ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

6/5/2005

Εισαγωγή στην Επιστήμη των Υπολογιστών

Εξάμηνο 4ο-ΣΗΜΜΥ

Σημειώσεις στα Συστήματα Αρίθμησης-Δυαδική Παράσταση Αριθμών

(καθηγητής: Νεκτάριος Κοζύρης)
(επιμέλεια σημειώσεων: Μαρία Αθανασάκη, Ευαγγελία Αθανασάκη)

Οι υπολογιστές αναπαριστούν όλα τα είδη πληροφορίας ως δυαδικά δεδομένα. Έτσι, για την ευκολότερη και ταχύτερη επεξεργασία των διαφόρων πληροφοριών, οι υπολογιστές χρησιμοποιούν αριθμητικά συστήματα διαφορετικά από το γνωστό μας *δεκαδικό* (decimal) σύστημα και κυρίως το *δυαδικό* (binary).

1. Αριθμητικά Συστήματα

Κάθε αριθμός N μπορεί να γραφεί με την ακόλουθη μορφή:

$$N = \sum_{i=-n}^{m-1} \alpha_i \cdot \beta^i = \underbrace{\alpha_{m-1} \cdot \beta^{m-1} + \alpha_{m-2} \cdot \beta^{m-2} + ... + \alpha_1 \cdot \beta + \alpha_0}_{\text{ακέραιο μέρος του αριθμού}} + \underbrace{\alpha_{-1} \cdot \frac{1}{\beta} + \alpha_{-2} \cdot \frac{1}{\beta^2} + ... + \alpha_{-n} \cdot \frac{1}{\beta^n}}_{\text{κλασματικό μέρος του αριθμού}}$$

Με β παριστάνουμε τη βάση του αριθμητικού συστήματος στο οποίο εκφράζεται ο αριθμός (β ≥ 2) και με α_i συμβολίζουμε τα ψηφία του αριθμού ($0 \le \alpha_i \le \beta-1$). Το ψηφίο α_i πολλαπλασιάζεται με τον αριθμό β^i, γι' αυτό λέμε ότι η τάξη (order) του ψηφίου α_i είναι i. Αν ο αριθμός N έχει m ακέραια ψηφία, οι εκθέτες i παίρνουν θετικές τιμές από 0 έως m-1 για το ακέραιο μέρος του και αν τα κλασματικά του ψηφία είναι n, οι εκθέτες i παίρνουν αρνητικές τιμές από -1 έως –n για το κλασματικό του τμήμα.

> Ο δεκαδικός αριθμός 19,278 με τον τρόπο αυτό γράφεται ως
> $\mathbf{1}\cdot 10^1 + \mathbf{9}\cdot 10^0 + \mathbf{2}\cdot 10^{-1} + \mathbf{7}\cdot 10^{-2} + \mathbf{8}\cdot 10^{-3}$, δηλαδή $\alpha_1=1$, $\alpha_0=9$, $\alpha_{-1}=2$, $\alpha_{-2}=7$, $\alpha_{-3}=8$.

Ένα αριθμητικό σύστημα με βάση β χρειάζεται β διαφορετικά «ψηφία» για την παράσταση των αριθμών, που παίρνουν τις τιμές από 0 έως β-1. Ένας ακέραιος αριθμός που έχει m

ψηφία, στο σύστημα αυτό μπορεί να πάρει τιμές από 0 έως β^{m-1}, δηλαδή β^m διαφορετικές τιμές.

Δεκαδικά ψηφία		Δυαδικά ψηφία	Οκταδικά ψηφία	Δεκαεξαδικά ψηφία	
0	5	0	4	0	8
1	6	1	5	1	9
2	7		6	2	A
3	8		7	3	B
4	9			4	C
				5	D
				6	E
				7	F

Τα συνηθέστερα αριθμητικά συστήματα είναι αυτά που έχουν βάση τους αριθμούς 2 (δυαδικό σύστημα, binary system),8 (οκταδικό σύστημα, octal system),10 (δεκαδικό σύστημα, decimal system) και 16 (δεκαεξαδικό σύστημα, hexadecimal system). Στον παραπάνω πίνακα βλέπουμε τα ψηφία αυτών των αριθμητικών συστημάτων.

Το δεκαεξαδικό σύστημα χρειάζεται 16 ψηφία για την παράσταση των αριθμών, αλλά το γνωστό δεκαδικό αριθμητικό σύστημα παρέχει μόνο 10 ψηφία (0-9). Για τα επιπλέον 6 ψηφία χρησιμοποιούμε τους χαρακτήρες A-F, δηλαδή A=10, B=11, C=12, D=13, E=14 και F=15. Η σύμβαση αυτή ακολουθείται σε κάθε σύστημα που χρειάζεται περισσότερα από 10 ψηφία.

> Στο δεκαεξαδικό σύστημα, η ακολουθία ψηφίων «6F3» παριστάνει τον αριθμό $6 \cdot 16^2 + 15 \cdot 16^1 + 3$

Στην καθημερινή μας ζωή χρησιμοποιούμε το δεκαδικό σύστημα και είμαστε εξοικειωμένοι με αυτό, έτσι παριστάνουμε τους αριθμούς μόνο με τα ψηφία τους, π.χ. λέμε 15 και όχι $1 \cdot 10^1+5$. Το ίδιο μπορούμε να κάνουμε και με οποιοδήποτε άλλο αριθμητικό σύστημα, αρκεί να δηλώνουμε το σύστημα αυτό. Ο προσδιορισμός του συστήματος γίνεται συνήθως με ένα δείκτη που συνοδεύει τον αριθμό και δηλώνει τη βάση του αριθμητικού συστήματος.

> Η ακολουθία ψηφίων «321» παριστάνει διαφορετικούς αριθμούς ανάλογα με το σύστημα αρίθμησης που θα δηλώσουμε. Στο δεκαδικό σύστημα θα γράψουμε $321_{(10)}$ και θα εννοούμε $3 \cdot 10^2+2 \cdot 10+1$, ενώ στο οκταδικό σύστημα θα γράψουμε $321_{(8)}$ και θα εννοούμε $3 \cdot 8^2+2 \cdot 8+1$. Στο δεκαεξαδικό σύστημα θα γράψουμε $321_{(16)}$ και θα εννοούμε $3 \cdot 16^2+2 \cdot 16+1$.
>
> Ο αριθμός $1101_{(2)}$ είναι γραμμένος στο δυαδικό σύστημα, έτσι αντιστοιχεί στην έκφραση $1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1$.

Τα ψηφία ενός αριθμού γραμμένου στο δυαδικό σύστημα ονομάζονται bits (**b**inary dig**its**, δυαδικά ψηφία). Το πιο αριστερό ψηφίο του αριθμού ονομάζεται *περισσότερο σημαντικό ψηφίο* (Most Significant Bit, MSB), γιατί πολλαπλασιάζεται με το μεγαλύτερο συντελεστή, και το πιο δεξιό ψηφίο του αριθμού ονομάζεται *λιγότερο σημαντικό ψηφίο* (Least Significant Bit, LSB), γιατί πολλαπλασιάζεται με το μικρότερο συντελεστή

2. Μετατροπή αριθμών από ένα αριθμητικό σύστημα σε άλλο

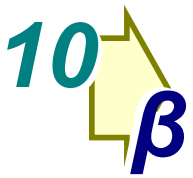
Η μετατροπή ενός αριθμού από ένα αριθμητικό σύστημα με βάση β προς το δεκαδικό σύστημα είναι πολύ απλή: υπολογίζουμε την τιμή της παράστασης

$$\alpha_{m-1}\cdot\beta^{m-1} + \ldots + \alpha_1\cdot\beta + \alpha_0 + \alpha_{-1}\cdot\beta^{-1} + \ldots + \alpha_{-n}\cdot\beta^{-n}.$$

Ο δυαδικός αριθμός $10011_{(2)}$ στο δεκαδικό σύστημα έχει την τιμή $1\cdot2^4 + 0\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1 = 16+2+1 = 19_{(10)}$.

Ο οκταδικός αριθμός $7123{,}35_{(8)}$ στο δεκαδικό σύστημα έχει την τιμή $7\cdot8^3 + 1\cdot8^2 + 2\cdot8 + 3 + 3\cdot8^{-1} + 5\cdot8^{-2} = 3584 + 64 +16 + 3 + 0{,}375 + 0{,}3125 = 3667{,}6875_{(10)}$.

Ο δεκαεξαδικός αριθμός $FC27_{(16)}$ είναι ισοδύναμος με το δεκαδικό $15\cdot16^3 + 12\cdot16^2 + 2\cdot16 + 7 = 61440 + 3072 + 32 + 7 = 64551_{(10)}$.



Πιο πολύπλοκη είναι η διαδικασία μετατροπής ενός αριθμού από το δεκαδικό σύστημα σε ένα άλλο σύστημα αρίθμησης με βάση β. Η μετατροπή γίνεται χωριστά για το ακέραιο και χωριστά για το κλασματικό μέρος.

