **Τύπος Μαθήματος:** Υποβάθρου, Ανάπτυξης Δεξιοτήτων
- **Είδος Μαθήματος:** Υποχρεωτικό (ΥΠ)
- **Διδασκαλία Θεωρίας:** 3 ώρες/εβδομάδα
- **Διδασκαλία Φροντιστήριο:** 1 ώρες/εβδομάδα
- **Πιστωτικές μονάδες ECTS: 7**
- **Γλώσσα διδασκαλίας και Εξετάσεων:** Ελληνικά

Πληροφορίες για το Μάθημα (Φόρτος)

Δραστηριότητα	Φόρτος εργασίας εξαμήνου
Διαλέξεις	78 ώρες
Φροντιστηριακές Ασκήσεις	26 ώρες
Γραπτές Εξετάσεις	2 ώρες
Γραπτές Εργασίες	34 ώρες
Αυτοτελής Μελέτη	35 ώρες
Σύνολο	**175 ώρες (7 ECTS)**

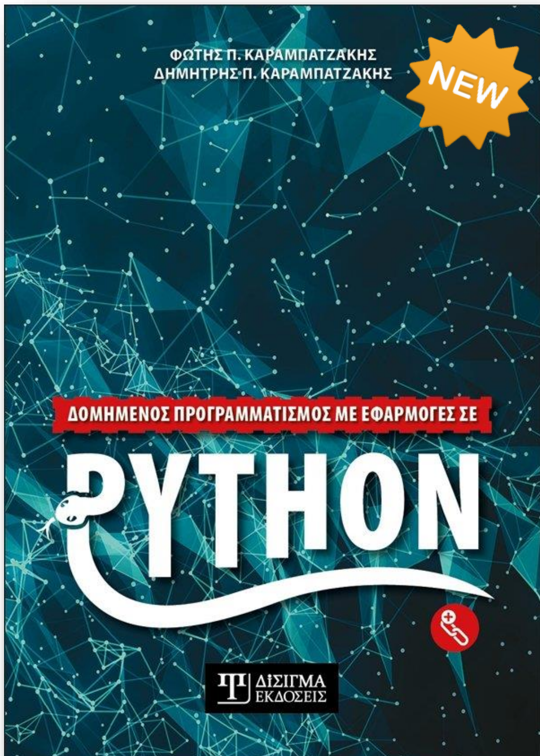
Κύριο Σύγγραμμα Μαθήματος (ΕΥΔΟΞΟΣ)

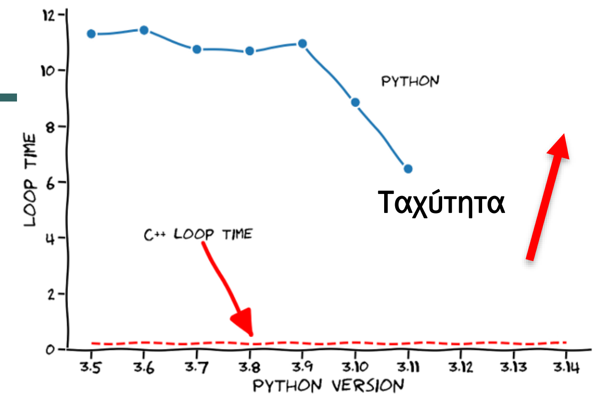
Ψηφιακή Σχεδίαση
με τις Γλώσσες VHDL και Verilog

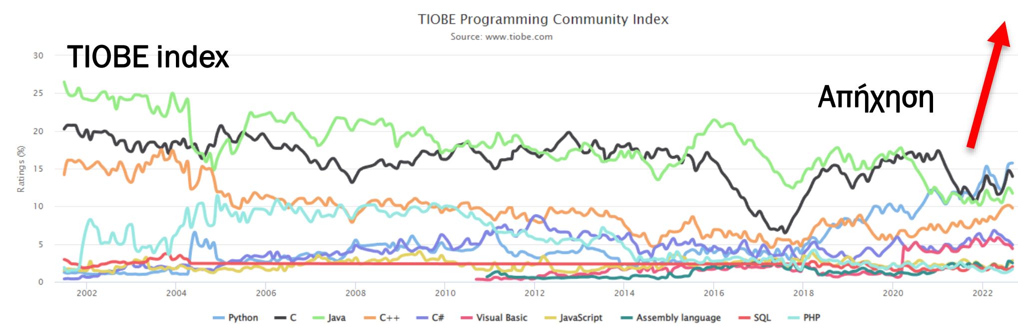
Συγγραφέας: Πογαρίδης Δημήτριος

Έτος Έκδοσης: 2019
Κωδικός στον Εύδοξο: **86192991**

Δομημένος Προγραμματισμός με εφαρμογές σε Python







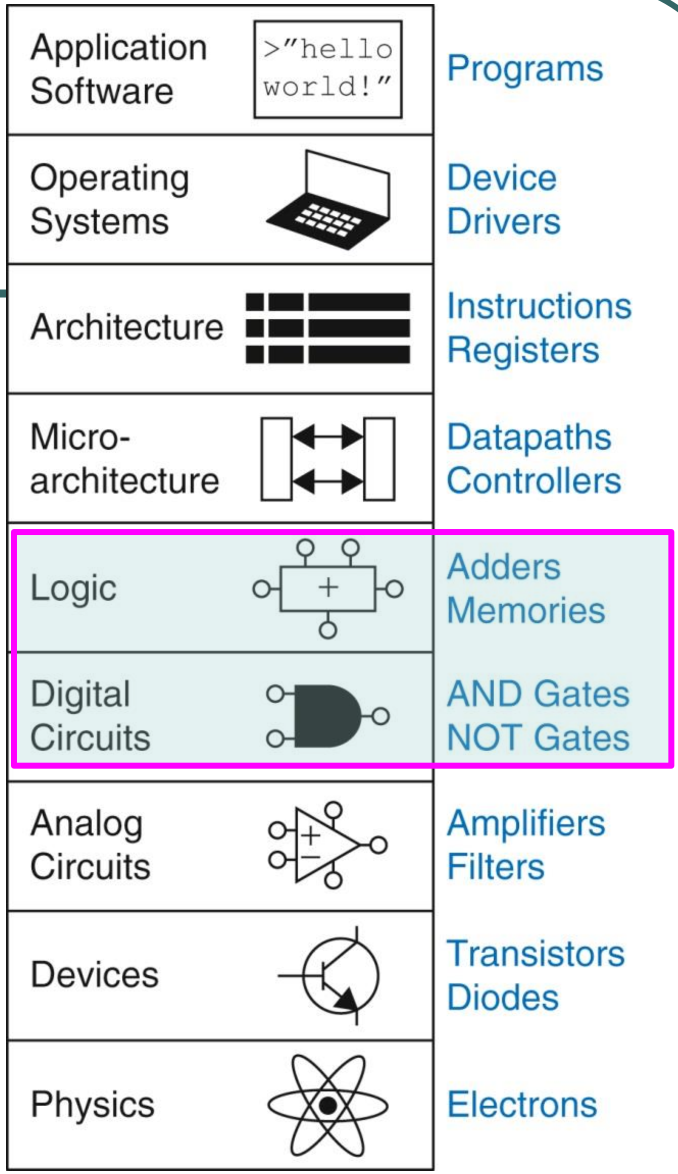
Sep 2022	Sep 2021	Change	Programming Language	Ratings	Change
1	2	^	Python	15.74%	+4.07%
2	1	v	C	13.96%	+2.13%
3	3		Java	11.72%	+0.60%
4	4		C++	9.76%	+2.63%
5	5		C#	4.88%	-0.89%

Συνοδευτικό Εκπαιδευτικό Υλικό

Μπορείτε να κατεβάσετε δωρεάν συνοδευτικό εκπαιδευτικό υλικό σε ψηφιακή μορφή (π.χ. κώδικες ΓΛΩΣΣΑ, Python, Circuitpython, επιλεγμένες λύσεις ασκήσεων, παροράματα κλπ.) από την ιστοσελίδα του βιβλίου στον διαδικτυακό τόπο **www.disigma.gr** εντοπίζοντας το συγκεκριμένο βιβλίο ή πληκτρολογώντας κατευθείαν την ιστοσελίδα **https://disigma.gr/products/domimenos-programmatismos-python**.

