

Ψηφιακή Σχεδίαση

Δεδομένα & Αναπαράσταση Πληροφορίας

ΕΛΕΥΘΕΡΙΟΣ ΚΟΣΜΑΣ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2019-2020 | ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Περίληψη

Σήμερα ...

- ▶ Θα μιλήσουμε για **δεδομένα**, **ψηφιακά δεδομένα** και **πληροφορίες**
- ▶ Θα αναφέρουμε τα **πληροφοριακά συστήματα**
- ▶ Θα συζητήσουμε τους **δυαδικούς αριθμούς**, την **εκτέλεση πράξεων** μεταξύ τους και τον τρόπο **μετατροπής** αριθμών σε συστήματα με άλλη **βάση**
- ▶ Θα μελετήσουμε τα **συμπληρώματα** αριθμών και θα συζητήσουμε τη χρησιμότητά τους
- ▶ Θα συζητήσουμε τη χρησιμότητα των **δυαδικών κωδίκων** για την αναπαράσταση (**κωδικοποίηση**) πληροφοριών στα ψηφιακά συστήματα

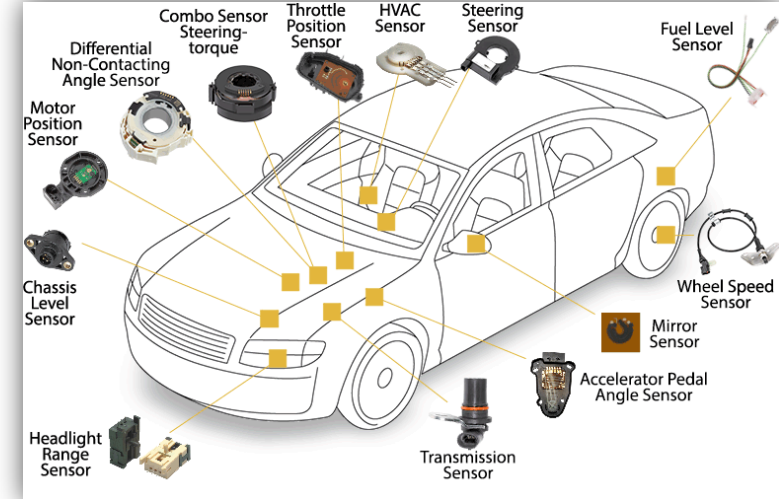
Δεδομένα & Πληροφορίες

Ποια η διαφορά τους;

Δεδομένα & Ψηφιακά Δεδομένα

Δεδομένα

- ❖ ένα σύνολο **διακριτών** στοιχείων σχετικά με ένα συμβάν ή μια διαδικασία **χωρίς** κάποια άμεση σημασία
- ❖ τα δεδομένα είναι ουσιαστικά **οντότητες**
 - ▶ χαρακτήρες, κείμενο, αριθμοί, σύμβολα, σχήματα, εικόνες, ...
- και **ιδιότητες**
 - ▶ που προκύπτουν από παρατηρήσεις, πειράματα ή υπολογισμούς



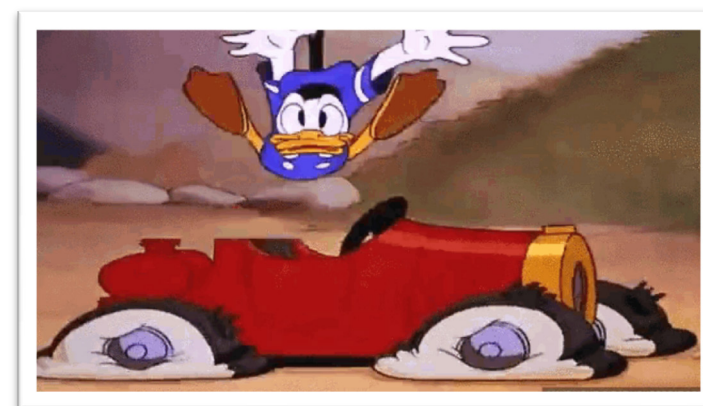
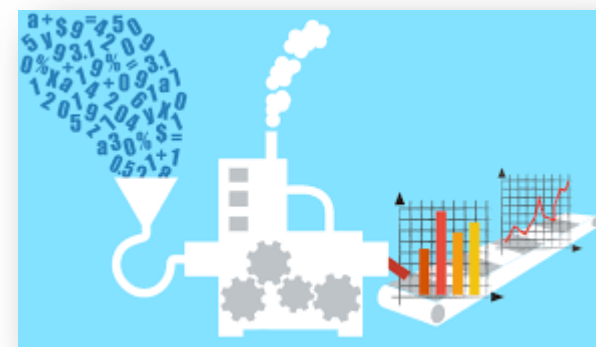
Ψηφιακά δεδομένα (data)

- ❖ δεδομένα που έχουν **αναπαρασταθεί** με κάποιο τρόπο (**κωδικοποίηση**)
- ❖ είναι **αποθηκευμένα** σε υπολογιστή
 - ▶ σε συλλογές δεδομένων που έχουν κάποια σχέση μεταξύ τους
 - ▶ κείμενο, εγγραφές και βάσεις δεδομένων
- ❖ είναι δυνατόν να **επεξεργαστούν**



Πληροφορίες

- ❖ παράγονται από την **επεξεργασία** (ή/και την **ερμηνεία**) των ψηφιακών δεδομένων
- ❖ **επεξεργασμένα** δεδομένα στα οποία έχει δοθεί κάποιο **νόημα**
- ❖ βοηθούν στη λήψη **αποφάσεων**
- ❖ **επεξεργασία** ψηφιακών δεδομένων:
 - ▶ συλλογή
 - ▶ αναζήτηση
 - ▶ ομαδοποίηση
 - ▶ ταξινόμηση
 - ▶ σύγκριση
 - ▶ επιλογή
 - ▶ εκτέλεση αριθμητικών/λογικών υπολογισμών



Δεδομένα & Πληροφορία

Παράδειγμα

έστω οι αριθμοί: 8, 5, 7, 6, 7

? τι σημαίνουν;

▶ π.χ. ημέρες, μήνες, κιλά, γραμμάρια, άτομα, αντικείμενα, ...

👉 ως **δεδομένα** → **δεν** έχουν κάποιο νόημα

❖ ως **βαθμοί** εργασιών ενός φοιτητή σε ένα μάθημα → **αποκτούν** σημασία!

? Γιατί;

▶ ο **μέσος όρος** αυτών μπορεί να μας οδηγήσει στην **απόφαση** αν ο φοιτητής περνά το μάθημα

▶ ο **υπολογισμός** του μέσου όρου αποτελεί την **επεξεργασία** των δεδομένων

▶ αν ο μέσος όρος είναι ≥ 5 τότε έχουμε την **πληροφορία** ότι ο φοιτητής περνάει το μάθημα

✂ **δεδομένα** + **επεξεργασία** → **πληροφορία**

Πληροφοριακά Συστήματα

Σύντομη αναφορά

Πληροφορική

- ❖ παροχή **έγκαιρης** και **έγκυρης** πληροφόρησης
 - ▶ καταχώριση στοιχείων
 - ▶ αποθήκευση
 - ▶ επεξεργασία ψηφιακών δεδομένων
 - ▶ μεταφορά ψηφιακών δεδομένων
 - ▶ άντληση πληροφοριών
- ❖ **εργαλεία**
 - ▶ υπολογιστές - περιφερειακά
 - ▶ μετάδοση ψηφιακών δεδομένων
- ❖ **προϊόν**: πληροφοριακό σύστημα
 - ▶ hardware (υπολογιστές, δίκτυα)
 - ▶ software (προγράμματα)

Πληροφορική

Γιατί ψηφιακά δεδομένα;

- 👍 είναι δυνατόν να επεξεργαστούν από ψηφιακά συστήματα (π.χ. ψηφιακούς υπολογιστές)
 - ▶ άρα και να κοστολογηθούν!
- 👍 αποθηκεύονται εύκολα
 - ▶ άρα είναι δυνατόν να επαναληφθούν
- 👍 διακινούνται με ταχύτητα και με αξιοπιστία
 - ▶ εξαιρετικά ενδιαφέρον για την υποστήριξη διαφόρων υπηρεσιών

Πληροφορική

Πληροφοριακό σύστημα

- ❖ προϊόν της πληροφορικής
- ❖ βοηθά στην επίλυση προβλημάτων που μας απασχολούν στη καθημερινή μας ζωή
- ❖ ένα πληροφοριακό σύστημα:
 - ▶ δέχεται ακατέργαστα στοιχεία
 - ▶ τα αποθηκεύει ως δεδομένα (data)
 - ▶ κάνει επεξεργασία αυτών των δεδομένων και παράγει πληροφορία
 - ▶ παρουσιάζει πληροφορίες
 - ▶ μεταδίδει δεδομένα
- ✍ αποτελείται από το υλικό και το λογισμικό

Πληροφοριακό Σύστημα

Απεικόνιση



Συστήματα Αρίθμησης

δεκαδικό, δυαδικό, οκταδικό, δεκαεξαδικό

Αναπαράσταση Αριθμών (από ανθρώπους)

Δεκαδικό σύστημα αρίθμησης

- ❖ (οι άνθρωποι) το χρησιμοποιούμε για να παραστήσουμε (δεκαδικούς) αριθμούς
- ❖ βάση: 10
- ❖ δέκα διαφορετικά ψηφία: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
π.χ.

- ▶ ο δεκαδικός αριθμός 7392 είναι μία συντομογραφία της μαθηματικής έκφρασης:

$$7 \cdot 10^3 + 3 \cdot 10^2 + 9 \cdot 10^1 + 2 \cdot 10^0$$

- ▶ ο δεκαδικός αριθμός 26.75 είναι μία συντομογραφία της μαθηματικής έκφρασης:

$$2 \cdot 10^1 + 6 \cdot 10^0 + 7 \cdot 10^{-1} + 5 \cdot 10^{-2}$$

- ❖ γενικά, ένας αριθμός A ($a_{n-1}a_{n-2} \dots a_0 . a_{-1}a_{-2} \dots a_{-m+1}a_{-m}$) με βάση 10 έχει συντελεστές που πολλαπλασιάζονται με δυνάμεις του 10:

$$a_{n-1} \cdot 10^{n-1} + a_{n-2} \cdot 10^{n-2} + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + a_{-2} \cdot 10^{-2} + \dots + a_{-m+1} \cdot 10^{-m+1} + a_{-m} \cdot 10^{-m}$$

όπου n, m φυσικοί αριθμοί με $n \geq 1$ και $m \geq 0$

Αναπαράσταση Πληροφορίας (στα ψηφιακά συστήματα)

Δυαδικά ψηφία (binary digits, bits)

- ❖ τα ψηφιακά συστήματα είναι σε θέση να αναγνωρίζουν **2 διακριτές** καταστάσεις:
 1. περνά ρεύμα
 2. δεν περνά ρεύμα

το **δυαδικό ψηφίο** (binary digit ή bit)

- ❖ είναι η **μικρότερη** μονάδα μέτρησης
- ❖ **εκφράζει** αυτές τις 2 διακριτές καταστάσεις
 - ▶ τιμή **1** → περνά ρεύμα
 - ▶ τιμή **0** → δεν περνά ρεύμα

Αναπαράσταση Πληροφορίας (στα ψηφιακά συστήματα)

Δυαδικό σύστημα αρίθμησης

- ❖ (τα ψηφιακά συστήματα) το χρησιμοποιούν για να παραστήσουν (δυαδικούς) αριθμούς
 - ❖ βάση: 2
 - ❖ δύο διαφορετικά ψηφία: 0, 1
- π.χ.