Για να μετατρέψουμε το ακέραιο μέρος του αριθμού Α σε βάση β, κάνουμε διαδοχικές διαιρέσεις του ακεραίου μέρους του Α με τον αριθμό β. Η διαδικασία μετατροπής είναι η εξής:

(1) Αρχικά ο νέος αριθμός Χ δεν έχει ψηφία και χρησιμοποιούμε μόνο το ακέραιο μέρος του Α.
(2) Διαιρούμε τον Α με τη βάση β και παίρνουμε το πηλίκο Π και το υπόλοιπο Υ.
(3) Γράφουμε το Υ στα αριστερά του νέου αριθμού Χ.
(4) Αντικαθιστούμε τον αριθμό Α με το πηλίκο Π.
(5) Επαναλαμβάνουμε τα βήματα (2), (3), (4) έως ότου το Α να γίνει 0.

Ας δούμε πώς μετατρέπεται ο αριθμός $A=53_{(10)}$ στο δυαδικό σύστημα (β=2) :

Α		**Π**	**Υ**	**Χ**
53	*Διαιρούμε το 53 με το 2* →	26	1	**1**
26	*Διαιρούμε το 26 με το 2* →	13	0	**0**1
13	*Διαιρούμε το 13 με το 2* →	6	1	**1**01
6	*Διαιρούμε το 6 με το 2* →	3	0	**0**101
3	*Διαιρούμε το 3 με το 2* →	1	1	**1**0101
1	*Διαιρούμε το 1 με το 2* →	0	1	**1**10101
0	→ **$53_{(10)} = 110101_{(2)}$**			

Ας δούμε και τη μετατροπή του αριθμού 312 στο οκταδικό σύστημα:

Α		Π	Υ	Χ
312	*Διαιρούμε το 312 με το 8* →	39	0	**0**
39	*Διαιρούμε το 39 με το 8* →	4	7	**7**0
4	*Διαιρούμε το 4 με το 8* →	0	4	**4**70
0	→ **$312_{(10)} = 470_{(8)}$**			

Για να μετατρέψουμε το κλασματικό μέρος ενός αριθμού Α από το δεκαδικό σύστημα σε ένα άλλο σύστημα αρίθμησης με βάση β, κάνουμε διαδοχικούς πολλαπλασιασμούς του κλασματικού μέρους του Α με τη βάση β. Το κλασματικό μέρος στο νέο σύστημα αρίθμησης μπορεί να έχει άπειρα ψηφία, γι' αυτό καθορίζουμε από πριν το μέγιστο αριθμό ψηφίων Ν που θα υπολογίσουμε για το νέο σύστημα αρίθμησης. Η διαδικασία μετατροπής είναι η ακόλουθη:

(1) Αρχικά το νέο κλασματικό μέρος Υ δεν έχει ψηφία και χρησιμοποιούμε μόνο το κλασματικό μέρος του Α.
(2) Πολλαπλασιάζουμε τον Α με τη βάση β. Το αποτέλεσμα έχει ακέραιο μέρος Μ και κλασματικό μέρος Κ.
(3) Γράφουμε το Μ στα δεξιά του νέου κλασματικού μέρους Υ.
(4) Αντικαθιστούμε τον αριθμό Α με το Κ.
(5) Επαναλαμβάνουμε τα βήματα (2), (3), (4) έως ότου το Α να γίνει 0 ή να έχουμε υπολογίσει Ν ψηφία.

Ας υπολογίσουμε την τιμή του κλασματικού αριθμού $0,625_{(10)}$ στο δυαδικό σύστημα:

Α		Μ	Κ	Υ
0,625	*0,625 × 2 = 1,25* →	1	0,25	0,**1**
0,25	*0,25 × 2 = 0,5* →	0	0,5	0,1**0**
0,5	*0,5 × 2 = 1* →	1	0	0,101
0	→ **$0,625_{(10)} = 0,101_{(2)}$**			

Επίσης ας υπολογίσουμε την τιμή του κλασματικού αριθμού $0,171875_{(10)}$ στο δεκαεξαδικό σύστημα:

Α		Μ	Κ	Υ
0, 171875	*0,171875 × 16 = 2,75* →	2	0,75	0,**2**
0,75	*0,75 × 16 = 12* →	12	0	0,2**C**
0	→ **$0,171875_{(10)} = 0,2C_{(16)}$**			

Ας υπολογίσουμε και την τιμή του κλασματικού αριθμού $0,4_{(10)}$ στο οκταδικό σύστημα. Καθορίζουμε από πριν ότι θα υπολογίσουμε το πολύ 5 οκταδικά ψηφία.

A		M	K	Y
0,4	$0,4 \times 8 = 3,2$	3	0,2	0,**3**
0,2	$0,2 \times 8 = 1,6$	1	0,6	0,3**1**
0,6	$0,6 \times 8 = 4,8$	4	0,8	0,31**4**
0,8	$0,8 \times 8 = 6,4$	6	0,4	0,314**6**
0,4	$0,4 \times 8 = 3,2$	3	0,2	0,31463
0,2	$0,4_{(10)} \cong 0,31463_{(8)}$			

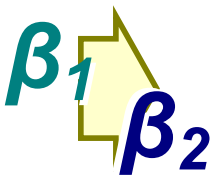
Εδώ σταματήσαμε τον υπολογισμό παρότι το A δεν είναι 0, γιατί φθάσαμε στο μέγιστο αριθμό κλασματικών ψηφίων που επιτρέπουμε. Αν προσέξουμε τις τιμές που παίρνει το A, παρατηρούμε ότι μετά από τον υπολογισμό του 4ου κλασματικού ψηφίου το A παίρνει την αρχική του τιμή που ήταν 0,4. Έτσι μπορούμε να συμπεράνουμε ότι στο οκταδικό σύστημα ο A είναι ένας περιοδικός αριθμός, ο $0,31463146\ldots_{(8)}$ ή αλλιώς $0,\overline{3146}_{(8)}$.

Όταν σταματάμε τον υπολογισμό του αριθμού μετά από n κλασματικά ψηφία, αν και το A δεν είναι 0, κάνουμε *αποκοπή* (truncation) του αριθμού. Για να είναι ο υπολογισμός μας όσο πιο ακριβής γίνεται, μπορούμε να κάνουμε *στρογγυλοποίηση* (rounding). Στη στρογγυλοποίηση υπολογίζουμε άλλο ένα κλασματικό ψηφίο. Αν αυτό είναι μικρότερο από το ½ της βάσης, τότε αφήνουμε τον αριθμό όπως είναι, με n ψηφία. Αν όμως το επιπλέον ψηφίο είναι μεγαλύτερο ή ίσο από το ½ της βάσης, τότε αυξάνουμε το τελευταίο (το n-οστό) ψηφίο κατά 1. Έτσι το σφάλμα του υπολογισμού είναι πιο μικρό.

Κατά τη μετατροπή του αριθμού $0,4_{(10)}$ στο δεκαεξαδικό σύστημα, ολοκληρώσαμε τον υπολογισμό μετά από 5 ψηφία. Το επόμενο ψηφίο που θα υπολογίζαμε είναι το 1, που είναι μικρότερο από το 4 (το ½ της βάσης), έτσι και μετά από τη στρογγυλοποίηση ο αριθμός μένει ο ίδιος.

Αν όμως κρατούσαμε μόνο n=3 κλασματικά ψηφία στρογγυλοποιώντας το αποτέλεσμα, θα υπολογίζαμε και το 4ο ψηφίο που έχει την τιμή 6 > 4. Θα αυξάναμε, λοιπόν, το 3ο ψηφίο κατά 1, και ο αριθμός θα ήταν τελικά ο $0,315_{(8)}$.

Για να μετατρέψουμε από το δεκαδικό σύστημα αρίθμησης σε άλλο, έναν αριθμό που έχει και ακέραιο και κλασματικό μέρος, μετατρέπουμε ξεχωριστά τα δύο μέρη του με τον τρόπο που είδαμε και μετά συνδυάζουμε τα αποτελέσματα.

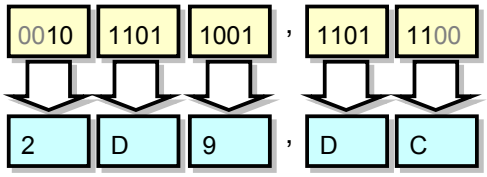


Η μετατροπή ενός αριθμού από ένα σύστημα με βάση β_1 σε ένα άλλο σύστημα με βάση β_2 γίνεται εύκολα άν χρησιμοποιήσουμε ενδιάμεσα το δεκαδικό σύστημα: μετατρέπουμε πρώτα τον αριθμό με βάση β_1 στο δεκαδικό σύστημα, και στη συνέχεια τον μετατρέπουμε από το δεκαδικό σύστημα στο σύστημα με βάση β_2. Η μέθοδος αυτή είναι πιο εύκολη από την απευθείας μετατροπή, γιατί είμαστε πιο εξοικειωμένοι με υπολογισμούς στο δεκαδικό σύστημα.