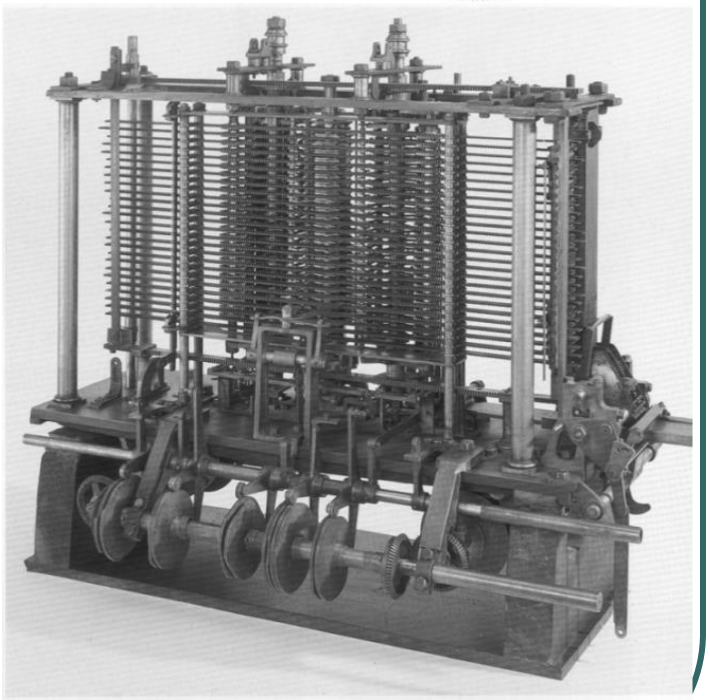
Φώτης Π. Καραμπατζάκης | Δημήτρης Π. Καραμπατζάκης
ISBN: 978-618-202-102-6, Έκδοση: **2022**

Επίπεδα Αφαίρεσης

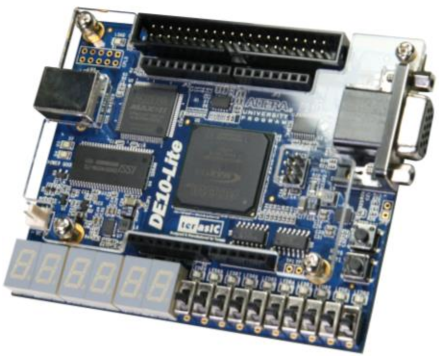


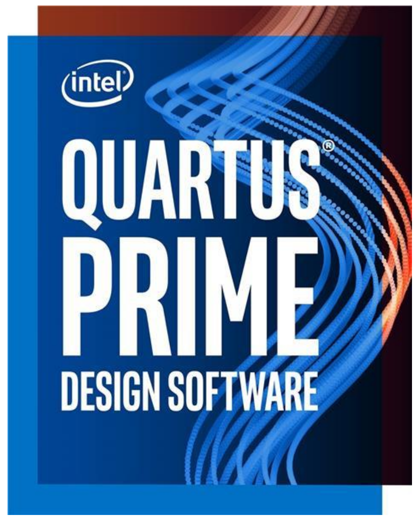
Η Αναλυτική Μηχανή

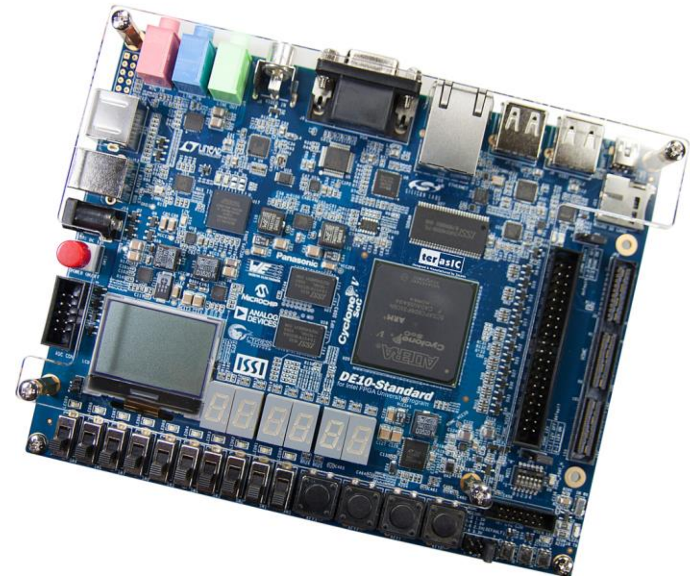
- Σχεδιάστηκε από τον **Charles Babbage**, από το 1834 έως το 1871
- Θωρείται ως ο πρώτος ψηφιακός υπολογιστής
- Χρησιμοποιούσε μηχανικά γρανάζια, όπου κάθε γρανάζι αναπαριστούσε μια διακριτή τιμή (0-9)
- Ο Babbage πέθανε προτού ολοκληρωθεί η κατασκευή της



Intel FPGA University Program







DE10-Lite

MAX10 FPGA

Arduino connector

10 τεμ.

DE10-Standard

Cyclone V FPGA SoC

Dual-core Cortex-A9 HPS ARM

10 τεμ.

Συστήματα Αριθμών

Παράσταση φυσικού αριθμού

Κάθε φυσικός αριθμός μπορεί να παρασταθεί με την παρακάτω σχέση:

$$N = \underbrace{d_{-n}r^{-n} + \ldots + d_{-2}r^{-2} + d_{-1}r^{-1}}_{N_1} + \underbrace{d_0 r^0 + d_1 r^1 + d_2 r^2 + \ldots + d_{m-1}r^{m-1}}_{N_2}$$

Όπου: ***N*** είναι ο φυσικός αριθμός με κλασματικό μέρος ***(N₁)*** και ακέραιο μέρος ***(N₂)***.

r είναι η βάση (radix ή base) του συστήματος, που πρέπει να είναι ένας αριθμός ακέραιος και θετικός και άρα μεγαλύτερος της μονάδας.

d είναι τα ψηφία (digits) του αριθμού που παίρνουν τιμές μεταξύ **0** και ***(r-1)***.

n είναι το πλήθος των κλασματικών ψηφίων του αριθμού

m είναι το πλήθος των ακεραίων ψηφίων του αριθμού ***N***

i η τάξη του ψηφίου $\boldsymbol{d_i}$

Δεκαδικό Σύστημα (DEC)

Η βάση του συστήματος αυτού είναι το **10**.

Δηλαδή ***r* = 10**.

Τα ψηφία του δεκαδικού συστήματος είναι συνολικά **10**,

από το **0** έως και το **9**

(*r*-1 = 10-1 = 9)

Δεκαδικό Σύστημα

Σύμφωνα με την εξίσωση:

$$N = \underbrace{d_{-n}r^{-n}+\ldots+d_{-2}r^{-2}+d_{-1}r^{-1}}_{N_1}+\underbrace{d_0r^0+d_1r^1+d_2r^2+\ldots+d_{m-1}r^{m-1}}_{N_2}$$

ο δεκαδικός αριθμός $N=197{,}268_{10}$ γράφεται σαν:

$$N=197{,}268_{10}=8x10^{-3}+6x10^{-2}+2x10^{-1}+7x10^{0}+9x10^{1}+1x10^{2}$$

$$=0{,}008+0{,}06+0{,}2+7+90+100=197{,}268_{10}$$

Δυαδικό Σύστημα (BIN)

Η βάση του συστήματος αυτού είναι το **2.**

Δηλαδή ***r*** **= 2**.

Τα ψηφία του δυαδικού συστήματος είναι συνολικά **2,**

το **0** και το **1**

(*r*-1 = 2-1 = 1)

Δυαδικό Σύστημα

Σύμφωνα με την εξίσωση:

$$N = \underbrace{d_{-n}r^{-n}+\ldots+d_{-2}r^{-2}+d_{-1}r^{-1}}_{N_1}+\underbrace{d_0r^0+d_1r^1+d_2r^2+\ldots+d_{m-1}r^{m-1}}_{N_2}$$

ο δυαδικός αριθμός $N=101{,}010_2$ γράφεται σαν:

$$N=101{,}010_2=0x2^{-3}+1x2^{-2}+0x2^{-1}+1x2^0+0x2^1+1x2^2$$

$$=0+0.25+0+1+0+4=5{,}25_{10}$$

Δυαδικό Σύστημα

Ο δυαδικός αριθμός π.χ. 10010_2 αποτελείται από πέντε δυαδικά ψηφία από τα οποία

το Λιγότερο Σημαντικό Ψηφίο ή LSB (αρχικά των λέξεων Least Significant Bit) είναι το 0 ενώ

το Περισσότερο Σημαντικό Ψηφίο ή MSB (αρχικά των λέξεων Most Significant Bit) είναι το 1

(η σημαντικότητα αυξάνει από τα δεξιά προς τ' αριστερά).

Δυαδικό Σύστημα

Δύο Διακριτές τιμές:

1’s και 0’s

1, TRUE, HIGH

0, FALSE, LOW

“Ψηφιακή Αφαίρεση”

1 & 0: επίπεδα τάσης, επίπεδα υγρού, περιστρεφόμενα γρανάζια κλπ. Τα ψηφιακά κυκλώματα χρησιμοποιούν τα επίπεδα τάσης (**voltage**) για την αναπαράσταση του 1 και του 0 (π.χ. 5V και 0V αντίστοιχα)

Το κάθε δυαδικό ψηφίο ονομάζεται και BIT (σύντμηση των λέξεων Binary digIT).

Μια ομάδα ψηφίων αποτελεί μια ψηφιολέξη.

Η ψηφιολέξη 4 ψηφίων ονομάζεται ένα nibble, και η ψηφιολέξη των 8 ψηφίων ένα byte.

