

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής  
Εισαγωγή στην Επιστήμη των Υπολογιστών  
2017-18

# Πράξεις με δυαδικούς αριθμούς

(αριθμητικές πράξεις)

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

Μ.Στεφανιδάκης



# Πράξεις με δυαδικούς αριθμούς

- Δυαδικοί Αριθμοί

- Ο υπολογιστής μπορεί να εκτελέσει
  - Λογικές πράξεις
  - Αριθμητικές πράξεις
- Οι πράξεις εκτελούνται
  - Σε ομάδες bits (bytes ή πολλαπλάσιά τους)

# Το Byte ως δυαδικός αριθμός

- 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

# Μετατροπή δεκαδικού σε δυαδικό

- Δυαδικοί αριθμοί

το λιγότερο  
σημαντικό  
bit

!

Προσοχή στη θέση  
του περισσότερο  
σημαντικού bit!

243 (δεκαδικό)  
= 11110011

το περισσότερο  
σημαντικό bit

# Δεκαεξαδικό Σύστημα

- Δυαδικοί αριθμοί

- 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

- Δυαδικοί αριθμοί

- Κάθε 4 δυαδικά ψηφία αντιστοιχούν σε ένα δεκαεξαδικό!

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

- Δυαδικοί αριθμοί

▪ 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
- Δεν αλλάζει τον αριθμό, όπως ακριβώς και στο δεκαδικό σύστημα

# Φυσικοί αριθμοί (χωρίς πρόσημο)

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί

- Άμεση αντιστοιχία

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  $2^n - 1$



# Ποια η χρήση των “φυσικών αριθμών”;

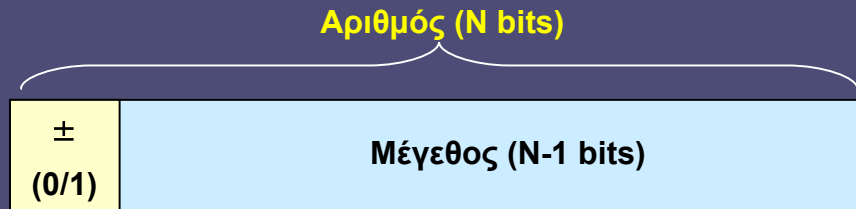
- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί

- Για αναπαράσταση
  - Διαφορετικών “πραγμάτων”
    - Συνήθως χωρίς αριθμητική έννοια
    - Αν και η ταξινόμηση είναι bonus!
  - Απαρίθμηση!
    - Παρέχοντας μοναδικούς αναγνωριστικούς αριθμούς
  - Παραδείγματα
    - Οι ξεχωριστές διευθύνσεις μνήμης
    - Οι χαρακτήρες σε ένα αλφάβητο
- Ξανά: με  $n$  bits απαριθμούνται
  - έως και  $2^n$  διαφορετικά “πράγματα”

# Ακέραιοι αριθμοί (με πρόσημο)

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- Ακέραιοι

- Πώς θα αναπαρασταθούν οι **αρνητικοί**;
  - Για να γίνονται εύκολα οι πράξεις!
- Όχι καλή ιδέα:
  - **Ξεχωριστό bit πρόσημου**



**Πρόσημο (1 bit)**

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

# Ακέραιοι αριθμοί (προσημασμένοι - signed)

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- **Ακέραιοι**

- Επίσης όχι καλή ιδέα:
  - **Συμπλήρωμα ως προς 1**
    - αντιστροφή όλων των bits του αριθμού
    - 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!
- Καλή ιδέα!
  - **Συμπλήρωμα ως προς 2**
    - Πώς υπολογίζεται;

# Συμπλήρωμα ως προς 2

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- **Ακέραιοι**

- 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”
  - Προσοχή στο 0 (και το 10000....0)
- Συμπλήρωμα ως προς 2: παραδείγματα
- 001011100  $\Rightarrow$  110100**100**
- 011111111  $\Rightarrow$  10000000**1**
- Προσοχή:
- 000000000  $\Rightarrow$  000000000

# Ακέραιοι σε συμπλήρωμα ως προς 2

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- Ακέραιοι

- Διάστημα τιμών για αριθμούς με  $n$  bits  
 $-(2^{n-1})$  έως  $+(2^{n-1}-1)$  (για  $n=8$ ,  $-128 \dots +127$ )
  - Μόνο το  $+(2^{n-1})$  δεν μπορεί να αναπαρασταθεί
- Ευκολία στις πράξεις
  - 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!!!

