ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

http://eclass.aueb.gr/courses/INF511/

Αποθήκευση Δεδομένων (ΚΕΦΑΛΑΙΟ 1)

Αλκμήνη Σγουρίτσα

Κοδριγκτώνος 12, 2ος όροφος

E-mail: alkmini@aueb.gr

Αποθήκευση Δεδομένων

- Τα bits και ο τρόπος που αποθηκεύονται
  - Πύλες, δισταθή κυκλώματα
- Δυαδικό σύστημα για αναπαράσταση αριθμών στον υπολογιστή
  - Ακέραιοι αριθμοί (θετικοί/αρνητικοί)
  - Πραγματικοί αριθμοί (Αριθμητική κινητής υποδιαστολής)
  - Προβλήματα: υπερχείλιση, σφάλμα στρογγυλοποίησης
- Αναπαράσταση πληροφορίας στον υπολογιστή
  - Δειγματοληψία σε αναλογική πηγή ήχου
  - Κωδικοποίηση πληροφορίας
- Συμπίεση δεδομένων
- Σφάλματα επικοινωνίας και κωδικοποίηση

Τα bit και η σημασία τους

- Bit = **bi**nary digi**t**
- “Το bit είναι η βασική μονάδα πληροφορίας στους υπολογιστές και στις ψηφιακές επικοινωνίες.” [Wikipedia]
- Ένα bit μπορεί να έχει μόνο μία από 2 τιμές και μπορεί ως εκ τούτου να υλοποιηθεί φυσικά με μια συσκευή δύο καταστάσεων.
  - Αυτές οι τιμές είναι 0 και 1.
- Μερικές πιθανές ερμηνείες για ένα bit:
  - Αριθμητική τιμή (1 ή 0)
  - Τιμές Boolean (αληθές ή ψευδές)
  - Τάση (υψηλή ή χαμηλή)
- Τα δεδομένα που αποθηκεύονται σε έναν υπολογιστή αναπαρίστανται με ακολουθίες bits (π.χ. 01101110) και μπορεί να είναι:
  - Αριθμοί
  - Χαρακτήρες κειμένου
  - Εικόνες
  - Ήχοι
  - Βίντεο
  - Εντολές

Πράξεις Boolean

- Οι πράξεις που χειρίζονται τιμές τύπου αληθής / ψευδής ονομάζονται λογικές πράξεις ή **πράξεις Boolean**.
  - Χρησιμοποιούνται για πράξεις σε bits
  - bit=1 (αληθές), bit=0 (ψευδές)
- Συγκεκριμένες πράξεις:
  - Σύζευξη: AND
  - Διάζευξη: OR
  - Αποκλειστική Διάζευξη: XOR
  - Άρνηση: NOT

Οι λογικές πράξεις σύζευξη (**AND**), διάζευξη (**OR**) και αποκλειστική διάζευξη (**XOR**)

Η πράξη της σύζευξης (AND)

$$\begin{array}{r} 0 \\ \text{AND} \quad 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ \text{AND} \quad 1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \text{AND} \quad 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \text{AND} \quad 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1011 \\ \text{AND} \quad 0010 \\ \hline 0010 \end{array}$$

Η πράξη της διάζευξης (OR)

$$\begin{array}{r} 0 \\ \text{OR} \quad 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ \text{OR} \quad 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ \text{OR} \quad 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ \text{OR} \quad 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1001 \\ \text{OR} \quad 0010 \\ \hline 1011 \end{array}$$

Η πράξη της αποκλειστικής διάζευξης (XOR)

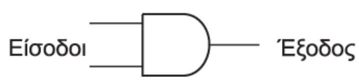
$$\begin{array}{r} 0 \\ \text{XOR} \quad 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ \text{XOR} \quad 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ \text{XOR} \quad 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ \text{XOR} \quad 1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1011 \\ \text{XOR} \quad 0010 \\ \hline 1001 \end{array}$$

Η πράξη της άρνησης (NOT): NOT 0 = 1, NOT 1 = 0

Πύλες

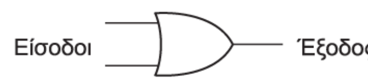
- Η συσκευή (κύκλωμα) που παράγει την έξοδο μίας λογικής πράξης για δεδομένες τιμές εισόδου ονομάζεται **πύλη** (gate).
  - Υλοποιείται από μικρά ηλεκτρονικά κυκλώματα, στα οποία τα ψηφία 0 και 1 αντιπροσωπεύονται από επίπεδα τάσης.

AND



Είσοδοι	Έξοδος
0 0	0
0 1	0
1 0	0
1 1	1

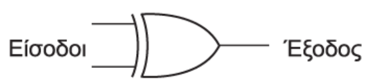
OR



Είσοδοι	Έξοδος
0 0	0
0 1	1
1 0	1
1 1	1

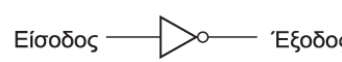
Πίνακες αλήθειας

XOR



Είσοδοι	Έξοδος
0 0	0
0 1	1
1 0	1
1 1	0

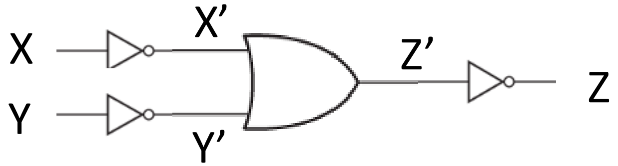
NOT



Είσοδος	Έξοδος
0	1
1	0

Πύλες

- Ερώτηση: Μπορώ να παράγω την πύλη AND χρησιμοποιώντας μόνο πύλες OR και NOT?



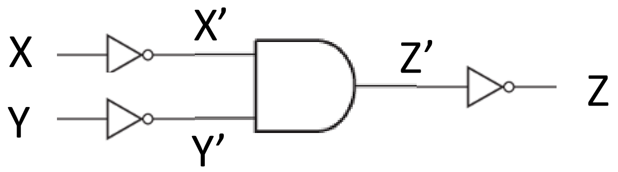
X	Y	X’	Y’	Z’	Z
0	0	1	1	1	0
0	1	1	0	1	0
1	0	0	1	1	0
1	1	0	0	0	1

- Ερώτηση για το σπίτι 1: Πώς μπορώ να παράγω την πύλη OR χρησιμοποιώντας μόνο πύλες AND και NOT?
- Ερώτηση για το σπίτι 2: Πώς μπορώ να παράγω την πύλη XOR χρησιμοποιώντας μόνο πύλες AND, OR και NOT?

  Χρειάζομαι και τις δύο πύλες AND και OR για το τελευταίο?

Πύλες

- <u>Ερώτηση για το σπίτι 1</u>: Πώς μπορώ να παράγω την πύλη OR χρησιμοποιώντας μόνο πύλες AND και NOT?

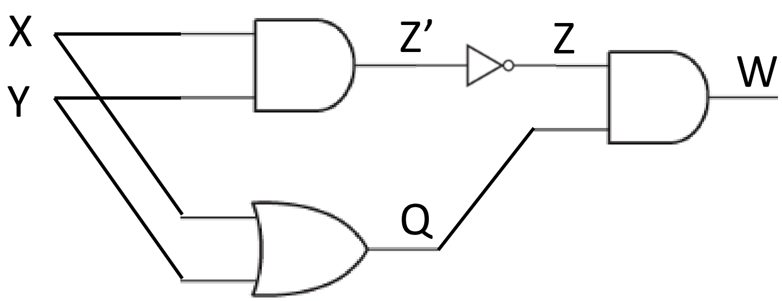


X	Y	X’	Y’	Z’	Z
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

Πύλες

- <u>Ερώτηση για το σπίτι 2</u>: Πώς μπορώ να παράγω την πύλη XOR χρησιμοποιώντας μόνο πύλες AND, OR και NOT?

Χρειάζομαι και τις δύο πύλες AND και OR για το τελευταίο?

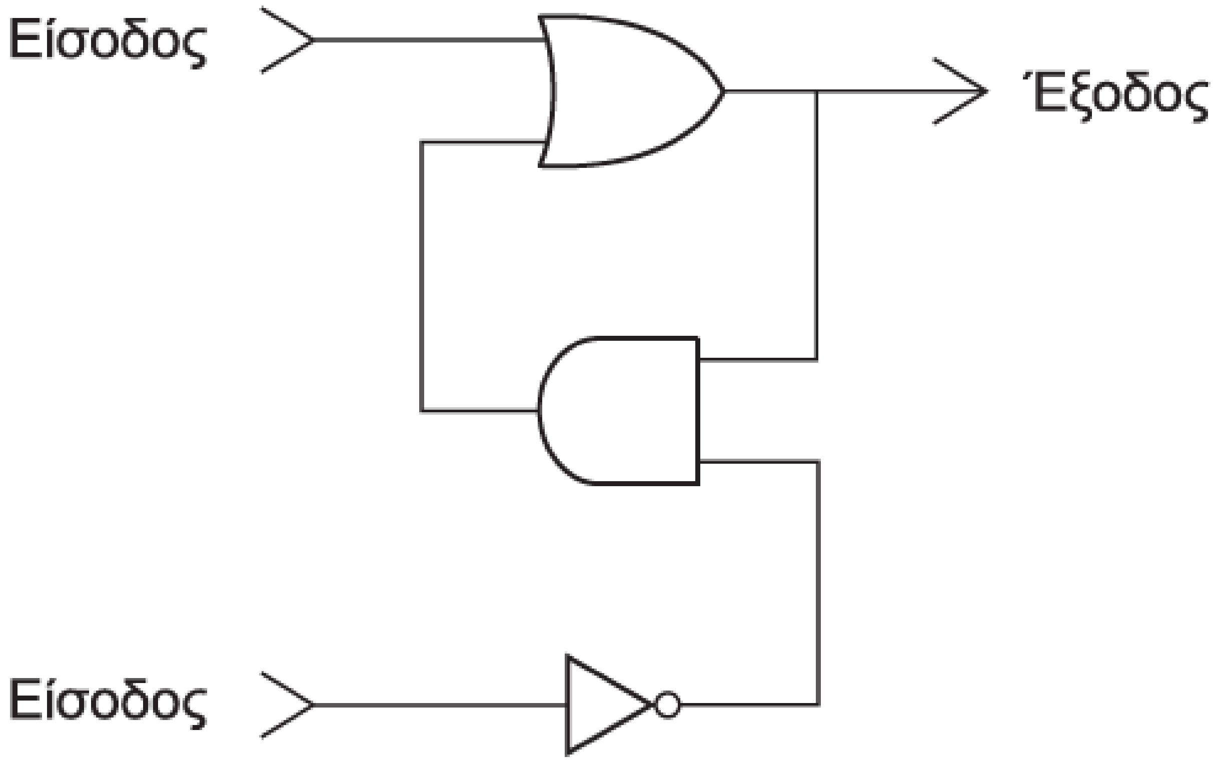


X	Y’	Z’	Z	Q	W
0	0	0	1	0	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	1	0	1	0

Μπορούμε να αντικαταστήσουμε είτε την πύλη AND είτε την πύλη OR με το ισοδύναμο κύκλωμά τους που είδαμε πριν.

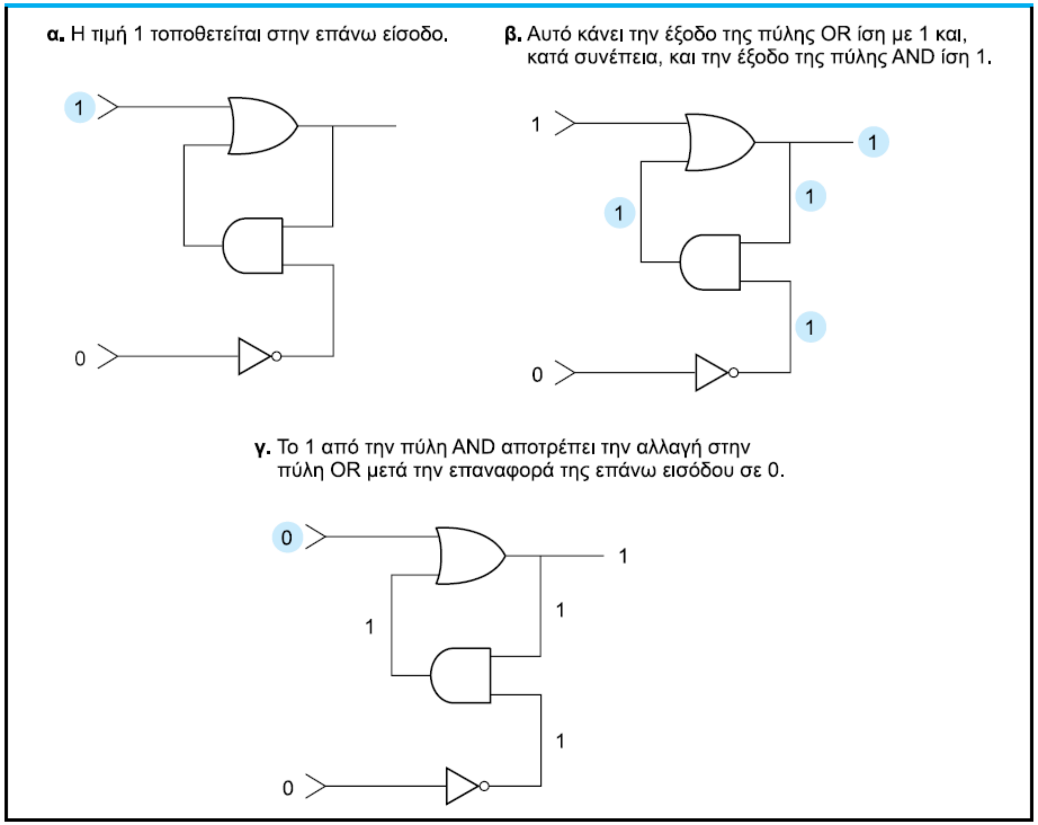
Δισταθές Κύκλωμα (flip-flop)