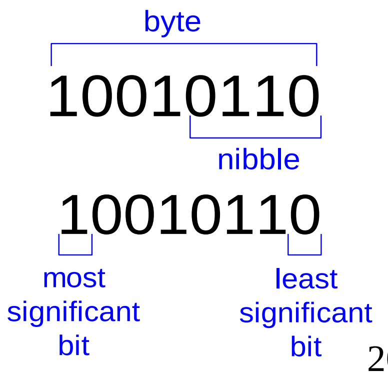


Table 1.1 Binary numbers and their decimal equivalent

1-Bit Binary Numbers	2-Bit Binary Numbers	3-Bit Binary Numbers	4-Bit Binary Numbers	Decimal Equivalents
0	00	000	0000	0
1	01	001	0001	1
	10	010	0010	2
	11	011	0011	3
		100	0100	4
		101	0101	5
		110	0110	6
		111	0111	7
			1000	8
			1001	9
			1010	10
			1011	11
			1100	12
			1101	13
			1110	14
			1111	15

Οκταδικό Σύστημα (OCT)

Η βάση του συστήματος αυτού είναι το **8**.

Δηλαδή ***r*** **= 8**.

Τα ψηφία του οκταδικού συστήματος είναι συνολικά **8**,

από το **0** έως και το **7**

(*r*-1 = 8-1 = 7)

Οκταδικό Σύστημα

Σύμφωνα με την εξίσωση:

$$N = \underbrace{d_{-n}r^{-n} + \ldots + d_{-2}r^{-2} + d_{-1}r^{-1}}_{N_1} + \underbrace{d_0r^0 + d_1r^1 + d_2r^2 + \ldots + d_{m-1}r^{m-1}}_{N_2}$$

ο οκταδικός αριθμός N=102,736$_8$ γράφεται σαν:

$$N=102{,}736_8=6x8^{-3}+3x8^{-2}+7x8^{-1}+2x8^0+0x8^1+1x8^2$$

$$=0{,}93359375+66=66{,}9335_{10}$$

Δεκαεξαδικό σύστημα

Η βάση του συστήματος αυτού είναι το 16.

Δηλαδή *r* = 16.

Τα ψηφία του δεκαεξαδικού συστήματος είναι συνολικά 16,

από το 0 έως και το 15

(*r*-1 = 16-1 = 15)

Χρησιμοποιούνται, επομένως, τα δέκα ψηφία (0-9) του δεκαδικού συστήματος και τα γράμματα του Λατινικού αλφάβητου A, B, C, D, E, F για τους αριθμούς 10, 11, 12, 13, 14 και 15 αντίστοιχα.

Table 1.2 **Hexadecimal number system**

Hexadecimal Digit	Decimal Equivalent	Binary Equivalent
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Δεκαεξαδικό σύστημα (HEX)

Σύμφωνα με την εξίσωση:

$$N = \underbrace{d_{-n}r^{-n} + \ldots + d_{-2}r^{-2} + d_{-1}r^{-1}}_{N_1} + \underbrace{d_0 r^0 + d_1 r^1 + d_2 r^2 + \ldots + d_{m-1}r^{m-1}}_{N_2}$$

ο δεκαεξαδικός αριθμός $N=A39{,}6CB_{16}$ γράφεται σαν:

$$\mathbf{N=11x16^{-3}+12x16^{-2}+6x16^{-1}+9x16^{0}+3x16^{1}+10x16^{2}}$$

$$\mathbf{=2617{,}424_{10}}$$

Δεκαδικός σε άλλο σύστημα

Οι μετατροπές από το δεκαδικό σ' ένα άλλο σύστημα γίνονται με τον παρακάτω τρόπο:

α. Χωρίζεται ο δεκαδικός αριθμός στο ακέραιο και το κλασματικό του μέρος.

β. Διαιρείται το ακέραιο μέρος με τη βάση του νέου συστήματος.

γ. Πολλαπλασιάζεται το κλασματικό μέρος με την βάση του νέου συστήματος.

Δεκαδικός σε δυαδικό

ΠΑΡΑΔΕΙΓΜΑ 1.1

Να μετατραπεί ο δεκαδικός αριθμός 21,375 στον αντίστοιχο δυαδικό.

Δεκαδικός σε δυαδικό

Ακέραιο μέρος

21:2

LSB 1 10:2

0 5:2

1 2:2

0 1:2

MSB 1 0

Επομένως, $21_{10}=10101_2$.

Κλασματικό μέρος

0,375

x2

MSB 0,750

x2

1,500

x2

LSB 1,000

Επομένως $0,375_{10}=011_2$

και $21,375_{10}=10101,011_2$

Δεκαδικός σε οκταδικό

ΠΙΣΩ

ΠΑΡΑΔΕΙΓΜΑ 1.3

Να μετατραπεί ο δεκαδικός αριθμός 67,458 στον αντίστοιχο οκταδικό.

Δεκαδικός σε οκταδικό

ΠΙΣΩ

Ακέραιο μέρος.

67:8

LSB 3 8:8

0 1:8

MSB 1 0

Επομένως, $67_{10}=103_8$

Κλασματικό μέρος.

0,458

x8

MSB 3,664

x8

5,312

x8

LSB 2,496

Επομένως, $67{,}458_{10}=103{,}352_{8}$.

ΠΙΣΩ

Δεκαδικός σε δεκαεξαδικό

ΠΑΡΑΔΕΙΓΜΑ 1.4

Να μετατραπεί στον αντίστοιχο δεκαεξαδικό ο δεκαδικός αριθμός $67{,}458_{10}$

Δεκαδικός σε δεκαεξαδικό

<u>Ακέραιο μέρος.</u>

67:16

LSB 3 4:16

MSB 4 0

Επομένως, $67_{10}=43_{16}$.

Κλασματικό μέρος.

0,458

x16

MSB 7,328

x16

5,248

x16

LSB 3,968

Επομένως, $0,458_{10}=0,753F7CE_{16}$

και $67,458_{10}=43,753F7CE_{16}$.

Δυαδικός σε οκταδικό

Επειδή η βάση του οκταδικού συστήματος είναι το οκτώ 8, δηλαδή η τρίτη δύναμη του δύο (2^3=8) που είναι η βάση του δυαδικού συστήματος, βγαίνει το συμπέρασμα ότι κάθε οκταδικό ψηφίο αντιστοιχεί σε τρία δυαδικά.

Η μετατροπή από το δυαδικό στο οκταδικό γίνεται με τον χωρισμό του δυαδικού αριθμού από την υποδιαστολή και προς τις δύο κατευθύνσεις σε ομάδες των τριών ψηφίων και την αντικατάσταση της κάθε μιας από το αντίστοιχο ψηφίο του οκταδικού συστήματος.

ΠΑΡΑΔΕΙΓΜΑ 1.5

Να μετατραπεί ο δυαδικός αριθμός

1011110010,101010100011 στον αντίστοιχο οκταδικό.

Ακέραιο Τμήμα ←　→ Δεκαδικό Τμήμα

001 011 110 010 , 101 010 100 110$_2$

1　3　6　2 , 5　2　4　6$_8$

Επομένως, 1011110010,101010100011$_2$=1362,5246$_8$

Οκταδικός σε δυαδικό

Η διαδικασία αυτή είναι ακριβώς αντίστροφη από την προηγούμενη, δηλαδή αντικαθίσταται κάθε ψηφίο του οκταδικού αριθμού με τον αντίστοιχο δυαδικό τριών ψηφίων.

Ο αριθμός που προκύπτει θα είναι ο αντίστοιχος δυαδικός.

Οκταδικός σε δυαδικό

ΠΑΡΑΔΕΙΓΜΑ 1.6

Να μετατραπεί ο οκταδικός αριθμός $2376{,}143_8$ στον αντίστοιχο δυαδικό.

2	**3**	**7**	**6**	**,**	**1**	**4**	**3_8**
010	**011**	**111**	**110**	**,**	**001**	**100**	**011_2**

Επομένως, $2376{,}143_8 = 010011111110{,}001100011_2$

Δυαδικός σε δεκαεξαδικό

Επειδή η βάση του δεκαεξαδικού συστήματος είναι το 16, δηλαδή είναι η τέταρτη δύναμη του 2 (2^4=16), συνεπάγεται ότι κάθε δεκαεξαδικό ψηφίο αντιστοιχεί σε τέσσερα δυαδικά.

Συνεπώς, εδώ η μετατροπή θα γίνει με το χωρισμό του δυαδικού αριθμού σε ομάδες των τεσσάρων ψηφίων.

Το ξεκίνημα γίνεται από την υποδιαστολή, και προς τις δύο κατευθύνσεις, με αντικατάσταση κάθε ομάδας δυαδικών ψηφίων με το αντίστοιχο δεκαεξαδικό ψηφίο.

ΠΑΡΑΔΕΙΓΜΑ 1.5

Να μετατραπεί ο δυαδικός αριθμός

1011111011,10101010011

στον αντίστοιχο δεκαεξαδικό.

Ακέραιο Τμήμα ← **→ Δεκαδικό Τμήμα**

0010 1111 1011 , 1010 1010 0110$_2$

2 F B , A A 6$_{16}$

Επομένως, 1011111011,10101010011$_2$=2FB,AA6$_{16}$

Δεκαεξαδικός σε δυαδικό

ΠΑΡΑΔΕΙΓΜΑ 1.8

Να μετατραπεί ο δεκαεξαδικός αριθμός 4B6C,2589 στον αντίστοιχο δυαδικό.