# Κλασματικοί αριθμοί

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- Ακέραιοι
- Κλασματικοί

- Θεωρητικά:
  - Θα μπορούσαμε να επεξεργαζόμαστε ξεχωριστά το ακέραιο και το κλασματικό μέρος
- Αλλά:
  - Δυσκολία στις πράξεις – απώλεια ακρίβειας κατά τις διαιρέσεις
  - 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

# Αριθμοί κινητής υποδιαστολής

- Δυαδικοί αριθμοί
- Φυσικοί αριθμοί
- Ακέραιοι
- Κλασματικοί

- 3 μέρη
  - Πρόσημο ( $\Pi$ ) (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 ( $\Sigma$ ) (23 ή 52 bits)
    - Κανονικοποίηση: μορφή  $1,xxxxxxxxxxxxx\dots$
    - Το '1,' εννοείται και δεν αποθηκεύεται
- Τελικός αριθμός:  $-1^{\Pi} \times 1.\Sigma \times 2^{E-127}$  (ή  $2^{E-1023}$ )
  - Ειδικοί αριθμοί: 0,  $\infty$ , NaN (Not a Number)

# Αριθμητικές πράξεις

- 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

- Αριθμητικές πράξεις

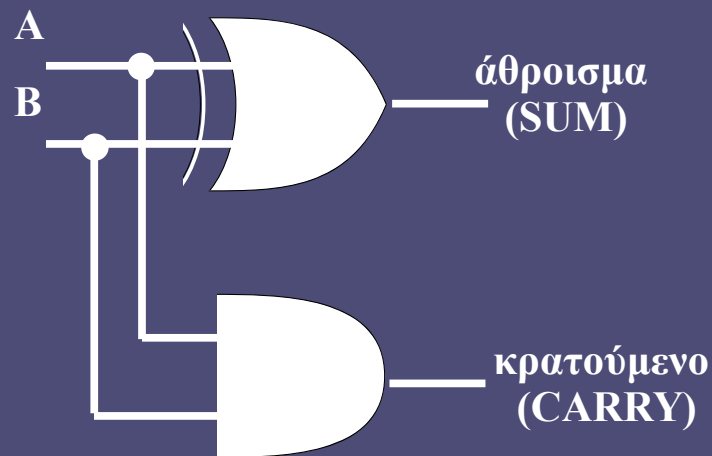
bits	άθροισμα	κρατούμενο
$0 + 0$	0	0
$0 + 1$	1	0
$1 + 0$	1	0
$1 + 1$	0	1

# Ημιαθροιστής (half-adder)

- Αριθμητικές πράξεις

;

Αν απαιτείται  
πρόσθεση αριθμών  
με περισσότερα  
bits;