- Ένα **δισταθές (bi-stable) κύκλωμα** (flip-flop) είναι ένα κύκλωμα που δημιουργείται από πύλες
- Παράγει μία τιμή εξόδου 0 ή 1 η οποία εναλλάσσεται (flip-flop) ανάλογα με τις τιμές εισόδου



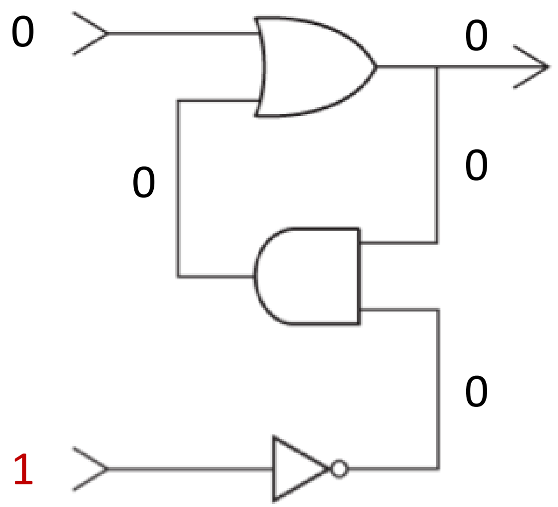
Βασικό παράδειγμα Flip-flop

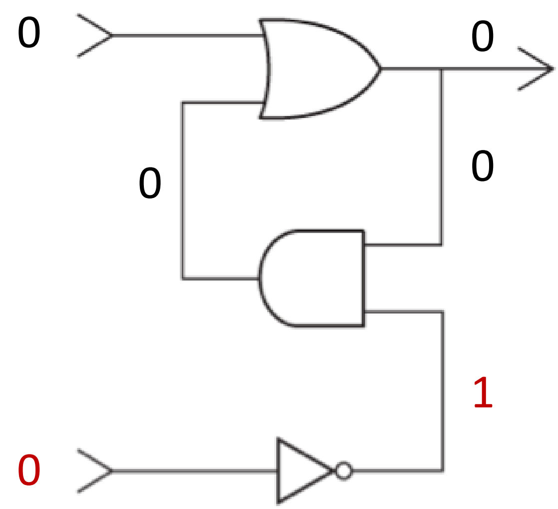
- Αν είναι 0 η κάτω είσοδος και βάλουμε 1 στην πάνω, η έξοδος γίνεται 1.
- Η έξοδος παραμένει 1 αν η πάνω είσοδος γίνει 0.
- Το flip-flop «αποθηκεύει» την έξοδο 1!



Βασικό παράδειγμα Flip-flop

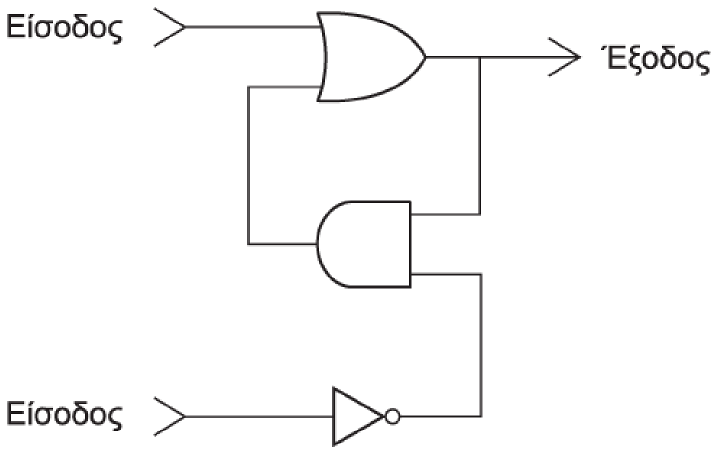
- Ομοίως, αν είναι 0 η πάνω είσοδος και βάλουμε 1 στην κάτω, η έξοδος γίνεται 0.
- Η έξοδος παραμένει 0 αν η κάτω είσοδος γίνει 0.
- Το flip-flop «αποθηκεύει» την έξοδο 0!

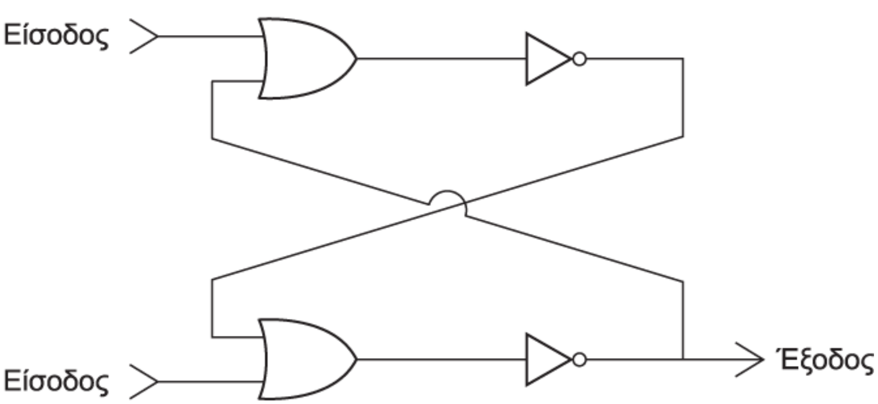




Το δισταθές κύκλωμα (flip-flop) μπορεί να αποθηκεύσει ένα bit δεδομένων

Flip-flops (συνέχεια)

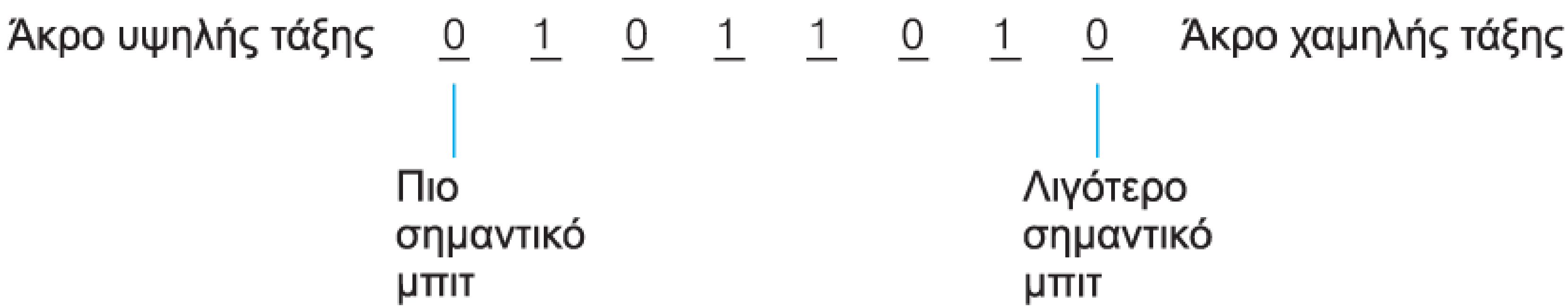




- Ένα flip-flop μπορεί να αποθηκεύσει ένα bit δεδομένων
- Δικτύωση πολλών τέτοιων μονάδων για κατασκευή κυκλωμάτων πολύ υψηλής κλίμακας ολοκλήρωσης (very large-scale integration, VLSI)
- Θεμελιώδεις μονάδες για κατασκευή περίπλοκων ολοκληρωμένων κυκλωμάτων
  - Αφαιρετικότητα: δεν μας ενδιαφέρουν τα κυκλώματα αλλά μόνο τι bit παράγεται στην έξοδο ανάλογα με τα bit στις εισόδους
- Σχεδιασμός ψηφιακών συστημάτων: χρήση πυλών για την κατασκευή συσκευών.

Κύρια μνήμη: κυψελίδες (κελιά)

- Η **κύρια μνήμη** είναι ένα σύνολο κυκλωμάτων (πχ δισταθή), καθένα από τα οποία μπορεί να αποθηκεύσει ένα bit.
- Τα κυκλώματα αυτά είναι διατεταγμένα σε θέσεις, συνήθως των 8 bits (1 byte) που λέγονται **κελιά** (cells).



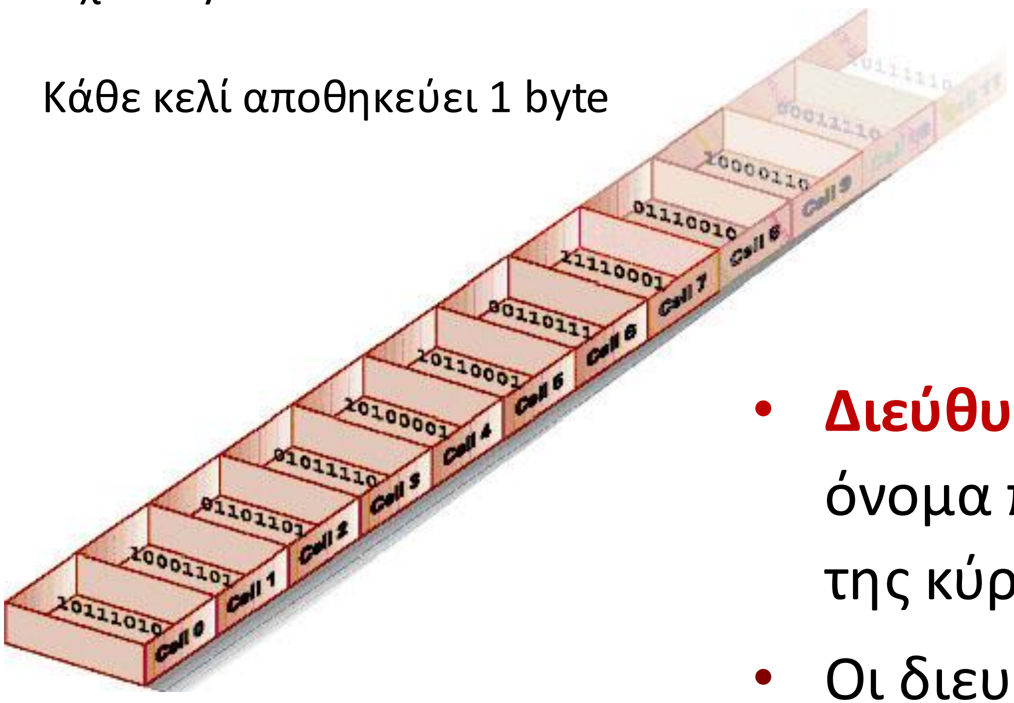
- **Άκρο υψηλής τάξης** (high-order end): το αριστερό άκρο της σειράς στην οποία είναι διατεταγμένα τα bit ενός κελιού μνήμης.
  - Το πιο σημαντικό bit (most significant bit, MSB) είναι το πρώτο bit στο άκρο υψηλής τάξης
- **Άκρο χαμηλής τάξης** (low-order end): είναι αντίστοιχα το δεξιό άκρο.
  - Το λιγότερο σημαντικό bit (least significant bit, LSB) είναι το τελευταίο bit στο άκρο χαμηλής τάξης

Διεύθυνση κύριας μνήμης

Μέγεθος κύριας μνήμης = αριθμός κελιών

π.χ. 1Kbyte = 1024 κελιά = 2^{10} κελιά

Κάθε κελί αποθηκεύει 1 byte



- **Διεύθυνση (address):** ένα μοναδικό όνομα που προσδιορίζει κάθε κελί της κύριας μνήμης
- Οι διευθύνσεις είναι αριθμητικές και ξεκινούν από το 0

Μέτρηση χωρητικότητας μνήμης

- Ο όρος Kilo- αναφέρεται στο ~1.000
  - Kilobyte (KB)= 2^{10} =1024 κελιά (bytes)~ 10^3
- Ο όρος Mega- αναφέρεται στο ~1.000.000
  - Megabyte (MB)= 2^{20} =1.048.576 ~ 10^6
- Ο όρος Giga- συνήθως αναφέρεται στο ~1.000.000.000
  - Gigabyte (GB)= 2^{30} =1.073.741.824 ~ 10^9
- Ο όρος Tera- συνήθως αναφέρεται στο ~1.000.000.000.000
  - Terabyte (TB)= 2^{40} ~ 10^{12}
- Ακόμα: Peta-byte: ~ 10^{15} , Exa-byte: ~ 10^{18}, Zetta-byte: ~ 10^{21} , Yotta-byte: ~ 10^{24}

Είναι βολικό η χωρητικότητα της μνήμης να είναι δύναμη του 2. Γιατί;
Η διεύθυνση κάθε κελιού αναπαρίσταται με ένα δυαδικό αριθμό

https://www.lifewire.com/terabytes-gigabytes-amp-petabytes-how-big-are-they-4125169