Δεκαεξαδικός σε δυαδικό

Εδώ κάθε δεκαεξαδικό ψηφίο αντικαθίσταται με τον αντίστοιχο τετραψήφιο δυαδικό.

4	**B**	**6**	**C**	**,**	**2**	**5**	**8**	**9_{16}**
0100	**1011**	**0110**	**1100**	**,**	**0010**	**0101**	**1000**	**1001_2**

Επομένως,

$4B6C,2589_{16}=100101101101100,0010010110001001_2$.

Πίνακας αριθμητικών συστημάτων

Δεκαδικός	Δυαδικός	Οκταδικός	Δεκαεξαδικός
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Πρόσθεση δυαδικών αριθμών

Η πρόσθεση δύο δυαδικών αριθμών γίνεται, όπως και των δεκαδικών, αρχίζοντας από το τέλος και προσθέτοντας τα ψηφία που βρίσκονται το ένα κάτω από το άλλο.

Χρησιμοποιούνται μόνο τα ψηφία που ανήκουν στο δυαδικό σύστημα, τα οποία, ως γνωστόν, είναι το "0" και το "1".

Κρατούμενο (carry) προκύπτει όταν το άθροισμα (sum) των δύο ψηφίων είναι μεγαλύτερο του "1". Το κρατούμενο μεταφέρεται και προστίθεται στο αμέσως επόμενο σε σημαντικότητα ζευγάρι των υπό πρόσθεση δυαδικών ψηφίων.

Πρόσθεση δυαδικών αριθμών

Η πρόσθεση δυαδικών είναι παρόμοια με αυτή των δεκαδικών.

Για παράδειγμα έχουμε 8+7=15, που είναι μεγαλύτερο από το 9 (σε κάθε θέση χωρά ένα δεκαδικό) άρα κρατάμε το 5 και το 1 μεταφέρεται ως κρατούμενο.

Στην πρόσθεση δυαδικών αριθμών έχουμε ένα δυαδικό ψηφίο σε κάθε θέση.

Επομένως αν έχουμε $1+1 = 2_{10} = 10_2$ κρατάμε 0_2 και το 1_2 μεταφέρεται ως κρατούμενο.

<u>Αν έχουμε $1+1+1 = 3_{10} = 11_2$ κρατάμε 1_2 και το 1_2 μεταφέρεται ως κρατούμενο.</u>

Πρόσθεση δυαδικών αριθμών

	Άθροισμα (Sum)	**Κρατούμενο (Carry)**
0+0	**0**	**0**
0+1	**1**	**0**
1+1	**0**	**1**

ΠΑΡΑΔΕΙΓΜΑ 1.9

Να υπολογιστεί το άθροισμα των αριθμών $+2_{10}$ και -6_{10}.

$$
\begin{array}{r}
+2 = 0010 \\
-6 = \underline{1110} \\
10000
\end{array}
$$

Παρατηρείται ότι το άθροισμα αποτελείται από πέντε δυαδικά ψηφία, υπερβαίνοντας τη δυνατότητα του αριθμητικού συστήματος που αποτελείται από το πρόσημο και τρία δυαδικά ψηφία.

Αυτό θα σημαίνει υπερχείλιση.

Επίσης, το αποτέλεσμα είναι λάθος, γιατί, ενώ έπρεπε να είναι -4 (1100), είναι 0.

Συμπλήρωμα «ως προς 1»

Το συμπλήρωμα "ως προς βάση μείον ένα" ενός θετικού αριθμού *N* με βάση *r*, *n* πλήθος κλασματικών ψηφίων και *m* ακεραίων ψηφίων είναι $r^m - r^{-n} - N$.

Έστω ο δυαδικός αριθμός 1011,0011:

$$1011{,}0011_2 = 1x2^{-4} + 1x2^{-3} + 0 + 0 + 1x2^{0} + 1x2^{1} + 0 + 1x2^{3}$$

$$= 0{,}1875 + 11 = 11{,}1875_{10}$$

Το συμπλήρωμα "ως προς βάση μείον ένα" του αριθμού αυτού θα είναι:

$$2^4 - 2^{-4} - 11{,}1875 = 16 - 0{,}0625 - 11{,}1875 = 4{,}75_{10} = 100{,}11_2$$

Συμπλήρωμα «ως προς 1»

Επομένως, το συμπλήρωμα "ως προς βάση μείον ένα" του δυαδικού 1011,0011 είναι ο δυαδικός αριθμός 0100,1100.

Αφού *r*=2 τότε *r*-1=1, οπότε το συμπλήρωμα ονομάζεται συμπλήρωμα "ως προς 1".

Παρατηρείται από τα παραπάνω ότι το συμπλήρωμα "ως προς 1" ενός δυαδικού αριθμού είναι ένας νέος δυαδικός αριθμός, που προκύπτει από τον πρώτο, αν τα "1" αντικατασταθούν με "0" και τα "0" με "1".

ΠΑΡΑΔΕΙΓΜΑ 1.10

Να βρεθεί το συμπλήρωμα ¨ως προς 1¨ του δυαδικού αριθμού 1011,0011.

Το συμπλήρωμα ¨ως προς 1¨

είναι ο δυαδικός αριθμός 0100,1100

Συμπλήρωμα «ως προς 2»

Το συμπλήρωμα "ως προς βάση *r*" ενός θετικού αριθμού *N* είναι r^m-*N*, όπου *m* το πλήθος των ακεραίων ψηφίων.

Έστω ο δυαδικός αριθμός *N*=1011,0011 με *m*=4 και *r*=2

Τότε το συμπλήρωμα ως προς βάση του αριθμού αυτού θα είναι:

$$r^4-N=2^4-11{,}1875=4{,}8125_{10}=100{,}1101_2$$

Αφού *r*=2, το συμπλήρωμα αυτής της μορφής λέγεται και συμπλήρωμα "ως προς 2" βρίσκεται δε ευκολότερα, αν στο συμπλήρωμα "ως προς 1" προστεθεί η μονάδα.

Κωδικοποίηση Συμπληρώματος ως προς 2

Για το διάνυσμα: $\vec{x} = [x_{w-1}, x_{w-2}, \dots, x_0]$

Ισχύει:

$$B2T_w(\vec{x}) \doteq - x_{w-1}2^{w-1} + \sum_{i=0}^{w-2} x_i 2^i$$

Το περισσότερο σημαντικό bit, x_{w-1}, ονομάζεται bit προσήμου (sign bit).
Όταν είναι 0 είναι θετικός, όταν είναι 1 είναι αρνητικός

Παράδειγμα:

$$B2T_4([0001]) = -0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0 + 0 + 0 + 1 = 1$$

$$B2T_4([0101]) = -0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0 + 4 + 0 + 1 = 5$$

$$B2T_4([1011]) = -1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = -8 + 0 + 2 + 1 = -5$$

$$B2T_4([1111]) = -1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = -8 + 4 + 2 + 1 = -1$$

[1 $\underline{\text{1 1 1}}$]

Πρόσημο - αρνητικό

Μέτρο/τιμή

[1 1 1]
[0 0 0]
+1

[0 0 1]

ΠΑΡΑΔΕΙΓΜΑ 1.11

Να βρεθεί το συμπλήρωμα ¨ως προς 2¨ του δυαδικού αριθμού 10110011.

Δυαδικός αριθμός	**10110011**
Συμπλήρωμα “ως προς ένα”	**01001100**
	+ 1
Συμπλήρωμα “ως προς δύο”	**01001101**

ΠΑΡΑΔΕΙΓΜΑ 1.12

Να γίνει η πρόσθεση των δυαδικών αριθμών:

α. 0011_2+1001_2 ***και***

β. $01100_2+10001_2$.

α. **0011= 3**

+ 1001=+9

1100=12

β. **01100= 12**

+10001=+17

11101= 29

Αφαίρεση δυαδικών αριθμών

Η αφαίρεση είναι λίγο πιο πολύπλοκη από την πρόσθεση αλλά δεν παύει να είναι απλή. Για την αφαίρεση ισχύει:

	Διαφορά (Difference)	Δανεικό (Borrow)
0-0	0	0
0-1	1	1
1-0	1	0
1-1	0	0

ΠΑΡΑΔΕΙΓΜΑ 1.13

Να γίνει η αφαίρεση των δυαδικών αριθμών

α) 1010_2-0101_2 και

β) 1110_2-1101_2.

α.	**1010= 10**	**β.**	**1110= 14**
	-0101= -5		**-1101= -13**
	0101= 5		**0001= 1**

Αφαίρεση με το συμπλήρωμα «ως προς 1»

Κατά την αφαίρεση με τη χρήση του συμπληρώματος "ως προς ένα" ακολουθείται η παρακάτω διαδικασία:

1. Αφήνεται ο αφαιρετέος όπως έχει.