αριθμός με
βάση 2

- ▶ το δεκαδικό ισοδύναμο του δυαδικού αριθμού **11011** προκύπτει ως εξής:

$$(11011)_2 = 1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = (27)_{10}$$

αριθμός με
βάση 10

- ▶ το δεκαδικό ισοδύναμο του δυαδικού αριθμού **101.01** προκύπτει ως εξής:

$$(101.01)_2 = 1*2^2 + 0*2^1 + 1*2^0 + 0*2^{-1} + 1*2^{-2} = (5.25)_{10}$$

- ❖ γενικά, ένας αριθμός **A** ($a_{n-1}a_{n-2} \dots a_0.a_{-1}a_{-2} \dots a_{-m+1}a_{-m}$) με βάση 2 έχει συντελεστές που πολλαπλασιάζονται με δυνάμεις του 2:

$$a_{n-1}*2^{n-1} + a_{n-2}*2^{n-2} + \dots + a_1*2^1 + a_0*2^0 + a_{-1}*2^{-1} + a_{-2}*2^{-2} + \dots + a_{-m+1}*2^{-m+1} + a_{-m}*2^{-m}$$

όπου **n**, **m** φυσικοί αριθμοί με **n** ≥ 1 και **m** ≥ 0

Άλλα συστήματα αρίθμησης

Οκταδικό

- ❖ βάση: 8
- ❖ οκτώ διαφορετικά ψηφία: 0, 1, 2, 3, 4, 5, 6, 7
π.χ.

► το δεκαδικό ισοδύναμο του οκταδικού αριθμού 127 προκύπτει ως εξής:

αριθμός με
βάση 8

$$(127)_8 = 1*8^2 + 2*8^1 + 7*8^0 = (87)_{10}$$

αριθμός με
βάση 10

► το δεκαδικό ισοδύναμο του οκταδικού αριθμού 127.4 προκύπτει ως εξής:

$$(127.4)_8 = 1*8^2 + 2*8^1 + 7*8^0 + 4*8^{-1} = (87.5)_{10}$$

- ❖ γενικά, ένας αριθμός A ($a_{n-1}a_{n-2}\dots a_0.a_{-1}a_{-2}\dots a_{-m+1}a_{-m}$) με βάση 8 έχει συντελεστές που πολλαπλασιάζονται με δυνάμεις του 8:

$$a_{n-1}*8^{n-1} + a_{n-2}*8^{n-2} + \dots + a_1*8^1 + a_0*8^0 + a_{-1}*8^{-1} + a_{-2}*8^{-2} + \dots + a_{-m+1}*8^{-m+1} + a_{-m}*8^{-m}$$

όπου n, m φυσικοί αριθμοί με $n \geq 1$ και $m \geq 0$

Άλλα συστήματα αρίθμησης

Δεκαεξαδικό

- ❖ βάση: 16
- ❖ δεκαέξι διαφορετικά ψηφία: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
 - ▶ όπου τα γράμματα A, B, C, D, E, F χρησιμοποιούνται για τα ψηφία τάξης 10, 11, 12, 13, 14, 15, αντίστοιχα

π.χ.

- ▶ το δεκαδικό ισοδύναμο του δεκαεξαδικού αριθμού B65F προκύπτει ως εξής:

αριθμός με
βάση 16

$$(B65F)_{16} = 11 \cdot 16^3 + 6 \cdot 16^2 + 5 \cdot 16^1 + 15 \cdot 16^0 = (46687)_{10}$$

αριθμός με
βάση 10

- ❖ γενικά, ένας αριθμός A ($a_{n-1}a_{n-2} \dots a_0 \cdot a_{-1}a_{-2} \dots a_{-m+1}a_{-m}$) με βάση 16 έχει συντελεστές που πολλαπλασιάζονται με δυνάμεις του 16:

$$a_{n-1} \cdot 16^{n-1} + a_{n-2} \cdot 16^{n-2} + \dots + a_1 \cdot 16^1 + a_0 \cdot 16^0 + a_{-1} \cdot 16^{-1} + a_{-2} \cdot 16^{-2} + \dots + a_{-m+1} \cdot 16^{-m+1} + a_{-m} \cdot 16^{-m}$$

όπου n, m φυσικοί αριθμοί με $n \geq 1$ και $m \geq 0$

Συστήματα αρίθμησης

Πίνακας αντιστοιχίας αριθμητικών συστημάτων

10-δικό	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2-δικό	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000
8-δικό	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20
16-δικό	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

Συστήματα αρίθμησης

Γενικά & Μετατροπή αριθμού στο δεκαδικό σύστημα

- ❖ βάση: r
- ❖ r διαφορετικά ψηφία: $0, 1, 2, 3, 4, \dots, r-2, r-1$
- ❖ ένας αριθμός A ($a_{n-1}a_{n-2}\dots a_0a_{-1}a_{-2}\dots a_{-m+1}a_{-m}$) με βάση r έχει συντελεστές που πολλαπλασιάζονται με δυνάμεις του r :

$$a_{n-1} * r^{n-1} + a_{n-2} * r^{n-2} + \dots + a_1 * r^1 + a_0 * r^0 + a_{-1} * r^{-1} + a_{-2} * r^{-2} + \dots + a_{-m+1} * r^{-m+1} + a_{-m} * r^{-m}$$

όπου n, m φυσικοί αριθμοί με $n \geq 1$ και $m \geq 0$

- ✍ με αυτό τον τρόπο μπορούμε να μετατρέψουμε έναν αριθμό (σε σύστημα αρίθμησης) οποιασδήποτε βάσης στον αντίστοιχο ισοδύναμο δεκαδικό αριθμό

Συστήματα αρίθμησης

Δυαδικό - Δυνάμεις του 2

n	2 ⁿ	n	2 ⁿ	n	2 ⁿ
0	1	8	256	16	65,536
1	2	9	512	17	131,072
2	4	10	1,024	18	262,144
3	8	11	2,048	19	524,288
4	16	12	4,096	20	1,048,576
5	32	13	8,192	21	2,097,152
6	64	14	16,384	22	4,194,304
7	128	15	32,768	23	8,388,608

❖ στην επιστήμη των υπολογιστών:

- ▶ το **2¹⁰** ονομάζεται **kilo (K)**
- ▶ το **2²⁰** ονομάζεται **mega (M)**
- ▶ το **2³⁰** ονομάζεται **giga (G)**
- ▶ το **2⁴⁰** ονομάζεται **tera (T)**
- ▶ ένα **byte** ισούται με **8 bit**
- ▶ ένας σκληρός δίσκος υπολογιστή με **4T** χώρου αποθήκευσης έχει χωρητικότητα το **2⁴² byte**
 - ▶ δηλαδή περίπου **4** τρισεκατομμύρια **byte**

Συστήματα αρίθμησης

Πράξεις

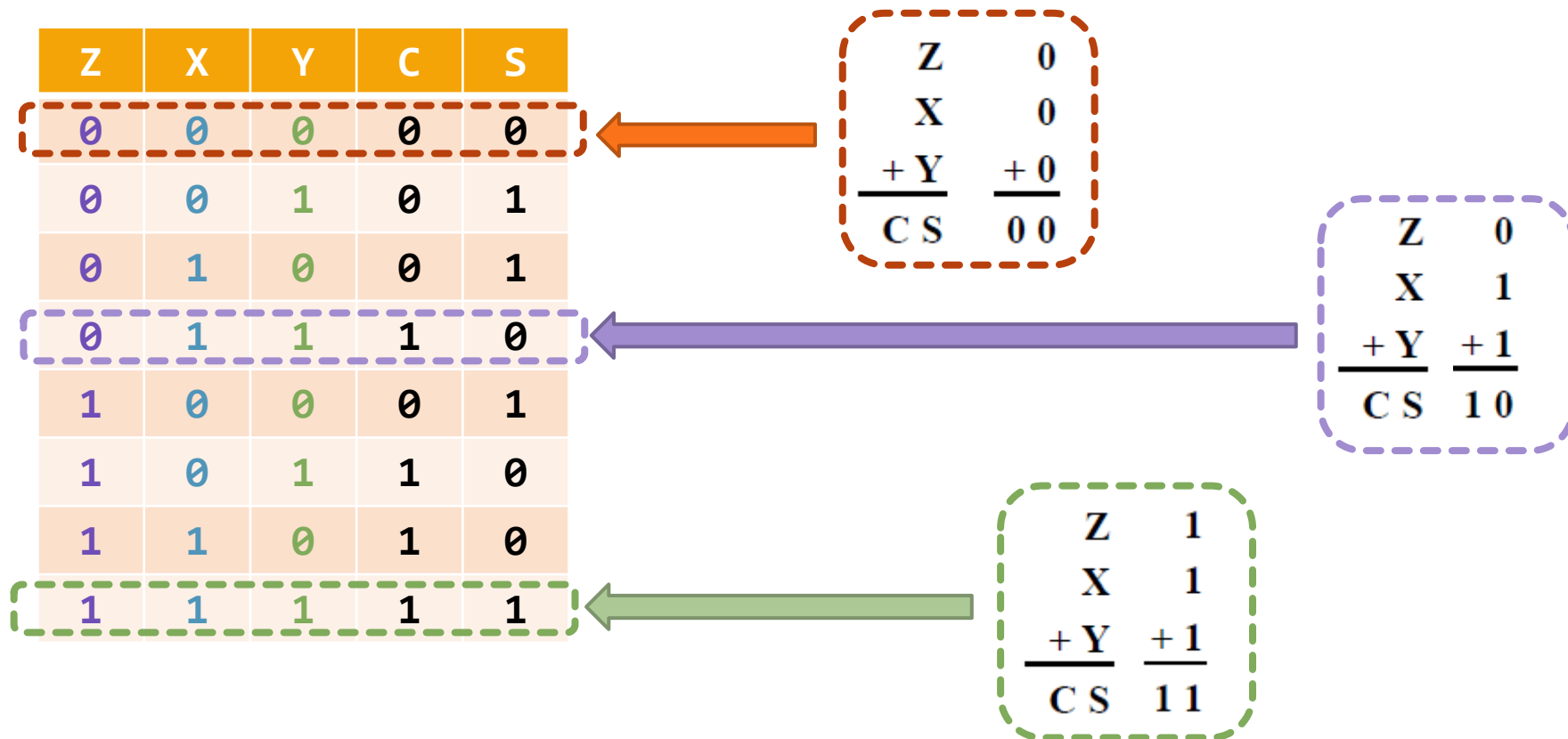
αριθμητικές πράξεις με αριθμούς σε βάση r

- ▶ γίνονται με χρήση των ίδιων κανόνων με αυτούς που χρησιμοποιούνται για δεκαδικούς αριθμούς
- ▶ εάν $r \neq 10$ πρέπει να προσέξουμε να χρησιμοποιούμε μόνο τα r επιτρεπόμενα ψηφία

Δυαδικό σύστημα αρίθμησης

Πρόσθεση ενός bit με κρατούμενο

- ▶ πρόσθεση δύο δυαδικών ψηφίων (**X**, **Y**) και ενός αρχικού κρατουμένου (**Z**) → δίνει ως αποτέλεσμα το άθροισμα (**S**) και ένα κρατούμενο (**C**)



Δυαδικό σύστημα αρίθμησης

Πρόσθεση αριθμών με πολλά bit

- ▶ π.χ. πρόσθεση των αριθμών $(01010)_2$ και $(10011)_2$

το 0 είναι εξ ορισμού το αρχικό κρατούμενο, στην περίπτωση πρόσθεσης δύο δυαδικών αριθμών