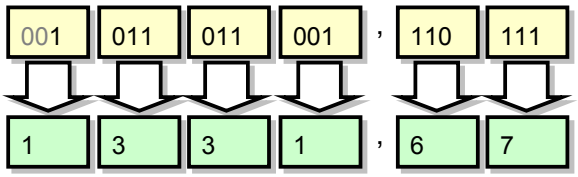
Μια ειδική περίπτωση, όμως, είναι η μετατροπή μεταξύ του δυαδικού και του οκταδικού ή του δεκαεξαδικού συστήματος. Οι μετατροπές αυτές είναι ιδιαίτερα εύκολες, γιατί οι βάσεις των δύο συστημάτων, το 8 και το 16, είναι δυνάμεις του 2.

Για να μετατρέψουμε ένα δυαδικό αριθμό στο δεκαεξαδικό σύστημα, χωρίζουμε τα ψηφία του σε τετράδες ξεκινώντας από την υποδιαστολή που χωρίζει ακέραιο και κλασματικό μέρος, και προχωρώντας προς τα «άκρα» του αριθμού. Κάθε τέτοια τετράδα αντιστοιχεί σε ένα μονοψήφιο δεκαεξαδικό αριθμό, και την αντικαθιστούμε με το ψηφίο αυτό. Η μετατροπή ενός δεκαεξαδικού αριθμού σε δυαδικό γίνεται αντικαθιστώντας κάθε ψηφίο του αριθμού με τον αντίστοιχο τετραψήφιο δυαδικό αριθμό.

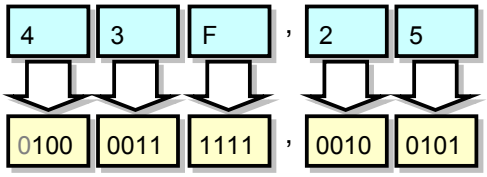
Η μετατροπή από το δυαδικό σύστημα προς το οκταδικό και αντίστροφα γίνεται με τον ίδιο τρόπο, αλλά χωρίζουμε τα δυαδικά ψηφία σε τριάδες αντί για τετράδες.



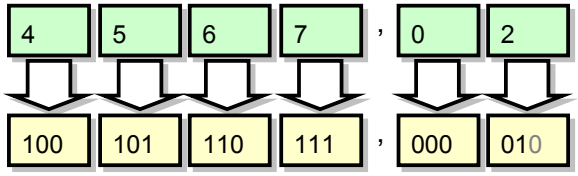
Για να μετατραπεί σε δεκαεξαδικό, χωρίζουμε το δυαδικό αριθμό $1011011001,110111_{(2)}$ πρώτα σε τετράδες ξεκινώντας από την υποδιαστολή. Στα «άκρα» του αριθμού προσθέτουμε όσα μηδενικά είναι απαραίτητα, έτσι ώστε να συμπληρωθούν οι τετράδες. Στη συνέχεια αντικαθιστούμε κάθε τετράδα με το αντίστοιχο δεκαεξαδικό ψηφίο. Π.χ. η αριστερότερη τετράδα που είναι 0010 ισοδυναμεί με το ψηφίο 2, ενώ η επόμενη τετράδα 1101 που έχει τιμή 13 ισοδυναμεί με το ψηφίο D. Ο ισοδύναμος δεκαεξαδικός αριθμός είναι ο $2D9,DC_{(16)}$.



Για να μετατρέψουμε τον ίδιο αριθμό σε οκταδικό τον χωρίζουμε σε τριάδες, προσθέτοντας και πάλι στα άκρα του μηδενικά, αν χρειαστεί. Κάθε τριάδα αντικαθίσταται από το αντίστοιχο οκταδικό ψηφίο. Π.χ. η δεξιότερη τριάδα που είναι 111 έχει την τιμή 7 ($1\cdot2^2 + 1\cdot2 + 1 = 4+2+1 = 7$) και αντικαθίσταται με το ψηφίο αυτό. Ο ισοδύναμος οκταδικός αριθμός είναι ο $1331,67_{(8)}$.



Αντίστροφα, η μετατροπή του δεκαεξαδικού αριθμού $43F,25_{(16)}$ γίνεται αντικαθιστώντας τα ψηφία του με τον αντίστοιχο τετραψήφιο δυαδικό αριθμό. Π.χ. το ψηφίο 4 θα αντικατασταθεί από τον δυαδικό αριθμό 0100 που έχει την τιμή 4. Ο δυαδικός αριθμός που προκύπτει είναι ο $010000111111,00100101_{(2)}$.



Στον οκταδικό αριθμό $4567,02_{(8)}$ θα αντικαταστήσουμε κάθε ψηφίο με τον αντίστοιχο *τριψήφιο* δυαδικό αριθμό, για να πάρουμε το δυαδικό αριθμό $100101110111,000010_{(2)}$.

Τα μηδενικά στα «άκρα» των δυαδικών αριθμών (όπως και σε όλα τα συστήματα αρίθμησης) μπορούν να παραλειφθούν.

3. Πράξεις θετικών ακεραίων αριθμών

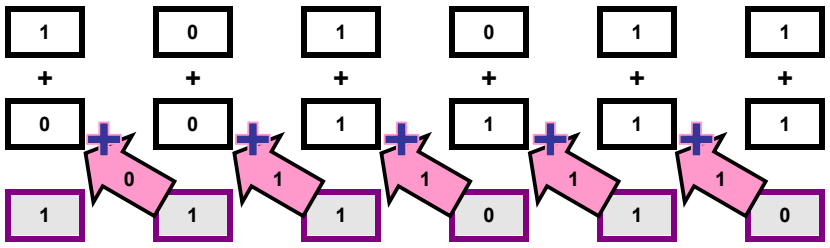
Στο δυαδικό σύστημα οι αριθμητικές πράξεις γίνονται με παρόμοιο τρόπο με το δεκαδικό σύστημα. Προσθέσεις και αφαιρέσεις γίνονται από δεξιά προς τα αριστερά, και χρησιμοποιούμε κρατούμενα ή δανεικά ψηφία, αντίστοιχα.

Για την πρόσθεση αριθμών με n bits, ξεκινάμε από τα δεξιά, προσθέτοντας τα δύο λιγότερο σημαντικά bits των αριθμών και συνεχίζουμε προς τα αριστερά. Αν οι αριθμοί είναι οι $x_n x_{n-1} \ldots x_i \ldots x_1$ και $y_n y_{n-1} \ldots y_i \ldots y_1$, ξεκινάμε από την πρόσθεση x_1+y_1, η οποία μας δίνει το ψηφίο z_1 του αποτελέσματος και το κρατούμενο K_1. Στη συνέχεια, προσθέτουμε τα ψηφία x_2, y_2 και το κρατούμενο K_1 για να πάρουμε το ψηφίο z_2 του αποτελέσματος και το κρατούμενο K_2, κλπ. Γενικά το κρατούμενο K_{i-1} που πιθανώς θα προκύψει από κάποια πρόσθεση προωθείται και προστίθεται με το επόμενο ζεύγος ψηφίων x_i και y_i των αριθμών. Αντίστοιχα, στην αφαίρεση λαμβάνουμε τα αντίστοιχα ζεύγη των ψηφίων x_i και y_i, κάνουμε την πράξη και προωθούμε το δανεικό ψηφίο B_i στο επόμενο ζεύγος ψηφίων x_{i+1} και y_{i+1}.

x_i	y_i	K_{i-1} ή B_{i-1}	$x_i+y_i+K_{i-1}$	K_i	$x_i-y_i-B_{i-1}$	B_i
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	1	0	0	1	0	0
1	1	1	1	1	1	1

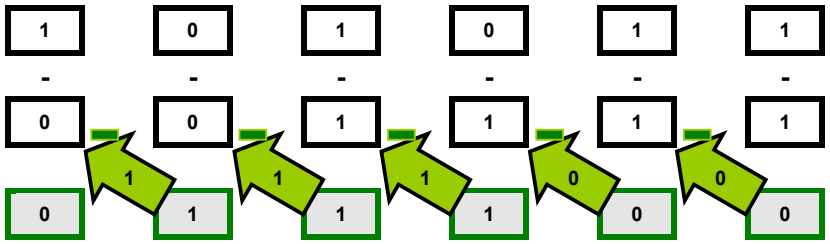
Οι υπολογισμοί που κάνουμε για αριθμούς των n bits αφορούν, λοιπόν, συνήθως 3 bits, γιατί έχει προστεθεί και το κρατούμενο ή το δανεικό ψηφίο. Στον πίνακα βλέπουμε τα αποτελέσματα της πρόσθεσης και αφαίρεσης δύο ψηφίων με κρατούμενο ή δανεικό ψηφίο αντίστοιχα.

Θέλουμε να προσθέσουμε τους αριθμούς 43 και 15, αλλά με τη δυαδική τους αναπαράσταση. Ο αριθμός 43 στο δυαδικό σύστημα είναι ο $101011_{(2)}$ ενώ ο 15 παριστάνεται ως $001111_{(2)}$.



