

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής
Εισαγωγή στην Επιστήμη των Η/Υ
2025-26

Πράξεις με δυαδικούς αριθμούς

(αριθμητικές πράξεις)

<https://mixstef.github.io/courses/csintro/>

Μ.Στεφανιδάκης



Πράξεις με δυαδικούς αριθμούς

- Ο υπολογιστής μπορεί να εκτελέσει
 - Λογικές πράξεις (το είδαμε στην προηγούμενη ενότητα)
 - Αριθμητικές πράξεις
- Υπενθύμιση: στις μονάδες επεξεργασίας
 - Οι πράξεις εκτελούνται σε ομάδες bits (bytes ή πολλαπλάσιά τους)

Σειρές από bits ως δυαδικοί αριθμοί

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

το περισσότερο
σημαντικό bit

το λιγότερο
σημαντικό
bit

1	1	1	1	0	0	1	1								
1x128	1x64	1x32	1x16	0x8	0x4	1x2	1x1								
128	+	64	+	32	+	16	+	0	+	0	+	2	+	1	=

243 (δεκαδικό)

- Μετατροπή από το δυαδικό στο δεκαδικό σύστημα
 - Εάν ο αριθμός διαθέτει περισσότερα bits, χρησιμοποιούμε μεγαλύτερες δυνάμεις του 2 στο αριστερό μέρος

Μετατροπή δεκαδικού σε δυαδικό



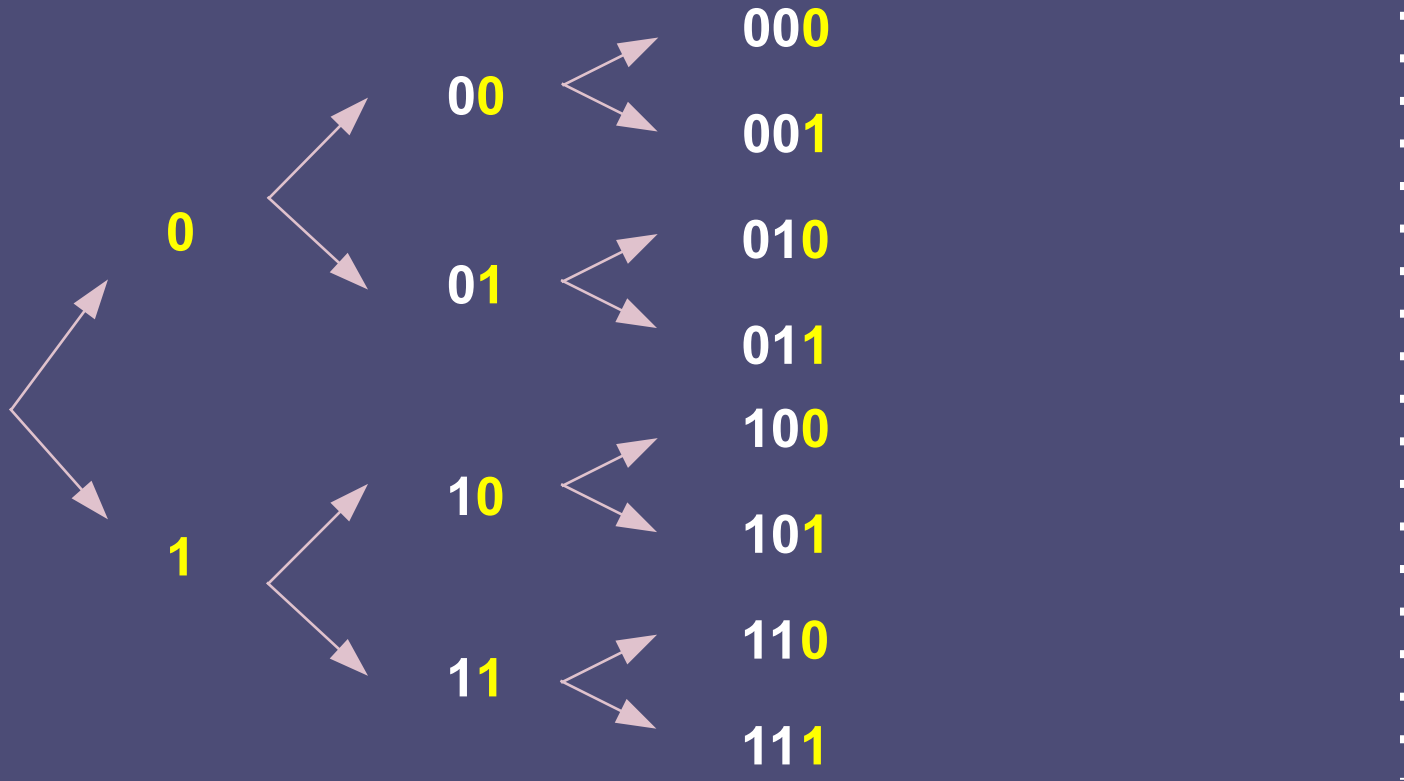
Πόσοι διαφορετικοί αριθμοί με n bits;