κρατούμενα	0	0	1	0	0
	0	1	0	1	0
+	1	0	0	1	1
άθροισμα	1	1	1	0	1

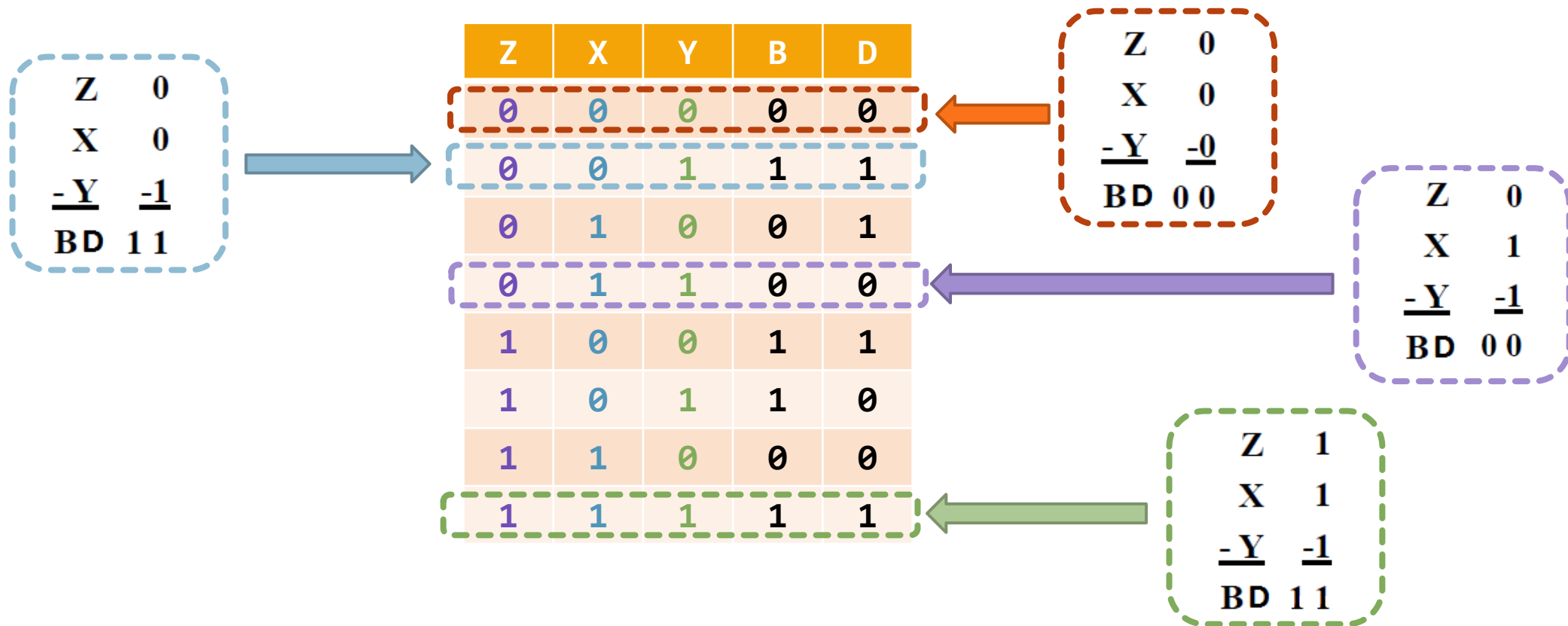
- ▶ π.χ. πρόσθεση των αριθμών $(1011)_2$ και $(1011)_2$

κρατούμενα	1	0	1	1	0
		1	0	1	1
+		1	0	1	1
άθροισμα	1	0	1	1	0

Δυαδικό σύστημα αρίθμησης

Αφαίρεση ενός bit με δανεικό

- ▶ αφαίρεση δύο δυαδικών ψηφίων (**X**, **Y**) και ενός δανεικού εισόδου (**Z**) → δίνει ως αποτέλεσμα τη διαφορά (**D**) και ένα δανεικό εξόδου (**B**)



Δυαδικό σύστημα αρίθμησης

Αφαίρεση αριθμών με πολλά bit

- π.χ. αφαίρεση των αριθμών $(10110)_2$ και $(10010)_2$

το 0 είναι εξ ορισμού το δανεικό εισόδο, στην περίπτωση αφαίρεσης δύο δυαδικών αριθμών

κρατούμενα	0	0	0	0	0
	1	0	1	1	0
-	1	0	0	1	0
διαφορά	0	0	1	0	0

- π.χ. αφαίρεση των αριθμών $(10110)_2$ και $(10011)_2$

κρατούμενα	0	0	1	1	0
	1	0	1	1	0
-	1	0	0	1	1
διαφορά	0	0	0	1	1

? πώς γίνεται η αφαίρεση εάν ο αφαιρετέος είναι μεγαλύτερος του μειωτέου;

1. τους αντιστρέφουμε
2. κάνουμε κανονικά την αφαίρεση
3. προσθέτουμε το αρνητικό πρόσημο (-) στο αποτέλεσμα

Δυαδικό σύστημα αρίθμησης

Πολλαπλασιασμός

- ❖ ο πολλαπλασιασμός δύο δυαδικών ψηφίων (**X**, **Y**) είναι απλός

X	Y	X*Y
0	0	0
0	1	0
1	0	0
1	1	1

- ❖ ο πολλαπλασιασμός δυαδικών αριθμών με πολλά bit, είναι επίσης απλός

			1	0	1	1
*				1	0	1
			1	0	1	1
		0	0	0	0	
+	1	0	1	1		
γινόμενο	1	1	0	1	1	1

Συστήματα αρίθμησης

Μετατροπή βάσης

- ❖ έχουμε ήδη συζητήσει τον τρόπο μετατροπής ενός αριθμού (σε σύστημα αρίθμησης) οποιασδήποτε βάσης → στον αντίστοιχο ισοδύναμο δεκαδικό αριθμό
- ❖ στη συνέχεια, θα μελετήσουμε τον τρόπο μετατροπής δεκαδικών αριθμών → στα εξής συστήματα αρίθμησης:
 - ▶ δυαδικό
 - ▶ οκταδικό
 - ▶ δεκαεξαδικό

Συστήματα αρίθμησης

Μετατροπή δεκαδικού αριθμού

- ▶ εάν ο αριθμός περιλαμβάνει **σημείο βάσης**, είναι απαραίτητο να διαχωρίσουμε τον αριθμό σε ένα **ακέραιο** και σε ένα **κλασματικό** μέρος
- ▶ κάθε μέρος θα υποστεί **διαφορετική** διαδικασία μετατροπής
- ❖ μετατροπή ενός **δεκαδικού ακεραίου** σε σύστημα αρίθμησης με **βάση r** :
 - ▶ διαιρούμε τον **αριθμό** και όλα τα επακόλουθα πηλικά με το **r**
 - ▶ συγκεντρώνουμε τα **υπόλοιπα**
- ❖ μετατροπή ενός **δεκαδικού κλάσματος** σε σύστημα αρίθμησης με **βάση r** :
 - ▶ πολλαπλασιάζουμε το **κλάσμα** και όλα τα επακόλουθα γινόμενα με το **r**
 - ▶ συγκεντρώνουμε τα **ακέραια μέρη**

Συστήματα αρίθμησης

Μετατροπή δεκαδικού (ακεραίου) σε δυαδικό

- ▶ π.χ. μετατροπή του $(41)_{10}$ σε δυαδικό αριθμό

	/	Πηλίκος ακέραιας διαίρεσης	Υπόλοιπο ακέραιας διαίρεσης
41	2	20	1
20	2	10	0
10	2	5	0
5	2	2	1
2	2	1	0
1	2	0	1

προκύπτει ότι $(41)_{10} = (101001)_2$

✍ γράφουμε τα ψηφία του
δυαδικού αριθμού από το
τελευταίο προς το πρώτο

Συστήματα αρίθμησης

Μετατροπή δεκαδικού (πραγματικού) σε δυαδικό

► π.χ. μετατροπή του $(41.6875)_{10}$ σε δυαδικό αριθμό

1. αρχικά, μετατρέπουμε τον ακέραιο $(41)_{10}$ σε $(101001)_2$
(ακολουθώντας τη διαδικασία της προηγούμενης διαφάνειας)
2. έπειτα, μετατρέπουμε το κλάσμα $(0.6875)_{10}$ σε δυαδικό αριθμό

	*	Ακέραιο μέρος		Κλασματικό μέρος
0.6875	2	1	+	0.3750
0.3750	2	0	+	0.7500
0.7500	2	1	+	0.5000
0.5000	2	1	+	0.0000

προκύπτει ότι $(0.6875)_{10} = (0.1011)_2$

✍ γράφουμε τα ψηφία του δυαδικού αριθμού από το πρώτο προς το τελευταίο

3. συνενώνουμε τους δύο δυαδικούς αριθμούς και τελικά προκύπτει ότι:

$$(41.6875)_{10} = (101001.1011)_2$$

Συστήματα αρίθμησης

Μετατροπή δεκαδικού (ακεραίου) σε οκταδικό

- ▶ π.χ. μετατροπή του $(153)_{10}$ σε οκταδικό αριθμό

	/	Πηλίκος ακέραιας διαίρεσης	Υπόλοιπο ακέραιας διαίρεσης
153	8	19	1
19	8	2	3
2	8	0	2

προκύπτει ότι $(153)_{10} = (231)_8$

✍ γράφουμε τα ψηφία του
οκταδικού αριθμού από το
τελευταίο προς το πρώτο

Συστήματα αρίθμησης

Μετατροπή δεκαδικού (πραγματικού) σε οκταδικό

► π.χ. μετατροπή του $(153.513)_{10}$ σε οκταδικό αριθμό

1. αρχικά, μετατρέπουμε τον ακέραιο $(153)_{10}$ σε $(231)_8$
(ακολουθώντας τη διαδικασία της προηγούμενης διαφάνειας)
2. έπειτα, μετατρέπουμε το κλάσμα $(0.513)_{10}$ σε οκταδικό αριθμό

	*	Ακέραιο μέρος		Κλασματικό μέρος
0.513	8	4	+	0.104
0.104	8	0	+	0.832
0.832	8	6	+	0.656
0.656	8	5	+	0.248
0.248	8	...	+	...