Βλέπουμε στο σχήμα πώς το κρατούμενο από κάθε ζεύγος ψηφίων προωθείται στο επόμενο. Το αποτέλεσμα είναι ο δυαδικός αριθμός $111010_{(2)}$ δηλαδή ο αριθμός 58.

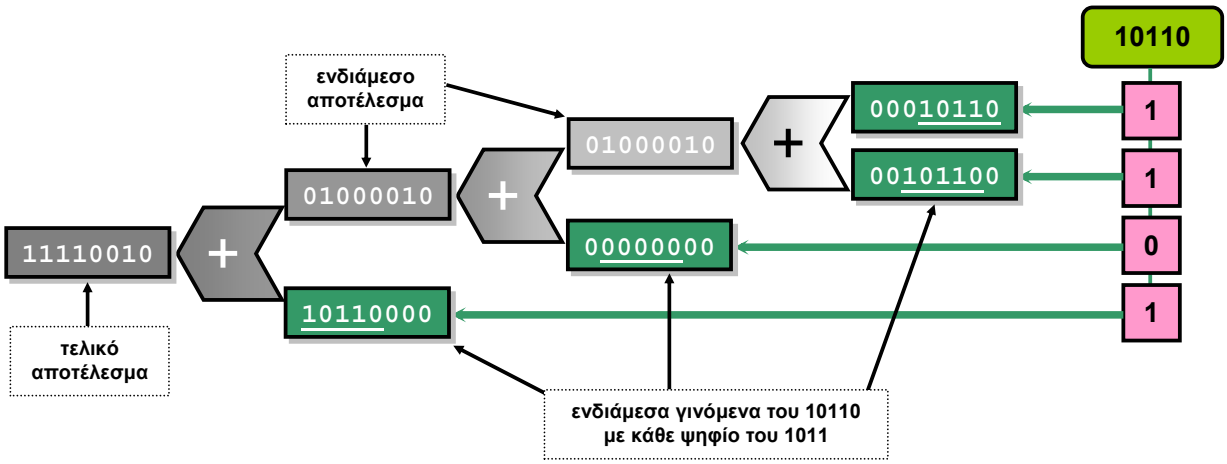
Αν θέλουμε να κάνουμε την αφαίρεση 43-15 αντίστοιχα θα έχουμε:



Το αποτέλεσμα είναι ο δυαδικός αριθμός $011100_{(2)}$ δηλαδή ο δεκαδικός 28. Και εδώ βλέπουμε τα δανεικά ψηφία του κάθε ζεύγους που μεταφέρονται στο επόμενο επίπεδο.

Ο πολλαπλασιασμός και η διαίρεση δυαδικών αριθμών γίνονται με διαδοχικές προσθέσεις και αφαιρέσεις αντίστοιχα.

Ας δούμε πώς πολλαπλασιάζουμε τους αριθμούς $22_{(10)} = 10110_{(2)}$ και $11_{(10)} = 1011_{(2)}$.

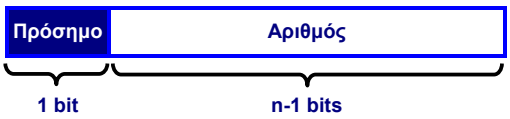


Στο σημείο αυτό πρέπει να σημειώσουμε ότι στον πολλαπλασιασμό εκτελούμε κάθε φορά τα επιμέρους αθροίσματα για να μην προκύπτουν κρατούμενα που πρέπει να ληφθούν υπ' όψη στις μεθεπόμενες βαθμίδες.

4. Παράσταση ακεραίων αριθμών

Η μνήμη κάθε υπολογιστή είναι οργανωμένη σε *λέξεις* (words), δηλαδή ομάδες των n bits (το n είναι συνήθως ένα πολλαπλάσιο του 8). Το n ονομάζεται *μήκος λέξης* (word length) του υπολογιστή. Κάθε αριθμός θα καταλαμβάνει χώρο όσο μία λέξη της μνήμης του υπολογιστή.

Όπως γνωρίζουμε, με n δυαδικά ψηφία μπορούμε να παραστήσουμε 2^n το πλήθος διαφορετικούς αριθμούς, τους 0 … 2^n-1. Στην περίπτωση που θέλουμε με n δυαδικά ψηφία να παραστήσουμε προσημασμένους ακέραιους αριθμούς, τότε εκμεταλλευόμαστε το αριστερότερο bit (δηλαδή το MSB) του αριθμού, στο οποίο κωδικοποιούμε το πρόσημό του. Αν το πρόσημο έχει την τιμή 0, τότε ο αριθμός είναι θετικός, ενώ αν έχει την τιμή 1 είναι αρνητικός. Με τα υπόλοιπα n-1 δυαδικά ψηφία κωδικοποιούμε την απόλυτη τιμή του αριθμού, δηλαδή το *μέτρο* του.



> Οι αριθμοί $01110010_{(2)}$ και $00001_{(2)}$ είναι θετικοί, ενώ οι αριθμοί $1110010_{(2)}$ και $100001_{(2)}$ είναι αρνητικοί.

Ένα θετικό αριθμό τον παριστάνουμε θέτοντας το πιο σημαντικό bit (δηλαδή το πρόσημο) στην τιμή 0, και τα υπόλοιπα n-1 bits στην τιμή του μέτρου του, δηλαδή στην τιμή του αριθμού. Επειδή ο μεγαλύτερος ακέραιος αριθμός που παριστάνεται με n-1 bits είναι ο 2^{n-1}-1, η τιμή του αριθμού δεν μπορεί να ξεπερνά το όριο αυτό.

> Σε έναν υπολογιστή όπου το μήκος λέξης είναι 8, ο αριθμός 23 θα παρασταθεί ως 00010111. Το αριστερότερο bit δηλώνει ότι ο αριθμός είναι θετικός, και τα υπόλοιπα bits περιέχουν τον αριθμό 23. Ο μεγαλύτερος θετικός αριθμός που μπορεί να αποθηκευθεί με 8 bits είναι ο 01111111, δηλαδή ο 127 (2^{8-1}-1 = 2^7-1 = 128-1).

Υπάρχουν τρεις διαφορετικοί τρόποι για να κωδικοποιήσουμε τους αρνητικούς προσημασμένους αριθμούς στα υπόλοιπα n-1 bits:

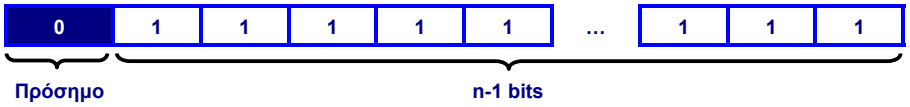
- Η παράσταση **μέτρου**
- Η παράσταση **συμπληρώματος ως προς 1**
- Η παράσταση **συμπληρώματος ως προς 2**

4.1. Παράσταση μέτρου

Το Most Significant Bit (MSB) δηλώνει αν ο αριθμός είναι θετικός ή αρνητικός, ενώ τα υπόλοιπα n-1 ψηφία παριστάνουν το μέτρο του αριθμού σε δυαδική μορφή.

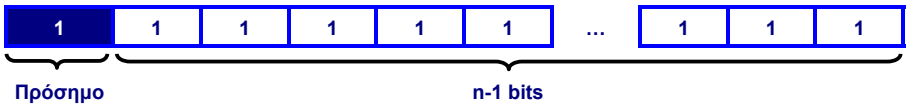
> Σε έναν υπολογιστή με λέξη 6 bits, ο αριθμός $+12_{(10)}$ παριστάνεται ως εξής: $001100_{(2)}$, ενώ ο αριθμός $-12_{(10)}$ παριστάνεται ως εξής: $101100_{(2)}$.

Ο μέγιστος θετικός αριθμός που μπορεί να παρασταθεί στο σύστημα αυτό με λέξη μήκους n bits έχει την παράσταση:



και είναι ο 2^{n-1}-1

Ο ελάχιστος αρνητικός που μπορεί να παρασταθεί με μήκος λέξης n bits έχει την παράσταση:



και είναι ο $-(2^{n-1}-1)$

Ο αριθμός 0 μπορεί να παρασταθεί με δύο τρόπους: σαν 00…00 και σαν 10…00.

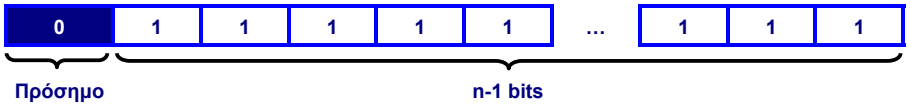
4.2. Παράσταση συμπληρώματος ως προς 1

Στην παράσταση αυτή, αν το MSB του αριθμού είναι 0, ο αριθμός είναι θετικός και το μέτρο του δίδεται από τα υπόλοιπα n-1 bits. Αν το MSB είναι 1, τότε ο αριθμός είναι αρνητικός και το συμπλήρωμα ως προς 1 των υπολοίπων n-1 bits δίνει το μέτρο του.