με 1 bit:
2 αριθμοί
($=2^1$)

με 2 bits:
4 αριθμοί
($=2^2$)

με 3 bits:
8 αριθμοί
($=2^3$)

με n bits:
 2^n
αριθμοί



Δυαδικοί αριθμοί χωρίς πρόσημο (φυσικοί)

- Για αναπαράσταση διαφορετικών «πραγμάτων»
 - Ως μοναδικοί αναγνωριστικοί αριθμοί
 - Συχνά χωρίς αριθμητική έννοια
 - Παραδείγματα
 - Οι ξεχωριστές διευθύνσεις μνήμης
 - Οι χαρακτήρες σε ένα αλφάβητο
- Ξανά: με n bits απαριθμούνται έως και 2^n διαφορετικά «πράγματα»

Φυσικοί αριθμοί (χωρίς πρόσημο)

Με κίτρινο φαίνεται
ο ελάχιστος αριθμός
bits που απαιτείται
για την αναπαράσταση
του αντίστοιχου
αριθμού

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
.....	...

- Με n bits περιγράφονται
 - Οι φυσικοί αριθμοί από 0 έως και $2^n - 1$

Δεκαεξαδικό Σύστημα

- 16 ψηφία
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
 - Αντιστοιχία με τους δεκαδικούς 0 έως 15
- Σε δυνάμεις του 16
 - $16^n \dots 16^4 \ 16^3 \ 16^2 \ 16^1 \ 16^0$
 - Π.χ. $16F(\text{hex}) = 1 \times 16^2 + 6 \times 16^1 + 15 \times 16^0$
 - $= 256 + 96 + 15 = 367$ (δεκαδικό)
- Χρήσιμο μόνο ως «συντομογραφία» δυαδικών αριθμών

Δεκαεξαδικό Σύστημα

- Κάθε 4 δυαδικά ψηφία αντιστοιχούν σε ένα δεκαεξαδικό ψηφίο

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Παράδειγμα στο δεκαεξαδικό σύστημα

- Παράδειγμα: 1100100110010100

1100 1001 1001 0100

C 9 9 4 = C994(hex)

- Παράδειγμα: 10000101011110

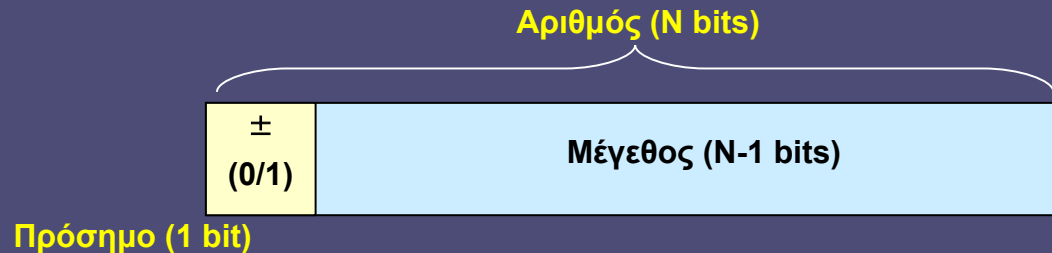
0010 0001 0101 1110

2 1 5 E = 215E (hex)

- Συμπλήρωση με 0 στα αριστερά για να έχουμε 4άδες
- Δεν αλλάζει τον αριθμό, όπως ακριβώς και στο δεκαδικό σύστημα

Ακέραιοι αριθμοί (με πρόσημο – signed)

- Πώς θα αναπαρασταθούν οι **αρνητικοί**;
 - Για να γίνονται εύκολα οι πράξεις
- Χρησιμοποιήθηκε παλαιότερα:
 - **Ξεχωριστό bit πρόσημου**