2. Βρίσκεται το συμπλήρωμα "ως προς 1" του αφαιρέτη και αυτό προστίθεται στον αφαιρετέο.

Αφαίρεση με το συμπλήρωμα «ως προς 1»

3. Αν το ψηφίο πριν από το περισσότερο σημαντικό ψηφίο του αποτελέσματος είναι 1 τότε αυτό μεταφέρεται και προστίθεται στο λιγότερο σημαντικό ψηφίο του αριθμού που προέκυψε από την πρόσθεση. Η ζητούμενη διαφορά θα είναι ένας θετικός αριθμός με μέτρο ίσο με το μέτρο του αποτελέσματος.

Αν το ψηφίο πριν από το περισσότερο σημαντικό ψηφίο του αποτελέσματος είναι 0 τότε η ζητούμενη διαφορά θα είναι ένας αρνητικός αριθμός με μέτρο ίσο προς το συμπλήρωμα "ως προς ένα" του αποτελέσματος.

ΠΑΡΑΔΕΙΓΜΑ 1.14

Να γίνει η παρακάτω αφαίρεση με τη χρήση του συμπληρώματος ¨ως προς ένα¨.

$$1010101_2 = 85_{10}$$

$$-\ 1000111_2 = 71_{10}$$

1010101 **Αφήνεται ο αφαιρετέος όπως έχει**

+ 0111000 **προστίθεται το συμπλήρωμα "ως προς 1" του αφαιρέτη στον αφαιρετέο**

10001101 **και επειδή το ψηφίο πριν το MSB είναι "1"**

+1 **αυτό προστίθεται στο LSB**

$0001110_2 = 14_{10}$ που είναι η ζητούμενη διαφορά.

ΠΑΡΑΔΕΙΓΜΑ 1.15

Να γίνει η παρακάτω αφαίρεση με τη χρήση του συμπληρώματος ¨ως προς ένα¨.

$$1000111_2 = 71_{10}$$

$$-1010101_2 = 85_{10}$$

1000111 **Αφήνεται ο αφαιρετέος όπως έχει**

+ 0101010 **προστίθεται το συμπλήρωμα "ως προς 1" του αφαιρέτη στον αφαιρετέο**

01110001 **και επειδή δεν υπάρχει ¨1¨ πριν το MSB, το αποτέλεσμα της αρχικής αφαίρεσης θα είναι ένας αρνητικός αριθμός**

$0001110_2 = -14_{10}$ **με μέτρο ίσο με το συμπλήρωμα "ως προς 1" του αθροίσματος (-14)**

Αφαίρεση με το συμπλήρωμα «ως προς 2»

Κατά την αφαίρεση με τη χρήση του συμπληρώματος "ως προς δύο" ακολουθείται η παρακάτω διαδικασία:

1. Αφήνεται ο αφαιρετέος όπως έχει.

2. Βρίσκεται το συμπλήρωμα ως προς δύο του αφαιρέτη και αυτό προστίθεται στον αφαιρετέο.

Αφαίρεση με το συμπλήρωμα «ως προς 2»

3. Αν το ψηφίο πριν από το περισσότερο σημαντικό ψηφίο του αποτελέσματος είναι 1 τότε η διαφορά είναι ένας θετικός αριθμός.

Αν το ψηφίο πριν από το περισσότερο σημαντικό ψηφίο του αποτελέσματος είναι 0 τότε η διαφορά είναι ένας αρνητικός αριθμός με μέτρο ίσο με το συμπλήρωμα "ως προς δύο" αποτελέσματος.

ΠΑΡΑΔΕΙΓΜΑ 1.16

Να γίνει η παρακάτω αφαίρεση με τη χρήση του συμπληρώματος ¨ως προς δύο¨.

$$1010101_2=85_{10}$$
$$-\ 1000111_2=71_{10}$$

1010101 Αφήνεται ο αφαιρετέος όπως είναι.

0111000 Βρίσκεται το συμπλήρωμα ως προς ένα του αφαιρέτη.

+1 Προστίθεται το 1 για να παρθεί το

0111001 συμπλήρωμα ¨ως προς δύο¨ του αφαιρέτη και αυτό προστίθεται στον αφαιρετέο.

1010101

+ 0111001

<u>1</u>0001110 Επειδή το ψηφίο πριν το MSB είναι το 1 συνεπάγεται ότι το άθροισμα αυτό είναι 14 και η ζητούμενη διαφορά είναι θετικός αριθμός.

ΠΑΡΑΔΕΙΓΜΑ 1.17

Να γίνει η παρακάτω αφαίρεση με τη χρήση του συμπληρώματος ¨ως προς δύο¨.

$$
\begin{array}{r}
1000111_2=71_{10} \\
-1010101_2=85_{10}
\end{array}
$$

1000111 Αφήνεται ο αφαιρετέος όπως είναι.

0101010 Βρίσκεται το συμπλήρωμα ¨ως προς ένα¨ του αφαιρέτη.

+1 Προστίθεται το ένα για να παρθεί το συμπλήρωμα ¨ως προς δύο¨ του αφαιρέτη και αυτό προστίθεται στον αφαιρετέο.

0101011

01110010 Επειδή το ψηφίο πριν το MSB είναι 0 η διαφορά είναι ένας αρνητικός αριθμός με μέτρο το συμπλήρωμα «ως προς δύο» του αθροίσματος.

0001101

+1

$0001110_2=-14_{10}$.

Πολλαπλασιασμός δυαδικών αριθμών

Είναι μια πράξη ακριβώς ίδια με την αντίστοιχη των δεκαδικών και ακόμη ευκολότερη γιατί τα ψηφία πού χρησιμοποιούνται είναι μόνο δύο, το "0" και το "1", οπότε το επιμέρους γινόμενο θα είναι είτε το 0, είτε ο πολλαπλασιαστέος. Αφού βρεθούν όλα τα επιμέρους γινόμενα τα προστίθενται και παίρνεται το τελικό αποτέλεσμα.

ΠΑΡΑΔΕΙΓΜΑ 1.18

Να γίνει ο πολλαπλασιασμός $1001_2 x 101_2 = 9_{10} x 5_{10}$.

$$
\begin{array}{r}
1001_2 = 9_{10} \\
x \quad 101_2 = 5_{10} \\
\hline
1001 \\
0000 \\
1001 \\
\hline
101101_2 = 45_{10}
\end{array}
$$

Διαίρεση δυαδικών αριθμών

Η διαίρεση αποτελείται από διαδοχικές αφαιρέσεις του διαιρέτη από τον διαιρετέο.

Κάθε φορά που ο *n*-ψήφιος διαιρέτης είναι μικρότερος από το *n*-ψήφιο μέρος του διαιρετέου που βρίσκεται ακριβώς από πάνω του, δηλαδή μπορεί να γίνει η αφαίρεση, τότε αυτή πραγματοποιείται και σημειώνεται το ψηφίο 1 στα δεξιά, ενώ σε αντίθετη περίπτωση σημειώνεται το ψηφίο 0

Το σύνολο των ψηφίων αυτών στο τέλος των αφαιρέσεων θα δώσει το πηλίκο.

Το αποτέλεσμα της κάθε αφαίρεσης γράφεται κάτω ακριβώς από το διαιρέτη και έχουμε στην καινούρια αφαίρεση μετατόπιση του διαιρέτη κατά μία θέση.

ΠΑΡΑΔΕΙΓΜΑ 1.19

Να γίνει η διαίρεση: $11100110/1011_2 = 230/11_{10}$

11100110		ΠΗΛΙΚΟ
-1011	Ν Α Ι	1 MSB
00110		
-1011	Ο Χ Ι	0
01101		
-1011	Ν Α Ι	1
00101		
-1011	Ο Χ Ι	0
01010		
-1011	Ο Χ Ι	0 LSB
01010 ← ΥΠΟΛΟΙΠΟ		

Επομένως, $11100110/1011_2=10100_2$ πηλίκο και 01010_2 υπόλοιπο ή $230/11_{10}=20_{10}$ πηλίκο και 10_{10} υπόλοιπο.

Προσημασμένοι αριθμοί

Στο δυαδικό σύστημα αριθμών το πρόσημο παριστάνεται με ένα ψηφίο, που ονομάζεται ψηφίο πρόσημου και τοποθετείται μπροστά από τον δυαδικό αριθμό. Το 1 χρησιμοποιείται για να παραστήσει το πλην (-) και το 0 για να παραστήσει το συν (+).

Παρατηρείται ότι υπάρχουν δύο παραστάσεις (+0 και -0) του αριθμού 0. Η περιοχή των ακεραίων που μπορούν να παρασταθούν με τρία δυαδικά ψηφία είναι από -3 έως +3.