Αναπαράσταση κειμένου

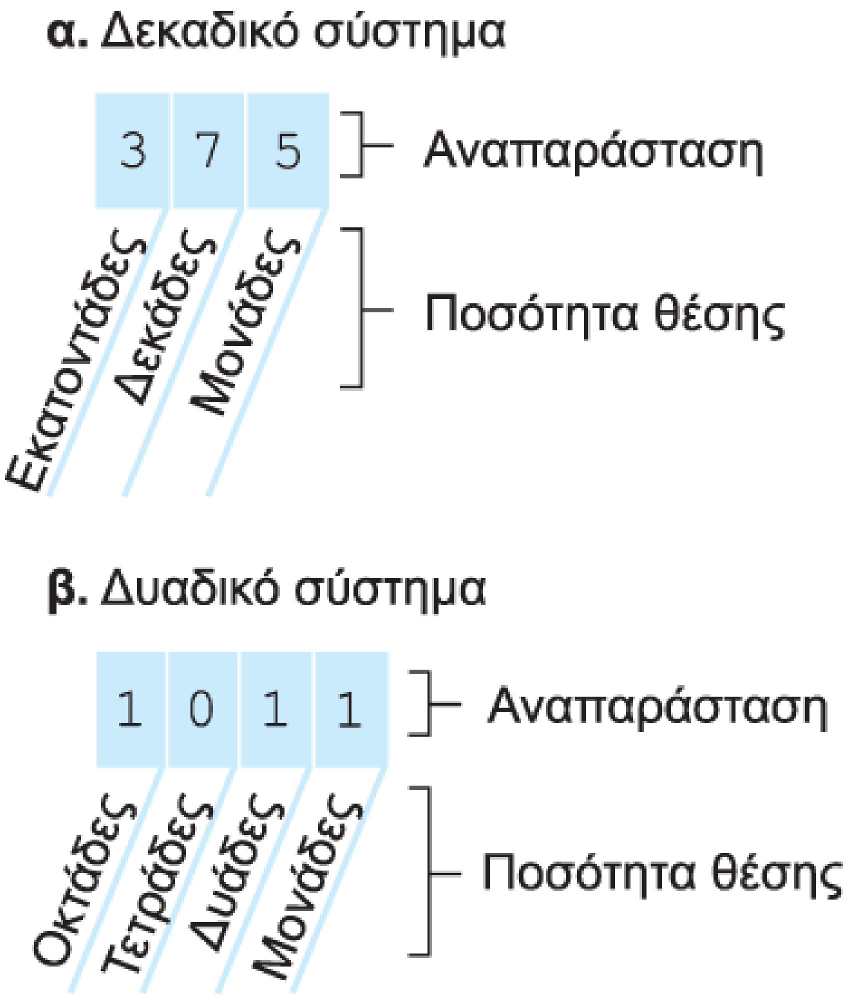
- Κάθε σύμβολο κειμένου (γράμμα αλφαβήτου, σημείο στίξης κτλ.) αντιστοιχίζεται σε μια <u>μοναδική</u> σειρά bit.
  - Αμερικανικό Πρότυπο Κώδικα για την Ανταλλαγή Πληροφοριών (**ASCII):** χρησιμοποιεί σειρές των 8 bit για να αναπαραστήσει σύμβολα του Αγγλικού αλφαβήτου.
  - Ο κώδικας **Unicode** χρησιμοποιεί σχήματα των 16 bit για τα σύμβολα των περισσοτέρων γλωσσών του κόσμου.
  - Ο Διεθνής Οργανισμός Τυποποίησης **ISO** χρησιμοποιεί σχήματα των 32 bit.
- Παράδειγμα: Το μήνυμα "Hello." σε κώδικα ASCII

01001000	01100101	01101100	01101100	01101111	00101110
H	**e**	**l**	**l**	**o**	**.**

Αναπαράσταση αριθμητικών τιμών

- Γιατί όχι με ASCII;
  - 1byte (8 bits) ανά ψηφίο θα σημαίνει ότι χρειαζόμαστε 16 bits για να παραστήσουμε διψήφιους αριθμούς (0-99)
    - Όχι αποδοτικό σε ό,τι αφορά τον αριθμό bits για αναπαράσταση
    - Όπως θα δούμε, με 16 bits μπορούν να παριστάνονται οι ακέραιοι 0 ως 65,535
- Βασικοί συμβολισμοί:
  - **Δυαδικός συμβολισμός (binary notation):** χρησιμοποιεί bits για αναπαράσταση θετικών ακεραίων
  - **Συμβολισμός συμπληρώματος ως προς 2 (two's complement notation)**: αναπαράσταση αρνητικών αριθμών
  - **Αναπαράσταση με υπέρβαση**: αναπαράσταση αρνητικών αριθμών
  - **Συμβολισμός κινητής υποδιαστολής (floating point):** για αναπαράσταση πραγματικών αριθμών και πολύ μεγάλων ακεραίων ή μικρών αριθμών
- Υπάρχουν περιορισμοί στο πόσα bits χρησιμοποιούνται στις αναπαραστάσεις αυτές

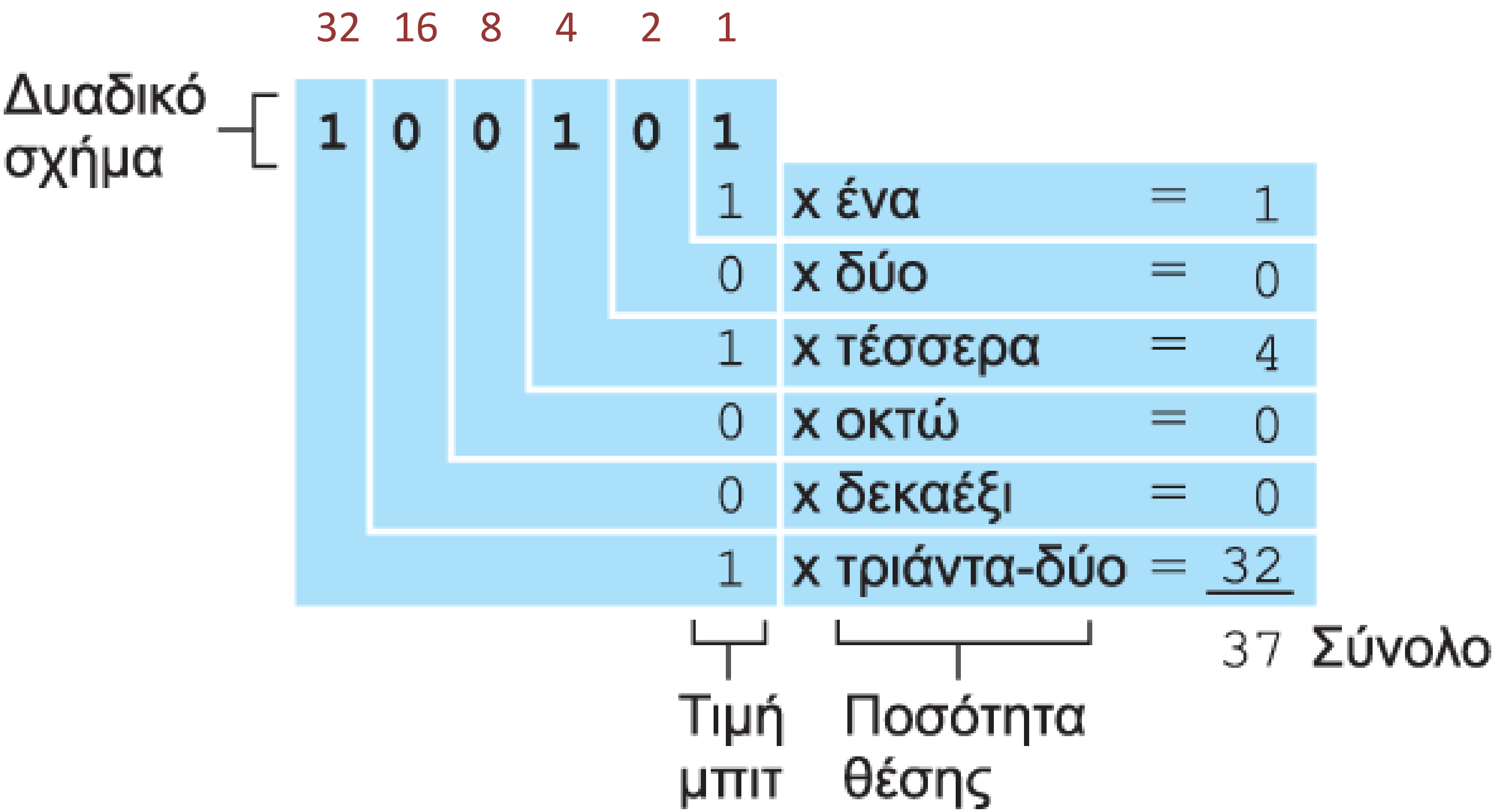
Το δεκαδικό και το δυαδικό σύστημα



Δυαδικός συμβολισμός με 3 bit

000
001
010
011
100
101
110
111

Μετατροπή από δυαδική σε δεκαδική αναπαράσταση



<u>Ερώτηση</u>: ποιος είναι ο αριθμός 10101_2

<u>Απάντηση</u>: $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 16 + 4 + 1 = 21$

Ένας αλγόριθμος για την εύρεση της δυαδικής αναπαράστασης ενός θετικού ακεραίου

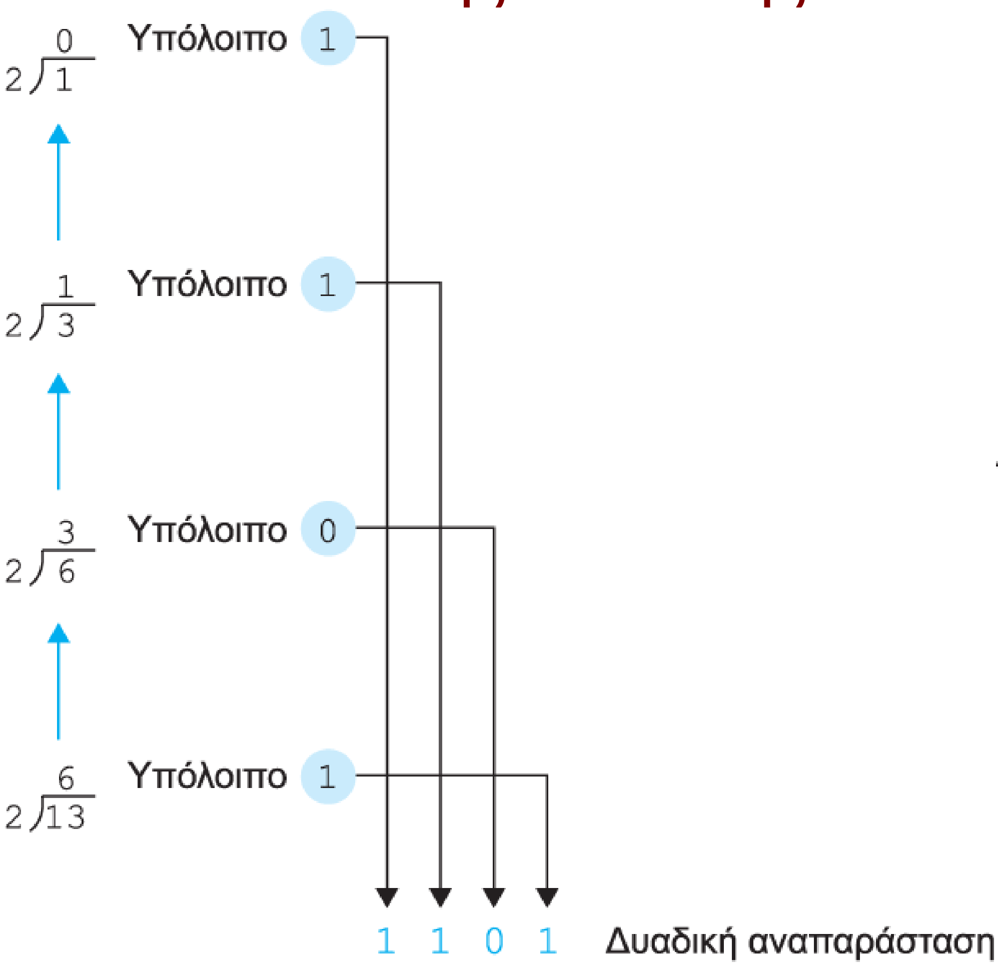
Βήμα 1. Διαίρεσε την τιμή με το δύο και σημείωσε το υπόλοιπο.

