Παράσταση αριθμών κινητής υποδιαστολής

Στην αναπαράσταση αριθμών με ακέραιο και κλασματικό μέρος που είδαμε στις προηγούμενες ενότητες, η θέση της υποδιαστολής είναι προκαθορισμένη και σταθερή. Οι αριθμοί που ακολουθούν αυτή την αναπαράσταση χαρακτηρίζονται ως **αριθμοί σταθερής υποδιαστολής**.

Το εύρος τιμών που μπορεί να εκφραστεί με αυτόν τον τρόπο αναπαράστασης είναι σχετικά μικρό. Για παράδειγμα, ο μεγαλύτερος ακέραιος μη-προσημασμένος αριθμός που μπορεί να παρασταθεί με 32 δυαδικά ψηφία (bits) είναι:

$$2^{32} – 1 \approx 4.3 \times 10^{9}.$$

Όμως στους επιστημονικούς υπολογισμούς χρησιμοποιούμαι αριθμούς πολύ μεγαλύτερης τάξης (π.χ. αριθμός Avogadro = 6.022×10^{23} mole^{-1}), καθώς και αριθμούς με πολύ μικρό κλασματικό μέρος (π.χ. σταθερά Boltzmann = 1.38×10^{-23} Joule/^{o}K), για την αναπαράσταση των οποίων θα απαιτούνταν πολύ μεγάλο πλήθος δυαδικών ψηφίων (bits).

Για να καλυφθεί αυτή η ανάγκη, στα ψηφιακά υπολογιστικά συστήματα έχει υιοθετηθεί η παράσταση **αριθμών κινητής υποδιαστολής**, στην οποία η θέση της υποδιαστολής των αριθμών δεν είναι προκαθορισμένη αλλά μετακινούμενη, για να μπορεί να προσαρμόζεται στις υπολογιστικές ανάγκες.

Για την αναπαράσταση αριθμών κινητής υποδιαστολής έχει επικρατήσει το πρότυπο του IEEE (***I****nstitute of* ***E****lectrical and* ***E****lectronic* ***E****ngineering*).

Σύμφωνα με αυτό το πρότυπο, οι αριθμοί κινητής υποδιαστολής αναπαριστάνονται με 32 δυαδικά ψηφία (**παράσταση απλής ακρίβειας**) ή με 64 δυαδικά ψηφία (**παράσταση διπλής ακρίβειας**) και περιλαμβάνουν τρία τμήματα:

- το ψηφίο προσήμου **S** (που έχει τιμή ‘0’ για θετικούς και τιμή ‘1’ για αρνητικούς αριθμούς),
- τον εκθέτη **E**, και
- το κλασματικό μέρος **M** (*mantissa*).

Στην παράσταση απλής ακρίβειας (32 bits):

- ο **εκθέτης** εκφράζεται με **8 δυαδικά ψηφία** που ακολουθούν μετά το πρόσημο, και
- το **κλασματικό μέρος** εκφράζεται με τα **23** λιγότερο σημαντικά ψηφία.

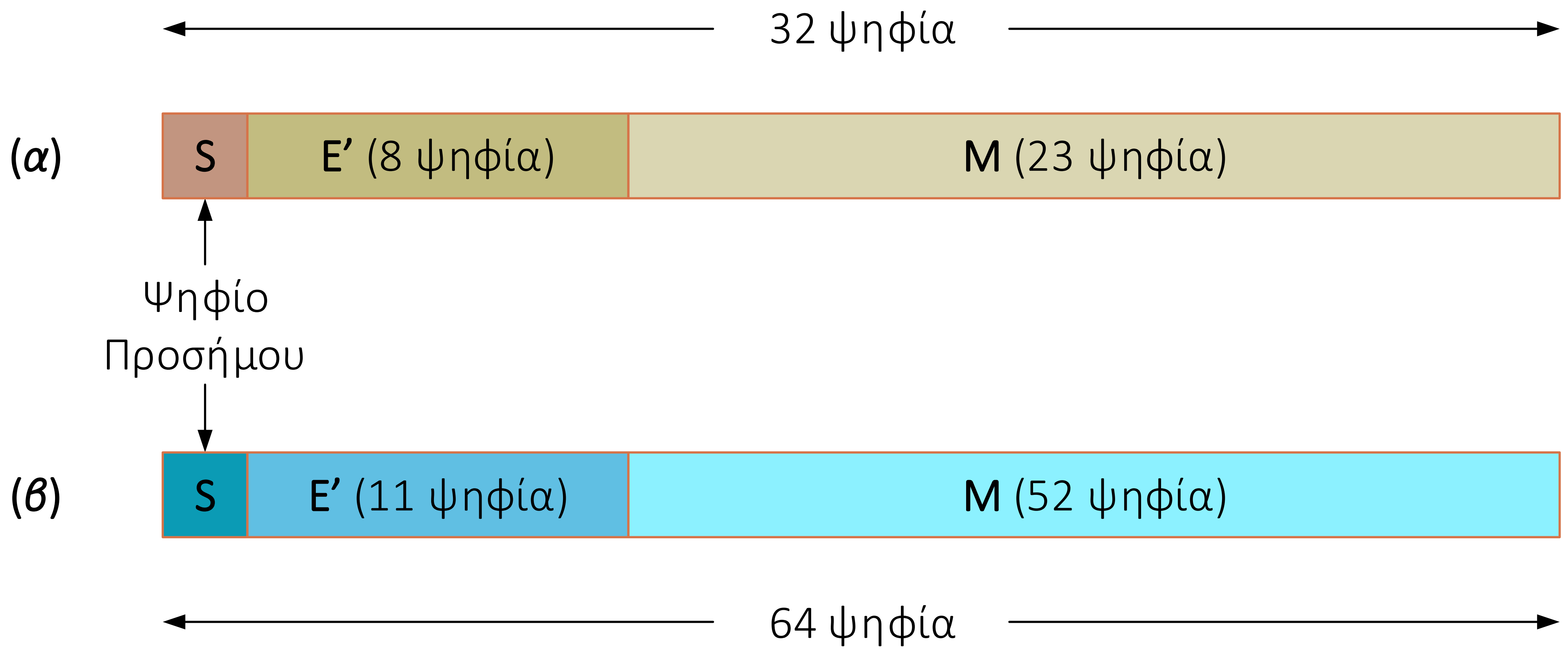
Στην πραγματικότητα, στην παράσταση αυτή, αντί για τον προσημασμένο εκθέτη **E**, χρησιμοποιείται ένας μη-προσημασμένος αριθμός **E'**, τέτοιος ώστε: **E'** = **E** + 127 και λαμβάνει τιμές από 0 έως 255.

Στην παράσταση διπλής ακρίβειας (64 bits):

- ο **εκθέτης** εκφράζεται με **11 δυαδικά ψηφία** που ακολουθούν μετά το πρόσημο, και
- το **κλασματικό μέρος** εκφράζεται με **52 δυαδικά ψηφία**.

Ο αριθμός **E'** είναι τέτοιος ώστε: **E'** = **E** + 1023 και λαμβάνει τιμές από 0 έως 2047.

Στο Σχήμα 2.1 παρουσιάζεται η δομή των αριθμών κινητής υποδιαστολή απλής και διπλής ακρίβειας, σύμφωνα με το πρότυπο του IEEE.