	Δεκαδικό ισοδύναμο			
Δυαδικός	**Μη-Προσημ.**	**Προσημ.**	**Συμπλήρ. ¨1¨**	**Συμπλήρ. ¨2¨**
0000	0	+0	+0	+0
0001	1	+1	+1	+1
0010	2	+2	+2	+2
0011	3	+3	+3	+3
0100	4	+4	+4	+4
0101	5	+5	+5	+5
0110	6	+6	+6	+6
0111	7	+7	+7	+7
1000	8	-0	-7	-8
1001	9	-1	-6	-7
1010	10	-2	-5	-6
1011	11	-3	-4	-5
1100	12	-4	-3	-4
1101	13	-5	-2	-3

Παρατηρείται ότι υπάρχουν δύο παραστάσεις (+0 και -0) του αριθμού 0. Η περιοχή των ακεραίων που μπορούν να παρασταθούν με τέσσερα δυαδικά ψηφία είναι από -7 έως +7.

Μη προσημασμένη - Προσημασμένη Αριθμητική

Στη μη-προσημασμένη αριθμητική ένα δυαδικός αριθμός ν ψηφίων παίρνει τιμές μεταξύ *0* και *2^{ν}-1*.

Επομένως, αν θεωρηθεί ένας δυαδικός αριθμός οκτώ ψηφίων (ένα byte), τότε αυτός παίρνει τιμές από 0-255_{10}(=2^8-1).

Στην προσημασμένη αριθμητική το περισσότερο σημαντικό ψηφίο της ψηφιολέξης παριστάνει το πρόσημο του αριθμού.

Επομένως στην περίπτωση του byte:

Όταν το περισσότερο σημαντικό ψηφίο είναι "0", ο αριθμός είναι θετικός και έχει μέτρο ίσο με το μέτρο των υπόλοιπων εφτά ψηφίων **(από $00000000_2=0_{10}$ έως $01111111_2=127_{10}$).**

Όταν το περισσότερο σημαντικό ψηφίο είναι "1", ο αριθμός είναι αρνητικός και έχει μέτρο ίσο με το συμπλήρωμα "ως προς 2" των υπόλοιπων εφτά ψηφίων **(από $10000000_2=-128_{10}$ έως $11111111_2=-1_{10}$).**

00000000_2=Θετικός αριθμός με μέτρο=0_{10}

01111111_2= Θετικός αριθμός με μέτρο=127_{10}

10000000_2=Αρνητικός αριθμός με μέτρο=-128_{10}

Συμπλήρωμα «ως προς 2»

$$\begin{array}{r} 1111111 \\ +\quad\quad\quad 1 \\ \hline 10000000 \end{array}=128_{10}$$

11111111_2=Αρνητικός αριθμός με μέτρο=1_{10}

Συμπλήρωμα «ως προς 2»

$$\begin{array}{r} 0000000 \\ +\quad\quad\quad 1 \\ \hline 0000001 \end{array}=1_{10}$$

ΠΑΡΑΔΕΙΓΜΑ 1.20

Να προστεθούν οι δυαδικοί αριθμοί 00001101_2 και 11010011_2

α. όταν ερμηνεύονται ως μη-προσημασμένοι αριθμοί και

β. όταν ερμηνεύονται ως προσημασμένοι αριθμοί.

Όταν ερμηνεύονται ως μη-προσημασμένοι αριθμοί:

$$\begin{array}{rr} 00001101 & 13 \\ +\underline{11010011} & +\underline{211} \\ 11100000_2 & 224_{10} \end{array}$$

Όταν ερμηνεύονται ως προσημασμένοι αριθμοί:

$$\begin{array}{rr} 00001101 & 13 \\ +\underline{11010011} & \underline{-45} \\ 11100000_2 & -32_{10} \end{array}$$

ΠΑΡΑΔΕΙΓΜΑ 1.21

Να προστεθούν οι δυαδικοί αριθμοί 11011101_2 και 10100011_2

α. όταν ερμηνεύονται ως μη-προσημασμένοι αριθμοί και

β. όταν ερμηνεύονται ως προσημασμένοι αριθμοί.

Όταν ερμηνεύονται ως μη-προσημασμένοι αριθμοί:

$$\begin{array}{rr} 11011101 & 221 \\ +\underline{10100011} & +\underline{163} \\ 1\,10000000_2 & 128_{10} \end{array}$$

Φαίνεται ότι το αποτέλεσμα είναι λάθος και αυτό οφείλεται στο γεγονός ότι οι δύο αριθμοί έχουν πραγματικό άθροισμα 384_{10} που είναι αριθμός μεγαλύτερος του μέγιστου αριθμού (255_{10}), που μπορεί να παρασταθεί με τα οκτώ ψηφία του byte.

Ταυτόχρονα C="1". Άρα, ύπαρξη κρατούμενου με βάρος 256_{10} που στην ουσία θα αποτελεί το ένατο ψηφίο του αποτελέσματος.

Όταν ερμηνεύονται ως προσημασμένοι αριθμοί:

$$\begin{array}{rr} 11011101 & -35_{10} \\ +\underline{10100011} & +\underline{(-93_{10})} \\ {\color{red}1}00000000_2 & -128_{10} \end{array}$$

Φαίνεται ότι το αποτέλεσμα είναι σωστό.

ΠΑΡΑΔΕΙΓΜΑ 1.22

Να προστεθούν οι δυαδικοί αριθμοί 10011101_2 και 10001011_2

α. όταν ερμηνεύονται ως μη-προσημασμένοι αριθμοί και

β. όταν ερμηνεύονται ως προσημασμένοι αριθμοί.

Όταν ερμηνεύονται ως μη-προσημασμένοι αριθμοί:

$$
\begin{array}{rr}
10011101 & 157 \\
+\underline{10001011} & +\underline{139} \\
\color{magenta}{1}00101000_2 & 40_{10}
\end{array}
$$

Φαίνεται ότι το αποτέλεσμα είναι λάθος και αυτό οφείλεται στο γεγονός ότι οι δύο αριθμοί έχουν πραγματικό άθροισμα 296_{10} που είναι αριθμός μεγαλύτερος του μέγιστου αριθμού (255_{10}), που μπορεί να παρασταθεί με τα οκτώ ψηφία του byte.

Ταυτόχρονα C="1". Άρα, ύπαρξη κρατούμενου (Carry) με βάρος 256_{10} που στην ουσία θα αποτελεί το ένατο ψηφίο του αποτελέσματος.

Όταν ερμηνεύονται ως προσημασμένοι αριθμοί:

$$\begin{array}{r@{\qquad}r} 10011101 & -99_{10} \\ +\underline{10001011} & +(\underline{-117_{10}}) \\ 1\,00101000_2 & +40_{10} \end{array}$$

Φαίνεται ότι το αποτέλεσμα είναι λάθος.

Το πραγματικό αποτέλεσμα (-216_{10}) είναι αριθμός μεγαλύτερος από το μεγαλύτερο αρνητικό αριθμό (-128_{10}), που μπορεί να παρασταθεί με τα οκτώ ψηφία του byte (2^7 γιατί το 8ο bit είναι το πρόσημο).

Αυτό φαίνεται από ένα δείκτη που ονομάζεται δείκτης υπερχείλισης (Overflow).

Ο κώδικας BCD

Ο BCD πήρε το όνομά του από τα ακρωνύμια των λέξεων Binary Coded Decimal (δυαδικά κωδικοποιημένος δεκαδικός) και οι αριθμοί 8421 συμβολίζουν τα "βάρη" των δυαδικών ψηφίων στην αντίστοιχη στήλη.

Δηλαδή ο αριθμός 0110_2 είναι $0x8+1x4+1x2+0x1=6_{10}$

Λόγω της ύπαρξης των βαρών ο κώδικας ονομάζεται ζυγοσταθμισμένος.

Ο BCD κάνει δυαδική μετατροπή αριθμών από το 0 έως και το 9.

Ο κώδικας BCD

Για παράδειγμα, ο δεκαδικός 67 είναι ο δυαδικός 1000011, ενώ ο αντίστοιχος δυαδικά-κωδικοποιημένος είναι ο 0110 0111, και ο

$375_{10}=101101111_2=0011\ 0111\ 0101_{BCD8421}$

Το βασικό που πρέπει να κατανοηθεί είναι, ότι ένας αριθμός BCD δεν είναι δυαδικός αλλά δυαδικά-κωδικοποιημένος αριθμός.

Πρόσθεση αριθμών BCD

Έστω, για παράδειγμα, ότι πρέπει να προστεθούν οι αριθμοί

52_{10}= 0101 0010$_{8421BCD}$ και 37_{10}= 0011 0111$_{8421BCD}$

$$\begin{array}{rl} 52_{10} & = 0101\ 0010_{8421BCD} \\ +37_{10} & = 0011\ 0111_{8421BCD} \\ \hline 89_{10} & = 1000\ 1001_{8421BCD} \end{array}$$

Παρατηρείται ότι, προστίθενται τα δύο λιγότερο σημαντικά ψηφία και, αν δεν προκύψει κρατούμενο, τα δύο περισσότερο σημαντικά ψηφία των δύο αριθμών.

Όταν προκύπτει κρατούμενο από την πρόσθεση των λιγότερο σημαντικών ψηφίων τότε αυτό πρέπει να προστεθεί στη στήλη των επόμενων σε σημαντικότητα ψηφίων.

Κρατούμενο προκύπτει όταν το άθροισμα των δύο ψηφίων είναι αριθμός μεγαλύτερος του $1001_2=9_{10}$. Δηλαδή 10, 11, 12, 13, 14, 15.