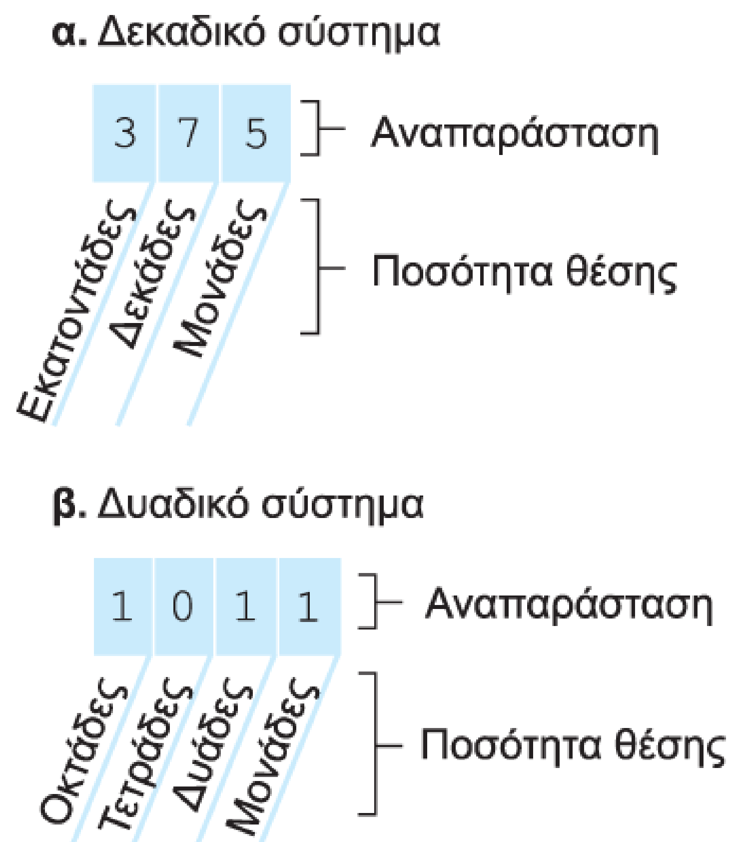
Βήμα 2. Όσο το πηλίκο είναι διάφορο του 0, συνέχισε να διαιρείς το νέο πηλίκο με το δύο και κατέγραφε το υπόλοιπο (γράφε από δεξιά προς αριστερά)

Βήμα 3. Αν προκύψει πηλίκο = 0, stop. Η δυαδική αναπαράσταση της αρχικής τιμής αποτελείται από τα υπόλοιπα, γραμμένα από τα δεξιά προς τα αριστερά με τη σειρά που σημειώθηκαν.

Παράδειγμα: Τρέξτε τον αλγόριθμο για το 13

Η εφαρμογή του αλγορίθμου για τον υπολογισμό της δυαδικής αναπαράστασης του 13





Οι μη αρνητικοί ακέραιοι μπορούν να αναπαρασταθούν ως άθροισμα δυνάμεων του 2

π.χ. 13 = 1 x 8 + 1 x 4 + 0 x 2+ 1 x 1 = 1 x 2^3 + 1 x 2^2 + 0 x 2^1 + 1 x 2^0 = 1101

π.χ. 25 = 1 x 16 (2^4)+ 1 x 8 (2^3) + 0 x 4 (2^2) + 0 x 2 (2^1) + 1 x 1 (2^0) = 11001

Οι κανόνες της δυαδικής πρόσθεσης

$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array}$$

Παράδειγμα:

 111010
+ 11011

1010101

Δεκαεξαδικός συμβολισμός

- Ο **δεκαεξαδικός συμβολισμός** (hexadecimal notation) είναι ένας συμβολισμός για αναπαράσταση μεγάλων ροών από bits
  - 4-άδες bit αντιστοιχίζονται σε 16-δικά σύμβολα
  - Πιο συμπαγής συμβολισμός

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

<u>Ερώτηση</u>: ποιος είναι ο αριθμός 0xA0 = $A0_{16}$

<u>Απάντηση</u>: $10 \cdot 16^1 + 0 \cdot 16^0 = 160$

Αναπαράσταση Ακεραίων (Θετικών και Αρνητικών)

- Οι ακέραιοι με πρόσημο μπορούν να αναπαρασταθούν:
  - Με **συμβολισμό συμπληρώματος ως προς δύο**
  - Με τον **συμβολισμό υπέρβασης**
    - λιγότερο διαδεδομένος – χρήσιμος στην αναπαράσταση αριθμών κινητής υποδιαστολής
- Το πρώτο bit είναι και στις δύο περιπτώσεις bit προσήμου

- Ποιοι αριθμοί μπορούν να αναπαρασταθούν με 2 bits;
  - $2^2 = 4$ αριθμοί: από το -2 έως το 1
- Ποιοι αριθμοί μπορούν να αναπαρασταθούν με 3 bits;
  - $2^3 = 8$ αριθμοί: από το -4 έως το 3
- Ποιοι αριθμοί μπορούν να αναπαρασταθούν με 4 bits;
  - $2^4 = 16$ αριθμοί: από το -8 έως το 7

Γενίκευση: Ποιοι αριθμοί μπορούν να αναπαρασταθούν με N bits;
  - 2^N αριθμοί: από το -2^{N-1} έως το $2^{N-1} - 1$

Σημείωση: Οι Η/Υ που χρησιμοποιούν 32 bits για θετικούς/αρνητικούς ακεραίους μπορούν να αναπαραστήσουν ακέραιους ως το $2^{31}-1 = 2,147,483,647$

Σύστημα συμπληρώματος ως προς δύο

- Θετικοί αριθμοί ξεκινούν πάντα με 0
- Αρνητικοί αριθμοί ξεκινούν πάντα με 1

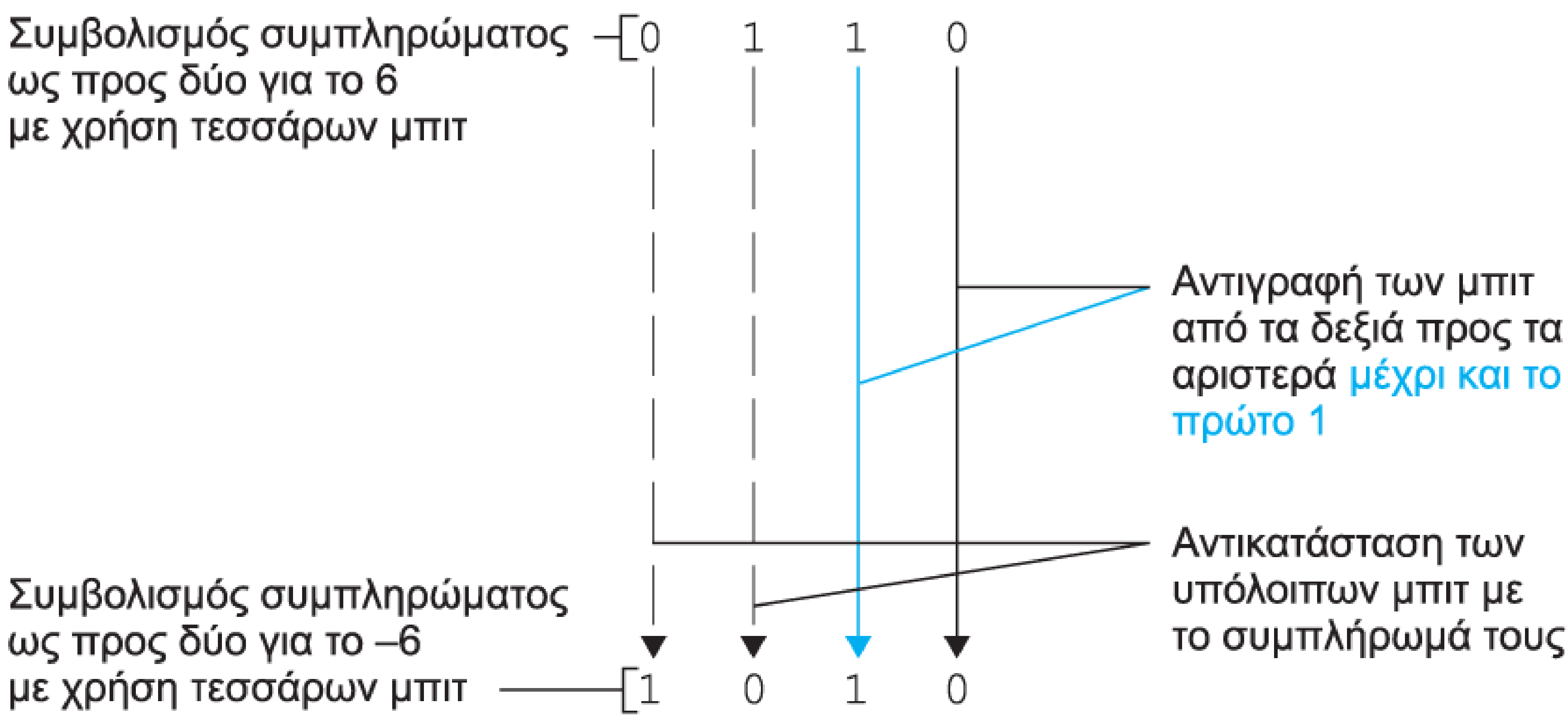
α. Χρήση σχημάτων μήκους τρία

Σχήμα μπιτ	Τιμή
011	3
010	2
001	1
000	0
111	−1
110	−2
101	−3
100	−4

β. Χρήση σχημάτων μήκους τέσσερα

Σχήμα μπιτ	Τιμή
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	−1
1110	−2
1101	−3
1100	−4
1011	−5
1010	−6
1001	−7
1000	−8

Αλγόριθμος αναπάραστασης του -6 σε σύστημα συμπληρώματος ως προς 2 με 4 bits



Ισοδύναμα, αντέστρεψε τα bits του 0110 (6) και πρόσθεσε δυαδικά + 1

Παρατήρηση: 0110 (6) + 1010 (-6) = 10000

Πρόσθεση σε σύστημα συμπληρώματος ως προς δύο

Πρόβλημα στο δεκαδικό	Πρόβλημα σε μορφή συμπληρώματος ως προς δύο	Αποτέλεσμα στο δεκαδικό
3 + 2	0011 + 0010 0101	5
−3 + −2	1101 + 1110 1011	−5
7 + −5	0111 + 1011 0010	2

Σύστημα αναπαράστασης με Υπέρβαση

Υπέρβαση κατά 4 (=2^{3-1})

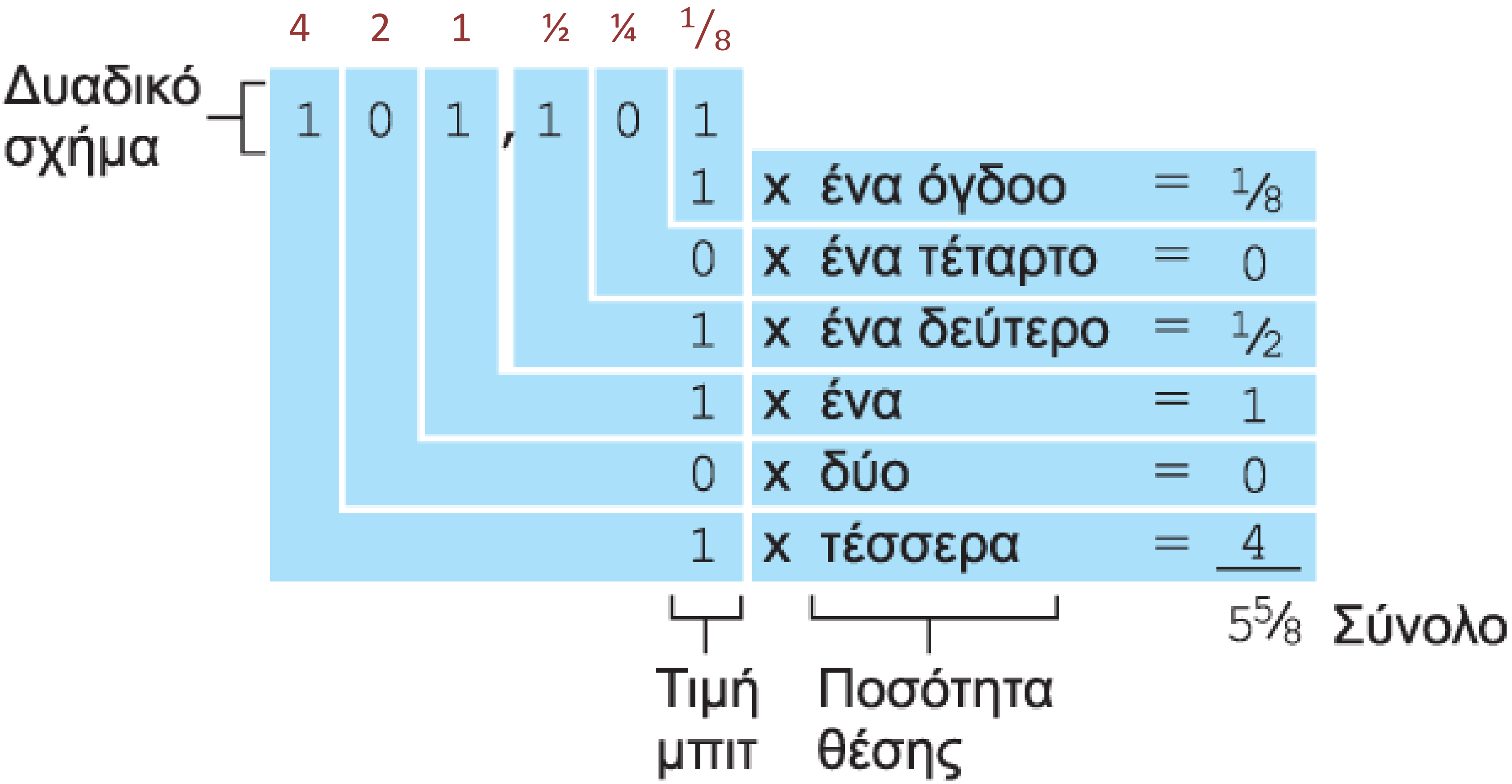
Σχήμα μπιτ	Τιμή
111	3
110	2
101	1
100	0
011	−1
010	−2
001	−3
000	−4

- Ανάθεση τετράδων bits από το κάτω όριο προς τα πάνω, σαν να ξεκινούσε η αρίθμηση από το 0 προς τους θετικούς
- Λέγεται «υπέρβαση κατά 4 (=2^{3-1})» γιατί αν υποθετικά είχαμε αναπαράσταση ΜΟΝΟ θετικών αριθμών, τότε η αναπαράστασή του θα ξεπερνούσε κατά 4 τον αντίστοιχο αριθμό που αναπαρίσταται με την μέθοδο υπέρβασης κατά 4
- Παρατήρηση: Σε σχέση με την αναπαράσταση με συμπλήρωμα ως προς 2, **μόνο το 1^{ο} bit κάθε τετράδας bits είναι ανάποδα** (δηλ. 1 αντί για 0, 0 αντί για 1) !