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

# Προσθέτοντας δυαδικούς αριθμούς (μη προσημασμένους)

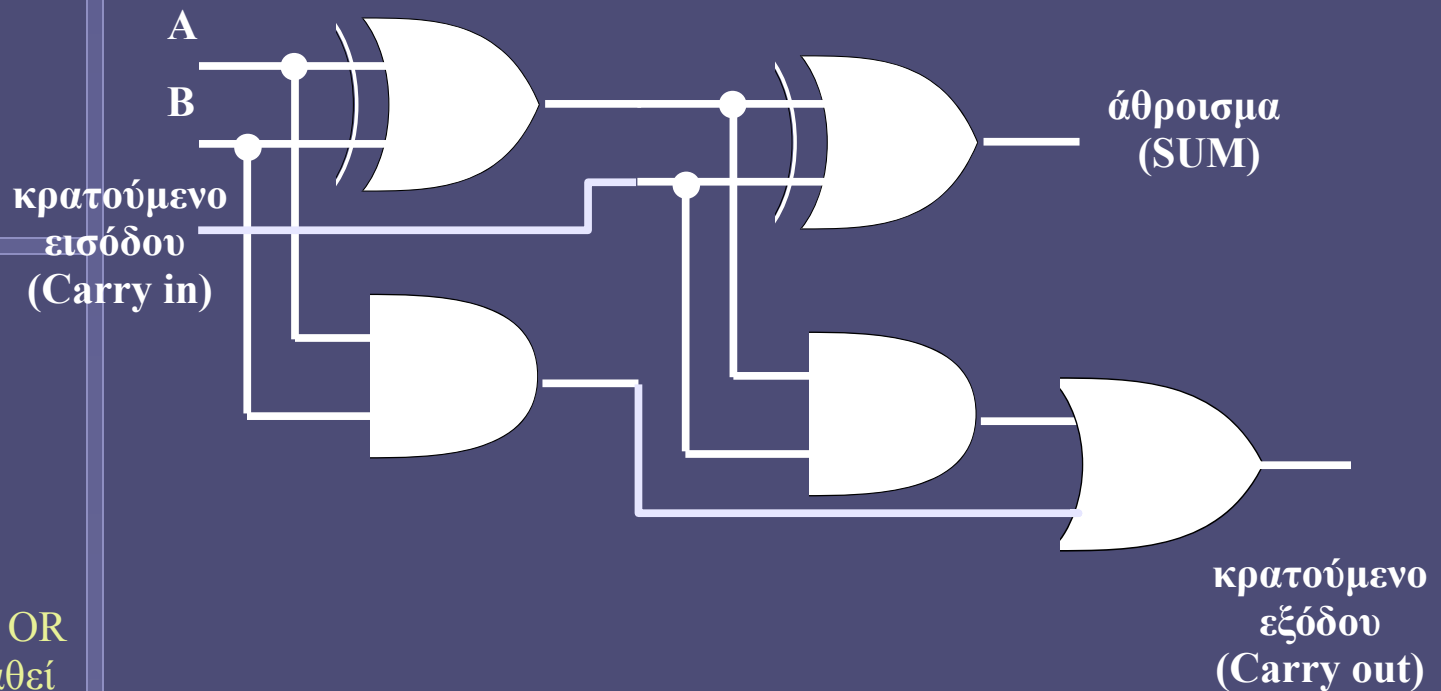
Κρατούμενο								
A' Αριθμός (119)	0	1	1	1	0	1	1	1
B' Αριθμός ( 88)	0	1	0	1	1	0	0	0
Άθροισμα (207)	1	1	0	0	1	1	1	1

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)

- Αριθμητικές πράξεις

- Μία από τις πιθανές υλοποιήσεις
  - με δύο ημιαθροιστές



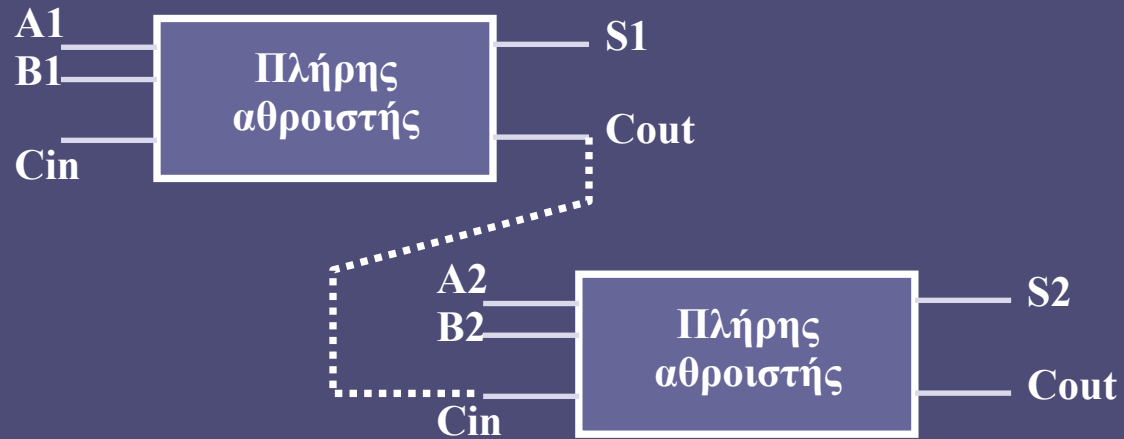
;