- Διάστημα τιμών για αριθμούς με n bits
 $-(2^{n-1}-1)$ έως $+(2^{n-1}-1)$ (π.χ. για $n=8$, $-127 \dots +127$)
 - ένα χρήσιμο bit λιγότερο
 - δυσκολία στις πράξεις
 - 2 αναπαραστάσεις του 0

Ακέρατοι αριθμοί (με πρόσημο – signed)

- Μια ακόμα μορφή που δεν χρησιμοποιείται σήμερα:
 - Συμπλήρωμα ως προς 1
 - Αντιστροφή όλων των bits του θετικού αριθμού για να πάρουμε τον αντίστοιχο αρνητικό
 - Πιο σημαντικό bit: 0 για θετικούς, 1 για αρνητικούς
 - Διάστημα τιμών για αριθμούς με n bits
 $-(2^{n-1}-1)$ έως $+(2^{n-1}-1)$ (γιατί;)
 - Τα ίδια προβλήματα με την χρήση ξεχωριστού bit πρόσημου

Ακέραιοι αριθμοί (με πρόσημο – signed)

- Η μορφή που χρησιμοποιεί το υλικό (hardware) σήμερα
 - Κάθε αρνητικός αριθμός είναι το «συμπλήρωμα ως προς 2» του αντίστοιχου θετικού
- Συμπλήρωμα ως προς 2
 - Τι σημαίνει «συμπλήρωμα ως προς 2»;
 - Πώς υπολογίζεται;

Συμπλήρωμα ως προς 2

- Ίσο με το «συμπλήρωμα ως προς 1» + 1
- Εμπειρικός κανόνας:
 - Αντιστροφή όλων των bits εκτός από τα δεξιότερα συνεχόμενα 0 και το πρώτο 1 αριστερά από αυτά
- Συμπλήρωμα ως προς 2: παραδείγματα
001011100 $\Rightarrow$ 110100100
011111111 $\Rightarrow$ 100000001
- Προσοχή στο 0000...00 και στο 1000...00

Ακέραιοι σε συμπλήρωμα ως προς 2

- Διάστημα τιμών για αριθμούς με n bits
 - (2^{n-1}) έως $+(2^{n-1}-1)$ (π.χ. για $n=8$, $-128 \dots +127$)
 - Υπάρχει μια **ασυμμετρία**: το $+(2^{n-1})$ δεν μπορεί να αναπαρασταθεί με n bits
- Ευκολία στις πράξεις
 - Το ίδιο κύκλωμα προσθέτει αριθμούς με και χωρίς πρόσημο
 - Αφαίρεση = πρόσθεση του συμπληρώματος ως προς 2
 - Μία και μοναδική αναπαράσταση του 0
- Πιο σημαντικό bit: 0 για θετικούς, 1 για αρνητικούς
 - **Δεν είναι όμως bit προσήμου!**

Μετατροπές ακεραίων δυαδικών σε δεκαδικό

- Γνωρίζουμε να μετατρέπουμε δυαδικούς χωρίς πρόσημο σε δεκαδικούς
 - Είδαμε π.χ. ότι το $11110011 = 2^7 + 2^6 + 2^5 + 2^4 + 2^1 + 2^0 = 128 + 64 + 32 + 16 + 2 + 1 = 243$ (δεκαδικός)
- Τι συμβαίνει όταν ο δυαδικός αριθμός έχει πρόσημο;

Μετατροπές ακεραίων δυαδικών σε δεκαδικό

- Εξετάζουμε το περισσότερο σημαντικό (αριστερότερο) bit
 - Αν είναι 0 → ο αριθμός είναι θετικός: μετατρέπουμε σε δεκαδικό όπως στους αριθμούς χωρίς πρόσημο
 - Π.χ. ο αριθμός 01101100 είναι θετικός $= 2^6 + 2^5 + 2^3 + 2^2 = +108$
 - Αν είναι 1 → ο αριθμός είναι αρνητικός: πρώτα συμπληρώνουμε ως προς 2 για να βρούμε τον αντίστοιχο θετικό
 - Π.χ. ο 11110011 είναι αρνητικός, με θετικό τον 00001101 $= 2^3 + 2^2 + 2^0 = +13$
 - Άρα ο ζητούμενος αρνητικός (11110011) είναι ο -13