Το συμπλήρωμα ως προς 1 ενός δυαδικού αριθμού βρίσκεται εύκολα αν αντικατασταθούν όλα τα 1 του αριθμού με 0 και όλα τα 0 με 1.

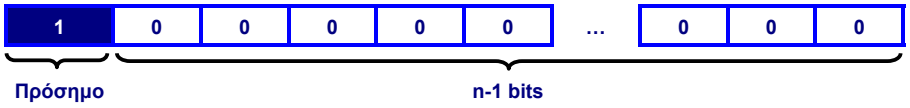
> Για παράδειγμα, το συμπλήρωμα ως προς 1 του αριθμού 11010110 είναι 00101001.
>
> Επομένως, με βάση τα παραπάνω, ο αριθμός $+12_{(10)}$ παριστάνεται σε υπολογιστή με λέξη μήκους n=6 bits σαν $001100_{(2)}$, ενώ ο αριθμός $-12_{(10)}$ παριστάνεται σαν $110011_{(2)}$.

Ο μέγιστος θετικός αριθμός που μπορεί να παρασταθεί στο σύστημα αυτό με λέξη μήκους n bits έχει την παράσταση:



και είναι ο $2^{n-1}-1$

Ο ελάχιστος αρνητικός που μπορεί να παρασταθεί με μήκος λέξης n bits έχει την παράσταση:



και είναι ο $-(2^{n-1}-1)$

Ο αριθμός 0 μπορεί να παρασταθεί με δύο τρόπους: σαν 00…00 και σαν 11…11.

4.3. Παράσταση συμπληρώματος ως προς 2

Η παράσταση συμπληρώματος ως προς 2 είναι αυτή που χρησιμοποιείται περισσότερο, γιατί διευκολύνει και απλοποιεί πολύ την εκτέλεση των αριθμητικών πράξεων, τόσο για τους θετικούς, όσο και για τους αρνητικούς αριθμούς.

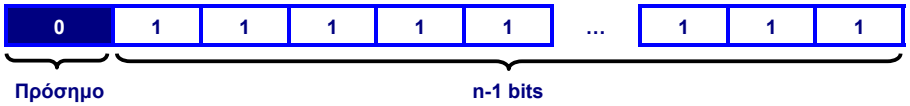
Όπως και στις προηγούμενες παραστάσεις, αν το MSB του αριθμού είναι 0, ο αριθμός είναι θετικός και το μέτρο του δίδεται από τα υπόλοιπα n-1 bits. Εάν το MSB του αριθμού είναι 1, τότε ο αριθμός είναι αρνητικός. Για να βρούμε το μέτρο του αριθμού, πρέπει να υπολογίσουμε το συμπλήρωμα ως προς 2 και των n ψηφίων του (δηλαδή λαμβάνουμε

υπόψη και το πρόσημο). Το συμπλήρωμα ως προς 2 ενός δυαδικού αριθμού βρίσκεται, εάν αντικαταστήσουμε το 0 με 1 και το 1 με 0 και στη συνέχεια προσθέσουμε 1.

Για να μετατρέψουμε έναν αρνητικό στην παράσταση συμπληρώματος του 2, ακολουθούμε παρόμοια διαδικασία: γράφουμε το μέτρο του σε δυαδική μορφή, αντικαθιστούμε το 0 με 1 και το 1 με 0 και στη συνέχεια προσθέτουμε 1. Αν δεν υπάρχει ήδη ως κρατούμενο, τοποθετούμε στα αριστερά του αριθμού το ψηφίο 1 του προσήμου.

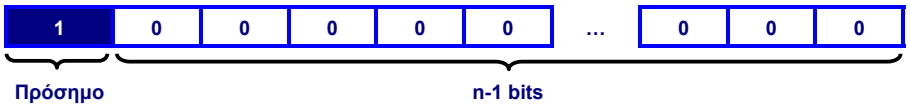
> Για να βρούμε την παράσταση συμπληρώματος ως προς 2 του αριθμού -17 σε ένα υπολογιστή με μήκος λέξης 16 bits, αρχικά θα γράψουμε τον αντίστοιχο θετικό (17) σε δυαδική μορφή, δηλαδή 0000000000010001. Στη συνέχεια θα αντικαταστήσουμε το 0 με 1 και το 1 με 0 στον αριθμό αυτό, και θα πάρουμε 1111111111101110. Στον αριθμό αυτό θα προσθέσουμε τον 1. Η τελική του παράσταση θα είναι λοιπόν 1111111111101111.
>
> Για να βρούμε την τιμή που παριστάνει ο αριθμός 11100110 θα υπολογίσουμε το συμπλήρωμά του ως προς 2 *μαζί με το πρόσημο*. Αρχικά αντιστρέφουμε όλα τα ψηφία του και παίρνουμε 00011001. Μετά προσθέτουμε τον αριθμό 1, και έχουμε 00011001 + 1 = 00011010. Άρα το μέτρο του αριθμού είναι το $00011010_{(2)} = 26_{(10)}$ και ο αριθμός είναι ο -26.

Ο μέγιστος θετικός αριθμός που μπορεί να παρασταθεί στο σύστημα αυτό με λέξη μήκους n bits έχει την παράσταση:



και είναι ο 2^{n-1}-1

Ο ελάχιστος αρνητικός που μπορεί να παρασταθεί με μήκος λέξης n bits έχει την παράσταση:



Για να βρούμε την τιμή του αριθμού αυτού θα υπολογίσουμε το συμπλήρωμά του ως προς 2. Αντιστρέφουμε τα ψηφία του και παίρνουμε τον αριθμό $01111\ldots1111_{(2)}$ και μετά προσθέτουμε 1, για να πάρουμε τον $10000\ldots000_{(2)} = 2^{n-1}$. Άρα ο ελάχιστος αριθμός είναι ο -2^{n-1}.

Η παράσταση συμπληρώματος του 2 έχει, από ό,τι βλέπουμε, μία ιδιαιτερότητα: ο μικρότερος αρνητικός αριθμός που μπορούμε να παραστήσουμε έχει μεγαλύτερη απόλυτη τιμή (2^{n-1}) από το μεγαλύτερο θετικό (2^{n-1}-1). Αυτό συμβαίνει γιατί ο αντίστοιχός του θετικός, ο 2^{n-1}, χρειάζεται και το ψηφίο του προσήμου για να παρασταθεί.

> Ας μην ξεχνάμε ότι οι έννοιες «συμπλήρωμα ως προς 2» και «παράσταση συμπληρώματος ως προς 2» είναι διαφορετικές. Το συμπλήρωμα ως προς 2 ενός αριθμού είναι το αποτέλεσμα της αντιστροφής των ψηφίων του αριθμού από 0 σε 1 και από 1 σε 0, και της πρόσθεσης σε αυτόν του 1. Η παράσταση συμπληρώματος ως προς δύο χρησιμοποιεί το συμπλήρωμα ως προς 2 για να παραστήσει τους αρνητικούς αριθμούς. Ανάλογα ισχύουν και για τις έννοιες «συμπλήρωμα ως προς 1» και «παράσταση συμπληρώματος ως προς 1».

5. Πρόσθεση προσημασμένων ακεραίων αριθμών

Η παραστάσεις του συμπληρώματος ως προς 1 και ως προς 2 έχουν το πλεονέκτημα: η πρόσθεση μεταξύ των αριθμών γίνεται απευθείας, χωρίς να χρειάζεται μετατροπή τους, ανεξάρτητα από το πρόσημό τους. Έτσι η διαδικασία της πρόσθεσης που είδαμε στην παράγραφο 3 εφαρμόζεται αυτούσια σχεδόν και σε προσημασμένους αριθμούς.

Εάν, όμως, προκύψει **κρατούμενο** από την πρόσθεση των MSB, **αγνοείται στην περίπτωση συμπληρώματος ως προς δύο**, ή **προστίθεται στο αποτέλεσμα στην περίπτωση συμπληρώματος ως προς 1**.

Στη συνέχεια δίδονται κάποια παραδείγματα προσθέσεων σε παράσταση συμπληρώματος ως προς 2:

00010101 (21)	11001010 (-54)	00010010 (18)	11111101 (-3)
+ 00110011 (51)	+ 01000100 (68)	+ 10001101 (-115)	+ 11110010 (-14)
01001000 (72)	100001110 (14)	10011111 (-97)	111101111 (-17)

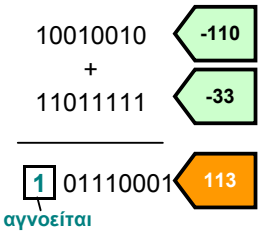
Στο δεύτερο άθροισμα, το αποτέλεσμα της πρόσθεσης έχει και ένα επιπλέον bit, γιατί $11001010_{(2)} + 01000100_{(2)} = 100001110_{(2)}$. Αυτό το επιπλέον bit το αγνοούμε και παίρνουμε το σωστό αποτέλεσμα. Το ίδιο ισχύει και το τέταρτο άθροισμα.