Σχήμα 2.1 Δομή των αριθμών κινητής υποδιαστολής (α) απλής και (β) διπλής ακρίβειας, σύμφωνα με το πρότυπο του IEEE

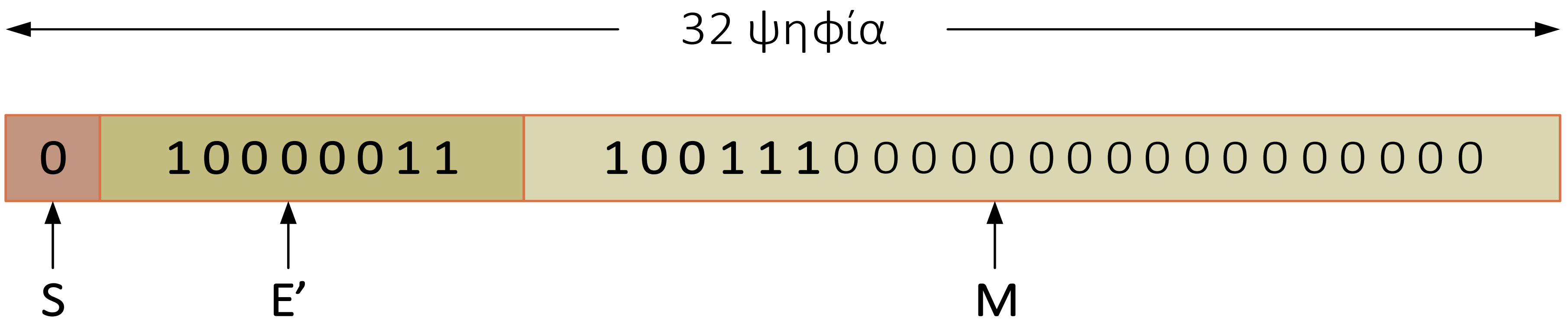
Για παράδειγμα, ας δούμε τον δεκαδικό αριθμό $\mathbf{(25.75)_{10}}$ σε παράσταση κινητής υποδιαστολής απλής ακρίβειας:

- Ο αριθμός είναι θετικός, επομένως **S = 0**.
- Ακολούθως προσδιορίζουμε τον αριθμό σε δυαδική μορφή:
  - Το ακέραιο μέρος του αριθμού σε δυαδική μορφή είναι $\mathbf{25_{10} = 11001_2}$
  - Το κλασματικό μέρος του αριθμού είναι $\mathbf{0.75_{10} = 0.11_2}$

  Άρα, ο ισοδύναμος δυαδικός αριθμός είναι $\mathbf{25.75_{10} = 11001.11_2}$
- Μετακινούμε την υποδιαστολή προς τα αριστερά, <u>αφήνοντας μόνο μια μονάδα αριστερά της</u>, δηλαδή, στην περίπτωσή μας, μετακινούμε την υποδιαστολή <u>4 bits προς τα αριστερά</u>.

  Επομένως ο δυαδικός αριθμός θα είναι $\mathbf{1.100111 \times 2^4}$
- Ο εκθέτης θα είναι **E = 4**, οπότε **E’ = E + 127 = 131**, δηλαδή: **E’ = 10000011**.
- <u>Το πρώτο bit</u>, αυτό που κατέχει τη θέση του ακέραιου μέρους αυτού του αριθμού και το οποίο έχει τιμή 1, <u>δεν αποθηκεύεται</u>.
- Τα υπόλοιπα bits αποτελούν το κλασματικό μέρος του αριθμού **M**, τη **mantissa**, που συμπληρώνεται με τόσα μηδενικά προς τα δεξιά, ώστε να έχουμε συνολικά 23 bits.

Στην περίπτωσή μας θα είναι **M = 100111**00000000000000000 (συμπληρώνουμε 17 μηδενικά).



Σχήμα 2.2 Αναπαράσταση του αριθμού $(25.75)_{10}$ *στο πρότυπο IEEE κινητής υποδιαστολής απλής ακρίβειας.*

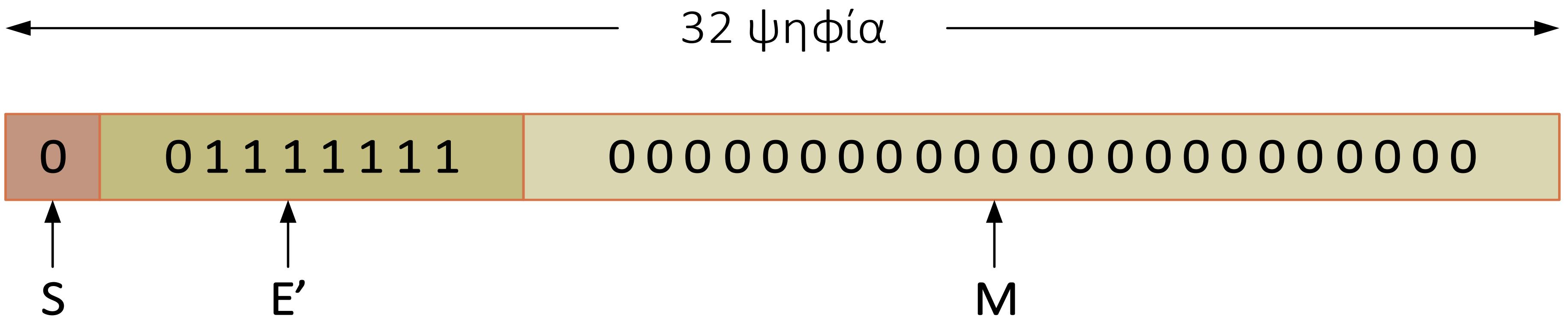
Επισημαίνεται ότι, το ψηφίο που βρίσκεται αριστερά της υποδιαστολής (**ακέραιο μέρος**) θα πρέπει να είναι **πάντοτε 1 και δεν αναπαριστάνεται**, αλλά θεωρείται ότι υπάρχει στη θέση αυτή.

Επομένως, οι αριθμοί που αναπαριστάνονται στο πρότυπο IEEE θα πρέπει να διαμορφώνονται με τέτοιο τρόπο ώστε **να υπάρχει μόνο μια μονάδα στη θέση αριστερά της υποδιαστολής**, δηλαδή, το ακέραιο μέρος να είναι πάντα ίσο με 1.

Η διαδικασία αυτή αναφέρεται ως **κανονικοποίηση**.

Σύμφωνα με τα παραπάνω βήματα, για να εκφράσουμε τον θετικό αριθμό $\mathbf{(1)_{10}}$ στο πρότυπο αυτό, θα πρέπει να γράψουμε τον αριθμό σε μορφή με ακέραιο και κλασματικό μέρος, για παράδειγμα, $\mathbf{(1.00)_{10}}$.