Στην περίπτωση αυτή θα πρέπει στο άθροισμα που προκύπτει να προστεθεί ο αριθμός έξι, όσος και ο αριθμός των δυαδικών συνδυασμών που δεν χρησιμοποιούνται για την παράσταση των αριθμών BCD. Η διαδικασία αυτή λέγεται '*διόρθωση κρατούμενου*' και φαίνεται στο παράδειγμα που ακολουθεί.

Ας υποτεθεί ότι πρόκειται να προστεθούν οι αριθμοί, 427_{10}= 0100 0010 $0111_{8421BCD}$ και 238_{10}= 0010 0011 $1000_{8421BCD}$

427_{10}= 0100 0010 $0111_{8421BCD}$

+ 238_{10}= 0010 0011 $1000_{8421BCD}$

1111 ⇒ άκυρο άθροισμα

Κρατούμενο ⇒ 1 0110 ⇒ πρόσθεση 6 για διόρθωση κρατούμενου

= 665_{10}= 0110 0110 $0101_{8421BCD}$

Παρατηρείται εδώ ότι τα δύο λιγότερο σημαντικά ψηφία θα δώσουν άθροισμα 15 που είναι αριθμός μεγαλύτερος του 9_{10}. Όταν στα ψηφία αυτά προστίθεται το έξι τότε προκύπτει ο αριθμός 15_{10} (άθροισμα 5 και κρατούμενο 1). Το ψηφίο 5 θα παραμείνει ως το λιγότερο σημαντικό ψηφίο του αθροίσματος και το κρατούμενο θα προστεθεί στα επόμενα περισσότερο σημαντικά ψηφία των αριθμών για να δώσει άθροισμα 6 κ.ο.κ.

Ας υποτεθεί ότι πρόκειται να προστεθούν οι αριθμοί, 429_{10}= 0100 0010 1001$_{8421BCD}$ και 238_{10}= 0010 0011 1000$_{8421BCD}$.

429_{10}= 0100 0010 1001$_{8421BCD}$

+238_{10}= 0010 0011 1000$_{8421BCD}$

Κρατούμενο ⇒1 0001 ⇒ άκυρο άθροισμα

0110 ⇒ πρόσθεση 6 για διόρθωση κρατούμενου

= 667_{10}= 0110 0110 0111$_{8421BCD}$

Εδώ παρατηρείται ότι τα δύο λιγότερο σημαντικά ψηφία δίνουν άθροισμα 17 (μεγαλύτερο του 15). Αυτό σημαίνει ότι το κρατούμενο που θα προστεθεί ήδη έχει παραχθεί επομένως ο αριθμός έξι προστίθεται μόνο για να γίνει η σχετική διόρθωση.

Κώδικας Gray

Στον κώδικα GRAY κάθε δυαδική θέση δεν έχει καθορισμένο "βάρος", γι' αυτό ο κώδικας αυτός είναι μη ζυγοσταθμισμένος.

Επίσης ονομάζεται κυκλικός και ανακλαστικός.

Το βασικό χαρακτηριστικό του είναι ότι μεταξύ δυο διαδοχικών αριθμών αλλάζει μόνο ένα ψηφίο.

Αυτό έχει ως αποτέλεσμα την αποφυγή των σφαλμάτων.

ΔΕΚΑΔΙΚΟΣ	**BCD_{8421}**	**GRAY**
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	0001 0000	1111
11	0001 0001	1110
12	0001 0010	1010
13	0001 0011	1011
14	0001 0100	1001
15	0001 0101	1000

Άλλοι δυαδικοί κώδικες

Δεκ.	BCD_{8421}	4221	2421	Υπέρβ 3	5421	5211	GRAY	7421
0	0000	0000	0000	0011	0000	0000	0000	0000
1	0001	0001	0001	0100	0001	0001	0001	0001
2	0010	0010	0010	0101	0010	0100	0011	0010
3	0011	0011	0011	0110	0011	0110	0010	0011
4	0100	1000	0100	0111	0100	0111	0110	0100
5	0101	0111	0101 ή 1011	1000	1000	1000	0111	0101
6	0110	1100	0110 ή 1100	1001	1001	1001	0101	0110
7	0111	1101	0111 ή 1101	1010	1010	1011	0100	1000
8	1000	1110	1110 ή 1110	1011	1011	1110	1100	1001
9	1001	1111	1111 ή 1111	1100	1100	1111	1101	1010

Κώδικας ASCII

	Στήλη	0	1	2	3	4	5	6	7
Γραμμή	Ψηφία	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
10	1010	LF	SUB	*	:	J	Z	j	z
11	1011	VT	ESC	+	;	K	[	k	{
12	1100	FF	FS	,	<	L	\	l	|
13	1101	CR	GS	-	=	M	]	m	}
14	1110	SO	RS	.	>	N	^	n	~
15	1111	SI	US	/	?	O	_	o	DEL

Σφάλμα ισοτιμίας

Ο **έλεγχος ισοτιμίας (parity check)** είναι ένας τρόπος -μόνο- ανίχνευσης λαθών κατά τη διάρκεια μετάδοσης δυαδικών πληροφοριών.

Το **ψηφίο ισοτιμίας (parity bit)** είναι ένα επιπρόσθετο ψηφίο, που παίρνει τις τιμές ¨1¨ και ¨0¨ και που τοποθετείται σε μία δυαδική πληροφορία έτσι ώστε το πλήθος των ¨1¨ της πληροφορίας, μαζί με το ψηφίο ισοτιμίας να γίνει είτε άρτιο, οπότε μιλάμε **για άρτια ισοτιμία (even parity)**, είτε περιττό, οπότε μιλάμε για **περιττή ισοτιμία (odd parity)**.

Σφάλμα ισοτιμίας

Έτσι μπορεί να ελεγχθεί αν η εκπομπή και η λήψη ενός μηνύματος έγιναν σωστά.

Το κύκλωμα που παράγει το ψηφίο ισοτιμίας στον πομπό λέγεται γεννήτρια ισοτιμίας (parity generator) και το κύκλωμα που ελέγχει την ισοτιμία στο δέκτη ελεγκτής ισοτιμίας (parity checker).

ΠΛΗΡΟΦΟΡΙΑ	ΚΩΔΙΚΕΣ	
	Με περιττή ισοτιμία	**Με άρτια ισοτιμία**
0000	00001	00000
0001	00010	00011
0010	00100	00101
0011	00111	00110
0100	01000	01001
0101	01011	01010
0110	01101	01100
0111	01110	01111
1000	10000	10001
1001	10011	10010
1010	10101	10100
1011	10110	10111
1100	11001	11000
1101	11010	11011
1110	11100	11101
1111	11111	11110

Inside a computer

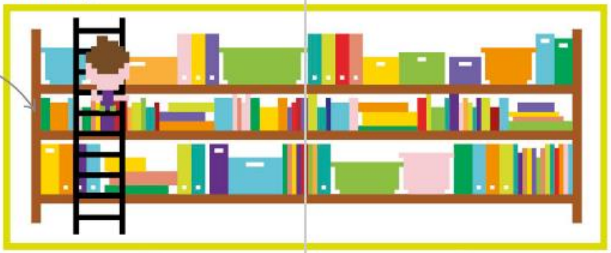
The earliest computers were simple calculators. At a basic level, computers haven't changed much since then. They take in data (input), perform calculations, and give out answers (output).

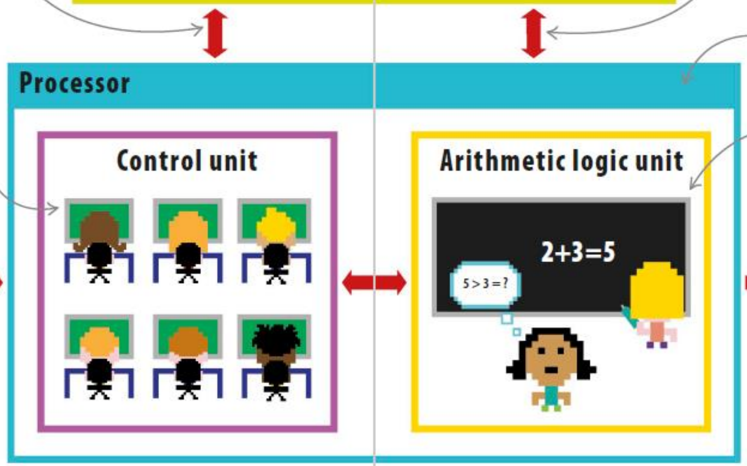
SEE ALSO
Storing data in files 192–193 ›
The internet 194–195 ›
Mini computers 214–215 ›

Basic elements

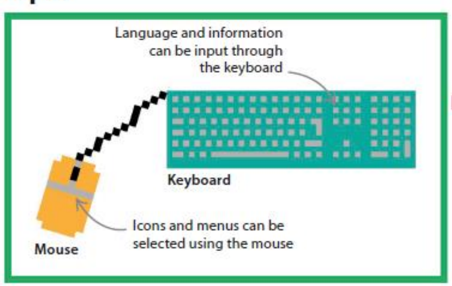
A computer consists of four main parts: input, memory, processor, and output. Input devices gather data, similar to the way your eyes or ears collect information about the world around you. Memory stores the data, while processors examine and alter it, just like a human brain. Output devices show the results of the processor's calculations, like a person speaking or moving after deciding what to do.