(με ακρίβεια 4 ψηφίων)
προκύπτει ότι $(0.513)_{10} = (0.4065)_8$

✍ γράφουμε τα ψηφία του οκταδικού αριθμού από το πρώτο προς το τελευταίο

3. συνενώνουμε τους δύο δυαδικούς αριθμούς και τελικά προκύπτει ότι:

$$(153.513)_{10} = (231.4065)_8$$

με ακρίβεια 4 ψηφίων,
μετά το σημείο βάσης

Συστήματα αρίθμησης

Μετατροπή μεταξύ δυαδικών και οκταδικών

❖ καθώς $2^3 = 8 \rightarrow$ κάθε οκταδικό ψηφίο αντιστοιχεί σε τρία δυαδικά ψηφία

- ▶ για να μετατρέψουμε έναν αριθμό από το οκταδικό στο δυαδικό σύστημα, αντιστοιχούμε κάθε ψηφίο του οκταδικού αριθμού με τον αντίστοιχο δυαδικό αριθμό τριών ψηφίων

π.χ.

$$(673.124)_8 = (\overset{6}{110} \overset{7}{111} \overset{3}{011} . \overset{1}{001} \overset{2}{010} \overset{4}{100})_2$$

- ▶ για να μετατρέψουμε έναν αριθμό από το δυαδικό στο οκταδικό σύστημα

1. χωρίζουμε το δυαδικό αριθμό σε ομάδες των τριών bit, ξεκινώντας από το δυαδικό σημείο (υποδιαστολή), προχωρώντας προς τα αριστερά στο ακέραιο μέρος και προς τα δεξιά στο κλασματικό
2. αντικαθιστούμε κάθε ομάδα τριών bit με το αντίστοιχο οκταδικό ψηφίο

π.χ.

$$\begin{array}{ccccccc} & 2 & 6 & 1 & 5 & 3 & . & 7 & 4 & 0 & 6 \\ \hline (10 & 110 & 001 & 101 & 011 & . & 111 & 100 & 000 & 110) & _2 = (26153.7406)_8 \end{array}$$

Συστήματα αρίθμησης

Μετατροπή μεταξύ δυαδικών και δεκαεξαδικών

❖ καθώς $2^4 = 16 \rightarrow$ κάθε δεκαεξαδικό ψηφίο αντιστοιχεί σε τέσσερα δυαδικά ψηφία

- ▶ για να μετατρέψουμε έναν αριθμό από το δεκαεξαδικό στο δυαδικό σύστημα, αντιστοιχούμε κάθε ψηφίο του δεκαεξαδικού αριθμού με τον αντίστοιχο δυαδικό αριθμό τεσσάρων ψηφίων

π.χ.

$$(306.D)_{16} = (\overset{3}{0011} \overset{0}{0000} \overset{6}{0110} . \overset{D}{1101})_2$$

- ▶ για να μετατρέψουμε έναν αριθμό από το δυαδικό στο δεκαεξαδικό σύστημα

1. χωρίζουμε το δυαδικό αριθμό σε ομάδες των τεσσάρων bit, ξεκινώντας από το δυαδικό σημείο (υποδιαστολή), προχωρώντας προς τα αριστερά στο ακέραιο μέρος και προς τα δεξιά στο κλασματικό
2. αντικαθιστούμε κάθε ομάδα τεσσάρων bit με το αντίστοιχο δεκαεξαδικό ψηφίο

π.χ.

$$(\overset{2}{10} \overset{C}{1100} \overset{6}{0110} \overset{B}{1011} . \overset{F}{1111} \overset{7}{0111})_2 = (2C6B.F7)_{16}$$

Συμπληρώματα

Συμπλήρωμα

- ❖ τα συμπληρώματα χρησιμοποιούνται στους ψηφιακούς υπολογιστές
 - ▶ για την **απλοποίηση** της πράξης της **αφαίρεσης** → απλούστερα και πιο οικονομικά κυκλώματα
 - ▶ για τις **λογικές** πράξεις
 - ▶ θα μάθουμε σύντομα τι είναι οι λογικές πράξεις ☺
- ❖ υπάρχουν **δύο τύποι** συμπληρωμάτων για κάθε σύστημα **βάσης r**
 1. συμπλήρωμα ως προς τη **βάση**
 - ▶ συμπλήρωμα ως προς **r**
 2. συμπλήρωμα ως προς την **ελαττωμένη βάση**
 - ▶ συμπλήρωμα ως προς **$r-1$**

Συμπλήρωμα

ως προς ελαττωμένη βάση

έστω, ο αριθμός N βάσης r με n ψηφία

❖ το συμπλήρωμα του N ως προς $r-1$ ορίζεται ως: $(r^n - 1) - N$

▶ εάν N είναι δεκαδικός αριθμός, δηλαδή $r = 10$

▶ το r^n είναι ένας αριθμός που αποτελείται από μία μονάδα και n μηδενικά (π.χ. $10^4 = 10000$)

▶ το $r^n - 1$ είναι ένας αριθμός που αποτελείται n το πλήθος ψηφία 9 (π.χ. $10^4 - 1 = 9999$)

☞ άρα, το συμπλήρωμα ως προς 9 ενός δεκαδικού αριθμού προκύπτει εάν αφαιρέσουμε κάθε ψηφίο του από το 9, π.χ.

▶ (συμπλήρωμα ως προς 9 του 546700) $\rightarrow 999999 - 546700 = 453299$

▶ (συμπλήρωμα ως προς 9 του 012398) $\rightarrow 999999 - 012398 = 987601$

Συμπλήρωμα

ως προς ελαττωμένη βάση II

έστω, ο αριθμός N βάσης r με n ψηφία

❖ το συμπλήρωμα του N ως προς $r-1$ ορίζεται ως: $(r^n - 1) - N$

▶ εάν N είναι δυναδικός αριθμός, δηλαδή $r = 2$

▶ το r^n είναι ένας αριθμός που αποτελείται από μία μονάδα και n μηδενικά (π.χ. $2^4 = 10000$)

▶ το $r^n - 1$ είναι ένας αριθμός που αποτελείται n το πλήθος ψηφία 1 (π.χ. $2^4 - 1 = 1111$)

☞ άρα, το συμπλήρωμα ως προς 1 ενός δυναδικού αριθμού προκύπτει εάν αφαιρέσουμε κάθε ψηφίο του από το 2 (ή ισοδύναμα εάν απλά αλλάξουμε τα 1 σε 0 και τα 0 σε 1) π.χ.

▶ (συμπλήρωμα ως προς 1 του 1011000) $\rightarrow 0100111$

▶ (συμπλήρωμα ως προς 1 του 0101101) $\rightarrow 1010010$

▶ εάν N είναι οκταδικός αριθμός, δηλαδή $r = 8$, το συμπλήρωμα ως προς 7 του N προκύπτει εάν αφαιρέσουμε κάθε ψηφίο του από το 7

▶ εάν N είναι δεκαεξαδικός αριθμός, δηλαδή $r = 16$, το συμπλήρωμα ως προς 15 του N προκύπτει εάν αφαιρέσουμε κάθε ψηφίο του από το F (δηλαδή το δεκαδικό 15)

Συμπλήρωμα ως προς βάση

έστω, ο αριθμός N βάσης r με n ψηφία

❖ το συμπλήρωμα του N ως προς r ορίζεται ως: $r^n - N$

► επομένως, το συμπλήρωμα του N ως προς r προκύπτει εάν προσθέσουμε 1 στο συμπλήρωμα του N ως προς $r-1$ π.χ.

1. (συμπλήρωμα ως προς 10 του δεκαδικού 546700) $\rightarrow 999999 - 546700 = 453300$

► καθώς (συμπλήρωμα ως προς 9 του δεκαδικού 546700) $\rightarrow 999999 - 546700 = 453299$

2. (συμπλήρωμα ως προς 10 του δεκαδικού 012398) $\rightarrow 999999 - 012398 = 987602$

► (συμπλήρωμα ως προς 9 του δεκαδικού 012398) $\rightarrow 999999 - 012398 = 987601$

1. (συμπλήρωμα ως προς 2 του δυαδικού 1011000) $\rightarrow 0101000$

► καθώς (συμπλήρωμα ως προς 1 του δυαδικού 1011000) $\rightarrow 0100111$

2. (συμπλήρωμα ως προς 2 του δυαδικού 0101101) $\rightarrow 1010011$

► καθώς (συμπλήρωμα ως προς 1 του δυαδικού 0101101) $\rightarrow 1010010$

Αφαίρεση

με χρήση συμπληρώματος ως προς βάση

έστω δύο (μη προσημασμένοι) αριθμοί M και N σε σύστημα με βάση r , με n ψηφία

❖ η διαφορά $M-N$ υπολογίζεται ως εξής:

1. προσθέτουμε το μειωτέο M στο συμπλήρωμα ως προς r του αφαιρετέου N

▶ δηλαδή υπολογίζουμε το $M + (r^n - N) = M - N + r^n$

2. εάν $M \geq N$,

▶ το άθροισμα θα έχει ένα τελικό κρατούμενο, το οποίο μπορεί να παραληφθεί

▶ ο αριθμός που απομένει είναι το ζητούμενο αποτέλεσμα ($M-N$)

3. εάν $M < N$,

▶ το άθροισμα έχει μηδενικό τελικό κρατούμενο και ισούται με $r^n - (N-M)$, το οποίο είναι το συμπλήρωμα ως προς r του ($N-M$)

▶ για να πάρουμε το αποτέλεσμα σε πιο οικεία μορφή,

i. υπολογίζουμε το συμπλήρωμα ως προς r του αθροίσματος

ii. και τοποθετούμε στην αρχή το αρνητικό πρόσημο (-)

Αφαίρεση με χρήση συμπληρώματος ως προς βάση

Παραδείγματα

- ❖ χρησιμοποιώντας το συμπλήρωμα ως προς 10, κάνετε την αφαίρεση: $72532 - 3250$

72532	(M)
+ 96750	(συμπλήρωμα ως προς 10 του 03250)
169282	(άθροισμα)
-100000	(ισχύει ότι $M \geq N \rightarrow$ παραλείπουμε το τελικό κρατούμενο: 10^5)
69282	(αποτέλεσμα)

επειδή οι αριθμοί M και N πρέπει να έχουν το ίδιο πλήθος ψηφίων, προσθέσαμε ένα 0 στην αρχή του N

- ❖ χρησιμοποιώντας το συμπλήρωμα ως προς 10, κάνετε την αφαίρεση: $3250 - 72532$

03250	(M)
+ 27468	(συμπλήρωμα ως προς 10 του 72532)
30718	(άθροισμα) (ισχύει ότι $M < N \rightarrow$ δεν προκύπτει κρατούμενο \rightarrow αποτέλεσμα)
-69282	(αποτέλεσμα σε οικεία μορφή: -(συμπλήρωμα ως προς 10 του 30718))

Αφαίρεση με χρήση συμπληρώματος ως προς βάση

Παραδείγματα II

- ❖ χρησιμοποιώντας το συμπλήρωμα ως προς 2, κάνετε την αφαίρεση: $1010100 - 1000011$

1010100	(M)
$+ 0111101$	(συμπλήρωμα ως προς 2 του 1000011)
<hr/>	
10010001	(άθροισμα)
-10000000	(ισχύει ότι $M \geq N \rightarrow$ παραλείπουμε το τελικό κρατούμενο: 2^7)
<hr/>	
0010001	(αποτέλεσμα)

- ❖ χρησιμοποιώντας το συμπλήρωμα ως προς 2, κάνετε την αφαίρεση: $1000011 - 1010100$

1000011	(M)
$+ 0101100$	(συμπλήρωμα ως προς 2 του 1010100)
<hr/>	
1101111	(άθροισμα) (ισχύει ότι $M < N \rightarrow$ δεν προκύπτει κρατούμενο \rightarrow αποτέλεσμα)
-0010001	(αποτέλεσμα σε οικεία μορφή: $-(\text{συμπλήρωμα ως προς 2 του } 1101111))$