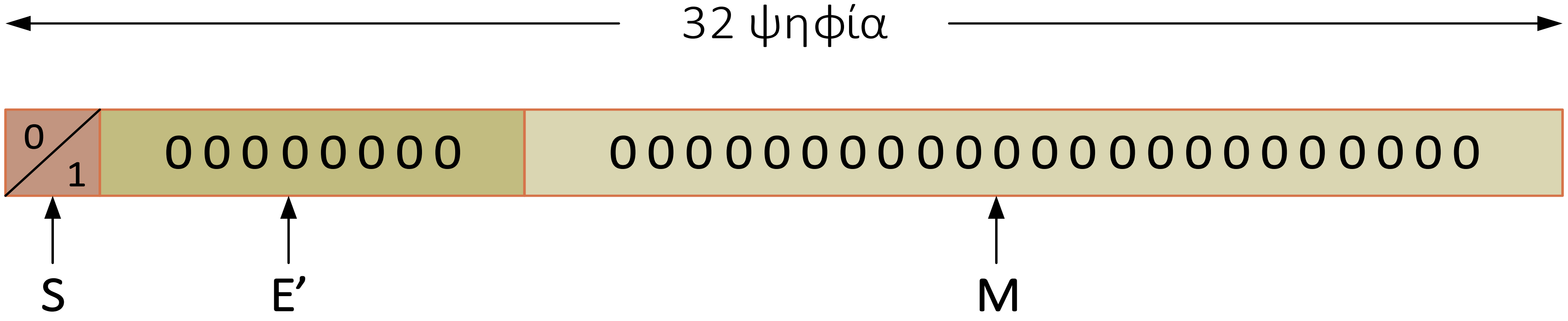
- το πεδίο προσήμου είναι **S = 0**,
- το πεδίο **M** (mantissa), που αντιστοιχεί στο κλασματικό μέρος του αριθμού, θα αναπαριστάνεται με **23 μηδενικά ψηφία**,
- η τιμή του εκθέτη είναι **E = 0** (αφού $1 = 1 \times 10^0$), οπότε θα είναι **E’** = E + 127 = 0 + 127 **= 127**,
- το πεδίο του εκθέτη θα έχει τιμή **01111111**.



Σχήμα 2.3 Αναπαράσταση του αριθμού $(1)_{10}$ στο πρότυπο IEEE κινητής υποδιαστολής απλής ακρίβειας.

Αν προσπαθήσουμε να αναπαραστήσουμε τον αριθμό **0** σύμφωνα με την παραπάνω διαδικασία, προκύπτει πρόβλημα, αφού θα έχουμε την ίδια ακριβώς αναπαράσταση όπως και για τον αριθμό **1**.

Για το λόγο αυτό, έχει καθοριστεί (:σύμβαση) στο πρότυπο παράστασης αριθμών κινητής υποδιαστολής του IEEE, **η τιμή του 0 να αναπαριστάνεται με τις τιμές E’ = 0 (8 bits) και M = 0 (23 bits)** , ενώ **το πρόσημο μπορεί να είναι 0 ή 1**, όπως φαίνεται στο Σχήμα 2.4.



Σχήμα 2.4 Αναπαράσταση του αριθμού 0 στο πρότυπο IEEE κινητής υποδιαστολής απλής ακρίβειας.

Υπάρχουν ακόμα δύο σημαντικές συμβάσεις για τις αναπαραστάσεις αριθμών κινητής υποδιαστολής στο πρότυπο IEEE.

Η πρώτη αφορά στην περίπτωση που είναι **E’ = 255** (και τα 8 ψηφία έχουν τιμή 1) και **η mantissa είναι διάφορη του 0 (M ≠ 0).**

Τότε, η αναπαριστώμενη τιμή θεωρείται ως **μη-αριθμός** (Not a Number – NaN) και αντιστοιχεί στο αποτέλεσμα μιας άκυρης πράξης (π.χ. η ρίζα ενός αρνητικού αριθμού).

Η δεύτερη αφορά στην περίπτωση που είναι **E’ = 255 και M = 0**. Η αναπαράσταση αυτή αντιστοιχεί στη **διαίρεση ενός αριθμού με το 0**, το αποτέλεσμα της οποίας είναι η τιμή του απείρου (**∞**).

Παρατηρούμε λοιπόν ότι **οι ακραίες τιμές του εκθέτη (0 ή 255) χρησιμοποιούνται** κατά σύμβαση στο πρότυπο IEEE **για την αναπαράσταση ειδικών τιμών**.

Επομένως, το πραγματικό εύρος του εκθέτη E’ για τις υπόλοιπες (εκτός των ακραίων) τιμές είναι από 1 έως και 254, δηλαδή **ο πραγματικός εκθέτης E λαμβάνει τιμές από -126 έως και 127.**

Όλα τα προαναφερθέντα για την αναπαράσταση αριθμών στο πρότυπο κινητής υποδιαστολής απλής ακρίβειας του IEEE, ισχύουν παρομοίως και στην αναπαράσταση διπλής ακρίβειας, με τις αντίστοιχες βεβαίως προσαρμογές ως πρός τις ακραίες τιμές του εκθέτη, που στην περίπτωση αυτή αποτελείται από 11 bits και λαμβάνει τιμές στο διάστημα 0 έως 2047.

Αριθμητικές πράξεις με αριθμούς σε παράσταση κινητής υποδιαστολής

Για να εκτελέσουμε την πρόσθεση ή την αφαίρεση δύο αριθμών εκφρασμένων σε παράσταση κινητής υποδιαστολής ακολουθούμε την εξής διαδικασία:

1. Θα πρέπει <u>και οι δύο αριθμοί να έχουν τον ίδιο εκθέτη</u>.
   - Αν έχουν διαφορετικό εκθέτη, υπολογίζουμε τη διαφορά των εκθετών των δύο αριθμών και μετατοπίζουμε προς τα δεξιά τον αριθμό με τον μικρότερο εκθέτη (το κλασματικό μέρος του <u>και το ψηφίο που βρίσκεται αριστερά της υποδιαστολής</u>) τόσες θέσεις όσες είναι η διαφορά.
   - Ως εκθέτης του αποτελέσματος λαμβάνεται ο μεγαλύτερος από τους εκθέτες των δύο αριθμών.

2. Ακολούθως, εκτελούμε την πρόσθεση ή αφαίρεση των κλασματικών μερών των δύο αριθμών και καθορίζουμε το πρόσημο του αθροίσματος ή της διαφοράς.

3. Τέλος, κανονικοποιούμε το αποτέλεσμα, εάν χρειάζεται.

Παράδειγμα 15: Δίνονται οι δεκαδικοί αριθμοί $\mathbf{A = (28.625)_{10}}$ και $\mathbf{B = (118.5)_{10}}$. Να εκφραστούν οι αριθμοί σε παράσταση κινητής υποδιαστολής απλής ακρίβειας και να υπολογιστούν το άθροισμά τους, **A + B**, και η διαφορά τους, **A – B**.

Στο πρώτο βήμα, εκφράζουμε τους αριθμούς σε παράσταση κινητής υποδιαστολής απλής ακρίβειας, σύμφωνα με τη διαδικασία που έχουμε ήδη περιγράψει:

$$\mathbf{A = (28.625)_{10} = (11100.101)_2}$$

Μετακινούμε την υποδιαστολή 4 θέσεις αριστερά, ώστε να έχουμε μόνο μια μονάδα αριστερά της (ακέραιο μέρος), οπότε θα έχουμε $\mathbf{A = 1.1100101 \times 2^4}$.