Αντίθετα, στην περίπτωση πράξεων με παράσταση συμπληρώματος ως προς 1, το επιπλέον bit προστίθεται στο αποτέλεσμα:

00010101 (21)	11001001 (-54)	00010010 (18)	11111100 (-3)
+ 00110011 (51)	+ 01000100 (68)	+ 10001100 (-115)	+ 11110001 (-14)
01001000 (72)	100001101	10011110 (-97)	111101101
	+ 1		+ 1
	00001110 (14)		11101110 (-17)

Η μόνη περίπτωση στην εκτέλεση των πράξεων που χρειάζεται προσοχή είναι όταν το αποτέλεσμα μίας πράξης είναι πολύ μεγάλο ή πολύ μικρό και δεν μπορεί να παρασταθεί με το πλήθος των bits που έχουμε στη διάθεσή μας. Τότε λέμε ότι η πράξη προκάλεσε *υπερχείλιση* (overflow) του αποτελέσματος.

Για παράδειγμα, σε περίπτωση που χρησιμοποιείται η παράσταση συμπληρώματος ως προς 2:



Το άθροισμα των αριθμών -110 και -33, που είναι -143, δεν μπορεί να παρασταθεί με 8 bits, γιατί ο μικρότερος αριθμός που μπορεί να παρασταθεί με 8 bits είναι ο -128. Αν προσθέσουμε αυτούς τους δύο αριθμούς, όπως βλέπουμε και στο σχήμα, το αποτέλεσμα είναι λανθασμένο.

Στη συνέχεια, θα αποδείξουμε μαθηματικά ότι, πράγματι, ο τρόπος που περιγράφηκε παραπάνω για την άθροιση προσημασμένων αριθμών οδηγεί σε σωστό αποτέλεσμα.

Έστω ότι θέλουμε να προσθέσουμε τις **παραστάσεις συμπληρώματος ως προς 2** δύο προσημασμένων αριθμών x και y. Πριν ξεκινήσουμε τη μαθηματική θεμελίωση του παραπάνω ισχυρισμού, επισημαίνουμε ότι το συμπλήρωμα ως προς 2 ενός οποιουδήποτε αριθμού α είναι ο αριθμός 2^n-α. Αν ο α είναι μη αρνητικός, τότε σε παράσταση συμπληρώματος ως προς 2 γράφεται ως |α|=α, ενώ αν είναι αρνητικός γράφεται ως 2^n-|α|=2^n+α.

- Αν και οι δύο αριθμοί x, y είναι θετικοί, κατά την πρόσθεση θα προστεθούν τα μέτρα τους |x| και |y|. Αφού $0 \le |x|, |y| \le 2^{n-1} - 1$, θα ισχύει $0 \le |x| + |y| \le 2^n - 2$. Δηλαδή το MSB του αριθμού μπορεί να προκύψει 0, αλλά μπορεί να προκύψει και 1. Αν βγει 1, τότε το αποτέλεσμα διαβάζεται ως αρνητικός αριθμός. Το λάθος οφείλεται στο γεγονός ότι ο αριθμός |x| + |y| έχει τιμή μεγαλύτερη από το μεγαλύτερο θετικό αριθμό που μπορεί να παρασταθεί με n bits στην παράσταση συμπληρώματος ως προς 2. Επομένως, η τιμή του αθροίσματος διαβάζεται κατά 2^n μικρότερη από την πραγματική.
- Αν ο ένας αριθμός (έστω ο x) είναι μη αρνητικός και ο άλλος αρνητικός, κατά την πρόσθεση θα προστεθούν οι αριθμοί |x| και 2^n-|y|. Επομένως, το αποτέλεσμα θα είναι το $z = 2^n + |x| - |y|$.
  - Έστω ότι $|x| \ge |y|$. Επειδή $0 \le |x| - |y| \le 2^{n-1} - 1$, ο αριθμός |x| − |y| μπορεί να γραφεί με n-1 δυαδικά ψηφία, ενώ ο 2^n γράφεται με ένα 1 ακολουθούμενο από n μηδενικά. Επομένως, κατά την πρόσθεση των |x| + (2^n-|y|), αυτό το 1 βγαίνει σαν κρατούμενο κατά την πρόσθεση των MSB των δύο προσθετέων. Αγνοώντας το κρατούμενο αυτό (είναι σαν να διαγράφουμε το 2^n από το z) παραμένει στο αποτέλεσμα ένας n-ψήφιος δυαδικός αριθμός με το MSB ίσο με 0, που είναι ο αριθμός |x| − |y|. (σωστό)
  - Έστω ότι $|x| < |y|$.Τότε ο z γράφεται ως $z = 2^n - (|y| - |x|)$ και ισχύει $0 < |y| - |x| \le 2^{n-1} \Rightarrow 2^{n-1} \le 2^n - (|y| - |x|) < 2^n$. Επομένως, δε θα προκύψει κρατούμενο κατά την άθροιση των αριθμών |x| και 2^n-|y|, ενώ το αποτέλεσμα θα διαβάζεται σαν αρνητικός αριθμός, αφού το MSB θα βγει ίσο με 1. (σωστό)
- Ανάλογα ισχύουν και όταν ο x είναι αρνητικός και ο y μη αρνητικός.
- Αν οι x και y είναι αρνητικοί, κατά την άθροιση θα προστεθούν οι αριθμοί $2^n - |x|$ και $2^n - |y|$, οπότε θα προκύψει ο αριθμός $z = 2^n + (2^n - (|x| + |y|))$. Όμως, $1 \le |x|, |y| \le 2^{n-1} \Rightarrow 2^n - 2 \ge 2^n - (|x| + |y|) \ge 0$. Δηλαδή, ο αριθμός $2^n - (|x| + |y|)$ μπορεί να παρασταθεί με n bits, ενώ η άλλη δύναμη 2^n που υπάρχει στην έκφραση του z είναι ένα 1 μια θέση αριστερά από το MSB του $2^n - (|x| + |y|)$, δηλαδή είναι κρατούμενο και περισσεύει, αφού ο υπολογιστής

διαθέτει μόνο n bits για κάθε αριθμό του. Αγνοώντας το κρατούμενο, το αποτέλεσμα γίνεται $z' = 2^n - (|x| + |y|)$. Αν $|x| + |y| \le 2^{n-1}$, το MSB του z΄ είναι 1 και ο z΄ παριστάνει έναν αρνητικό αριθμό με απόλυτη τιμή $|x| + |y|$. (σωστό) Αν, όμως, $|x| + |y| > 2^{n-1}$, τότε $0 \le 2^n - (|x| + |y|) \le 2^{n-1} - 1$ και ο z΄ έχει MSB ίσο με 0. Δηλαδή ο z΄ θα διαβάζεται σαν θετικός αριθμός, που δεν είναι σωστό. Αυτός ο θετικός αριθμός θα είναι κατά 2^n μεγαλύτερος από την πραγματική τιμή του αποτελέσματος. Το λάθος οφείλεται στο γεγονός ότι ο αριθμός $x + y = -|x| - |y|$ είναι έξω από το διάστημα των αριθμών που μπορούν να γραφούν με n bits στην παράσταση συμπληρώματος ως προς 2.

Ας δούμε τώρα τι συμβαίνει όταν προσθέτουμε τις **παραστάσεις συμπληρώματος ως προς 1** δύο προσημασμένων αριθμών x και y. Πριν ξεκινήσουμε τη διαδικασία αυτή επισημαίνουμε ότι το συμπλήρωμα ως προς 1 του αριθμού α ισούται με $2^n - 1 - \alpha$.

- Αν $x \ge 0$ και $y \ge 0$, κατά την άθροιση δεν προκύπτει κρατούμενο. Το αποτέλεσμα που παίρνουμε είναι σωστό εφόσον το x+y δεν είναι τόσο μεγάλο που να βγαίνει έξω από τα όρια του διαστήματος $[-2^{n-1} + 1, 2^{n-1} - 1]$ των αριθμών που μπορούν να γραφούν με n bits σε παράσταση συμπληρώματος ως προς 1.
- Αν $x \ge 0$ και $y \le 0$, προστίθενται τα $|x|$ και $2^n - 1 - |y|$, οπότε δίνουν αποτέλεσμα $z= 2^n - 1 + |x| - |y|$.
  - Αν $|x| > |y|$, ισχύει $|x| - |y| - 1 \ge 0$, οπότε ο 2^n που υπάρχει στην προηγούμενη έκφραση του z, δίνει τελικό κρατούμενο 1 στην άθροιση. Παίρνοντας αυτό το κρατούμενο από τον z και προσθέτοντάς το σαν 1 στο υπόλοιπο τμήμα του z, προκύπτει ο αριθμό $z' = |x| - |y|$, που είναι το σωστό αποτέλεσμα στο δυαδικό σύστημα
  - Αν $|x| \le |y|$, τότε $z= 2^n - 1 - (|y| - |x|)$, με $0 \le |y| - |x| \le 2^{n-1} - 1$. Επομένως, κατά την άθροιση δεν προκύπτει τελικό κρατούμενο και το αποτέλεσμα είναι το σωστό x + y, γραμμένο στην παράσταση συμπληρώματος ως προς 1.
- Ανάλογα με τα παραπάνω ισχύουν και όταν $x \le 0$ και $y \ge 0$.
- Αν $x \le 0$ και $y \le 0$, τότε προστίθενται τα $2^n - 1 - |x|$ και $2^n - 1 - |y|$ και δίνουν αποτέλεσμα $z= 2^n - 1 + 2^n - 1 - (|x| + |y|)$. Αφού $0 \le |x|, |y| \le 2^{n-1} - 1$, ισχύει $2^n - 2 \ge A= -1 + 2^n - 1 - (|x| + |y|) \ge 0$, που σημαίνει ότι ο A μπορεί να γραφεί με n bits. Επομένως, το 2^n που απομένει στην έκφραση του z, δίνει κατά την πρόσθεση τελικό κρατούμενο 1. Αν αυτό το κρατούμενο προστεθεί σαν μία μονάδα στο τέλος του z, προκύπτει ο αριθμός $z' = 2^n - 1 - (|x| + |y|)$. Αυτός είναι η σωστή παράσταση συμπληρώματος ως προς 1 του αρνητικού αριθμού x+y, εφόσον το άθροισμα αυτό δεν είναι μικρότερο από $-2^{n-1} + 1$ (δηλαδή δεν υπερχειλίζει).