Ποιος πίνακας αλήθειας υλοποιείται;

Μπορεί η πύλη OR να αντικατασταθεί από XOR;

# Πρόσθεση αριθμών με πλήρεις αθροιστές

- Αριθμητικές πράξεις



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

# Προσθέτοντας δυαδικούς αριθμούς (μη προσημασμένους)

- Υπερχείλιση
  - Στον υπολογιστή το πλήθος των bits ανά αριθμό είναι προκαθορισμένο
  - Το αποτέλεσμα της πρόσθεσης θα πρέπει να χωρά στα διαθέσιμα bits ενός καταχωρητή
  - Μη προσημασμένοι αριθμοί:
    - αριθμός με  $N$  bits  $\Rightarrow$  πεδίο τιμών  $[ 0 \dots 2^N - 1 ]$
    - π.χ. για αριθμούς με 8 bits, από 0 έως 255

**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 (προσημασμένους)







- Προσημασμένοι ακέραιοι
  - 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 σε μη προσημασμένους
  - Τελικό κρατούμενο αγνοείται
    - Πώς γίνεται τώρα ο έλεγχος υπερχείλισης;
  - Αφαίρεση = πρόσθεση του συμπληρώματος ως προς 2 του αφαιρετέου
    - $A - B = A + (-B)$
    - χωρίς πρόσθετα κυκλώματα για την αφαίρεση!

# Προσθέτοντας δυαδικούς αριθμούς (προσημασμένους)

Κρατούμενο								
Α' Αριθμός (+17)	0	0	0	1	0	0	0	1
Β' Αριθμός (+22)	0	0	0	1	0	1	1	0
Άθροισμα (+39)	0	0	1	0	0	1	1	1