Ζητήματα αναπαράστασης

- Προσοχή: η σειρά bits **11110011** έχει διαφορετική σημασία ανάλογα με την επιλεγμένη **αναπαράσταση**
 - Αν το **11110011** αναπαριστά **αριθμό χωρίς πρόσημο** τότε πρόκειται για τον δεκαδικό 243 (φυσικός αριθμός)
 - Αν το **11110011** αναπαριστά **αριθμό με πρόσημο** τότε πρόκειται για τον δεκαδικό -13 (ακέραιος αριθμός)
- Συνεπώς δεν μπορούμε να ξέρουμε με τι ισούται μια σειρά από bits παρά μόνο αν γνωρίζουμε την **αναπαράσταση** που ισχύει!

Επέκταση προσήμου

- Συχνά στους υπολογιστές υπάρχει ανάγκη να γράψουμε έναν δυαδικό αριθμό με μεγαλύτερο αριθμό bits απ'όσα έχει
 - Αν πρόκειται για αριθμό χωρίς πρόσημο, αρκεί να συμπληρώσουμε με 0 έως τον απαιτούμενο αριθμό bits
 - Π.χ. για να επεκτείνουμε τον 8-bit αριθμό 01101100 στα 16 bits θα προσθέσουμε στα αριστερά 8 μηδενικά: 0000000001101100
- Αν όμως ο δυαδικός αριθμός έχει πρόσημο και είναι αρνητικός, η προσθήκη μηδενικών αλλοιώνει τον αριθμό
 - Π.χ. το 0000000011110011 δεν είναι το -13 (έγινε θετικός!)

Επέκταση προσήμου

- Όταν ο αριθμός έχει πρόσημο (θετικός ή αρνητικός) συμπληρώνουμε τα bits που λείπουν με το πιο σημαντικό (αριστερότερο) bit του αρχικού αριθμού
 - 01101100 → 0000000001101100
 - 11110011 → 1111111111110011
 - Η λειτουργία αυτή ονομάζεται επέκταση προσήμου

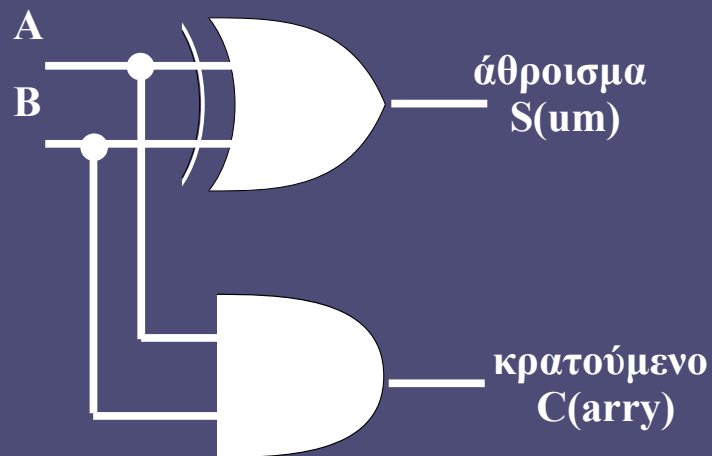
Αριθμητικές πράξεις

- Οι βασικές πράξεις
 - Πρόσθεση
 - Αφαίρεση
- Άλλες πράξεις
 - Πολλαπλασιασμός
 - Διαίρεση
 - Επίσης:
 - Τετραγωνική ρίζα, τριγωνομετρικές συναρτήσεις, εκθετικά, λογάριθμοι κλπ..
 - Υλοποίηση σε υλικό με διάφορες τεχνικές
 - Π.χ με πολυώνυμα

Προσθέτοντας 2 bits

bits	άθροισμα	κρατούμενο
$0 + 0$	0	0
$0 + 1$	1	0
$1 + 0$	1	0
$1 + 1$	0	1

Ημιαθροιστής (half-adder)



A	B	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Πώς γίνεται η πρόσθεση αριθμών με περισσότερα bits;

Προσθέτοντας δυαδικούς αριθμούς (χωρίς πρόσημο)

Κρατούμενο

A' Αριθμός (119)

0 1 1 1 0 1 1 1

B' Αριθμός (88)