Αφαίρεση

με χρήση συμπληρώματος ως προς ελαττωμένη βάση

έστω δύο (μη προσημασμένοι) αριθμοί M και N σε σύστημα με βάση r , με n ψηφία

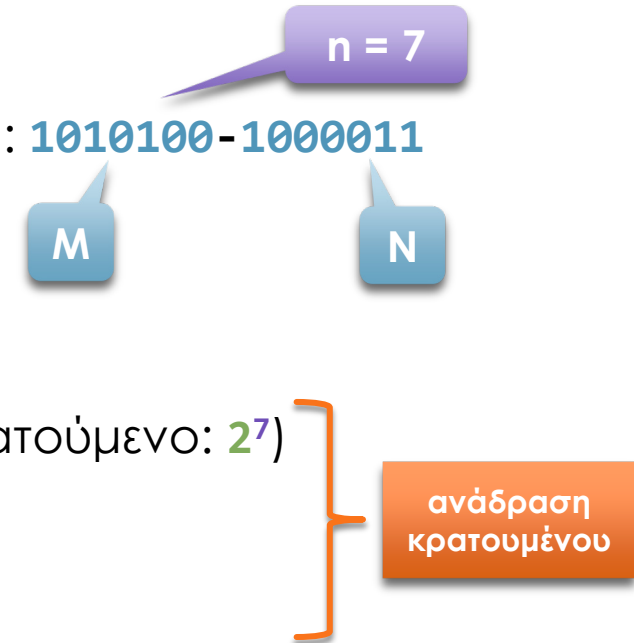
- ❖ η διαφορά $M - N$ μπορεί να υπολογιστεί χρησιμοποιώντας το συμπλήρωμα ως προς $r - 1$ του αφαιρετέου N
 - ▶ η διαδικασία είναι παρόμοια με τη χρήση του συμπληρώματος ως προς r του αφαιρετέου N
 - ▶ με τη διαφορά ότι συμπεριλαμβάνει την ανάδραση κρατουμένου (end-around carry):
 - ✍ (το συμπλήρωμα ως προς $r - 1$ είναι κατά 1 μικρότερο από το συμπλήρωμα ως προς r)
 - ▶ εάν παραχθεί άθροισμα με τελικό κρατούμενο (δηλαδή όταν ισχύει $M \geq N$) \rightarrow το άθροισμα είναι κατά ένα μικρότερο από το τελικό αποτέλεσμα
 - ✍ οπότε, μετά την αφαίρεση του τελικού κρατουμένου, είναι απαραίτητη η πρόσθεση του 1 στο άθροισμα
- ✍ ανάδραση κρατουμένου: η αφαίρεση κρατουμένου και η πρόσθεση του 1 στο άθροισμα

Αφαίρεση με χρήση συμπληρώματος ως προς ελαττωμένη βάση

Παραδείγματα

- ❖ χρησιμοποιώντας το συμπλήρωμα ως προς **1**, κάνετε την αφαίρεση: **1010100** - **1000011**

1010100	(M)
+ 0111100	(συμπλήρωμα ως προς 1 του 1000011)
<hr/>	
10010000	(άθροισμα)
- 10000000	(ισχύει ότι M ≥ N → παραλείπουμε το τελικό κρατούμενο: 2⁷)
<hr/>	
0010000	
+ 0000001	(ισχύει ότι M ≥ N → προσθέτουμε το 1)
<hr/>	
0010001	(αποτέλεσμα)



- ❖ χρησιμοποιώντας το συμπλήρωμα ως προς **1**, κάνετε την αφαίρεση: **1000011** - **1010100**
- | | |
|------------------|---|
| 1000011 | (M) |
| + 0101011 | (συμπλήρωμα ως προς 1 του 1010100) |
| <hr/> | |
| 1101110 | (άθροισμα) (ισχύει ότι M < N → δεν προκύπτει κρατούμενο → αποτέλεσμα) |
| <hr/> | |
| - 0010001 | (αποτέλεσμα σε οικεία μορφή: - (συμπλήρωμα ως προς 1 του 1101110)) |

Αναπαράσταση ακεραίων αριθμών

δυαδική αναπαράσταση

Αναπαράσταση ακεραίων αριθμών

- ❖ όλοι οι φυσικοί αριθμοί με πρόσημο και το μηδέν
 - ▶ εύρος: $-\infty \dots, -2, -1, 0, 1, 2, \dots +\infty$
- ✗ δεν υπάρχει υπολογιστικό σύστημα που να αναπαριστά όλους τους ακεραίους
 - ▶ κάθε υπολογιστικό σύστημα χρησιμοποιεί συγκεκριμένο και περιορισμένο πλήθος από bits για την αναπαράσταση των αριθμών
 - ▶ το πλήθος αυτό καθορίζει και το εύρος των τιμών των ακεραίων
- ❖ υπάρχουν διαφορετικές αναπαραστάσεις για:
 - ▶ ακεραίους χωρίς πρόσημο (μη προσημασμένοι: από 0 έως ∞)
 - ▶ ακεραίους με πρόσημο (προσημασμένοι: από $-\infty$ έως $+\infty$)

Αναπαράσταση ακεραίων αριθμών

Μη προσημασμένοι

- ❖ ο αριθμός (**N**) των bits που χρησιμοποιείται για την αναπαράσταση καθορίζει και τον μεγαλύτερο αριθμό που μπορεί να αναπαρασταθεί

- ▶ **4** bits:

- ▶ ο μεγαλύτερος ακέραιος είναι $1111_2 = 15_{10} = 2^4 - 1$

- ▶ εύρος: από **0** έως **15**

αριθμός με
βάση 2

αριθμός με
βάση 10

- ▶ **8** bits:

- ▶ ο μεγαλύτερος ακέραιος είναι $11111111_2 = 255_{10} = 2^8 - 1$

- ▶ εύρος: από **0** έως **255**

- ▶ γενικά, για **N** bits το εύρος των τιμών που έχουμε για ακέραιους χωρίς πρόσημο είναι:
από **0** έως **$2^N - 1$**

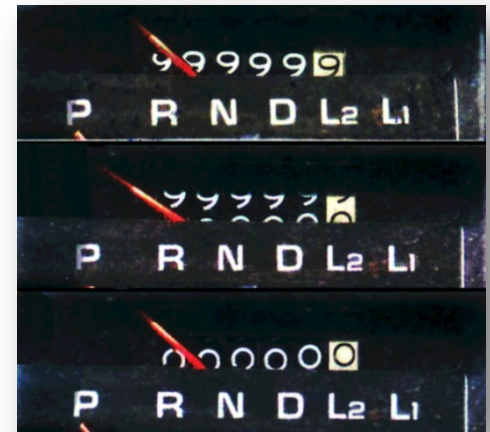
👉 για να αποθηκευτεί σε υπολογιστή ένας μη προσημασμένος ακέραιος αριθμός απλά **μετατρέπεται στο δυαδικό σύστημα**

Αναπαράσταση ακεραίων αριθμών

Μη προσημασμένοι - Παραδείγματα

έστω **N** ο αριθμός των bits που χρησιμοποιείται για την αναπαράσταση

- ✓ αν **N** = 8 → ο αριθμός 12_{10} (1100_2) αποθηκεύεται ως: **00001100**
- ✓ αν **N** = 16 → ο αριθμός 12_{10} (1100_2) αποθηκεύεται ως: **0000000000001100**
- ✗ αν **N** = 8 → ο αριθμός 260_{10} (100000100_2) **δε** μπορεί να αποθηκευτεί
 - ? αν προσπαθήσουμε να τον αποθηκεύσουμε σε 8 bits;
 - 👉 θα αποθηκευτεί **0000100**, δηλαδή ο αριθμός 4_{10} (100_2)
 - ✍ έχουμε το φαινόμενο της **υπερχείλισης** (overflow) → οδηγεί σε **απώλεια** πληροφορίας
- ✓ αν **N** = 16 → ο αριθμός 260_{10} (100000100_2) αποθηκεύεται ως: **0000000100000100**
- ✓ αν **N** = 32 → ο αριθμός 260_{10} (100000100_2) αποθηκεύεται ως: **000000000000000000000000100000100**



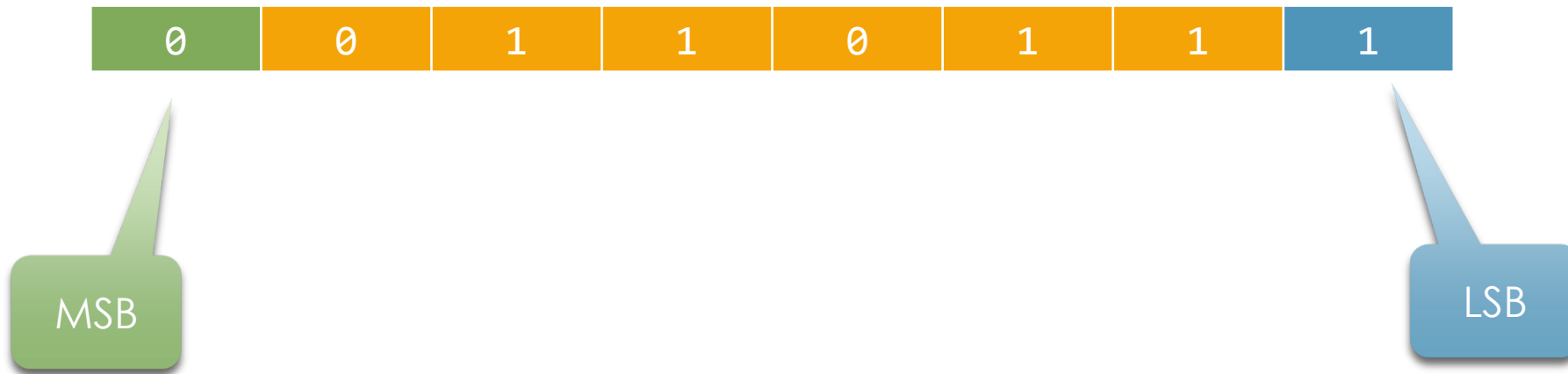
Αναπαράσταση ακεραίων αριθμών

Προσημασμένοι

- ❖ αφιερώνεται **ένα** bit για την ένδειξη του **πρόσημου** του ακεραίου:
 - ▶ **0** → θετικός
 - ▶ **1** → αρνητικός
- ❖ τα **υπόλοιπα** bits χρησιμοποιούνται για την αναπαράσταση του **μέτρου** του ακεραίου (απόλυτη τιμή)
 - ✍ με **ένα** bit **λιγότερο** (αυτό του πρόσημου) το εύρος των αριθμών **υποδιπλασιάζεται**
- ❖ στις **περιπτώσεις** που θα εξετάσουμε στη συνέχεια αλλάξει **μόνο** ο τρόπος αναπαράστασης των **αρνητικών** αριθμών
(η αναπαράσταση των **θετικών** ακεραίων είναι **ίδια**)
 1. προσημασμένου μεγέθους (signed-magnitude)
 2. προσημασμένου συμπληρώματος (signed-complement)
 - a. συμπλήρωμα ως προς 1
 - b. συμπλήρωμα ως προς 2

Δυαδική Αναπαράσταση

- ❖ δυαδική αναπαράσταση (με κατεύθυνση από αριστερά προς τα δεξιά)
 - ▶ το **πρώτο** ψηφίο είναι το **περισσότερο σημαντικό ψηφίο** (Most Significant Bit – MSB)
 - ▶ το **τελευταίο** ψηφίο είναι το **λιγότερο σημαντικό ψηφίο** (Least Significant Bit - LSB)



Αναπαράσταση ακεραίων αριθμών

Προσημασμένοι - Σύμβαση: προσημασμένου μεγέθους

❖ το πιο σημαντικό bit (**MSB**) χρησιμοποιείται για την ένδειξη του πρόσημου

❖ τα εναπομείναντα ψηφία χρησιμοποιούνται για το μέτρο του αριθμού:

• $+12_{10} \rightarrow 00001100$ (8 bits αναπαράσταση)

• $-12_{10} \rightarrow 10001100$ (8 bits αναπαράσταση)

• $+260_{10} \rightarrow 0000000100000100$ (16 bits αναπαράσταση)

• $-260_{10} \rightarrow 1000000100000100$ (16 bits αναπαράσταση)

❖ έστω ότι χρησιμοποιούμε **N** bits

▶ ο μέγιστος προσημασμένος ακέραιος είναι:

$$011111\dots1111_2 = +(2^{N-1} - 1)_{10}$$

▶ ο ελάχιστος προσημασμένος ακέραιος είναι:

$$111111\dots1111_2 = -(2^{N-1} - 1)_{10}$$

Αναπαράσταση ακεραίων αριθμών

Προσημασμένοι - Σύμβαση: προσημασμένου μεγέθους (II)