Υπέρβαση κατά 8 ($=2^{4-1}$)

Σχήμα μπιτ	Τιμή
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	−1
0110	−2
0101	−3
0100	−4
0011	−5
0010	−6
0001	−7
0000	−8

Αποκωδικοποίηση της δυαδικής αναπαράστασης 101.101



Ερώτηση: ποιος είναι ο αριθμός 10.1011_2

Απάντηση: $1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} = 2 + 1/2 + 1/8 + 1/16 = 2.6875$

Κωδικοποίηση του δεκαδικού αριθμού 0.188

$0.188 \times 2 = 0.376$ carry = 0 MSB

$0.376 \times 2 = 0.752$ carry = 0

$0.752 \times 2 = 1.504$ carry = 1

$0.504 \times 2 = 1.008$ carry = 1

$0.008 \times 2 = 0.016$ carry = 0

Answer = .00110 (for five significant digits)

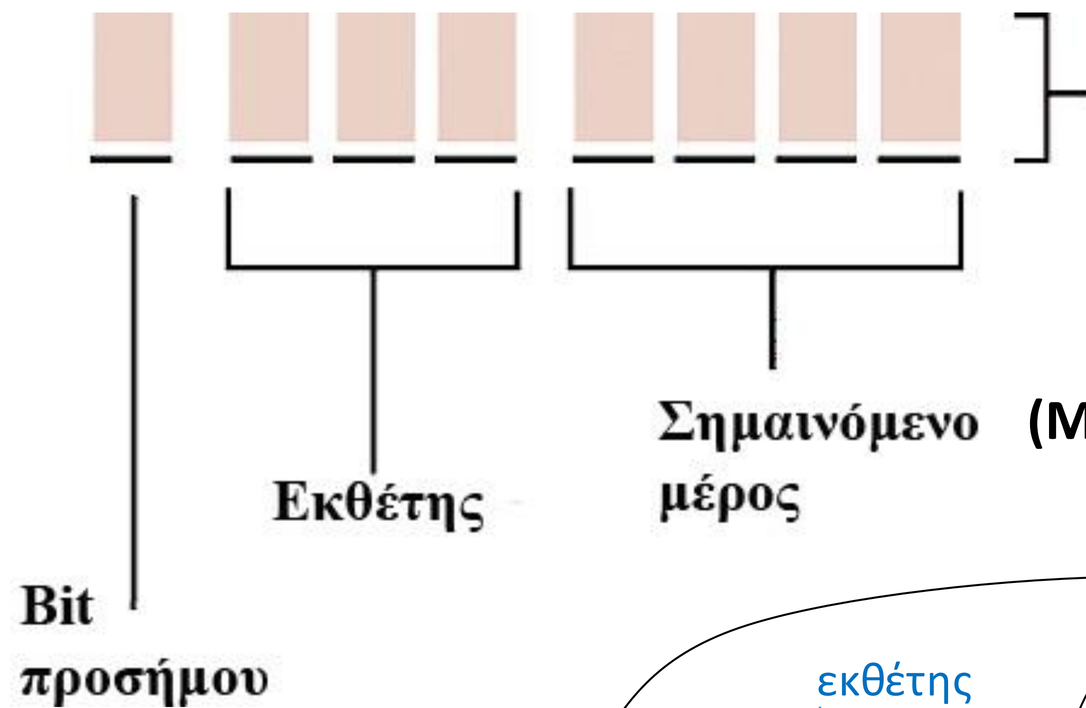
Ερώτηση: πώς κωδικοποιείται ο δεκαδικός αριθμός 5.625 στο δυαδικό

Απάντηση: $5 = 101_2$ με τον αλγόριθμο που είδαμε

$0.625 \cdot 2 = 1.250$, $0.25 \cdot 2 = 0.5$, $0.25 \cdot 2 = 1.0$

Συνολικά: 101.101

Συμβολισμός Κινητής Υποδιαστολής για πραγματικούς αριθμούς



Την mantissa xxxx θα θεωρούμε ότι είναι 1.xxxx (εκτός και αν ειπωθεί κάτι διαφορετικό)

(Mantissa)

εκθέτης

$$-110.11 \leftrightarrow -1.1011 \times 2^{2} \leftrightarrow \{-,2,1011\} \leftrightarrow 11101011$$

Σημ: Το 110 στον εκθέτη είναι το 2 σε συμβολισμό υπέρβασης

Αναπαράσταση για πραγματικούς (ακέραιους και μη ακέραιους) αριθμούς

- και για πολύ μικρούς ή πολύ μεγάλους αριθμούς

Μορφή κινητής υποδιαστολής

Single precision: 8 bit
double precision: 11 bit

Single precision: 23 bit
double precision: 52 bit

S	Εκθέτης	Κλάσμα (mantissa)

- Εκθέτης (exponent) , Κλάσμα (fraction)
  - <u>Εξισορρόπηση</u> μεταξύ <u>ακρίβειας</u> (κλάσμα) και <u>εύρους αναπαράστασης</u> (εκθέτης)
  - S: bit προσήμου (0 $\Rightarrow$ μη αρνητικός, 1 $\Rightarrow$ αρνητικός)

$$x = (-1)^{S} \times (1.\text{Κλάσμα}) \times 2^{\text{Εκθέτης}}$$

Άρα η αναπαράσταση (bit) S E1 … EK S1 S2 S3 … SL
Όπου (K,L) =(8,23) ή (11,52) είναι ο αριθμός:
$z = (-1)^S \times 1.S1\ S2 \ldots SL \times 2^{E1\ E2\ ..EK}$

Κινητή υποδιαστολή (floating point)

- Επιστημονική σημειογραφία (scientific notation): <u>ένα ψηφίο</u> αριστερά της υποδιαστολής
- IEEE 754: αναπαράσταση σε επιστημονική σημειογραφία με το ψηφίο αριστερά της υποδιαστολής να είναι 1 (παραλείπεται).
  - Π.χ. $+1.001 \times 2^4$
- Δύο αναπαραστάσεις κινητής υποδιαστολής (floating point)
  - Απλή ακρίβεια – single precision (32 bit)
  - Διπλή ακρίβεια – double precision (64 bit)
  - Τύποι `float` και `double` της C

Κινητή υποδιαστολή Παραδείγματα

- Κωδικοποιείστε τους παρακάτω αριθμούς στο δεκαδικό σε κανονικοποιημένη μορφή των **8 bits**, με 3 bits για τον εκθέτη και 4 bits για τη mantissa (σημαινόμενο μέρος)

- Ποια είναι η κωδικοποίηση του 2.5;
- Μετατροπή σε δυαδικό: 2.5 = 10.1
- Κανονικοποιημένη μορφή 1.01 x 2^{1} -> Εκθέτης: 1, mantissa: 0100
- Κωδ.: 0(bit προσήμου) 101(εκθέτης) 0100(mantissa) -> 01010100

- Ποια είναι η κωδικοποίηση του -0.375;
- Μετατροπή σε δυαδικό: -0.375 = -0.011
- Κανονικοποιημένη μορφή -1.1 x 2^{-2} -> Εκθέτης: -2, mantissa: 1000
- Κωδ.: 1(bit προσήμου) 010(εκθέτης) 1000(mantissa) -> 10101000

Κινητή υποδιαστολή Παραδείγματα

- Υποθέστε αναπαραστάσεις σε κανονικοποιημένη μορφή των **8 bits**, με 3 bits για τον εκθέτη και 4 bits για τη mantissa (σημαινόμενο μέρος)

- Ποια είναι η αποκωδικοποίηση του σχήματος 00111000;
- Εκθέτης: -1, mantissa: 1000
- Αποκωδικοποίηση $(-1)^0 \times (1.1000) \times 2^{-1} = 0.11$ (=3/4)

- Ποια είναι η αποκωδικοποίηση του σχήματος 11000100;
- Εκθέτης: 0, mantissa: 0100
- Αποκωδικοποίηση $(-1)^1 \times (1.0100) \times 2^0 = -1.01$ (=-5/4)

Εύρος αναπαράστασης απλής ακρίβειας (**single-precision**): 8 bits για τον εκθέτη

- <u>**Εύρος αναπαράστασης IEEE 754**</u> **:** εξαρτάται από τον εκθέτη
- Οι εκθέτες 00000000 και 11111111 δεσμεύονται
  - Το 00..00 για την αναπαράσταση του “0”
  - Το 11…111 για σήμανση ότι ο αριθμός είναι εκτός εύρους αναπαράστασης
- Μικρότερη τιμή αριθμού (κατά απόλυτη τιμή) που μπορεί να αναπαρασταθεί:
  - Μικρότερος Εκθέτης: 00000001
    $\Rightarrow$ αντιστοιχεί σε πραγματική τιμή εκθέτη: $-2^{8-1}+1 = -128 + 1 = -127$ (λόγω της υπέρβασης κατά 128)
  - Μικρότερο Κλασματικό μέρος: 000…00 $\Rightarrow$ Σημαντικό μέρος = 1.0
  - $\pm 1.0 \times 2^{-127} \approx \mathbf{\pm 10^{-38}}$
- Μεγαλύτερη (κατά απόλυτη) τιμή που μπορεί να αναπαρασταθεί:
  - Μεγαλύτερος Εκθέτης: 11111110
    $\Rightarrow$ αντιστοιχεί σε πραγματική τιμή εκθέτη $= 2^{8-1} - 1 - 1 = 128 - 2 = +126$
  - Μεγαλύτερο Κλασματικό μέρος: 111…11 $\Rightarrow$ Σημαντικό μέρος ≈ 2.0
  - $\pm 2.0 \times 2^{+126} \approx \mathbf{\pm 10^{+38}}$

Εύρος αναπαράστασης διπλής ακρίβειας (double-precision): 11 bits για τον εκθέτη

- **Εύρος αναπαράστασης IEEE 754 :** εξαρτάται από τον εκθέτη
- Οι εκθέτες 0000…00 και 1111…11 δεσμεύονται
  - Το 00..00 για την αναπαράσταση του “0”
  - Το 11…111 για σήμανση ότι ο αριθμός είναι εκτός εύρους αναπαράστασης
- Μικρότερη (κατά απόλυτη) τιμή που μπορεί να αναπαρασταθεί:
  - Μικρότερος Εκθέτης: 00000000001
    $\Rightarrow$ πραγματικός μικρότερος εκθέτης $= -2^{11-1} + 1 = -1024 + 1 = -1023$
  - Μικρότερο Κλασματικό μέρος: 000…00 $\Rightarrow$ Σημαντικό μέρος = 1.0
  - $\pm 1.0 \times 2^{-1023} \approx \pm \mathbf{10^{-308}}$
- Μεγαλύτερη (κατά απόλυτη) τιμή που μπορεί να αναπαρασταθεί:
  - Μεγαλύτερος Εκθέτης: 11111111110
    $\Rightarrow$ πραγματικός μεγαλύτερος εκθέτης $= 2^{11-1} - 1 - 1 = 1024 - 2 = +1022$
  - Μεγαλύτερο Κλασματικό μέρος: 111…11 $\Rightarrow$ Σημαντικό μέρος ≈ 2.0
  - $\pm 2.0 \times 2^{+1022} \approx \pm \mathbf{10^{+308}}$

Ακρίβεια αναπαράστασης στην κινητή υποδιαστολή

- **<u>Ακρίβεια αναπάραστασης</u>**: εξαρτάται από τον **αριθμό bits στην mantissa**
- <u>Single-Precision Floating Point</u>: αναπαράσταση με 32 bits
  - 1, 8, 23 bits για πρόσημο, εκθέτη, mantissa αντίστοιχα
  - Ερώτηση: πόσα δεκαδικά ψηφία ακρίβεια δίνει η παραπάνω αναπαράσταση;
  - Hint: Δείτε τον **αριθμό bits για mantissa**
  - Ο μικρότερος αριθμός που μπορεί να αναπαρασταθεί με 23 ψηφία mantissa είναι ο 2^{-23} (αυτή είναι η μικρότερη δυνατή διαφορά μεταξύ δύο αριθμών)
  - Με πόσα δεκαδικά ψηφία αντιστοιχεί αυτός στο δεκαδικό σύστημα;
  - Ο αριθμός των δεκαδικών ψηφίων ενός αριθμού x είναι $-\log_{10}x$
  - Π.χ. Το $10^{-5} = 0.00001$ έχει 5 δεκαδικά ψηφία
  - Το 2^{-23} είναι ισοδύναμο με $23 \times \log_{10}2 \approx 23 \times 0.3 \approx 6.9$: 6 δεκαδικά ψηφία ακρίβειας