0 1 0 1 1 0 0 0

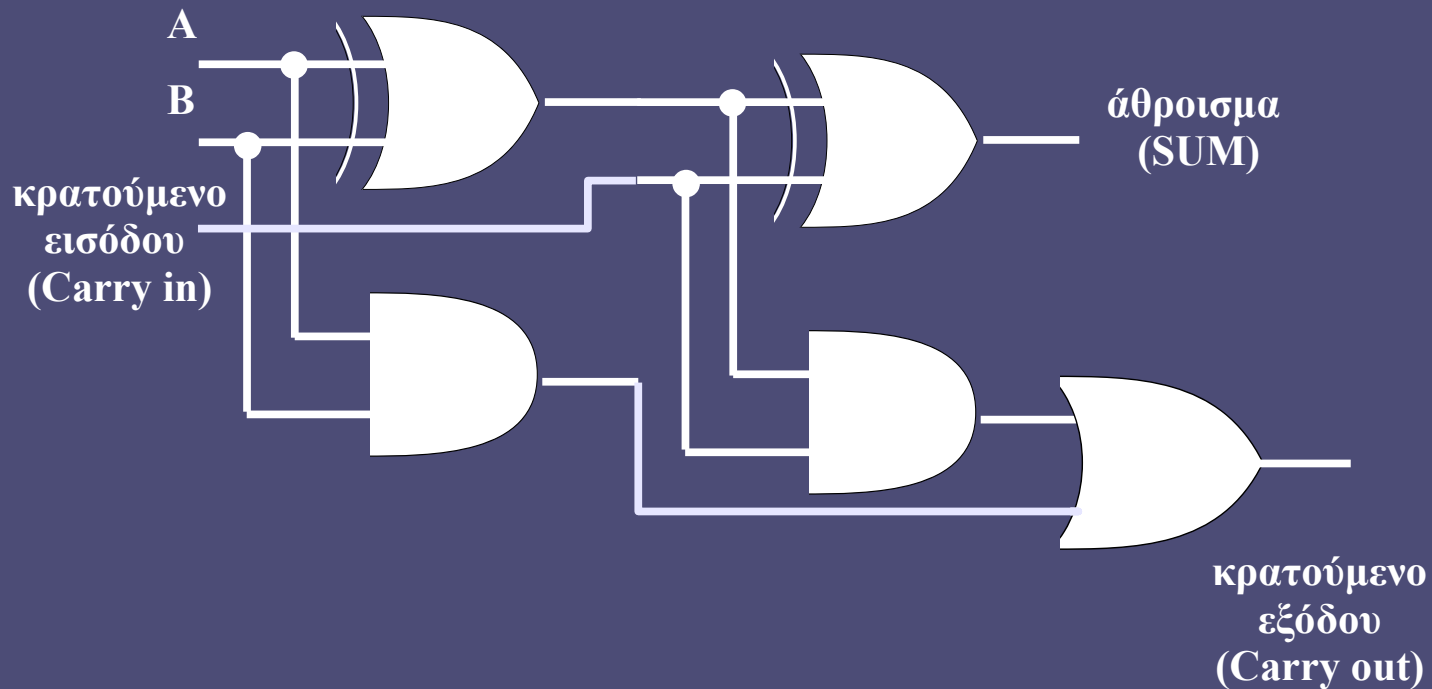
Άθροισμα (207)