❖ έστω ότι χρησιμοποιούμε **N** bits,

▶ το **εύρος τιμών** που έχουμε είναι: από $-(2^{(N-1)}-1)_{10}$ έως $+(2^{(N-1)}-1)_{10}$

παραδείγματα:

▶ με **8** bits το εύρος τιμών είναι: -2^7-1 έως $+2^7-1$ (-127 έως +127)

▶ με **16** bits το εύρος τιμών είναι: $-2^{15}-1$ έως $+2^{15}-1$ (-32767 έως +32767)

✍ έχουμε **δύο** αναπαραστάσεις για το **μηδέν**:

▶ **000...000** ($+0_{10}$) και

▶ **100...000** (-0_{10})

Αναπαράσταση ακεραίων αριθμών

Προσημασμένοι - Σύμβαση: προσημασμένου συμπληρώματος ως προς 1

- ❖ (πάλι) το πιο **σημαντικό** bit (**MSB**) χρησιμοποιείται για την **ένδειξη** του **πρόσημου**
- ❖ για την αναπαράσταση των **αρνητικών** ακεραίων χρησιμοποιούμε **το συμπλήρωμα ως προς 1** της δυαδικής αναπαράστασης του αριθμού
 - ✍ το **συμπλήρωμα ως προς 1** ενός δυαδικού αριθμού βρίσκεται εύκολα αν αλλάξουμε
 - ▶ όλα τα ψηφία 1 σε 0 και
 - ▶ όλα τα ψηφία 0 σε 1

παραδείγματα:

- $+12_{10}$ (1100_2) \rightarrow **00001100** (8 bits αναπαράσταση)
 - ▶ **όμοια** με την αναπαράσταση προσημασμένου μεγέθους
- -12_{10} θα παρασταθεί ως: **11110011**
 - ▶ καθώς το **συμπλήρωμα ως προς 1** του 0001100_2 είναι το 1110011_2

Αναπαράσταση ακεραίων αριθμών

Προσημασμένοι - Σύμβαση: προσημασμένου συμπληρώματος ως προς 1 (II)

❖ έστω ότι χρησιμοποιούμε **N** bits

- ▶ μέγιστος προσημασμένος ακέραιος: **01111...1111** = $+(2^{(N-1)} - 1)_{10}$
- ▶ ελάχιστος προσημασμένος ακέραιος: **10000...0000** = $-(2^{(N-1)} - 1)_{10}$
 - ▶ όπου το **0000...0000** είναι συμπλήρωμα ως προς 1 του **1111...1111**
- ▶ το εύρος τιμών που έχουμε είναι: από $-(2^{(N-1)}-1)_{10}$ έως $+(2^{(N-1)}-1)_{10}$

✍ (και πάλι) έχουμε **δύο** αναπαραστάσεις για το **μηδέν**:

- ▶ **000000...000000** ($+0_{10}$) και
- ▶ **11111...11111** (-0_{10})
 - ▶ όπου το **1111...1111** είναι το συμπλήρωμα ως προς 1 του **0000...0000**

Αναπαράσταση ακεραίων αριθμών

Προσημασμένοι - Σύμβαση: προσημασμένου συμπληρώματος ως προς 2

- ❖ (πάλι) το πιο σημαντικό bit (**MSB**) χρησιμοποιείται για την ένδειξη του πρόσημου
- ❖ για την αναπαράσταση των αρνητικών ακεραίων χρησιμοποιούμε το συμπλήρωμα ως προς 2 της δυαδικής αναπαράστασης του αριθμού
 - ▶ το συμπλήρωμα ως προς 2 ενός δυαδικού αριθμού βρίσκεται εύκολα αν
 - αλλάξουμε όλα τα ψηφία 1 σε 0 και όλα τα ψηφία 0 σε 1, και
 - στον αριθμό που προκύψει, προσθέσουμε τον αριθμό 1_2

το συμπλήρωμα ως προς 1 του αριθμού

παραδείγματα:

- $+12_{10}$ (1100_2) \rightarrow **00001100** (8 bits αναπαράσταση)
 - ▶ όμοια με την αναπαράσταση συμπληρώματος ως προς 1
- -12_{10} θα παρασταθεί ως: **1110100**
 - καθώς το συμπλήρωμα ως προς 1 του 0001100_2 είναι το: 1110011_2 , και
 - το συμπλήρωμα ως προς 2 του 0001100_2 είναι το: $1110011_2 + 1_2 = 1110100_2$

Αναπαράσταση ακεραίων αριθμών

Προσημασμένοι - Σύμβαση: προσημασμένου συμπληρώματος ως προς 2 (II)

- ❖ έστω ότι χρησιμοποιούμε **N** bits
 - ▶ μέγιστος προσημασμένος ακέραιος: **01111...1111** = $+(2^{(N-1)} - 1)_{10}$
 - ▶ ελάχιστος προσημασμένος ακέραιος: **10000...0001** = $-(2^{(N-1)} - 1)_{10}$
 - ▶ όπου το **10000...0001** είναι συμπλήρωμα ως προς 2 του **11111...1111**
 - ▶ το εύρος τιμών που έχουμε είναι: από $-(2^{(N-1)})_{10}$ έως $+(2^{(N-1)}-1)_{10}$
- ✍ έχουμε μία αναπαράσταση για το μηδέν: **000000...000000** ($+0_{10}$)

Αναπαράσταση ακεραίων αριθμών

Παράδειγμα με N=4 bits

	προσημασμένου μεγέθους	συμπληρώματος ως προς 1	συμπληρώματος ως προς 2
7	0111	0111	0111
6	0110	0110	0110
5	0101	0101	0101
4	0100	0100	0100
3	0011	0011	0011
2	0010	0010	0010
1	0001	0001	0001
0	0000, 1000	0000, 1111	0000
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	-	-	1000

οι θετικοί αριθμοί:

- είναι ίδιοι (και στις τρεις αναπαραστάσεις)
- έχουν το πιο σημαντικό τους bit 0

οι αρνητικοί αριθμοί:

- έχουν το πιο σημαντικό τους bit 1

Αναπαράσταση ακεραίων αριθμών

Παράδειγμα με N=8 bits

	προσημασμένου μεγέθους	συμπληρώματος ως προς 1	συμπληρώματος ως προς 2
127	01111111	01111111	01111111
126	01111110	01111110	01111110
...
2	00000010	00000010	00000010
1	00000001	00000001	00000001
0	00000000, 10000000	00000000, 11111111	00000000
-1	10000001	11111110	11111111
-2	10000010	11111101	11111110
...
-126	11111110	10000001	10000010
-127	11111111	10000000	10000001
-128	-	-	10000000

Αναπαράσταση ακεραίων αριθμών

Προσημασμένοι - Σύνοψη

- ❖ η σύμβαση προσημασμένου μεγέθους
 - ✓ χρησιμοποιείται στη συμβατική αριθμητική
 - ✗ δεν είναι πολύ πρακτικό στην αριθμητική υπολογιστών
 - ▶ διαφορετική αντιμετώπιση προσήμου και μεγέθους
- ❖ η σύμβαση συμπληρώματος ως προς 1
 - ▶ δημιουργεί κάποιες δυσκολίες και χρησιμοποιείται σπάνια για αριθμητικές πράξεις
 - ✓ χρήσιμο στις λογικές πράξεις (όπως θα συζητήσουμε σύντομα ☺)
- ✍ η σύμβαση συμπληρώματος ως προς 2 αποτελεί τον πιο συνηθισμένο τρόπο αναπαράστασης ακεραίων με πρόσημο

Προσημασμένοι δυαδικοί αριθμοί

Αριθμητική πρόσθεση και αφαίρεση

Προσημασμένοι δυαδικοί αριθμοί

Αριθμητική πρόσθεση

- ❖ για να προσθέσουμε δύο προσημασμένους δυαδικούς αριθμούς (μπορεί να είναι και αρνητικοί) που παριστάνονται στο σύστημα προσημασμένου συμπληρώματος ως προς 2:

1. τους προσθέτουμε, συμπεριλαμβανομένων των bits προσήμου
2. αγνοούμε τυχόν κρατούμενο που παράγεται στη θέση του bit προσήμου

π.χ.

+	6	00000110
+	13	+ 00001101
<hr/>		
+	19	00010011

-	6	11111010
+	13	+ 00001101
<hr/>		
+	7	00000111

+	6	00000110
-	13	+ 11110011
<hr/>		
-	7	11111001

-	6	11111010
-	13	+ 11110011
<hr/>		
-	19	11101101

Προσημασμένοι δυαδικοί αριθμοί

Αριθμητική αφαίρεση

- ❖ η πράξη της αφαίρεσης μπορεί να αλλάξει σε πράξη πρόσθεσης, εάν
 - i. αλλάξουμε το πρόσημο του αφαιρετέου και
 - ii. εκτελέσουμε την πρόσθεση

- ❖ οπότε, για να αφαιρέσουμε δύο προσημασμένους δυαδικούς αριθμούς (μπορεί να είναι και αρνητικοί) που παριστάνονται στο σύστημα προσημασμένου συμπληρώματος ως προς 2
 - i. υπολογίζουμε το συμπλήρωμα ως προς 2 του αφαιρετέου, συμπεριλαμβανομένου του bit προσήμου
 - ii. εκτελούμε την πρόσθεση, δηλαδή:
 - a. προσθέτουμε τον αφαιρετέο στον μειωτέο, συμπεριλαμβανομένων των bits προσήμου
 - b. αγνοούμε τυχόν κρατούμενο που παράγεται στη θέση του bit προσήμου

Προσημασμένοι δυαδικοί αριθμοί

Αριθμητική πρόσθεση και αφαίρεση - Συμπέρασμα

- ❖ οι δυαδικοί αριθμοί στο σύστημα προσημασμένου συμπληρώματος ως προς 2, προστίθενται και αφαιρούνται με τους **ίδιους** βασικούς κανόνες (πρόσθεσης και αφαίρεσης) που ισχύουν για τους μη προσημασμένους αριθμούς
 - ✓ οι υπολογιστές χρειάζονται **ένα και μοναδικό** ηλεκτρονικό κύκλωμα για να εκτελέσουν και τις δύο πράξεις
 - ✍ ο **χρήστης** είναι υπεύθυνος να **ερμηνεύσει** σωστά το αποτέλεσμα της πράξης
 - ▶ ανάλογα με το εάν χρησιμοποιεί προσημασμένους ή μη προσημασμένους αριθμούς

Αναπαράσταση πληροφορίας

στα ψηφιακά συστήματα, με χρήση δυαδικών κωδίκων

Αναπαράσταση πληροφορίας

- ▶ στα ψηφιακά συστήματα χρησιμοποιούνται διακριτές ποσότητες πληροφορίας που παριστάνονται σε δυαδική μορφή
- ▶ οι αριθμοί που χρησιμοποιούνται στους υπολογισμούς παριστάνονται με χρήση του δυαδικού συστήματος αρίθμησης
 - ▶ διακριτά στοιχεία, όπως π.χ. τα δεκαδικά ψηφία και οι χαρακτήρες κειμένου, παριστάνονται με τη χρήση δυαδικών κωδίκων

Αναπαράσταση Πληροφορίας

Κωδικοποίηση

- ❖ όλα τα στοιχεία ενός ψηφιακού συστήματος **κωδικοποιούνται** σε bits
 - ▶ οι εντολές που εκτελούνται
 - ▶ τα δεδομένα (π.χ. κείμενο, αριθμοί, αναλογικό σήμα, εικόνες και video)

επομένως, η **κωδικοποίηση** των δεδομένων:

- ▶ βασίζεται στο δυαδικό σύστημα και χρησιμοποιεί **δυαδικούς αριθμούς** (→ **δυαδική αναπαράσταση**)
- ▶ συνιστά τρόπο **αναπαράστασης** δεδομένων στα ψηφιακά συστήματα (**ψηφιακά δεδομένα**)
- ▶ λαμβάνει υπόψιν της το **είδος** των δεδομένων που πρόκειται να αναπαρασταθούν
- ▶ αφιερώνει **συγκεκριμένο** αριθμό από bits για κάθε στοιχείο των δεδομένων που πρόκειται να αναπαραστήσει

Κωδικοποίηση

1^ο Παράδειγμα

? πόσα bits απαιτούνται για τα σημεία του ορίζοντα;

✎ βορράς, νότος, ανατολή, δύση

▶ απαιτούνται **4 διακριτές** καταστάσεις

❖ με 1 bit

▶ 1^η κατάσταση: 0

▶ 2^η κατάσταση: 1

✗ μπορούν να αναπαραστηθούν **2 καταστάσεις** συνολικά → άρα, το **1 bit δεν** επαρκεί!