Ακρίβεια αναπαράστασης στην κινητή υποδιαστολή

- **Ακρίβεια αναπάραστασης**: εξαρτάται από τον **αριθμό bits στην mantissa**
- Double-Precision Floating Point: αναπαράσταση με 64 bits
  - 1, 11, 52 bits για πρόσημο, εκθέτη, mantissa αντίστοιχα
  - Ερώτηση: πόσα δεκαδικά ψηφία ακρίβεια δίνει η παραπάνω αναπαράσταση;
  - Το 2^{-52} είναι ισοδύναμο με $52 \times \log_{10}2 \approx 52 \times 0.3 \approx 15.6$: 15 δεκαδικά ψηφία ακρίβειας

Προβλήματα αριθμητικών πράξεων

- Αδυναμία αναπαράστασης ενός αριθμού με το υπάρχον σύστημα αναπαράστασης
- <u>Υπερχείλιση</u>: ο αριθμός είναι εκτός του εύρους τιμών που μπορούν να αναπαρασταθούν
  - Πολύ μεγάλος ή πολύ μικρός
- <u>Σφάλμα στρογγυλοποίησης</u>: η ακρίβεια του συστήματος δεν αρκεί για να αναπαραστήσει έναν αριθμό
  - Ο αριθμός χρειάζεται περισσότερα ψηφία μετά την υποδιαστολή από όσα έχει η αναπαράσταση

Υπερχείλιση

- Μπορεί να προκύψει κατά την πρόσθεση 2 θετικών ή 2 αρνητικών αριθμών
  - Συμβαίνει όταν ένας αριθμός δεν μπορεί να παρασταθεί <u>με το υπάρχον σύστημα αναπαράστασης</u> γιατί είναι πολύ μεγάλος
  - Π.χ. για ένα σύστημα συμπληρώματος ως προς 2 με 4 bits, μπορούμε να παραστήσουμε τις ακέραιες τιμές {-8,-7,...,0,1, ...7} (όπως είδαμε νωρίτερα)
    - Αν κατά την πρόσθεση 2 θετικών αριθμών, προκύψει αριθμός με πρώτο ψηφίο 1 (δηλ. αρνητικός), έχουμε υπερχείλιση
    - Π.χ. δείτε τι γίνεται στην πρόσθεση: 4+4 (0100+0100)
  - Ομοίως για πρόσθεση 2 αρνητικών αριθμών π.χ. (-5) + (-4) (1011+1100)
- Οι Η/Υ χρησιμοποιούν 32 bits για θετικούς/αρνητικούς ακεραίους → μπορούν να αναπαραστήσουν ακέραιους ως 2,147,483,647
- Υπενθύμιση:
  - N bits: μπορούν να αναπαραστήσουν αριθμούς $\{-2^{N-1}, \ldots, 2^{N-1}-1\}$
  - Αν έχω να αναπαραστήσω μόνο θετικούς: $\{0,\ldots, 2^{N}-1\}$

Παράδειγμα υπερχείλισης

- Έστω 4 bit διαθέσιμα + 1 bit προσήμου (5 bit συνολικά)
- Συμπλήρωμα ως προς 2: αναπαριστά αριθμούς στο [-16, 15]
- Πρόσθεση 12 (0**1100**)

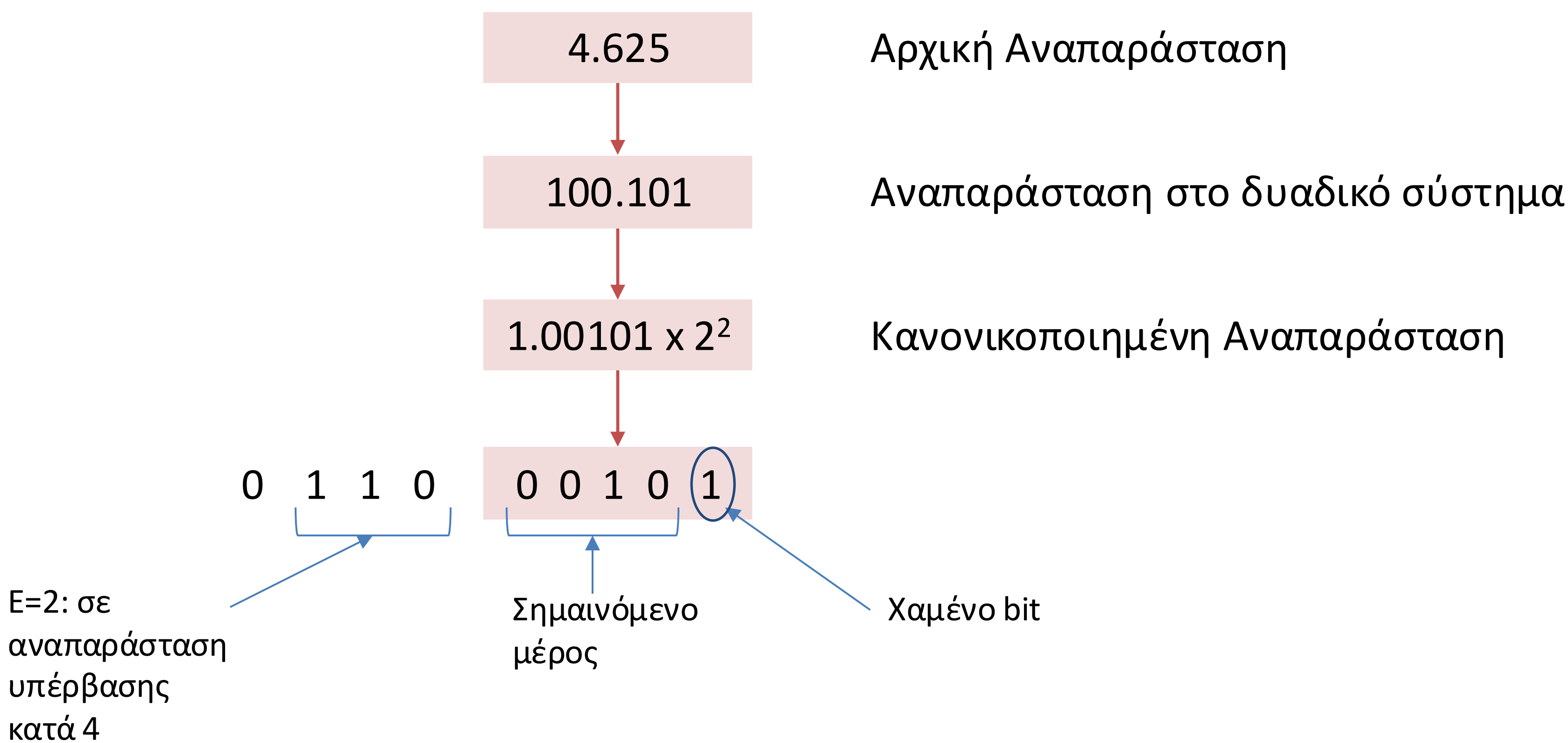
  \+ 5 (0**0101**)

  1**0001**
- Αποτέλεσμα (10001) < 0
- Σαν τιμή του bit προσήμου τίθεται όχι το πρόσημο, αλλά το bit που προκύπτει από την υπερχείλιση!
- Συμπέρασμα: Η υπερχείλιση μπορεί να ανιχνευτεί από το bit προσήμου!

Operation	Operand A	Operand B	Result indicating overflow
$A+B$	≥ 0	≥ 0	< 0
$A+B$	< 0	< 0	≥ 0
$A-B$	≥ 0	< 0	< 0
$A-B$	< 0	≥ 0	≥ 0

Σφάλμα Στρογγυλοποίησης

- **Περικοπή (στρογγυλοποίηση):** όταν ένας αριθμός βρίσκεται μεταξύ δύο διαδοχικών αναπαριστώμενων αριθμών
- Η mantissa **δεν είναι αρκετά μεγάλη** για να αναπαραστήσει τον αριθμό Υποθέστε για αυτό το παράδειγμα ότι ο αριθμός γράφεται 1.xxxx 2^E



Σφάλμα Στρογγυλοποίησης (2)

- Μπορεί ακόμα και η σειρά της πρόσθεσης να παίξει ρόλο!!
  - Πρόσθεση: πρέπει να είναι με τον ίδιο εκθέτη
  - Αν ένας πολύ «μεγάλος» αριθμός προστεθεί σε έναν πολύ «μικρό»...

- Δοκιμάστε την πρόσθεση 2.5 + 1/16 + 1/16 (με mantissa 4 bits)
  - 2.5 = 10.1, 1/16 = 0.0001
  - 2.5 + 1/16 = 10.1001 = 1.01001×2^1 = 2.5 Χάνεται το ‘1’ άρα το 1/16
- Δοκιμάστε τώρα την πρόσθεση (1/16 + 1/16) + 2.5
  - 1/16 = 0.0001 = 1.0×2^{-3}
  - 1/16 + 1/16 = 1/8 = = 0.001
  - 1/8 + 2.5 = 10.101 = 1.0101×2^1 Δεν χάνεται τίποτα!

- Επίσης σφάλματα λόγω ατέρμονου αριθμού ψηφίων π.χ. 1/3, 1/10

Σφάλματα

Ερώτηση για το σπίτι:

- Υποθέστε αναπαραστάσεις σε κανονικοποιημένη μορφή των **8 bits**, με 3 bits για τον εκθέτη και 4 bits για τη mantissa (σημαινόμενο μέρος).
- Επίσης υποθέστε ότι οι εκθέτες 000 και 111 **δεσμεύονται** για την αναπαράσταση του 0 και για τη σήμανση ότι ο αριθμός είναι εκτός του εύρους αναπαράστασης.
- **Τι σφάλμα** προκύπτει κατά την αναπαράσταση των παρακάτω αριθμών ?
  - 5.375
  - 5.25
  - 17
  - 9.5

Σφάλματα

Ερώτηση για το σπίτι:

- Υποθέστε αναπαραστάσεις σε κανονικοποιημένη μορφή των **8 bits**, με 3 bits για τον εκθέτη και 4 bits για τη mantissa (σημαινόμενο μέρος).
- Επίσης υποθέστε ότι οι εκθέτες 000 και 111 **δεσμεύονται** για την αναπαράσταση του 0 και για τη σήμανση ότι ο αριθμός είναι εκτός του εύρους αναπαράστασης.
- **Τι σφάλμα** προκύπτει κατά την αναπαράσταση των παρακάτω αριθμών ?
  - 5.375 = 101.011 = 1.01011 x 2^2 -> 0 110 0101 -> Σφάλμα περικοπής
  - 5.25 = 101.01 = 1.0101 x 2^2 -> 0 110 0101 -> Δεν υπάρχει σφάλμα
  - 17 = 10001 = 1.0001 x 2^4 -> Σφάλμα Υπερχείλισης
  - 9.5 = 1001.1 = 1.0011 x 2^3 -> Σφάλμα Υπερχείλισης

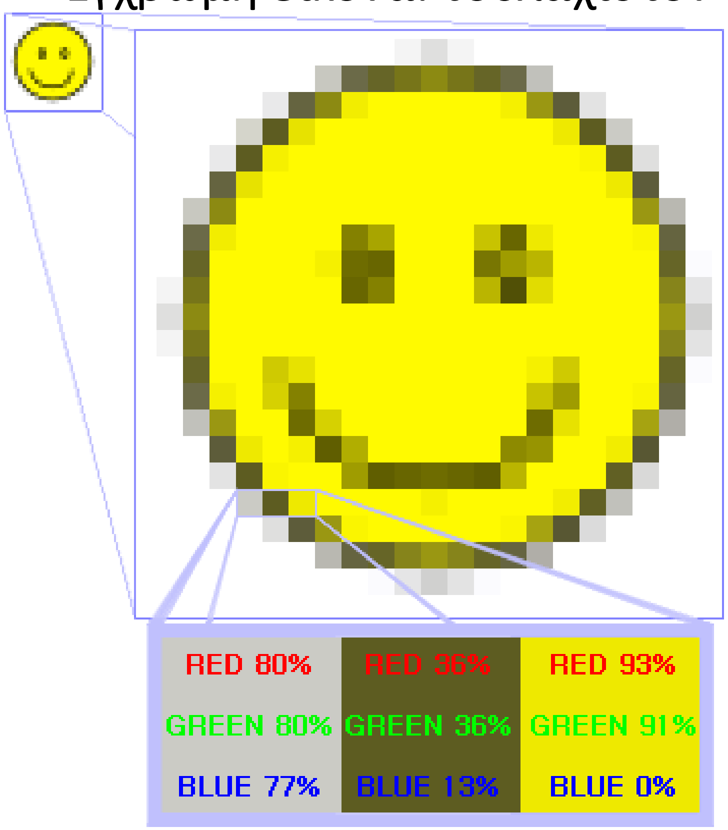
Αναλογικά και Ψηφιακά Συστήματα

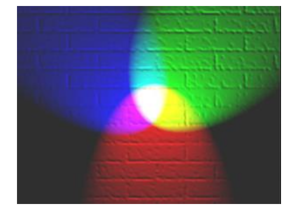
- Αναλογικά: παριστάνουν την ακριβή τιμή του μεγέθους (π.χ. την ένταση του ήχου)
- Ψηφιακά: αναπαριστούν την τιμή με bits {0,1}
  - https://learn.sparkfun.com/tutorials/analog-vs-digital