Επομένως, σε παράσταση κινητής υποδιαστολής απλής ακρίβειας, θα έχουμε:

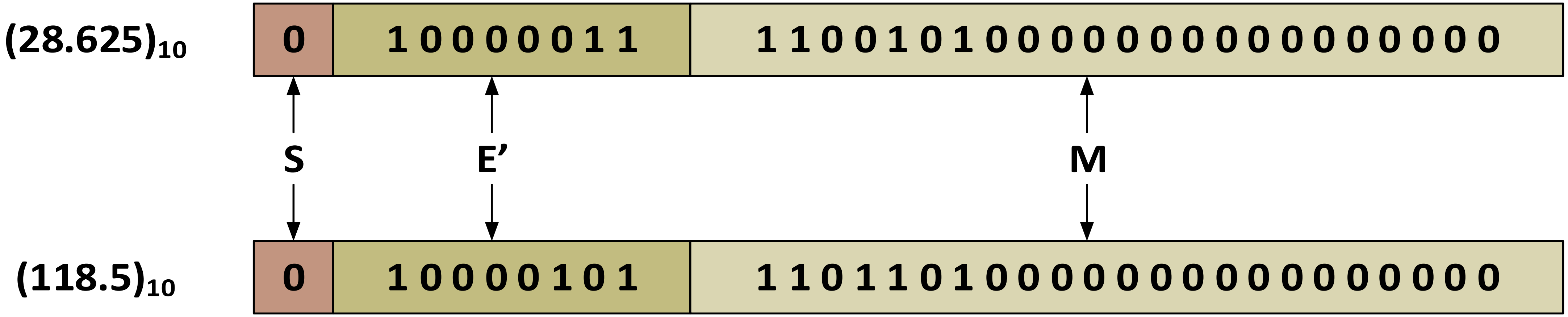
S = 0, **E = 4**, άρα **E’ = 4 + 127 = 131 = 10000011**, <u>το ψηφίο 1 αριστερά της υποδιαστολής δεν αποθηκεύεται</u> και **M = 11001010000000000000000**

(δεξιά από το κλασματικό μέρος του αριθμού, που αποτελείται από 7 ψηφία, προσθέτουμε 16 μηδενικά, ώστε να έχουμε συνολικά 23 ψηφία για να εκφράσουμε τη mantissa).

Ακολουθούμε την ίδια διαδικασία για τον αριθμό Β.

$B = (110.5)_{10} = (1110110.1)_2 = 1.1101101 \times 2^6$

Άρα, **S = 0, E = 6, E’ = 6 + 127 = 133 = 100000101, M = 11011010000000000000000**



Σχήμα 2.5 ***Αναπαράσταση των αριθμών $(28.625)_{10}$ και $(118.5)_{10}$ στο πρότυπο IEEE κινητής υποδιαστολής απλής ακρίβειας.***

Για να υπολογίσουμε το άθροισμα και τη διαφορά των δύο αριθμών θα πρέπει και **<u>οι δύο αριθμοί να έχουν τον ίδιο εκθέτη</u>**.

Επειδή ο αριθμός **Α έχει εκθέτη 4** και ο **Β έχει εκθέτη 6**, θα πρέπει να εκφραστεί και ο Α με τον μεγαλύτερο εκθέτη, δηλαδή 6.

Η διαφορά των εκθετών είναι **6 – 4 = 2**.

Άρα, θα πρέπει να μετατοπίσουμε κατά 2 θέσεις προς τα δεξιά τα ψηφία του αριθμού Α, οπότε θα έχουμε:

$$\mathbf{A = 1.1100101 \times 2^{4} = 0.011100101 \times 2^{6}}$$

Ακολούθως, εκτελούμε τις ζητούμενες αριθμητικές πράξεις.

<u>Πρόσθεση Α + Β :</u>

$$
\begin{array}{r}
0\,.\,0\,1\,1\,1\,0\,0\,1\,0\,1 \\
+\,1\,.\,1\,1\,0\,1\,1\,0\,1\,0\,0 \\
\hline
\underline{1}0\,.\,0\,1\,0\,0\,1\,1\,0\,0\,1
\end{array}
$$

Παρατηρούμε ότι, προκύπτει ψηφίο υπερχείλισης 1 και το αποτέλεσμα δεν είναι στην κανονική μορφή 1.xxxxxx.

Επομένως απαιτείται κανονικοποίηση του αποτελέσματος, δηλαδή θα πρέπει η μονάδα της υπερχείλισης να ληφθεί υπόψη και να αποτελέσει μέρος του αποτελέσματος.

Αυτό γίνεται μετατοπίζοντας την υποδιαστολή κατά μία θέση προς τα αριστερά, που ισοδυναμεί με την αύξηση της τιμής του εκθέτη κατά μία μονάδα, δηλαδή E= 6 + 1 =7, οπότε το τελικό αποτέλεσμα, που ήταν

$$\mathbf{10.010011001(\times 2^{6})}$$

θα γίνει

$$\mathbf{1.0010011001(\times 2^{7})}$$

και θα έχει

S = 0

E = 7

E’ = 7 + 127 = 134 = 100000110

M = 00100110010000000000000

$(28.625)_{10}$	0	1 0 0 0 0 0 1 1	1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ $(118.5)_{10}$	0	1 0 0 0 0 1 0 1	1 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
= $(147.125)_{10}$	0	1 0 0 0 0 1 1 0	0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

Σχήμα 2.6 ***Αναπαράσταση των αριθμών $(28.625)_{10}$, $(118.5)_{10}$ και του αθροίσματος αυτών $(147.125)_{10}$ στο πρότυπο IEEE κινητής υποδιαστολής απλής ακρίβειας.***

Αφαίρεση Α – Β :

Μετά την ολοκλήρωση του πρώτου βήματος της διαδικασίας που περιγράψαμε προηγουμένως (που είναι κοινό και για την πρόσθεση και για την αφαίρεση), στο δεύτερο βήμα, για την πράξη της αφαίρεσης, αφαιρούμε τα κλασματικά μέρη των δύο αριθμών.

Αυτό μπορεί να αναχθεί σε πρόσθεση του συμπληρώματος ως προς 2 του κλασματικού μέρους του αριθμού Β, στο κλασματικό μέρος του αριθμού Α.

Να σημειωθεί στο σημείο αυτό ότι, επειδή Β > Α, η διαφορά που προκύπτει θα είναι ένας αρνητικός αριθμός, οπότε **S = 1**.

Στο παράδειγμά μας, το κλασματικό μέρος του Β είναι 1101101, επομένως το συμπλήρωμά του ως προς 2 θα είναι:

$$\boldsymbol{\Sigma_2(1101101) = \Sigma_1(1101101) + 1 = 0010010 + 1 = 0010011}$$

Ακολούθως εκτελούμε την πρόσθεση του κλασματικού μέρους του αριθμού Α με το συμπλήρωμα ως προς 2 του κλασματικού μέρους του αριθμού Β:

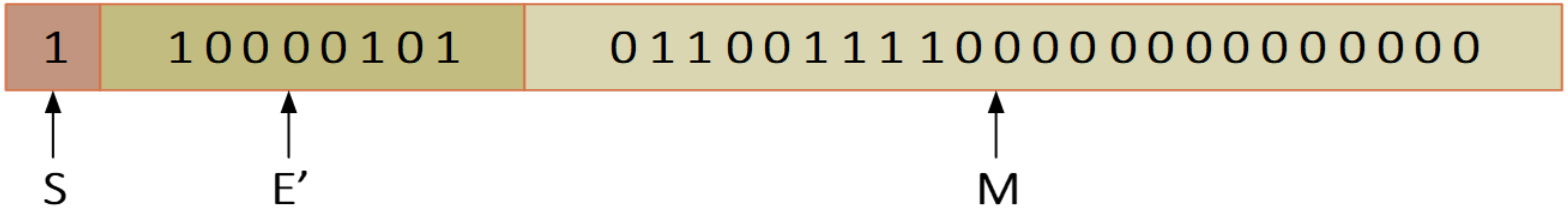
$$
\begin{array}{r}
0.011100101 \\
+\,0.001001100 \\
\hline
0.100110001
\end{array}
$$

Το αποτέλεσμα αυτό είναι εκφρασμένο σε μορφή συμπληρώματος ως προς 2, άρα, για να το πάρουμε σε κανονική μορφή, θα πρέπει να υπολογίσουμε το συμπλήρωμά του ως προς 2:

$$\Sigma_2(0.100110001) = \Sigma_1(0.100110001) + 1 = 1.011001110 + 1 = \mathbf{1.011001111}$$

Συνεπώς, η διαφορά που προκύπτει είναι **1.011001111** x $\mathbf{2^6}$ και αναπαριστάνεται ως αριθμός κινητής υποδιαστολής απλής ακρίβειας, όπως φαίνεται στο Σχήμα 1.7.

Η διαφορά των δύο αριθμών είναι: $\mathbf{A - B = (28.625)_{10} - (118.5)_{10} = (-\,89.875)_{10}}$ και μπορεί εύκολα να επαληθευτεί από την παράσταση αυτή.



Σχήμα 2.7 Αναπαράσταση του αριθμού $(-\,89.875)_{10}$ στο πρότυπο IEEE κινητής υποδιαστολής απλής ακρίβειας.

Για να εκτελέσουμε την πράξη του **πολλαπλασιασμού δύο αριθμών** εκφρασμένων σε παράσταση κινητής υποδιαστολής ακολουθούμε την εξής διαδικασία:

1. Προσθέτουμε τους εκθέτες των δύο αριθμών και από το άθροισμα που προκύπτει αφαιρούμε τον αριθμό 127, όταν πρόκειται για αριθμούς απλής ακρίβειας, ή τον αριθμό 1023, όταν πρόκειται για αριθμούς διπλής ακρίβειας.
2. Ακολούθως, εκτελούμε τον πολλαπλασιασμό των δύο αριθμών (το κλασματικό μέρος συν το ψηφίο που βρίσκεται αριστερά της υποδιαστολής), με τον ίδιο τρόπο όπως στους μη-προσημασμένους αριθμούς.
3. Στη συνέχεια, καθορίζουμε το πρόσημο του γινομένου.
4. Τέλος, κανονικοποιούμε το αποτέλεσμα, εάν χρειάζεται.

Αντίστοιχα, για να εκτελέσουμε την πράξη της **διαίρεσης δύο αριθμών** εκφρασμένων σε παράσταση κινητής υποδιαστολής ακολουθούμε την εξής διαδικασία:

1. Αφαιρούμε τους εκθέτες των δύο αριθμών και στη διαφορά που προκύπτει προσθέτουμε τον αριθμό 127, όταν πρόκειται για αριθμούς απλής ακρίβειας, ή τον αριθμό 1023, όταν πρόκειται για αριθμούς διπλής ακρίβειας.

2. Ακολούθως, εκτελούμε τη διαίρεση των δύο αριθμών (το κλασματικό μέρος συν το ψηφίο που βρίσκεται αριστερά της υποδιαστολής), με τον ίδιο τρόπο όπως στους μη-προσημασμένους αριθμούς.

3. Στη συνέχεια, καθορίζουμε το πρόσημο του πηλίκου.

4. Τέλος, κανονικοποιούμε το αποτέλεσμα, εάν χρειάζεται.

Δυαδικοί κώδικες

Στα ψηφιακά συστήματα, εκτός από τους δυαδικούς αριθμούς, αναπαριστάνονται, επεξεργάζονται, αποθηκεύονται και μεταδίδονται πολλά διακριτά στοιχεία πληροφορίας, όπως γράμματα, αριθμοί, χαρακτήρες, σύμβολα κλπ.

Κάθε διακριτό στοιχείο πληροφορίας μπορεί να παρασταθεί με τη χρήση ενός δυαδικού κώδικα, δηλαδή, με τη μορφή μιας μοναδικής ακολουθίας δυαδικών ψηφίων 0 και 1, για κάθε διακριτό στοιχείο πληροφορίας.

Ένας δυαδικός κώδικας με ***n*** το πλήθος bits είναι μια ομάδα από 2^n το πλήθος διαφορετικές μεταξύ τους ακολουθίες των ***n*** bits, δηλαδή η κάθε μια από αυτές τις ακολουθίες είναι ένας διαφορετικός συνδυασμός (διάταξη) των ***n*** bits.

Ο κάθε συνδυασμός παριστάνει ένα και μοναδικό (διακριτό) στοιχείο της πληροφορίας που κωδικοποιείται.

Απαγορεύεται η χρήση του ίδιου συνδυασμού bits για περισσότερα από ένα στοιχεία, γιατί σε τέτοια περίπτωση θα είχαμε έναν ασαφώς ορισμένο κώδικα.

Αν και ο ελάχιστος αριθμός bits που απαιτείται για την κωδικοποίηση $\boldsymbol{2^n}$ διακριτών στοιχείων είναι $\boldsymbol{n}$, δεν υπάρχει μέγιστος αριθμός bits που μπορούν να χρησιμοποιηθούν για τη δημιουργία ενός δυαδικού κώδικα.

Για παράδειγμα, μπορούμε να θεωρήσουμε ένα σύστημα κωδικοποίησης στο οποίο, τα 10 ψηφία του δεκαδικού αριθμητικού συστήματος (0, 1, 2, ..., 9) κωδικοποιούνται με 10 bits το καθένα, δηλαδή κάθε δεκαδικό ψηφίο αναπαριστάνεται με ένα συγκεκριμένο συνδυασμό από δέκα δυαδικά ψηφία 0 και 1.

Σε ένα τέτοιου τύπου δυαδικό κώδικα, το δεκαδικό ψηφίο 6 θα μπορούσε να αντιστοιχεί στο συνδυασμό bits 0001000000, το ψηφίο 3 στο συνδυασμό 0000001000 και το ψηφίο 0 στο συνδυασμό 0000000001.

Οι τρεις κύριοι τύποι δυαδικών κωδίκων είναι οι αριθμητικοί, οι αλφαριθμητικοί και οι κώδικες ανίχνευσης και διόρθωσης σφαλμάτων.

Αριθμητικός κώδικας BCD

Ο αριθμητικός κώδικας BCD, **B**inary **C**oded **D**ecimal, μετάφραση: **δυαδικά κωδικοποιημένοι δεκαδικοί**, είναι ένας κώδικας στον οποίο κάθε ψηφίο ενός δεκαδικού αριθμού εκφράζεται ξεχωριστά σε δυαδική μορφή.

Όπως θα δούμε στη συνέχεια, η παράσταση ενός δεκαδικού αριθμού σε κώδικα BCD απαιτεί περισσότερα δυαδικά ψηφία από την αντίστοιχη δυαδική παράσταση.