❖ με 2 bits

▶ 1^η κατάσταση: 00

▶ 2^η κατάσταση: 01

▶ 3^η κατάσταση: 10

▶ 4^η κατάσταση: 11

✓ μπορούν να αναπαραστηθούν **4 καταστάσεις** συνολικά → άρα, τα **2 bits επαρκούν!**

Σημείο	Κωδικοποίηση (δυναδικός αριθμός)
βορράς	00
νότος	01
ανατολή	10
δύση	11

Κωδικοποίηση

2^ο Παράδειγμα

? πόσα bits απαιτούνται για τα 24 κεφαλαία γράμματα του ελληνικού αλφαβήτου;

► απαιτούνται **24 διακριτές** καταστάσεις

✗ με **1** bit → έχουμε **2** διαφορετικές καταστάσεις – **δεν** επαρκούν!

✗ με **2** bits → έχουμε **4** διαφορετικές καταστάσεις – **δεν** επαρκούν!

✗ με **3** bits → έχουμε **8** διαφορετικές καταστάσεις – **δεν** επαρκούν!

✗ με **4** bits → έχουμε **16** διαφορετικές καταστάσεις – **δεν** επαρκούν!

✓ με **5** bits → έχουμε **32** διαφορετικές καταστάσεις – **επαρκούν**

► μάλιστα, περισσεύουν 8 καταστάσεις

✍ κάθε κωδική ακολουθία **πρέπει** να περιγράφει μόνο ένα γράμμα

✍ η επιλογή της κωδικής ακολουθίας (από bits) κάθε γράμματος **δεν είναι μοναδική**, π.χ. αντί για 00000, η κωδικοποίηση του 'Α' θα μπορούσε να είναι: 10000 ή 01000 ή ...

Γράμμα	Κωδικοποίηση (δυναδικός αριθμός)
Α	00000
Β	00001
Γ	00010
Δ	00011
...	...
Ψ	10110
Ω	10111

Κωδικοποίηση

? πόσα bits απαιτούνται;

✍ με k bits μπορούμε ν' αναπαραστήσουμε $N=2^k$ διαφορετικές καταστάσεις, όπου $k=\log_2 N$

- ❖ υπάρχουν πολλές **διαφορετικές** κωδικοποιήσεις για τα ίδια στοιχεία, αρκεί σε κάθε κωδικοποίηση:
 - ▶ μία κωδική ακολουθία να περιγράφει **ένα (και μόνο ένα)** στοιχείο των δεδομένων
 - ▶ (ισοδύναμα) κάθε στοιχείο να έχει **μία και μόνο μία** κωδική ακολουθία
 - ▶ αυτό δεν ισχύει πάντα
- ❖ για κάθε τύπο δεδομένων έχουν προταθεί **πρότυπα** κωδικοποίησης
 - ▶ για **χαρακτήρες**: **ASCII**, **ISO** και **UNICODE**
 - ▶ για δεκαδικά ψηφία: **BCD**

Αναπαράσταση Κειμένου

- ❖ αφορά:
 - ▶ **χαρακτήρες** που μπορούν να **εκτυπωθούν**:
A-Z, a-z, 0-9, !, @, #, \$, %, ^, &, *, (,), _, -, +, /
 - ▶ **χαρακτήρες ελέγχου** (που δεν εκτυπώνονται):
αλλαγή γραμμής, <enter>, <backspace>
- ❖ σε κάθε **κωδικοποίηση** χρησιμοποιείται ένας **συγκεκριμένος αριθμός** από bits
 - ▶ ανάλογα με τον **αριθμό** των χαρακτήρων που πρόκειται να αναπαρασταθούν
- ❖ σε κάθε **χαρακτήρα** αντιστοιχεί **ένας** συγκεκριμένος συνδυασμός από bits
 - ▶ δηλαδή ένας **δυαδικός αριθμός**
- ❖ υπάρχουν πολλές αναπαραστάσεις κειμένου
 - ▶ οι πιο **γνωστές**: **ASCII**, **ISO** & **UNICODE**

Αναπαράσταση Κειμένου

Κωδικοποίηση κατά ASCII

American standard Code for Information Interchange, ASCII

- ❖ το πρώτο **κοινό** πρότυπο αναπαράστασης χαρακτήρων κειμένου που χρησιμοποιήθηκε από **όλους** τους κατασκευαστές υπολογιστών
 - ▶ χρησιμοποιήθηκε για πρώτη φορά το 1963 και ευρέως από το 1968 και μετά
- ❖ έχει **επικρατήσει** σαν πρότυπο κωδικοποίησης αρχείων κειμένου
- ❖ χρησιμοποιεί **7** bits για την **κωδικοποίηση** και ένα **όγδοο** bit για λόγους **έλεγχου** στη μετάδοση των bits
 - ▶ άρα, υπάρχουν **128** (2^7) διαφορετικοί συνδυασμοί των 7 bits
 - ✍ οι **υπόλοιποι 128** ($2^8 = 256$ συνολικά) συνδυασμοί χρησιμοποιήθηκαν **αργότερα** για **ειδικούς** γραφικούς χαρακτήρες ή για χαρακτήρες **άλλων αλφάβητων**
 - ▶ από τον **ISO** (International Standardization Organization - Διεθνής Οργανισμός Τυποποίησης)

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Αναπαράσταση Κειμένου

Κωδικοποίηση κατά ISO και UNICODE

- ❖ η κωδικοποίηση ASCII
 - ▶ αναπαριστά ικανοποιητικά τους λατινικούς χαρακτήρες της αγγλικής γλώσσας
 - ▶ αλλά δεν συμπεριλαμβάνει εθνικούς χαρακτήρες άλλων χωρών
- ❖ το ISO 8859 είναι μια 8-bit επέκταση του ASCII που χρησιμοποιεί και τα 8 bits
 - ▶ για τους ελληνικούς χαρακτήρες (μονοτονικό σύστημα) υπάρχει το ISO 8859-7
- ❖ η κωδικοποίηση UNICODE χρησιμοποιεί 8, 16 ή και 32 bits για την αναπαράσταση χαρακτήρων (2^8 , 2^{16} και 2^{32} διαφορετικούς χαρακτήρες, αντίστοιχα)
 - ▶ το πρότυπο UTF-8 είναι η κυρίαρχη κωδικοποίηση στον Παγκόσμιο Ιστό (Word Wide Web, WWW)

Αναπαράσταση δεκαδικών ψηφίων

- ❖ γνωρίζουμε ότι:
 - ▶ οι υπολογιστές χρησιμοποιούν το δυαδικό σύστημα αρίθμησης
 - ▶ οι άνθρωποι είναι εξοικιωμένοι με το δεκαδικό σύστημα αρίθμησης
- ❖ ένας τρόπος αντιμετώπισης:
 1. μετατροπή των δεκαδικών αριθμών σε δυαδικούς
 2. εκτέλεση των αριθμητικών πράξεων με δυαδικό τρόπο
 3. μετατροπή αποτελέσματος στο δεκαδικό σύστημα
- ❖ απαραίτητη η αναπαράσταση των δεκαδικών ψηφίων με χρήση ενός δυαδικού κώδικα → κωδικοποίηση

π.χ.

- ▶ κωδικοποίηση ψηφίων με χρήση 10 bits, όπου κάθε ψηφίο αντιστοιχεί σε ένα συνδυασμό από εννέα 0 και ένα 1
 - ▶ κώδικας BCD
- ✍ απαιτούνται τουλάχιστον τέσσερα bits για την αναπαράσταση των δεκαδικών ψηφίων (γιατί;)

μία πιθανή κωδικοποίηση

δεκαδικό ψηφίο	κωδικοποιημένο ψηφίο
0	0000000001
1	0000000010
2	0000000100
3	0000001000
4	0000010000
5	0000100000
6	0001000000
7	0010000000
8	0100000000
9	1000000000

ακόμη μία πιθανή κωδικοποίηση

δεκαδικό ψηφίο	κωδικοποιημένο ψηφίο
0	1000000000
1	0100000010
2	0010000100
3	0001001000
4	0000100000
5	0000010000
6	0000001000
7	0000000100
8	0000000010
9	0000000001

Αναπαράσταση δεκαδικών ψηφίων

Κώδικας BCD

- ❖ ένας δυαδικός κώδικας αναπαράστασης δεκαδικών ψηφίων, όπου κάθε δεκαδικό ψηφίο **κωδικοποιείται** με χρήση **τεσσάρων** δυαδικών ψηφίων

❓ πόσες κωδικές λέξεις δε χρησιμοποιούνται; → 6

☞ 1010, 1011, 1100, 1101, 1110, 1111

- ❖ οι **αριθμοί** που κωδικοποιούνται με αυτό τον τρόπο ονομάζονται **δυναδικά κωδικοποιημένοι δεκαδικοί** (binary-coded decimal)

αριθμός σε δυαδική αναπαράσταση	δεκαδικός αριθμός	αριθμός σε κωδικοποίηση BCD
00000010	2	0010
00000101	5	0101
00001001	9	1001
00001010	10	0001 0000
10111001	185	0001 1000 0101
0001001001110110	4726	0100 0111 0010 0110
1100010100111100	50492	0101 0000 0100 1001 0010

Κώδικας BCD

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

σημαντικό:

για έναν **δεκαδικό αριθμό**, **μη συγχέετε:**

- τη **μετατροπή** του σε **δυναδικό**
- με την **κωδικοποίησή** του, χρησιμοποιώντας έναν **δυναδικό κώδικα**

Αναπαράσταση δεκαδικών ψηφίων

Κώδικας BCD (II)

- ❖ οι **αριθμοί** που κωδικοποιούνται με τον κώδικα BCD αυτό τον τρόπο ονομάζονται **δυναδικά κωδικοποιημένοι δεκαδικοί** (binary-coded decimal)
 - ▶ είναι **δεκαδικοί** και όχι **δυναδικοί** αριθμοί
- ❖ διαφορά **δεκαδικών αριθμών** και **αριθμών BCD**
 - ▶ για τους **δεκαδικούς** χρησιμοποιούνται τα σύμβολα: **0, 1, 2, ..., 9**
 - ▶ για τους **αριθμούς BCD** χρησιμοποιούνται τα στοιχεία του (δυναδικού) κώδικα BCD: **0000, 0001, 0010, ..., 1001**