1 1 0 0 1 1 1 1

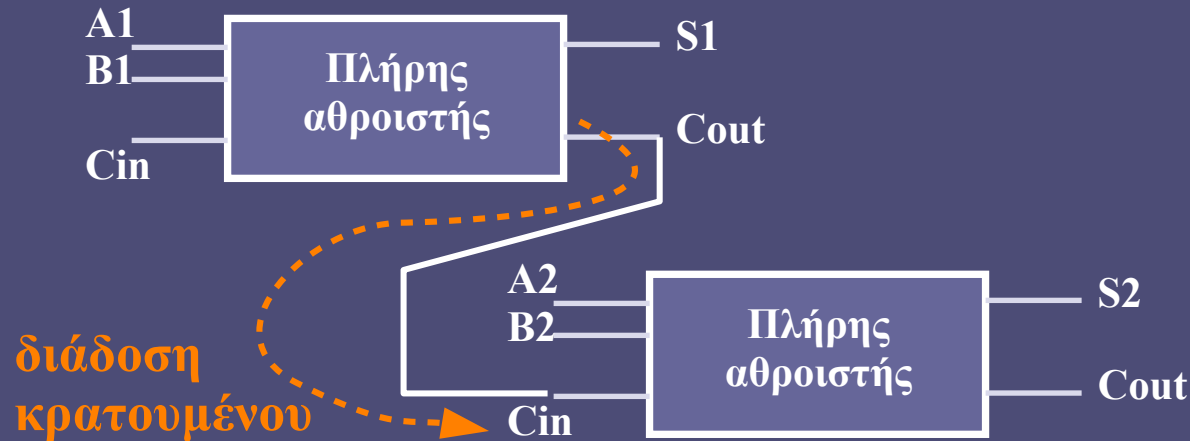
1. Αριθμοί με ίδιο μήκος (ίσος αριθμός bits)
2. Αρχίζοντας από το λιγότερο σημαντικό bit (το δεξιότερο)
3. Προσθέτουμε ζεύγη bits και μεταφέρουμε το κρατούμενο (αν υπάρχει) προς τα αριστερά
 - Το προσθέτουμε στο επόμενο ζεύγος bits

Πλήρης αθροιστής (full-adder)

- Μία από τις πιθανές υλοποιήσεις
 - με δύο ημιαθροιστές



Πρόσθεση αριθμών με πλήρεις αθροιστές



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

Προσθέτοντας δυαδικούς αριθμούς (χωρίς πρόσημο)

- Υπερχείλιση (overflow)
 - Στον υπολογιστή το πλήθος των bits ανά αριθμό είναι **προκαθορισμένο**
 - Το αποτέλεσμα της πρόσθεσης θα πρέπει να χωρά στα διαθέσιμα bits (π.χ. σε έναν **καταχωρητή**)
 - Αν δεν χωρά, έχουμε **υπερχείλιση**
- Αριθμοί χωρίς πρόσημο:
 - **Αριθμός με N bits $\rightarrow$ πεδίο τιμών $[0 \dots 2^N - 1]$**
 - Π.χ. για αριθμούς με 8 bits, από 0 έως 255

Προσθέτοντας δυαδικούς αριθμούς (χωρίς πρόσημο)

Κρατούμενο

A' Αριθμός (180)

B' Αριθμός (78)

Άθροισμα (258)

1 1 1 1 1 1

1 0 1 1 0 1 0 0

0 1 0 0 1 1 1 0

1 0 0 0 0 0 0 1 0

διαθέσιμος χώρος

ύπαρξη τελικού κρατουμένου = υπερχείλιση



Προσθέτοντας δυαδικούς αριθμούς (με πρόσημο)

- Ακέρατοι (με πρόσημο)
 - Οι αρνητικοί αριθμοί είναι σε συμπλήρωμα ως προς 2
 - Το περισσότερο σημαντικό bit **υποδηλώνει** το πρόσημο
 - 0=θετικός, 1=αρνητικός
 - **Αριθμός με N bits $\Rightarrow$ πεδίο τιμών $[-2^{N-1} \dots 0 \dots +2^{N-1} - 1]$**
 - π.χ. για αριθμούς με 8 bits, από -128 έως +127
- Πρόσθεση
 - Όπως ακριβώς με τους αριθμούς χωρίς πρόσημο
 - Αλλά: το τελικό κρατούμενο αγνοείται
 - **Δεν είναι ένδειξη υπερχείλισης**

Προσθέτοντας δυαδικούς αριθμούς (με πρόσημο)

Κρατούμενο

A' Αριθμός (+17)

0 0 0 1 0 0 0 1

B' Αριθμός (+22)

0 0 0 1 0 1 1 0

Άθροισμα (+39)