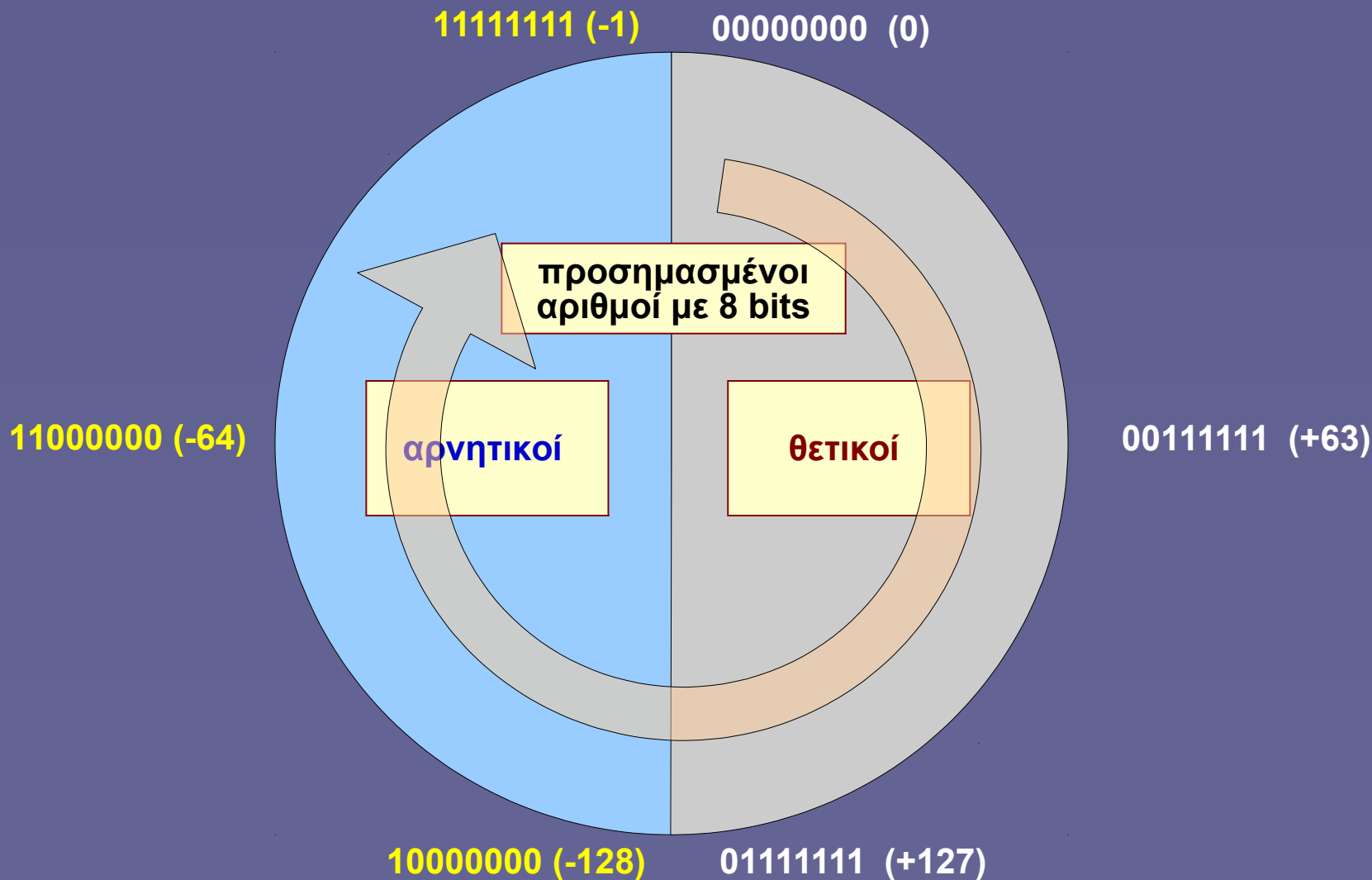


# Προσθέτοντας δυαδικούς αριθμούς (προσημασμένους)

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

- το κρατούμενο αγνοείται

# Υπερχείλιση σε προσημασμένους αριθμούς



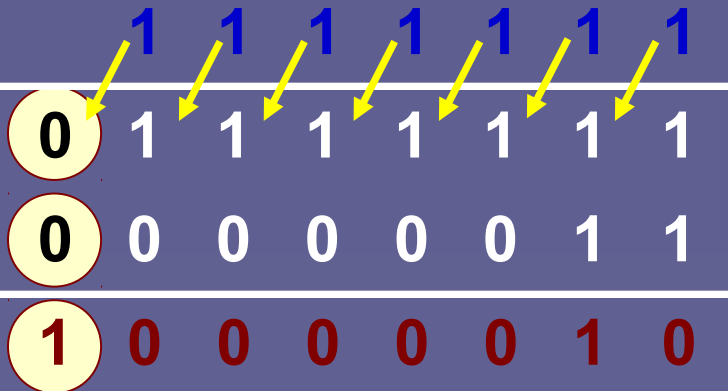
# Υπερχείλιση σε προσημασμένους αριθμούς

Κρατούμενο

Α' Αριθμός (+127)

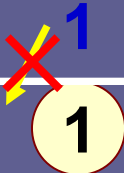



Β' Αριθμός ( +3)

Άθροισμα (-126;)



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

# Υπερχείλιση σε προσημασμένους αριθμούς

Κρατούμενο								
A' Αριθμός (-126)		0	0	0	0	0		0
B' Αριθμός (-5)		1	1	1	1	0	1	0
Άθροισμα (+124;)		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**
  - πώς θα ήταν ένα κύκλωμα με πύλες για ανίχνευση υπερχείλισης;

# Πράξεις με αριθμούς κινητής υποδιαστολής

- Αριθμητικές πράξεις

- 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
  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
<hr/>			
=	$+2^5$	x	10,00001
κανονικοποίηση	$+2^6$	x	<b>1,000001</b>

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