Το πλεονέκτημα, ωστόσο, της κωδικοποίησης αυτής είναι η ευχέρεια με την οποία ο άνθρωπος μπορεί να χρησιμοποιεί τους δεκαδικούς αριθμούς λόγω της εξοικείωσής του με αυτούς, σε συνδυασμό με τη χρήση του δυαδικού συστήματος που χρησιμοποιείται στα ψηφιακά συστήματα.

Στα ψηφιακά υπολογιστικά συστήματα εισάγουμε (: πληκτρολογούμε) δεκαδικούς αριθμούς, αυτοί μετατρέπονται σε δυαδική μορφή, εκτελούνται οι αριθμητικοί υπολογισμοί με δυαδικό τρόπο και τα αποτελέσματα μετατρέπονται και λαμβάνονται σε δεκαδική μορφή.

Επειδή το μεγαλύτερο δεκαδικό ψηφίο (9), εκφράζεται με 4 δυαδικά ψηφία (1001), όλα τα δεκαδικά ψηφία θα εκφράζονται επίσης με 4 δυαδικά ψηφία, όπως παρουσιάζεται στον πίνακα που ακολουθεί.

Κώδικας BCD	
Δεκαδικό ψηφίο	**Κωδικοποίηση BCD**
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Παρατηρούμε ότι, από τους 16 δυνατούς συνδυασμούς που παράγονται από 4 δυαδικά ψηφία, χρησιμοποιούνται μόνο οι 10 πρώτοι, από 0000 έως και 1001, ενώ οι υπόλοιποι 6 συνδυασμοί από 1010 έως και 1111, δεν χρησιμοποιούνται στην κωδικοποίηση BCD και, επομένως, δεν έχουν νόημα στον κώδικα αυτό.

Ένας δεκαδικός αριθμός με ***n*** ψηφία, σε BCD κωδικοποίηση εκφράζεται με ***n*** x ***4*** δυαδικά ψηφία, δηλαδή, μια τετράδα δυαδικών ψηφίων για κάθε δεκαδικό ψηφίο. Για παράδειγμα, ο δεκαδικός αριθμός 295_{10}, σε δυαδική παράσταση και σε BCD κωδικοποίηση εκφράζεται ως:

$$\mathbf{(295)_{10} = (100100111)_2 = (\ 0010\ 1001\ 0101\ )_{BCD}.}$$

Παρατηρούμε ότι, η παράσταση BCD απαιτεί 12 bits, ενώ ο ισοδύναμος δυαδικός αριθμός απαιτεί 9 bits. Γενικά, η παράσταση ενός δεκαδικού αριθμού σε κωδικοποίηση BCD απαιτεί περισσότερα δυαδικά ψηφία από την αντίστοιχη δυαδική παράσταση.

Είναι σημαντικό να επισημάνουμε ότι **οι αριθμοί BCD** αναπαριστούν δεκαδικούς αριθμούς και **δεν είναι δυαδικοί αριθμοί**. Ωστόσο, επειδή εκφράζονται σε δυαδική μορφή, μπορούμε να εφαρμόζουμε τους κανόνες που ισχύουν στο δυαδικό αριθμητικό σύστημα, λαμβάνοντας φυσικά υπόψη μας τους περιορισμούς που τίθενται από τον τρόπο που γίνεται η κωδικοποίηση.

Για παράδειγμα, κατά την πρόσθεση δύο δεκαδικών ψηφίων κωδικοποιημένων κατά BCD, όταν το δυαδικό άθροισμα είναι ίσο ή μικρότερο του 1001 (δηλαδή του 9), τότε ταυτίζεται με το άθροισμα κατά BCD.

Στην περίπτωση όμως που το δυαδικό άθροισμα είναι μεγαλύτερο του 1001 (και προφανώς δεν περιλαμβάνεται στον κώδικα BCD), για να ληφθεί το σωστό άθροισμα κατά BCD, θα πρέπει να προσθέσουμε στο δυαδικό άθροισμα τον αριθμό 0110 (= 6_{10}), που είναι ίσο με το πλήθος των μη χρησιμοποιούμενων στον κώδικα BCD συνδυασμών των τεσσάρων δυαδικών ψηφίων.

Αυτό θα έχει αποτέλεσμα τη δημιουργία κρατούμενου, το οποίο θα πρέπει να ληφθεί υπόψη κατά την πρόσθεση των κωδικοποιημένων κατά BCD ψηφίων της αμέσως επόμενης πιο σημαντικής θέσης.

Η διαδικασία αυτή λέγεται **διόρθωση κρατούμενου**.

Παράδειγμα 16. Να εκφραστούν οι δεκαδικοί αριθμοί 53_{10} και 38_{10} σε κώδικα BCD και να εκτελεστεί η πράξη της πρόσθεσης.

Η BCD κωδικοποίηση των δύο αριθμών είναι:

53_{10} = 0101 0011   και 38_{10} = 0011 1000

Η πρόσθεση των δυαδικών ψηφίων γίνεται σύμφωνα με τους κανόνες του δυαδικού αριθμητικού συστήματος. Επομένως:

	0	1	0	1		0	0	1	1	←	53_{10}
+	0	0	1	1	+	1	0	0	0	←	38_{10}
	1	0	0	0		1	0	1	1	←	Μη αποδεκτή παράσταση BCD
+				**1**	+	0	1	1	0	←	$+ 6_{10}$
	1	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**1**	←	$\mathbf{91_{10}}$

Αριθμητικός κώδικας GRAY

Ο κώδικας Gray είναι ένας κώδικας αναπαράστασης αριθμών με δυαδικά ψηφία κατά τέτοιο τρόπο ώστε, κατά τη μετάβαση μεταξύ δύο διαδοχικών αριθμών να αλλάζει μόνο ένα ψηφίο.

Ο κώδικας Gray χρησιμοποιείται σε εφαρμογές όπου η κανονική ακολουθία δυαδικών αριθμών μπορεί να παραγάγει ένα σφάλμα κατά τη μετάβαση μεταξύ δύο διαδοχικών αριθμών.

Ας θεωρήσουμε, για παράδειγμα, τη μετάβαση από τον αριθμό **0111** (7_{10}) στον επόμενο **1000** (8_{10}) μιας κανονικής δυαδικής ακολουθίας, κατά την οποία, εάν η αλλαγή τιμής του λιγότερο σημαντικού ψηφίου γίνει καθυστερημένα, θα προκύψει προσωρινά η λανθασμένη τιμή **1001** (9_{10}).

Με χρήση της ακολουθίας του κώδικα Gray αυτό αντιμετωπίζεται, αφού αλλάζει η τιμή μόνο ενός ψηφίου κατά τη μετάβαση μεταξύ διαδοχικών αριθμών.

Ο κώδικας Gray για τέσσερα δυαδικά ψηφία, παρουσιάζεται στον πίνακα που ακολουθεί.