▷ **Von Neumann architecture**
A scientist called John von Neumann first came up with the standard layout for a computer in 1945. His plan is still followed today, with some improvements.

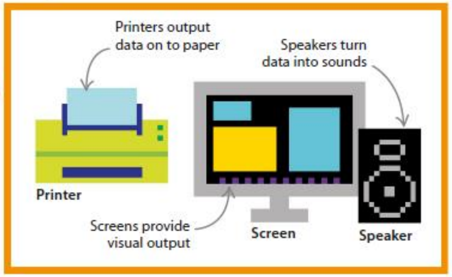




Input

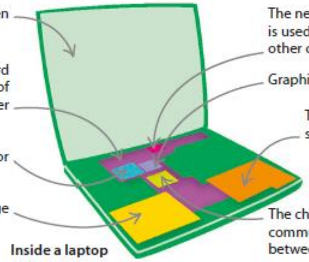


Output



Computer hardware

Hardware is the physical parts of a computer. Computers contain many different bits of hardware working together. As computer makers pack more and more features into smaller machines, the hardware components have to be smaller, generate less heat, and use less power.



Inside a laptop

LINGO

GIGO

"Garbage in, garbage out" ("GIGO" for short) is a computing phrase meaning that even the best programs will output nonsense if they receive the wrong input.

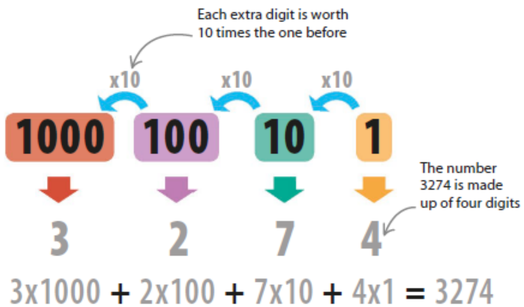
Binary and bases

SEE ALSO
Symbols and codes 184–185 ❯
Logic gates 186–187 ❯

How can computers solve complex calculations when all they understand is electrical signals? Binary numbers are used to translate these signals into numbers.

What is a base number?

A "base" is the number of values that can be shown using only one digit. Each extra digit increases the number of values that can be shown by a multiple of the base.

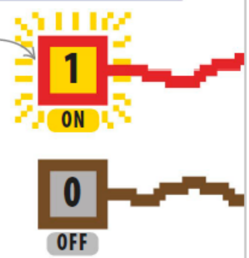


▷ **Decimal system**
The decimal system is the most familiar counting system, and has a base of 10. It can show 10 values with one digit, 100 values with two digits, and 1000 with three digits.

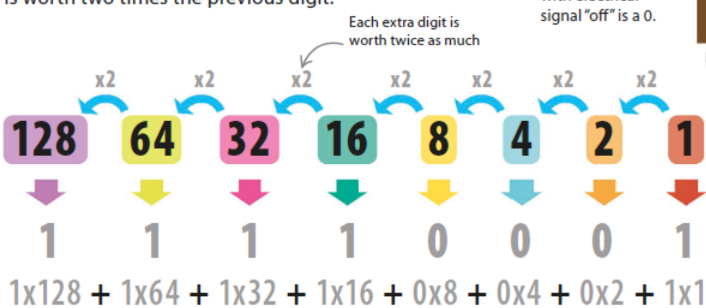
Binary code

At the most basic level, computers only understand two values: electrical signals that are "on" and "off". As there are only two values, computers deal with numbers using a base of two, or "binary". Each digit is either a 1 or a 0, and each extra digit in the number is worth two times the previous digit.

A wire with a current



▷ **1 and 0**
A wire with electrical signal "on" is a 1. A wire with electrical signal "off" is a 0.

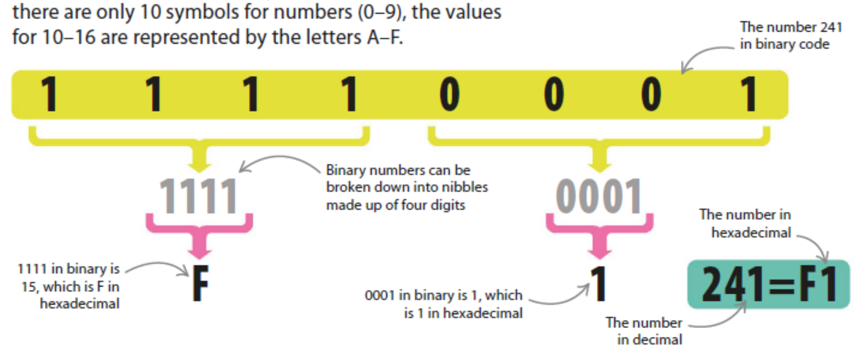


◁ **Binary**
The range of values that can be written doubles with each digit added.

Hexadecimal

When using numbers in computer programs, a base of 16 is often used because it's easy to translate from binary. As there are only 10 symbols for numbers (0–9), the values for 10–16 are represented by the letters A–F.

▽ **Understanding nibbles**
A "nibble" is made up of four binary digits, which can be represented by one hexadecimal digit.



▽ **Comparing base systems**
Using this table, you can see that expressing numbers in hexadecimal gives the most information with the fewest digits.

DIFFERENT BASES

Decimal	Binary	Hexadecimal
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	2
3	0 0 1 1	3
4	0 1 0 0	4
5	0 1 0 1	5
6	0 1 1 0	6
7	0 1 1 1	7
8	1 0 0 0	8
9	1 0 0 1	9
10	1 0 1 0	A
11	1 0 1 1	B
12	1 1 0 0	C
13	1 1 0 1	D
14	1 1 1 0	E
15	1 1 1 1	F

REMEMBER

Bits, nibbles, and bytes

A binary digit is known as a "bit", and is the smallest unit of memory in computing. Bits are combined to make "nibbles" and "bytes". A kilobit is 1024 bits. A megabit is 1024 kilobits.

Bits: Each bit is a single binary digit – a 1 or 0.

Nibbles: Four bits make up a nibble – enough for one hexadecimal digit.

Bytes: Eight bits, or two hexadecimal digits, make up a byte. This gives us a range of values from 0 to 255 (00 to FF).

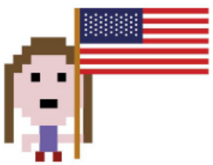
Symbols and codes

Computers use binary code to translate numbers into electrical signals. But how would a computer use binary to store the words and characters on this page?

SEE ALSO
❮ 180–181 Inside a computer
❮ 182–183 Binary and bases

ASCII

The first computers each stored characters in their own unique way. This worked fine until data needed to be moved between computers. At this point, a common system was chosen, called the American Standard Code for Information Interchange (ASCII, pronounced "askey").



▷ **ASCII table**
In ASCII, a decimal number value is given to each character in the upper- and lower case alphabets. Numbers are also assigned to punctuation and other characters, such as a space.

▷ **ASCII in binary**
As each character has a number, that number then needs to be converted to binary to be stored in a computer.

R = 82 = 1010010

r = 114 = 1110010

▽ **ASCII in Python**
You can convert between ASCII and binary in most languages, including Python.

This command prints the character, the ASCII value, and the binary value for each letter in the name "Sam"

```
>>> name = "Sam"
>>> for c in name:
        print(c, ord(c), bin(ord(c)))

S 83 0b1010011
a 97 0b1100001
m 109 0b1101101
```

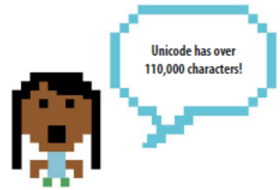
Here are the results. The beginning of each binary number is marked "0b"

ASCII

32	SPACE	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_	127	DELETE

Unicode

As computers across the world began to share data, the limits of ASCII began to show. Thousands of characters used in hundreds of languages had to be represented, so a universal standard called Unicode was agreed on.



▷ **International code**
Unicode represents all the languages of the world. For example, the Arabic characters are represented in the range 0600–06FF.

▷ **Unicode characters**
Unicode characters are represented by their hexadecimal value, which appears as a series of letters and numbers (see pp.182–183). Each character has its own code. More characters are added all the time, and there are some unusual ones, such as a mini umbrella.

REMEMBER

Hexadecimals

Hexadecimal numbers have a base of 16. Ordinary decimal numbers are used for 0 to 9, and the values 10–15 are represented by the letters A to F. Each hexadecimal number has an equivalent binary value.

The Unicode value of ë as hexadecimal

The same value as binary

ë = 00EB = 11100111

▽ **Unicode in Python**
Unicode can be used to display special characters in Python. Simply type a string containing a Unicode character code.

Putting "\u" before the hexadecimal code tells the computer this is Unicode

```
>>> "Zo\u00EB"
"Zoë"
```

The code is translated into the character "ë"