6. Παράσταση πραγματικών αριθμών

Εκτός από τους ακέραιους αριθμούς, θέλουμε να παραστήσουμε στον υπολογιστή και «πραγματικούς» αριθμούς, δηλαδή αριθμούς με ακέραιο και κλασματικό μέρος.

6.1. Παράσταση σταθερής υποδιαστολής

Ένας απλός τρόπος για την παράσταση των πραγματικών αριθμών με λέξεις μήκους n bits είναι να μοιράσουμε τα ψηφία του αριθμού μεταξύ του ακέραιου και του κλασματικού μέρους. Δηλαδή, υποθέτουμε ότι υπάρχει η υποδιαστολή στο δεξιό μέρος κάποιου συγκεκριμένου ψηφίου από τα n που χρησιμοποιούνται για την παράσταση του αριθμού. Αυτή είναι η παράσταση *σταθερής υποδιαστολής* (fixed point representation). Στην περίπτωση αυτή ισχύουν όσα αναφέραμε στις προηγούμενες παραγράφους για την παράσταση των προσημασμένων αριθμών και τις πράξεις μεταξύ τους.

Για την παράσταση πραγματικών αριθμών έστω ότι διαθέτουμε 8 ψηφία. Αυτά μοιράζονται ως εξής: 5 ψηφία για το ακέραιο μέρος και 3 ψηφία για το κλασματικό. Η παράσταση των αρνητικών αριθμών γίνεται με το συμπλήρωμα του 2. Ο μεγαλύτερος θετικός αριθμός που μπορούμε να παραστήσουμε έχει την παράσταση 01111,111 και είναι ο 15,875. Ο μικρότερος αρνητικός αριθμός που μπορούμε να παραστήσουμε είναι ο 10000,000 που έχει την τιμή -16.

Οι πλησιέστεροι αριθμοί στο 0 που μπορούμε να παραστήσουμε (εκτός από το 0 βέβαια, που παριστάνεται ως 00000,000) είναι:

$00000{,}001_{(2)} = 0{,}125$ για τους θετικούς και

$11111{,}111_{(2)} = -0{,}125$ για τους αρνητικούς.

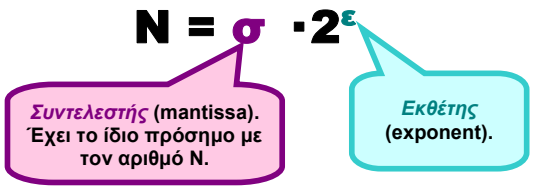
Το άθροισμα των αριθμών $01001{,}110_{(2)} = 9{,}75$ και $10010{,}001_{(2)} = -13{,}875$ υπολογίζεται απευθείας και είναι $11011{,}111_{(2)} = -4{,}125$

Το σπουδαιότερο μειονέκτημα της παραστάσεως σταθερής υποδιαστολής είναι ότι το διάστημα των αριθμών που μπορούν να παρασταθούν δεν είναι πολύ μεγάλο.

6.2. Παράσταση κινητής υποδιαστολής

Συνήθως στους υπολογιστές χρησιμοποιείται η παράσταση *κινητής υποδιαστολής* (floating point representation).

Στην παράσταση κινητής υποδιαστολής ο δυαδικός αριθμός N που θέλουμε να παραστήσουμε εκφράζεται σε *εκθετική μορφή* (exponential representation), σαν ένα γινόμενο, δηλαδή, ενός κλασματικού αριθμού και μιας δύναμης του 2.



Ο αριθμός $101{,}011_{(2)}$ σε εκθετική μορφή μπορεί να γραφτεί με διάφορες μορφές:

$0{,}101011 \cdot 2^3$ ⇨ $\sigma = 0{,}101011$, $\varepsilon = 3_{(10)} = 11_{(2)}$

$1{,}01011 \cdot 2^2$ ⇨ $\sigma = 1{,}01011$, $\varepsilon = 2_{(10)} = 10_{(2)}$

$10{,}1011 \cdot 2^1$ ⇨ $\sigma = 10{,}1011$, $\varepsilon = 1_{(10)} = 1_{(2)}$

> Το ίδιο συμβαίνει και στο δεκαδικό σύστημα: κάθε αριθμός μπορεί να γραφτεί με πολλές εκθετικές μορφές. Το 1023 π.χ. μπορεί να γραφτεί σαν $1,023 \cdot 10^3$, σαν $10,23 \cdot 10^2$, σαν $0,001023 \cdot 10^6$ κλπ.

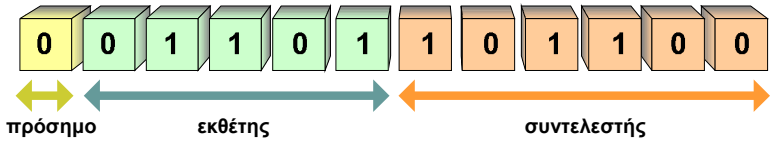
Βλέπουμε, λοιπόν, ότι υπάρχουν πολλές εναλλακτικές παραστάσεις ενός αριθμού σε εκθετική μορφή. Στους υπολογιστές έχει επιλεγεί μία από τις παραστάσεις αυτές, η οποία έχει την ιδιότητα $½ \le σ < 1$ και ονομάζεται *κανονική μορφή* (normal form).

Όταν ο συντελεστής είναι μεταξύ ½ και 1, έχει δύο χαρακτηριστικά:

- Το ακέραιο μέρος του είναι πάντα 0. Έτσι δε χρειάζεται να το αποθηκεύουμε, γιατί η τιμή του είναι γνωστή και δεδομένη.
- Το πρώτο του κλασματικό ψηφίο είναι πάντα 1. Αυτό συμβαίνει, γιατί, οι κλασματικοί αριθμοί που είναι μεγαλύτεροι από ½ ($=2^{-1}$), στο δυαδικό σύστημα περιέχουν πάντα τον προσθετέο 2^{-1}.

> Από όλες τις εκθετικές μορφές του αριθμού $101,011_{(2)}$ που είδαμε πιο πριν, η κανονική μορφή είναι η $0,101011 \cdot 2^3$.
>
> Η κανονική μορφή του αριθμού $0,000100_{(2)}$ είναι η $0,100 \cdot 2^{-3}$. Εδώ ο εκθέτης πρέπει να είναι αρνητικός, για να ικανοποιεί ο συντελεστής τη συνθήκη $½ \le σ < 1$.

Αν, λοιπόν, όλοι οι αριθμοί είναι εκφρασμένοι στην κανονική εκθετική μορφή, μπορούμε να τους κωδικοποιήσουμε αφιερώνοντας n_1 δυαδικά ψηφία στον εκθέτη και n_2 δυαδικά ψηφία στο συντελεστή, κρατώντας και ένα ψηφίο που θα κωδικοποιεί το πρόσημο του αριθμού, όπως βλέπουμε στο σχήμα.



Το κλασματικό μέρος του συντελεστή παριστάνεται σαν ένας δυαδικός αριθμός με n_1 ψηφία. Εάν ο συντελεστής έχει λιγότερα από n_1 ψηφία, προσθέτουμε μηδενικά στο τέλος, ενώ αν έχει περισσότερα από n_1 ψηφία, τότε τον στρογγυλοποιούμε. Στη *στρογγυλοποίηση* (rounding), αγνοούμε τα ψηφία που περισσεύουν, αλλά, εάν το πρώτο ψηφίο που περισσεύει είναι 1, τότε προσθέτουμε 1 στο λιγότερο σημαντικό ψηφίο του συντελεστή. Ο εκθέτης παριστάνεται και αυτός σαν ένας δυαδικός αριθμός με n_2 ψηφία.