Δεκαδικό ψηφίο	Κωδικοποίηση GRAY			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Ένας πρακτικός τρόπος συμπλήρωσης του παραπάνω πίνακα στηρίζεται στην “ανακλαστική” ιδιότητα:

- Κατά τη δημιουργία της πρώτης από δεξιά στήλης ξεκινάμε με ένα μηδενικό και μία μονάδα,

Δεκαδικό ψηφίο	**Κωδικοποίηση GRAY**
0	0
1	1
2	1
3	0
4	0
5	1
6	1
7	0
8	0
9	1
10	1
11	0
12	0
13	1
14	1
15	0

Ένας πρακτικός τρόπος συμπλήρωσης του παραπάνω πίνακα στηρίζεται στην “ανακλαστική” ιδιότητα:

- Κατά τη δημιουργία της πρώτης από δεξιά στήλης ξεκινάμε με ένα μηδενικό και μία μονάδα,
- κατά τη δημιουργία της δεύτερης στήλης ξεκινάμε με δύο μηδενικά και δύο μονάδες,

Δεκαδικό ψηφίο	**Κωδικοποίηση GRAY**	
0	0	0
1	0	1
2	1	1
3	1	0
4	1	0
5	1	1
6	0	1
7	0	0
8	0	0
9	0	1
10	1	1
11	1	0
12	1	0
13	1	1
14	0	1
15	0	0

Ένας πρακτικός τρόπος συμπλήρωσης του παραπάνω πίνακα στηρίζεται στην "ανακλαστική" ιδιότητα:

- Κατά τη δημιουργία της πρώτης από δεξιά στήλης ξεκινάμε με ένα μηδενικό και μία μονάδα,
- κατά τη δημιουργία της δεύτερης στήλης ξεκινάμε με δύο μηδενικά και δύο μονάδες,
- κατά τη δημιουργία της τρίτης στήλης με τέσσερα μηδενικά και τέσσερις μονάδες κ.ο.κ.

Δεκαδικό ψηφίο	**Κωδικοποίηση GRAY**		
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0
8	1	0	0
9	1	0	1
10	1	1	1
11	1	1	0
12	0	1	0
13	0	1	1
14	0	0	1
15	0	0	0

Ένας πρακτικός τρόπος συμπλήρωσης του παραπάνω πίνακα στηρίζεται στην “ανακλαστική” ιδιότητα:

- Κατά τη δημιουργία της πρώτης από δεξιά στήλης ξεκινάμε με ένα μηδενικό και μία μονάδα,
- κατά τη δημιουργία της δεύτερης στήλης ξεκινάμε με δύο μηδενικά και δύο μονάδες,
- κατά τη δημιουργία της τρίτης στήλης με τέσσερα μηδενικά και τέσσερις μονάδες,
- κατά τη δημιουργία της τέταρτης στήλης με οκτώ μηδενικά και οκτώ μονάδες.

Δεκαδικό ψηφίο	Κωδικοποίηση GRAY			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Όπως θα δούμε αργότερα, η παραγωγή του κώδικα Gray είναι μια εύκολη διαδικασία με εφαρμογή της λογικής πράξης ΑΠΟΚΛΕΙΣΤΙΚΟ ‘Η (EXCLUSIVE OR – XOR, σύμβολο: $\oplus$), ακολουθώντας τα εξής βήματα:

1. Εκφράζουμε τον δεκαδικό αριθμό σε δυαδική μορφή.
2. Το πλέον αριστερό ψηφίο (μέγιστο σημαντικό ψηφίο) παραμένει το ίδιο, ενώ τα ψηφία μιας οποιασδήποτε άλλης θέσης ***n*** προκύπτουν από την πράξη **XOR** του ψηφίου αυτού με το ψηφίο που βρίσκεται στην αμέσως πιο σημαντική θέση ***n***+***1*** :

$$A_n(Gray) = A_{n+1}(Bin) \oplus A_n(Bin).$$

Αλφαριθμητικός κώδικας ASCII

Η επεξεργασία οποιαδήποτε πληροφορίας στα ψηφιακά συστήματα πραγματοποιείται αποκλειστικά με τη χρήση των δυαδικών ψηφίων 0 και 1.

Για το λόγο αυτό προέκυψε η ανάγκη κωδικοποίησης (: αναπαράστασης) χαρακτήρων, όπως τα γράμματα και τα σύμβολα, αλλά και "εντολών" για την εκτέλεση συγκεκριμένων ενεργειών, όπως, για παράδειγμα, την αλλαγή γραμμής ή παραγράφου σε ένα κείμενο, ως μία ακολουθία (συνδυασμό) συγκεκριμένου πλήθους ψηφίων 0 και 1.

Η πλέον χρησιμοποιούμενη κωδικοποίηση είναι η ASCII (American Standard Code for Information Interchange), ένας δυαδικός κώδικας αλφαριθμητικών χαρακτήρων που **χρησιμοποιεί 7 δυαδικά ψηφία** για την **αναπαράσταση 128 χαρακτήρων**, 94 εκτυπώσιμων (26 κεφαλαία και 26 μικρά λατινικά γράμματα, τα 10 δεκαδικά ψηφία και 32 ειδικά σύμβολα), καθώς και 34 μη εκτυπώσιμων (για πράξεις ελέγχου).

Για παράδειγμα, σε κωδικοποίηση ASCII έχουμε τις ακόλουθες αναπαραστάσεις:

- η λέξη yes! (με το θαυμαστικό): 1111001 1100101 1110011 0100001
- το δεκαδικό ψηφίο 7 σαν χαρακτήρας (όχι ο αριθμός): 0110111
- η αλλαγή παραγράφου (Enter/CR – carriage return): 0001101
- η διαγραφή ενός χαρακτήρα (DEL – delete): 1111111

Στους δύο πίνακες που ακολουθούν παρουσιάζεται η κωδικοποίηση ASCII για τους 94 εκτυπώσιμους χαρακτήρες και τους χαρακτήρες ελέγχου αντίστοιχα.

$b_4 b_3 b_2 b_1$	$b_7 b_6 b_5$	$b_7 b_6 b_5$	$b_7 b_6 b_5$	$b_7 b_6 b_5$	$b_7 b_6 b_5$	$b_7 b_6 b_5$	$b_7 b_6 b_5$	$b_7 b_6 b_5$
	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	STX	DC2	“	2	B	R	b	r
0 0 1 1	ETX	DC3	#	3	C	S	c	s
0 1 0 0	EOT	DC4	$	4	D	T	d	t
0 1 0 1	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	ACK	SYN	&	6	F	V	f	v
0 1 1 1	BEL	ETB	'	7	G	W	g	w
1 0 0 0	BS	CAN	(	8	H	X	h	x
1 0 0 1	HT	EM	)	9	I	Y	i	y
1 0 1 0	LF	SUB	*	:	J	Z	g	z
1 0 1 1	VT	ESC	+	;	K	[	k	{
1 1 0 0	FF	FS	,	<	L	\	l	|
1 1 0 1	CR	GS	–	=	M	]	m	}
1 1 1 0	SO	RS	.	>	N	^	n	~
1 1 1 1	SI	US	/	?	O	_	o	DEL

Χαρακτήρες ελέγχου			
NUL	Null	DLE	Data-link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronous idle
BEL	Bell	ETB	End-of-transmission block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
SP	Space	DEL	Delete

Ο Ελληνικός Οργανισμός Τυποποίησης (ΕΛΟΤ) έχει αναπτύξει τον κώδικα **ΕΛΟΤ-928** (επέκταση του ASCII), ο οποίος **χρησιμοποιεί 8 δυαδικά ψηφία** και παρέχει **ενιαία κωδικοποίηση λατινικών και ελληνικών χαρακτήρων**.