Αναπαράσταση δεκαδικών ψηφίων

Κώδικας BCD - Πρόσθεση

- ❖ οι μεγαλύτερες τιμές κατά την πρόσθεση δύο δεκαδικών ψηφίων (**A**, **B**) και ενός αρχικού κρατουμένου (**Z**), μπορεί να είναι:

$$Z + A + B = 1 + 9 + 9$$

✍️ καθώς, κανένα δεκαδικό ψηφίο δεν υπερβαίνει το 9

- ❖ έστω ότι προσθέτουμε τα ψηφία BCD (**A**, **B**, **Z**) σαν να ήταν δυαδικοί αριθμοί → η τιμή του αθροίσματος ανήκει στο διάστημα **[0, 19]**, το οποίο:
 - ▶ στο δυαδικό σύστημα είναι **[0000, 10011]**
 - ▶ στο BCD σύστημα είναι **[0000, 1 1001]**, όπου:
 - ▶ τα 4 λιγότερο σημαντικά bits είναι το BCD άθροισμα
 - ▶ το 5^ο bit (όπου υπάρχει) είναι το κρατούμενο

άθροισμα	δυαδικό σύστημα	BCD σύστημα
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	1 0000
11	1011	1 0001
12	1100	1 0010
13	1101	1 0011
14	1110	1 0100
15	1111	1 0101
16	10000	1 0110
17	10001	1 0111
18	10010	1 1000
19	10011	1 1001

το δυαδικό άθροισμα παριστάνει σωστά το BCD ψηφίο

το δυαδικό άθροισμα δεν παριστάνει σωστά το BCD ψηφίο

Αναπαράσταση δεκαδικών ψηφίων

Κώδικας BCD - Πρόσθεση (II)

- ▶ εάν το **δυναδικό άθροισμα** είναι **μεγαλύτερο ή ίσο** του 1010_2 δεν παριστάνει σωστά το **BCD ψηφίο**
- ▶ για να το διορθώσουμε προσθέτουμε το 110_2 (6_{10}), ώστε
 1. το **δυναδικό άθροισμα** να μετατραπεί στο σωστό **BCD ψηφίο**
 2. να παραχθεί ένα επιπλέον σωστό **BCD κρατούμενο**
- ✍ αυτό συμβαίνει διότι
 - ένα **κρατούμενο** στην πιο σημαντική θέση bit του δυναδικού αθροίσματος και
 - ένα δεκαδικό **κρατούμενο**
 διαφέρουν κατά $16 - 10 = 6$

το δυναδικό άθροισμα παριστάνει σωστά το BCD ψηφίο

το δυναδικό άθροισμα δεν παριστάνει σωστά το BCD ψηφίο

άθροισμα	δυναδικό σύστημα	BCD σύστημα	
0	0000	0000	
1	0001	0001	
2	0010	0010	
3	0011	0011	
4	0100	0100	
5	0101	0101	
6	0110	0110	
7	0111	0111	
8	1000	1000	
9	1001	1001	
10	1010	1 0000	διορθωμένο δυναδικό σύστημα (+110 ₂)
11	1011	1 0001	10000
12	1100	1 0010	10001
13	1101	1 0011	10010
14	1110	1 0100	10011
15	1111	1 0101	10100
16	10000	1 0110	10101
17	10001	1 0111	10110
18	10010	1 1000	10111
19	10011	1 1001	11000
			11001

Αναπαράσταση δεκαδικών ψηφίων

Κώδικας BCD - Πρόσθεση Ψηφίων - Παραδείγματα

4	0100
+ 5	+ 0101
9	1001

(επειδή το δυαδικό άθροισμα είναι μικρότερο του $1010_2 \rightarrow$ είναι σωστό BCD ψηφίο)

4	0100
+ 8	+ 1000
12	1100
	+ 0110
	10010

(επειδή το δυαδικό άθροισμα είναι μεγαλύτερο του $1010_2 \rightarrow$ δεν είναι σωστό BCD ψηφίο)
(προσθέτουμε το 110_2 ώστε να το διορθώσουμε)
(προκύπτει το σωστό BCD ψηφίο και ένα κρατούμενο)

8	1000
+ 9	+ 1001
17	10001
	+ 0110
	10111

(επειδή το δυαδικό άθροισμα είναι μεγαλύτερο του $1010_2 \rightarrow$ δεν είναι σωστό BCD ψηφίο)
(προσθέτουμε το 110_2 ώστε να το διορθώσουμε)
(προκύπτει το σωστό BCD ψηφίο και ένα κρατούμενο)

Αναπαράσταση δεκαδικών ψηφίων

Κώδικας BCD - Πρόσθεση αριθμών - Παράδειγμα

- ❖ η πρόσθεση δύο μη προσημασμένων αριθμών BCD των n ψηφίων γίνεται με την εκτέλεση πρόσθεσης BCD των ψηφίων, με μεταφορά των κρατούμενων BCD που προκύπτουν

		1	1		
184	0001	1000		0100	
+ 576	+ 0101	+ 0111		+ 0110	
<hr/>	<hr/>	<hr/>		<hr/>	
760	0111	10000	(μεγαλύτερο του 1010_2)	1010	(ίσο με 1010_2)
		+ 0110	(προσθέτουμε το 110_2)	+ 0110	(προσθέτουμε το 110_2)
		<hr/>		<hr/>	
		10110		10000	
	<hr/>				
	(0111	0110		0000) _{BCD}

Αναπαράσταση δεκαδικών ψηφίων

Κώδικας BCD - Προσημασμένοι αριθμοί

- ❖ η αναπαράσταση προσημασμένων δεκαδικών αριθμών σε BCD είναι παρόμοια με την αναπαράσταση προσημασμένων αριθμών στο δυαδικό σύστημα
 - ▶ σύστημα προσημασμένου μεγέθους
 - ▶ σύστημα προσημασμένου συμπληρώματος
 - i. ως προς 9
 - ii. ως προς 10
- ❖ οι αριθμητικές πράξεις μεταξύ προσημασμένων δεκαδικών αριθμών σε BCD γίνονται με παρόμοια τρόπο όπως στις αριθμητικές πράξεις μεταξύ προσημασμένων αριθμών στο δυαδικό σύστημα
 - ▶ π.χ. χρησιμοποιώντας το συμπλήρωμα ως προς 10, κάνετε την αφαίρεση: $375_{\text{BCD}} - 240_{\text{BCD}}$

$$\begin{array}{r} 375_{\text{BCD}} \\ + 760_{\text{BCD}} \\ \hline 135_{\text{BCD}} \end{array} \quad \begin{array}{l} \text{(συμπλήρωμα ως προς 10 του } 240_{\text{BCD}}) \\ \text{(διαφορά)} \end{array}$$

Αναπαράσταση δεκαδικών ψηφίων

Άλλοι δεκαδικοί κώδικες

- ❖ ο **BCD** είναι κώδικας με **συντελεστές βαρών**: 8, 4, 2, 1
 - ▶ αντιστοιχούν στις αντίστοιχες δυνάμεις του 2 (δηλαδή 2^3 , 2^2 , 2^1 , 2^0)
 - ▶ π.χ. $(1101)_{\text{BCD}} = 8*1 + 4*1 + 2*0 + 1*1 = (13)_{10}$
- ❖ ο **κώδικας 2421** είναι ένας ακόμη κώδικας με **συντελεστές βαρών**
 - ▶ π.χ. $(1101)_{2421} = 2*1 + 4*1 + 2*0 + 1*1 = (7)_{10}$
 - ▶ κάποια ψηφία μπορούν να κωδικοποιηθούν με δύο δυνατούς τρόπους, π.χ.

$$(0100)_{2421} = (1010)_{2421} = (4)_{10}$$

- ❖ στον κώδικα **συν-3** κάθε κωδικοποιημένο ψηφίο προκύπτει από τη δυαδική του αναπαράσταση **συν 3**
- ❖ ο **κώδικας 8,4,-2,-1** είναι ένας ακόμη κώδικας με **συντελεστές βαρών** (και με **αρνητικούς** συντελεστές)
 - ▶ π.χ. $(1101)_{8,4,-2,-1} = 8*1 + 4*1 + (-2)*0 + (-1)*1 = (11)_{10}$

δεκαδικό ψηφίο	BCD (8421)	2421	συν-3	8,4,-2,-1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
Συνδυασμοί bits που δε χρησιμοποιούνται	1010	0101	0000	0001
	1011	0110	0001	0010
	1100	0111	0010	0011
	1101	1000	1101	1100
	1110	1001	1110	1101
	1111	1010	1111	1110

Αναπαράσταση δεκαδικών ψηφίων

Άλλοι δεκαδικοί κώδικες (II)

❖ οι κώδικες **2421** και **συν-3** είναι αυτοσυμπληρωματικοί

- ▶ το συμπλήρωμα ως προς 9 ενός δεκαδικού αριθμού, προκύπτει απευθείας αν εναλλάξουμε τα 1 και τα 0 της δυαδικής αναπαράστασης

π.χ.

□ $(395)_{10} = (0110\ 1100\ 1000)_{\text{συν-3}}$

□ συμπλήρωμα ως προς 9 του $(395)_{10} = (604)_{10}$

□ $(604)_{10} = (1001\ 0011\ 0111)_{\text{συν-3}}$

δεκαδικό ψηφίο	BCD (8421)	2421	συν-3	8,4,-2,-1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
Συνδυασμοί bits που δε χρησιμοποιούνται	1010	0101	0000	0001
	1011	0110	0001	0010
	1100	0111	0010	0011
	1101	1000	1101	1100
	1110	1001	1110	1101
	1111	1010	1111	1110

Άλλοι δυαδικοί κώδικες

Ανίχνευση σφαλμάτων - bit ισοτιμίας

- ✍ συχνά προκύπτουν **σφάλματα** κατά τη μετάδοση ενός συνόλου bits
 - ▶ π.χ. μετάδοση πακέτων πληροφορίας στα δίκτυα
 - ☞ χρησιμοποιούμε δυαδικούς κώδικες για να **ανιχνεύσουμε** (και να **επιδιορθώσουμε** σφάλματα)
- ❖ ένα απλός **κώδικας ανίχνευσης σφαλμάτων** προκύπτει με την προσθήκη του **bit ισοτιμίας**
 - ▶ ένα **επιπλέον bit** (0 ή 1) που περιλαμβάνεται σε ένα μήνυμα
 - ▶ η **τιμή** του bit ισοτιμίας **επιλέγεται** έτσι ώστε ο **συνολικός** αριθμός των άσων του μηνύματος να γίνει είτε **άρτιος** (άρτια ισοτιμία) είτε **περριτός** (περιττή ισοτιμία)

χαρακτήρας	κωδική λέξη ASCII	με άρτια ισοτιμία	με περιττή ισοτιμία
A	1000001	01000001	11000001
T	1010100	11010100	01010100

Σύνοψη

- ❖ Δεδομένα, ψηφιακά δεδομένα και πληροφορίες
- ❖ Πληροφοριακά συστήματα
- ❖ Συστήματα αρίθμησης
 - ▶ δεκαδικό, δυαδικό, οκταδικό, δεκαεξαδικό
 - ▶ μετατροπή αριθμών σε συστήματα με άλλη βάση
- ❖ Συμπληρώματα
 - ▶ ως προς βάση και ως προς ελαττωμένη βάση
 - ▶ αφαίρεση με συμπληρώματα
- ❖ Αναπαράσταση πληροφορίας
 - ▶ Κωδικοποίηση
 - ▶ Αναπαράσταση κειμένου (ASCII, ISO, UNICODE)
 - ▶ Αναπαράσταση δεκαδικών ψηφίων
 - ▶ Κώδικας BCD & Αριθμητικές πράξεις μεταξύ αριθμών BCD