0 0 1 0 0 1 1 1

1

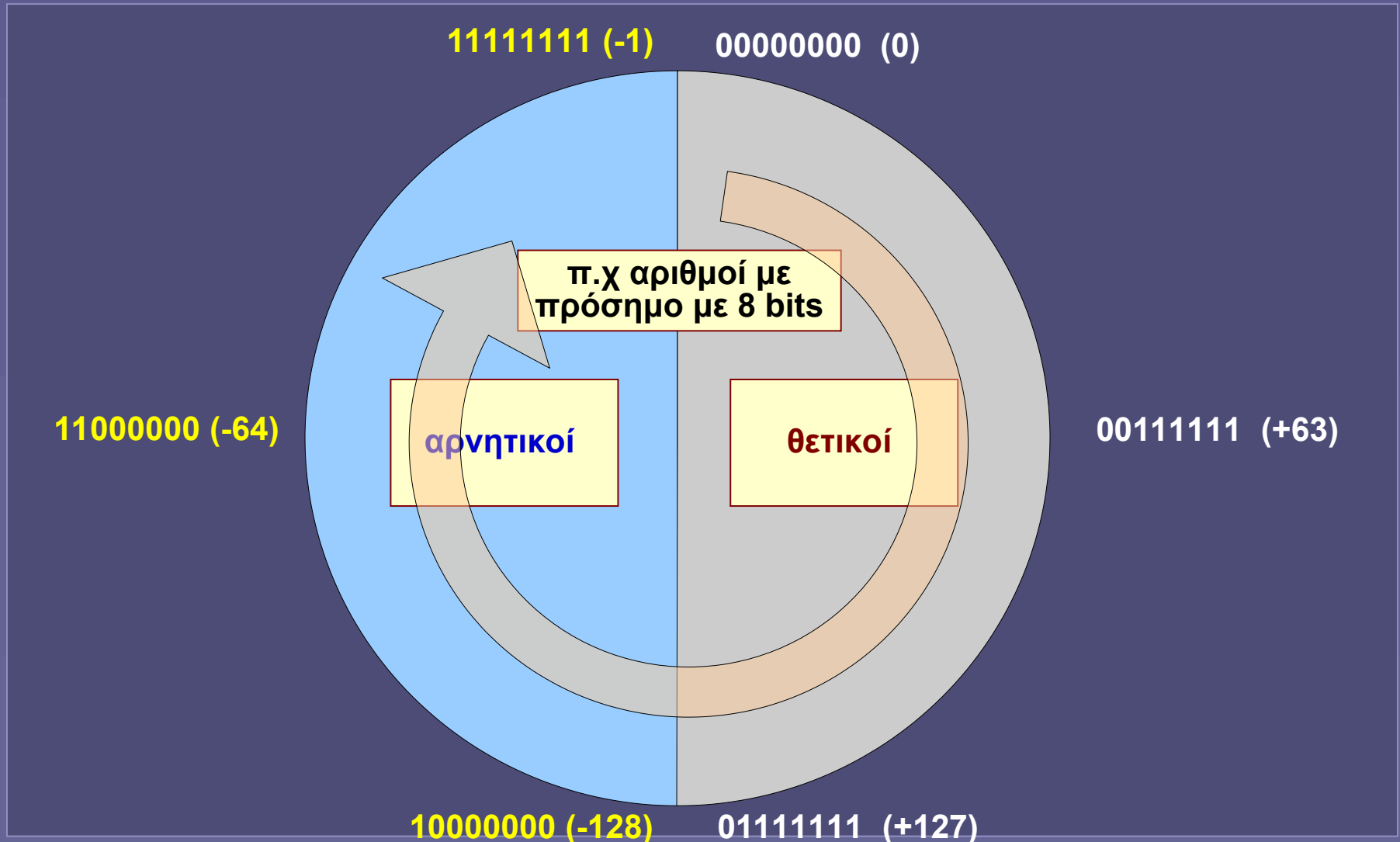


Προσθέτοντας δυαδικούς αριθμούς (με πρόσημο)

Κρατούμενο									
A' Αριθμός (+24)									
B' Αριθμός (-17)									
Άθροισμα (+7)									

- το κρατούμενο αγνοείται

Υπερχείλιση σε αριθμούς με πρόσημο



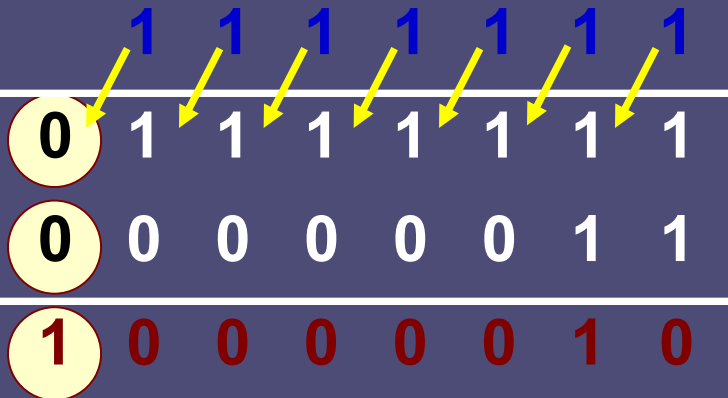
Υπερχείλιση σε αριθμούς με πρόσημο

Κρατούμενο

A' Αριθμός (+127)