Έως σήμερα έχουν γίνει πολλές προσπάθειες ανάπτυξης κωδίκων αλφαριθμητικών χαρακτήρων, με σημαντικότερη αυτήν της κοινοπραξίας **Unicode**, η οποία αναπτύσσει το ομώνυμο πρότυπο, για την κωδικοποίηση των χαρακτήρων όλων των υπαρχόντων συστημάτων γραφής, χρησιμοποιώντας έως και **32 δυαδικά ψηφία ανά χαρακτήρα**.

Κώδικες ανίχνευσης σφαλμάτων

Κατά τη μετάδοση ή την επεξεργασία ψηφιακής πληροφορίας είναι πιθανό, λόγω ηλεκτρονικού "**θορύβου**", δηλαδή ανεπιθύμητων διακυμάνσεων των ψηφιακών σημάτων, να συμβεί αλλοίωση της τιμής ενός ή περισσότερων δυαδικών ψηφίων.

Σφάλματα τέτοιου είδους είναι δυνατό να εντοπισθούν και να διορθωθούν χρησιμοποιώντας **κώδικες ανίχνευσης-μόνο** ή **ανίχνευσης και διόρθωσης σφαλμάτων.**

Ένας τρόπος ανίχνευσης-μόνο των σφαλμάτων είναι η προσθήκη ενός επιπλέον ψηφίου (**ψηφίο ισοτιμίας, parity bit**) σε κάθε κωδικοποιημένο χαρακτήρα ή αριθμό, έτσι ώστε **το πλήθος των ψηφίων 1 που περιέχονται σε αυτόν να είναι περιττό ή άρτιο**, δηλαδή να δημιουργείται κώδικας με **περιττή ή άρτια ισοτιμία**, αντίστοιχα.

Στα συστήματα που χρησιμοποιούν κωδικοποίηση ASCII (στην οποία, όπως είδαμε, χρησιμοποιούνται 7 δυαδικά ψηφία για την αναπαράσταση 128 χαρακτήρων), αυτό γίνεται με την προσθήκη ενός επιπλέον ψηφίου στην πιο σημαντική θέση κάθε κωδικοποιημένου χαρακτήρα.

Αυτό **το όγδοο ψηφίο** ονομάζεται **ψηφίο ισοτιμίας** (parity bit).

Ο **έλεγχος ισοτιμίας** (parity check) είναι ένας τρόπος **ανίχνευσης-μόνο** σφαλμάτων.

Το ψηφίο ισοτιμίας μπορεί να πάρει τιμή είτε “1”, είτε “0”, και τοποθετείται στη δυαδική ακολουθία έτσι ώστε, **το πλήθος των “1” στα 8 συνολικά ψηφία της ακολουθίας** να γίνεται είτε **άρτιο**, οπότε γίνεται λόγος για **άρτια ισοτιμία** (even parity), είτε **περιττό**, οπότε προκύπτει **περιττή ισοτιμία** (odd parity).

Για παράδειγμα, η λέξη “TO” (γραμμένη με λατινικά κεφαλαία γράμματα) συνίσταται από τους δύο χαρακτήρες “T” και “O” των οποίων η κωδικοποίηση σε κώδικα ASCII (7 δυαδικά ψηφία) είναι 1010100 και 1001111, αντίστοιχα. Στην πιο σημαντική θέση προσθέτουμε ένα ψηφίο ισοτιμίας με τιμή 1, προκειμένου να υποδηλώσουμε άρτια ισοτιμία (το πλήθος των “1” είναι άρτιος αριθμός), οπότε θα έχουμε **1**1010100 και **1**1001111, με άρτιο πλήθος ψηφίων με τιμή 1.

Στη συνέχεια, οι κωδικοποιημένοι χαρακτήρες που συνίστανται πλέον από 8 δυαδικά ψηφία μεταδίδονται στον προορισμό τους και η ισοτιμία τους ελέγχεται από το δέκτη. Εάν η ισοτιμία των χαρακτήρων που ελήφθησαν δεν είναι άρτια, αυτό σημαίνει ότι κατά τη διάρκεια της μετάδοσης έχει αλλάξει η τιμή τουλάχιστον ενός δυαδικού ψηφίου.

Ωστόσο, όταν έχουμε άρτιο πλήθος σφαλμάτων (π.χ. εάν έχουμε δύο σφάλματα ταυτόχρονα), δεν εξασφαλίζεται η ανίχνευσή τους, οπότε απαιτούνται διαφορετικού είδους κώδικες ανίχνευσης σφαλμάτων.

Όλοι οι κώδικες διόρθωσης / ανίχνευσης σφαλμάτων προσθέτουν πλεονασματική πληροφορία στα δεδομένα που αποστέλλονται.

Πληρέστερος κώδικας ανίχνευσης σφαλμάτων είναι ο **κώδικας Hamming**. Εκτός από τη δυνατότητα ανίχνευσης της ύπαρξης σφαλμάτων, έχει τη δυνατότητα προσδιορισμού της θέσης των σφαλμάτων σε έναν κωδικοποιημένο χαρακτήρα, έτσι ώστε να μπορούν να ανακτηθούν τα αρχικά δεδομένα.

Με βάση τη μέθοδο αυτή, δημιουργούνται κωδικοποιημένοι χαρακτήρες στους οποίους τα ψηφία ισοτιμίας συνδυάζονται με επιλεγμένες ομάδες ψηφίων των αρχικών κωδικοποιημένων χαρακτήρων.

Κατά τη λήψη κάθε χαρακτήρα, ανιχνεύονται τυχόν σφάλματα μέσω ελέγχου ισοτιμίας και σχηματίζεται ένας δυαδικός αριθμός με ψηφία ελέγχου που δηλώνει τη θέση του ψηφίου του οποίου η τιμή έχει αλλάξει, ώστε αυτό να μπορεί να διορθωθεί.

Πηγές

Για τη δημιουργία των σημειώσεων των διαλέξεων του μαθήματος έχει χρησιμοποιηθεί υλικό από την παρακάτω βιβλιογραφία:

- Morris Mano, M. & Ciletti, M.D. (2013). Ψηφιακή Σχεδίαση (5η έκδοση). Εκδόσεις Παπασωτηρίου.
- Nelson, V.P., Troy Nagle, H., David Irwin, J. & Carrol, B.D. (2007). Ανάλυση και Σχεδίαση Κυκλωμάτων Ψηφιακής Λογικής. Εκδόσεις Επίκεντρο.
- Ρουμελιώτης, Μ. & Σουραβλάς, Στ. (2013). Ψηφιακή Σχεδίαση: Αρχές & Εφαρμογές. Εκδόσεις Τζιόλα.
- Wakerly, J.F. (2005). Ψηφιακή Σχεδίαση: Αρχές & Πρακτικές (3η έκδοση). Εκδόσεις Κλειδάριθμος.
- Brown, S., Vranezic, Z. (2001). Σχεδίαση Ψηφιακών Συστημάτων με τη Γλώσσα VHDL. Εκδόσεις Τζιόλα.
- Μπισδούνης, Λ. (2015). Βασικές Εξειδικεύσεις σε Αρχιτεκτονική και Δίκτυα Υπολογιστών – Τόμος Α’. Εκδόσεις Ελληνικό Ανοικτό Πανεπιστήμιο.