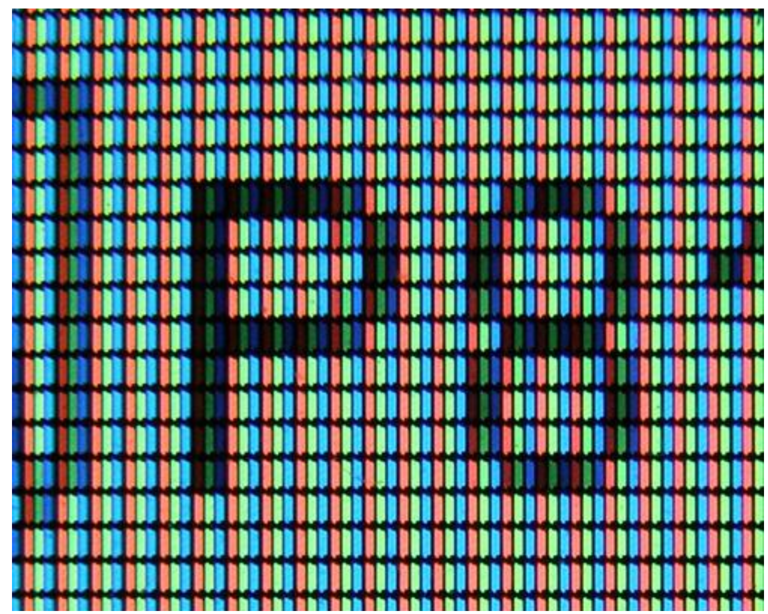
- Παράδειγμα βιβλίου: θέλω να αναπαραστήσω την πληροφορία: υπάρχει 1 κουβάς με νερό, γεμάτος κατά τα ¾.
  - Αναλογική Αναπαράσταση: ένας κουβάς γεμάτος κατά τα ¾
  - Δυαδική (ψηφιακή) αναπαράσταση: 0.11
    - 1 γεμάτος κουβάς για το ½ (1 bit) και 1 γεμάτος κουβάς για το ¼ (1 bit)
  - Συγκρίνετέ τα προς την ανθεκτικότητα στα σφάλματα κάνοντας ένα ταρακούνημα στους κουβάδες - και προσπαθώντας να καταλάβετε ποια είναι η ακριβής τιμή (3/4)
- Συμπέρασμα: Τα ψηφιακά συστήματα είναι **πολύ πιο ανθεκτικά** σε σφάλματα (λόγω εξωγενών παραγόντων) από τα αναλογικά

Ψηφιακή Αναπαράσταση εικόνων

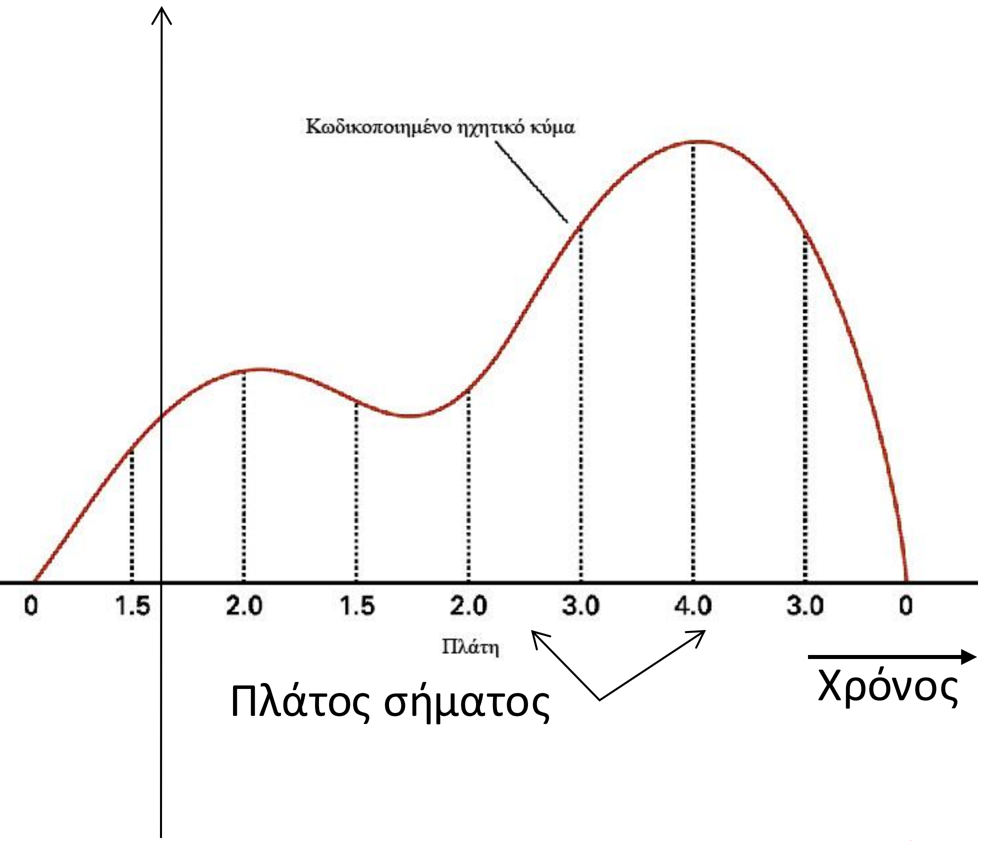
- Εικονοστοιχεία (pixels – [picture elements]
- Pixel = σημείο με χρωματική πληροφορία = (%R,%G,%B)
- Εικόνα (bit map): ορθογώνιο πλέγμα από pixels
- Μαυρόασπρη εικόνα: 1b/pixel, 1Byte/pixel
- Έγχρωμη εικόνα: τουλάχιστον 3B/pixel (RGB χρωμ. μοντέλο)







Ψηφιακή αναπαράσταση ήχου: δειγματοληψία



- 1. **Δειγματοληψία*** (sampling) ηχητικού σήματος ανά τακτά χρονικά διαστήματα (milliseconds)
- 2. **Καταγραφή πλάτους** (έντασης) ηχητικού σήματος σε κάθε χρονική στιγμή
- 3. **Απεικόνιση τιμής πλάτους με bits**
- 4. **Απεικόνιση σήματος ως ακολουθία <u>τιμών</u>**, καθεμιά από τις οποίες απεικονίζεται με έναν αριθμό bits
- Ψηφιακή επεξεργασία ήχου:
  - Κάθε δείγμα αναπαρίσταται με 16 bit (ή 32 bits για στέρεο)
  - Ενδεικτικοί ρυθμοί δειγματοληψίας:
  - Τηλέφωνο: 8.000 δείγματα / δευτερόλεπτο
  - CD: 44.100 δείγματα / sec

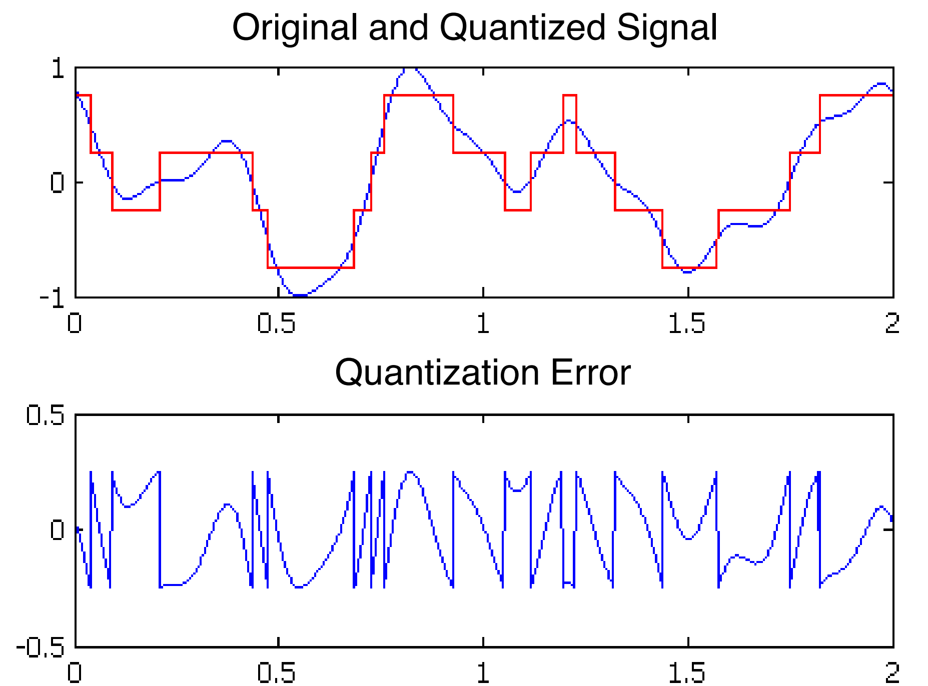
- Η δειγματοληψία πρέπει να γίνεται **αρκετά συχνά** ώστε να μπορεί να γίνει ανακατασκευή του αρχικού σήματος (που είναι συνεχές)
- Θεώρημα Nyquist: **Τουλάχιστον 2 φορές** πιο συχνά από τη μεγαλύτερη συχνότητα του σήματος

Ερώτηση

- Πόσα bits χρειάζονται για να αναπαραστήσουν ένα μουσικό κομμάτι διάρκειας 4 λεπτών σε ένα CD;
- 4 min x 60 sec/min x 44,100 samples/sec x 16 bit/sample = 169,344,000 ~ 169 Mbits ~ 21 Mbytes

Συμπίεση δεδομένων (Κωδικοποίηση πηγής)

- Αντικείμενο της Θεωρίας Πληροφορίας
- **Στόχος: Μείωση όγκου δεδομένων προς μετάδοση**
- Συμπίεση ή κωδικοποίηση πηγής (source coding)
- Με απώλειες (lossy) ή χωρίς απώλειες (lossless)
- Συμπίεση με απώλειες: Κβαντοποίηση (quantization)
  - στρογγυλοποίηση



MIT video lecture by Bob Gallager

https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-450-principles-of-digital-communications-i-fall-2006/video-lectures/lecture-6-quantization/

Πολύ καλό tutorial https://courses.engr.illinois.edu/ece110/fa2017/content/courseNotes/files/?samplingAndQuantization

Συμπίεση δεδομένων χωρίς απώλειες

1. **Κωδικοποίηση τρέχοντος μήκους** (run-length coding): π.χ. ακολουθία από 253 1's, 118 0's και 87 1's
   - Πώς κωδικοποιείται αυτή η ακολουθία με «λίγα» bits ;
2. **Κωδικοποίηση με βάση την συχνότητα:** π.χ. Κώδικας Huffman
   - Κάθε σύμβολο κωδικοποιείται ως μια ακολουθία από bits
   - Το μήκος της ακολουθίας bit που αναπαριστά ένα σύμβολο είναι <u>αντιστρόφως ανάλογο</u> με την συχνότητα εμφάνισης του συμβόλου
     - Τα **συχνότερα** εμφανιζόμενα σύμβολα αναπαρίστανται με **μικρότερου** μήκους ακολουθίες bits
     - Τα σπανιότερα εμφανιζόμενα σύμβολα αναπαρίστανται με μεγαλύτερου μήκους ακολουθίες bits
     - Ίδια λογική με τον κώδικα Morse
   - Ανήκει στην ομάδα κωδικών μεταβλητού μήκους (variable-length codes)

Κώδικας Huffman (1)

Τα **συχνότερα** εμφανιζόμενα σύμβολα αναπαρίστανται με **μικρότερου** μήκους ακολουθίες bits.

Παράδειγμα 1

- Υποθέστε ότι έχετε 4 «σύμβολα» A, B, C, D και το προς συμπίεση κείμενο είναι:

  BAAACDCCABACABBDAACBAADACABCCAACAACDCACA
- Το A εμφανίζεται 18 φορές, το B 6 φορές, το C 12 φορές, και το D 4 φορές.
- Προσέγγιση με **ίσο μήκος bits** για κωδικοποίηση.
  - Π.χ. A→00, B→01, C→10, D→11
  - Χρειάζονται 40 x 2 = **80 bits** για την κωδικοποίηση του κειμένου
- **Κώδικας Huffman**: **μεταβλητό μήκος bits** για κωδικοποίηση,
  - A→0, B→101, C→11, D→100
  - Χρειάζονται 18 x 1 + 12 x 2 + 10 x 3 = 72 bits για την κωδικοποίηση του κειμένου
- Ερώτηση: Γιατί δεν είναι καλή η κωδικοποίηση A→0, B→10, C→1, D→01 ;
- Απάντηση: Το σχήμα 10 αντιστοιχεί και στο B και στο CA!

Κώδικας Huffman (2)

Τα **συχνότερα** εμφανιζόμενα σύμβολα αναπαρίστανται με **μικρότερου** μήκους ακολουθίες bits.