B' Αριθμός (+3)

Άθροισμα (-126;)



- Το άθροισμα αριθμών με ίδιο πρόσημο θα πρέπει να έχει επίσης το ίδιο πρόσημο
 - στην αντίθετη περίπτωση: **υπερχείλιση**

Υπερχείλιση σε αριθμούς με πρόσημο

Κρατούμενο								
A' Αριθμός (-126)			0	0	0	0	0	
B' Αριθμός (-5)	1	1	1	1	1	0	1	0
Άθροισμα (+124;)	0	1	1	1	1	1	0	0

- Το άθροισμα αριθμών με ίδιο πρόσημο θα πρέπει να έχει επίσης το ίδιο πρόσημο
 - στην αντίθετη περίπτωση: **υπερχείλιση**

Κλασματικοί αριθμοί

- **Θεωρητικά**
 - Θα μπορούσαμε να επεξεργαζόμαστε ξεχωριστά το ακέραιο και το κλασματικό μέρος
- **Αλλά**
 - Αδυναμία αναπαράστασης πολύ μεγάλων και πολύ μικρών αριθμών
- **Η λύση**
 - Αριθμοί κινητής υποδιαστολής (**floating point**)
 - Εύκολη αναπαράσταση τόσο του **1.000.000.000.000** όσο και του **0,000000000000000001**
 - Προσοχή: οι αριθμοί κινητής υποδιαστολής είναι **προσεγγιστικοί**
 - Η αναπαράσταση κάποιων αριθμών είναι «στο περίπου»

Αριθμοί κινητής υποδιαστολής

- 3 μέρη
 - Πρόσημο (Π) (1 bit)
 - $0 = +$ $1 = -$
 - Εκθέτης (E) (8 ή 11 bits)
 - Η βάση είναι το 2 (εννοείται)
 - Θετικοί και αρνητικοί εκθέτες με πλεόνασμα 127 ή 1023 (π.χ. αντί -55, $E = -55 + 127 = 72$!)
 - Σημαινόμενο τμήμα (Σ) (23 ή 52 bits)
 - Κανονικοποίηση: μορφή $1,xxxxxxxxxxxxxxxxx...$
 - Το '1,' εννοείται και δεν αποθηκεύεται
- Τελικός αριθμός: $-1^{\Pi} \times 1.\Sigma \times 2^{E-127}$ (ή 2^{E-1023})
 - Ειδικοί αριθμοί: 0, ∞ , NaN (Not a Number)
- Στις εφαρμογές AI χρησιμοποιούνται και μορφές με λιγότερα bits σε εκθέτη και σημαινόμενο τμήμα

Πράξεις με αριθμούς κινητής υποδιαστολής

- Σύνθετη διαδικασία
- Η γενική μορφή της πρόσθεσης:
 1. Σύγκριση προσήμων
 - αν είναι ίδια $\Rightarrow$ πρόσθεση
 - αλλιώς $\Rightarrow$ αφαίρεση
 2. Εξίσωση εκθετών
 - «μετακίνηση» υποδιαστολής
 3. Πρόσθεση ή αφαίρεση σημαινόμενων τμημάτων
 - ακέραιο και κλασματικό μέρος
 4. Κανονικοποίηση αποτελέσματος
 - στη μορφή 1,xxxxxxxxxxxx
 5. Έλεγχος για υπερχείλιση

Πράξεις με αριθμούς κινητής υποδιαστολής

A' αριθμός: 0 **10000100** **1011**000000000000000000000000

+ $2^{132-127} \times 1,1011$ ($+2^5 \times 1,1011$)

B' αριθμός: 0 **10000010** **0110**000000000000000000000000

+ $2^{130-127} \times 1,011$ ($+2^3 \times 1,011$)

A	$+2^5$	x	1,10110
+ B	$+2^5$	x	0,01011
=	$+2^5$	x	10,00001
κανονικοποίηση	$+2^6$	x	1,000001

αποτέλεσμα: 0 **10000101** **000001**000000000000000000000000