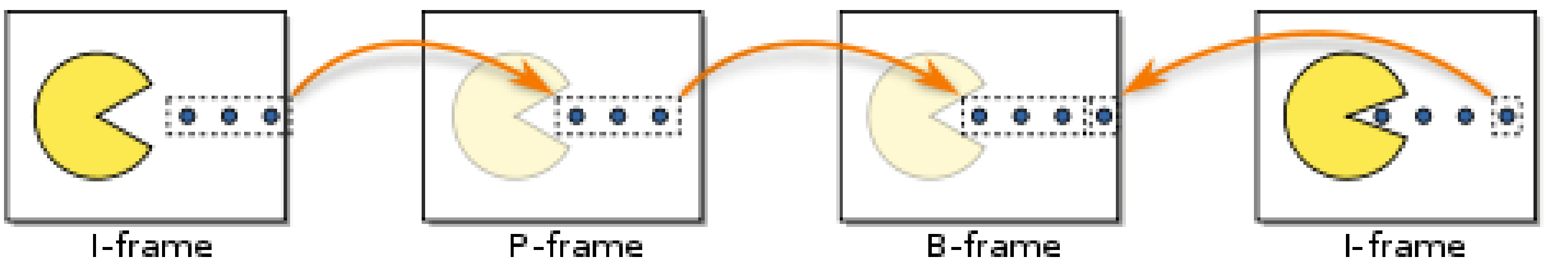
Παράδειγμα 2

- Γενικότερα υποθέστε ότι έχετε 2 «σύμβολα» A, B, και στο προς συμπίεση κείμενο το A εμφανίζεται 90% των φορών και το B 10%
- Υποθέστε ότι έχετε 2 επιλογές: κωδικοποίηση με ακολουθία bit με μήκος 10 bits ή με ακολουθία bit με μήκος 20 bits
  - Αν: A→10 bits, B→20 bits, ο μέσος όρος του αριθμού bits = **11**

    (=10 x 0.9 + 20 x 0.1)
  - Αλλά αν κάναμε την αντιστοιχία A→ 20 bits, B→ 10 bits, ο μέσος όρος του αριθμού bits = **19** (= 20 x 0.9 + 10 x 0.1)

Συμπίεση δεδομένων

- 3. **Διαφορική (ή σχετική) κωδικοποίηση** (differential or relative encoding):
  - Κωδικοποιείται η διαφορά σε σχέση π.χ. με την προηγούμενη εικόνα



Συμπίεση δεδομένων

- 4. **Κωδικοποίηση προσαρμοζόμενου λεξικού** (adaptive dictionary encoding)
  - Κωδικοποίηση κατά Lembel-Ziv-Welsh (LZW)
  - Υπάρχει ένα λεξικό από ζεύγη (μπλοκ χαρακτήρων + αναπαράσταση)
  - Αρχίζουμε με αναπαράσταση απλών δομικών μπλοκ
  - Καθώς βρίσκονται μεγαλύτερα μπλοκ, κωδικοποιούνται και προστίθεται στο λεξικό

xyx xyx xyx xyx

Κωδικοποίηση:
1 2 1 3 4 3 4 3 4

Λεξικό: x = 1
y = 2
“(κενό)“ = 3
Εύρεση λέξης: xyx = 4

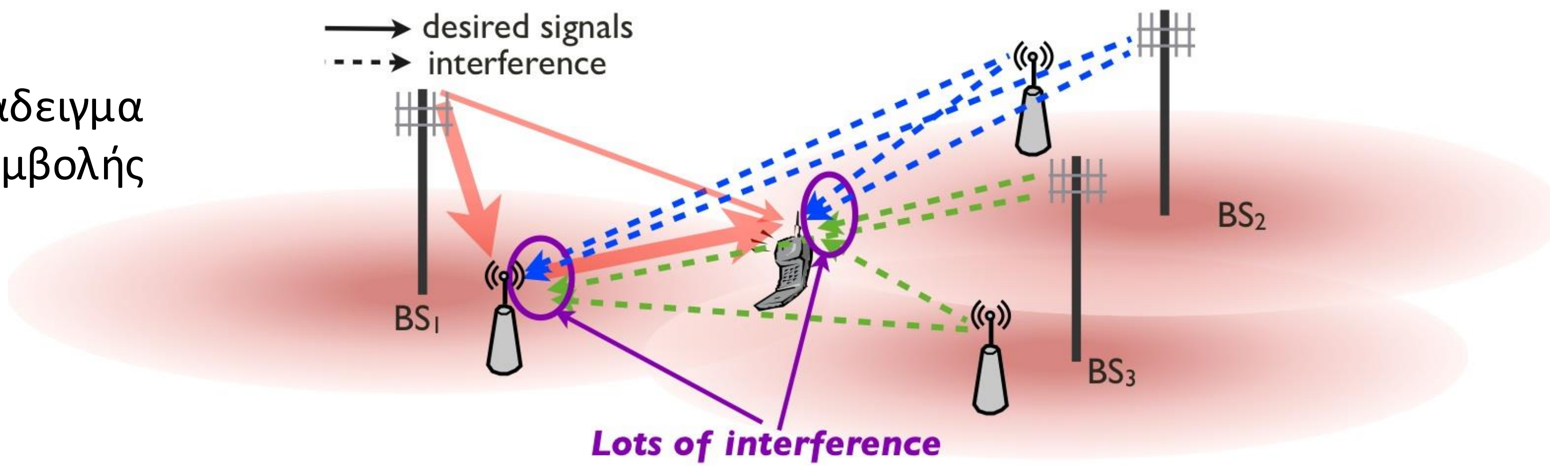
Από δω και στο εξής, για το xyx χρησιμοποιείται ένα σύμβολο και όχι τρία

Σφάλματα bit κατά την μεταφορά από σημείο Α σε σημείο Β

Προκαλούνται κατά την μετάδοση πληροφορίας στο κανάλι ή στα ηλεκτρονικά κυκλώματα του δέκτη Β

- Μέσο διάδοσης (κανάλι): αέρας στις ασύρματες επικοινωνίες, οπτική ίνα ή σύρμα στις ενσύρματες
  - Παρεμβολή (στις ασύρματες επικοινωνίες) λόγω γειτονικών μεταδόσεων
  - Εξασθένιση σήματος με την απόσταση
- Σφάλμα στα ηλεκτρονικά κυκλώματα του δέκτη
  - Θερμικός θόρυβος (thermal noise): Λόγω παλινδρομικής κίνησης των ηλεκτρονίων στο εναλλασσόμενο ρεύμα

Παράδειγμα παρεμβολής

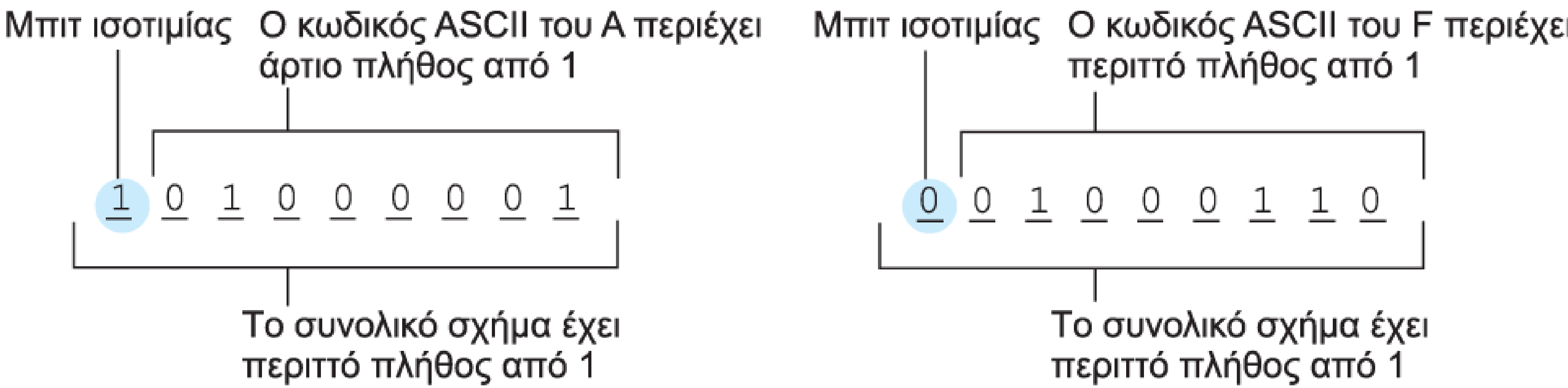


Σφάλματα bit κατά την μεταφορά από σημείο Α σε σημείο Β

- Κατά τη διάρκεια της μεταφοράς του από το Α στο Β ένα bit 0 μπορεί να αλλάξει σε 1 ή από 1 σε 0
  - Ο δέκτης θα το λάβει λάθος!
- Ο δέκτης πρέπει να κάνει:
  - Ανίχνευση σφάλματος
  - Διόρθωση σφάλματος
- Για να γίνει αυτό, πρέπει το σήμα πληροφορίας (bits) να ΠΡΟΣΤΑΤΕΥΤΕΙ εκ των προτέρων πριν την μετάδοσή του από τον πομπό

Κωδικοποίηση Καναλιού για ανίχνευση σφαλμάτων: Κώδικας ισοτιμίας (parity code)

Ρύθμιση εκ των προτέρων σε περιττή ισοτιμία



Ο κώδικας είναι το επιπλέον ψηφίο που εισάγεται στην αρχή (στον πομπό)

Αν στον δέκτη βρεθεί άρτιος αριθμός από 1's, τότε έχει συμβεί σφάλμα!!

Η χρήση των μπιτ ισοτιμίας χρησιμοποιείται και στην αποθήκευση δεδομένων

Κώδικας Ισοτιμίας

- Μετάδοση του F : 0 0 1 0 0 0 1 1 0
- Έστω στο δέκτη : 0 0 1 0 1 0 1 1 0 (ένα σφάλμα)
- Επειδή ο αριθμός των 1 είναι ζυγός, ο δέκτης καταλαβαίνει ότι έχει συμβεί σφάλμα

Ερώτηση: Τι είδους σφάλματα ανιχνεύονται με τον παραπάνω κώδικα;
Απάντηση: Περιττό πλήθος σφαλμάτων

Ερώτηση: Τι είδους σφάλματα ΔΕΝ ανιχνεύονται;
Απάντηση: Άρτιο πλήθος σφαλμάτων
Εναλλακτικά: Χρήση πολλών μπιτ ισοτιμίας (checkbyte)

Ερώτηση: Πώς γνωρίζουμε που έγινε το σφάλμα για να το διορθώσουμε;

Κώδικες Διόρθωσης Σφαλμάτων

- Αφού ανιχνευτεί ένα σφάλμα, πρέπει να διορθωθεί
- Κώδικες Διόρθωσης Σφαλμάτων: Error Correcting Code (ECC) ή Forward Error Correcting Code (FEC)
- Μπλοκ κώδικες διόρθωσης σφαλμάτων:
  - Κώδικας Hamming
  - Κώδικας Reed-Solomon
  - Κώδικες Turbo και κώδικες LDPC (Low-Density Parity Check)

Ένας απλός Κώδικας διόρθωσης σφαλμάτων: Κώδικας Hamming

Σύμβολο	Κώδικας
A	000000
B	001111
C	010011
D	011100
E	100110
F	101001
G	110101
H	111010

- Χρήσιμος ορισμός: Απόσταση Hamming (Hamming Distance) μεταξύ δυο ακολουθιών bits = αριθμός των bits που διαφέρουν
- Στο παράδειγμα, η απόσταση Hamming κάθε ζευγαριού είναι τουλάχιστον 3.

Κώδικας Hamming: Πως αποφασίζει ο δέκτης ποιο μήνυμα στάλθηκε;

Αποκωδικοποίηση με βάση την μικρότερη απόσταση (minimum-distance decoding)

Χαρα-κτήρας	Κώδικας	Σχήμα που λαμβάνεται	Απόσταση μεταξύ σχήματος και κωδικού
A	0 0 0 0 0 0	0 **1** 0 **1** 0 0	2
B	0 0 1 1 1 1	0 **1** **0** 1 **0** **0**	4
C	0 1 0 0 1 1	0 1 0 **1** **0** **0**	3
D	0 1 1 1 0 0	0 1 **0** 1 0 0	**1**
E	1 0 0 1 1 0	**0** **1** 0 1 **0** 0	3
F	1 0 1 0 0 1	**0** **1** **0** **1** **0** 0	5
G	1 1 0 1 0 1	**0** 1 0 1 0 **0**	2
H	1 1 1 0 1 0	**0** 1 **0** **1** **0** 0	4

Μικρότερη απόσταση

- Ο κλάδος των Ψηφιακών Επικοινωνιών μελετά σχετικά θέματα
- Error Detection and Correction

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

Τέλος Κεφαλαίου 1

- Είδαμε πώς αποθηκεύονται τα δεδομένα, τρόπους κωδικοποίησης-συμπίεσης και αντιμετώπιση σφαλμάτων

- Στο επόμενο κεφάλαιο θα ασχοληθούμε με τον τρόπο που η μηχανή χειρίζεται το δεδομένα και τις πληροφορίες που είναι αποθηκευμένες.