> Το πιο σημαντικό από τα ψηφία της λέξης, που παίζει το ρόλο του πρόσημου: έχει την τιμή 0, αν ο αριθμός είναι θετικός και την τιμή 1 αν είναι αρνητικός. Αυτό είναι το πρόσημο του αριθμού· ο εκθέτης, όπως είδαμε, μπορεί να είναι αρνητικός, οπότε έχει το δικό του πρόσημο.

Οι πράξεις με πραγματικούς αριθμούς κινητής υποδιαστολής είναι πιο πολύπλοκες από ό,τι με τους ακεραίους.

Για να προσθέσουμε δύο πραγματικούς αριθμούς κινητής υποδιαστολής, πρέπει πρώτα να τους μετατρέψουμε ώστε να έχουν τον ίδιο εκθέτη. Αν ο ένας αριθμός έχει εκθέτη $ε_1$ και ο άλλος $ε_2$, και ισχύει $ε_1 < ε_2$, τότε αυξάνουμε τον $ε_1$ κατά $ε_2$-$ε_1$ και «ολισθαίνουμε» το

συντελεστή του αριθμού αυτού προς τα δεξιά κατά ε_2-ε_1 ψηφία. Κατά την ολίσθηση αυτή, τα δεξιότερα ψηφία του αριθμού χάνονται, έτσι ο αριθμός μπορεί να μεταβληθεί. Το τελικό αποτέλεσμα λοιπόν μπορεί να μην είναι ακριβές.

Στη συνέχεια προσθέτουμε τους συντελεστές των αριθμών και γράφουμε ξανά το αποτέλεσμα στην κανονική μορφή στρογγυλοποιώντας το συντελεστή. Σε όλες τις μετατροπές, όμως, το πλήθος των ψηφίων του συντελεστή παραμένει σταθερό.

Στην παράσταση κινητής υποδιαστολής με 8 ψηφία για το συντελεστή και 4 ψηφία για τον εκθέτη, ο αριθμός $x = 16{,}125_{(10)}$ παριστάνεται ως

$$0{,}10000001 \cdot 2^5$$

και ο αριθμός $y = 4{,}3125_{(10)}$ παριστάνεται ως

$$0{,}1000101 \cdot 2^3.$$

x = **0100000010101**
y = **0100010100011**

Πρώτα μετατρέπουμε τον αριθμό με το μικρότερο εκθέτη, που είναι ο y. Αυξάνουμε τον εκθέτη του κατά 2 και ολισθαίνουμε το συντελεστή του προς τα δεξιά κατά 2 ψηφία.

y = **0001000100101**

Στη συνέχεια προσθέτουμε τους δύο συντελεστές. Το άθροισμα δε χρειάζεται κανονικοποίηση, άρα είναι και το τελικό αποτέλεσμα. Η τιμή του είναι

$$0{,}10100.011 \cdot 2^5,$$

x+y = **0101000110101**

δηλαδή 20,375. Το σωστό αποτέλεσμα της πρόσθεσης όμως είναι 20,4375. Το σφάλμα οφείλεται στη μετατροπή του y ώστε να έχει τον ίδιο εκθέτη με το x.

Οι πράξεις του πολλαπλασιασμού και της διαίρεσης με πραγματικούς αριθμούς κινητής υποδιαστολής είναι πιο εύκολες.

Για να πολλαπλασιάσουμε δύο αριθμούς, προσθέτουμε τους εκθέτες τους και πολλαπλασιάζουμε τους συντελεστές. Μετά φέρνουμε πάλι το αποτέλεσμα στην κανονική μορφή. Για να διαιρέσουμε δύο αριθμούς, αφαιρούμε τους εκθέτες και διαιρούμε τους συντελεστές.

$$(\sigma_1 \cdot 2^{\varepsilon 1}) \cdot (\sigma_2 \cdot 2^{\varepsilon 2}) = (\sigma_1 \cdot \sigma_2) \cdot 2^{\varepsilon 1+\varepsilon 2}$$
$$(\sigma_1 \cdot 2^{\varepsilon 1}) / (\sigma_2 \cdot 2^{\varepsilon 2}) = (\sigma_1 / \sigma_2) \cdot 2^{\varepsilon 1-\varepsilon 2}$$

Όπως είδαμε και στο προηγούμενο παράδειγμα, πολλές φορές η μετατροπή ή η στρογγυλοποίηση που κάνουμε στους αριθμούς κατά την εκτέλεση των πράξεων επιφέρει ένα μικρό σφάλμα στο αποτέλεσμα. Μετά από μία μεγάλη σειρά πράξεων, λοιπόν, τα σφάλματα αυτά «συσσωρεύονται», οπότε τα αποτελέσματα μπορεί να μην είναι ικανοποιητικά.

6.2.1. Ακρίβεια αριθμών σε παράσταση κινητής υποδιαστολής

Το πλήθος n_1 των ψηφίων που χρησιμοποιούνται για τον συντελεστή καθορίζουν την ακρίβεια παραστάσεως των αριθμών και το πλήθος n_2 των ψηφίων που χρησιμοποιούνται για τον εκθέτη καθορίζουν το διάστημα των αριθμών που μπορούμε να παραστήσουμε.

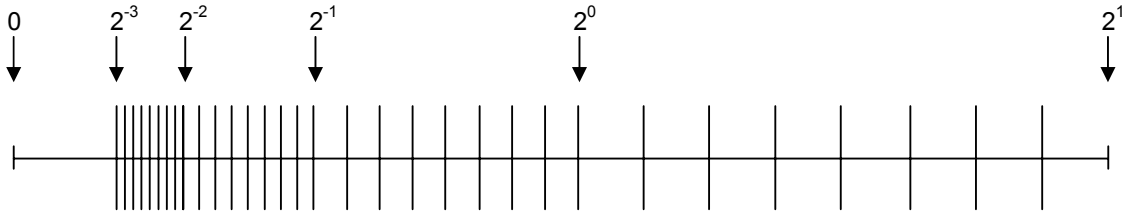
Συγκεκριμένα, η ακρίβεια της παράστασης εξαρτάται από το πλήθος των διαφορετικών αριθμών που μπορούν να αναπαρασταθούν σε κανονική μορφή με τον ίδιο εκθέτη ε. Το πλήθος των διαφορετικών αριθμών που έχουν τον ίδιο εκθέτη ε εξαρτάται από το πλήθος n_1 των ψηφίων του συντελεστή.

Όταν, λοιπόν, χρησιμοποιούμε για την αναπαράσταση ενός πραγματικού αριθμού τον πιο κοντινό σε τιμή κανονικοποιημένο δυαδικό $0,1\underbrace{ddddd}_{n_1 \text{ bits}}\times 2^{\varepsilon}$, το απόλυτο σφάλμα λόγω στρογγυλοποίησης μπορεί να είναι το πολύ $0,\underbrace{0000001}_{n_1+1 \text{ bits}}\times 2^{\varepsilon}=2^{-n1-1}\times 2^{\varepsilon}$. Όλοι οι αριθμοί της μορφής $0,1\underbrace{ddddd}_{n_1 \text{ bits}}\times 2^{\varepsilon}$, που έχουν τον ίδιο εκθέτη ε, έχουν το ίδιο απόλυτο σφάλμα ακρίβειας, που είναι το πολύ $2^{-n1-1}\times 2^{\varepsilon}$.

Οι αριθμοί αυτοί κυμαίνονται μεταξύ $2^{\varepsilon-1}$ και 2^{ε}. Είναι φανερό ότι αν διαθέτουμε n_1 ψηφία συντελεστή, η απόσταση μεταξύ δύο διαδοχικών αριθμών μεγαλώνει όσο μεγαλώνει ο εκθέτης ε. Επομένως, με την παράσταση κινητής υποδιαστολής, η ακρίβεια είναι μεγάλη για μικρούς αριθμούς (μικρές τιμές του ε), ενώ μικραίνει όσο μεγαλώνουν οι αριθμοί (μεγάλες τιμές του ε). Πάντα, όμως, είναι σταθερή (απόλυτη ακρίβεια) για τους 2^{n1-1} διαφορετικούς αριθμούς που ανήκουν στην περιοχή μεταξύ $2^{\varepsilon-1}$ και 2^{ε}.

Τέλος, το πλήθος n_2 των ψηφίων του εκθέτη προσδιορίζει τον μέγιστο και ελάχιστο εκθέτη ε, άρα το εύρος των αριθμών που μπορούν να παρασταθούν με τη συγκεκριμένη μορφή παράστασης κινητής υποδιαστολής.

Οι έννοιες αυτές δίδονται παραστατικά στο επόμενο σχήμα, στο οποίο έχουμε αναπαραστήσει σε μία ευθεία γραμμή τους θετικούς πραγματικούς αριθμούς που μπορούν να αναπαρασταθούν αν αφιερώνονται 2 bits για τον εκθέτη και 4 bits για το συντελεστή.



Έστω ότι έχουμε n_1=4 bits για το συντελεστή.

Ο δεκαδικός αριθμός 0,0625 γράφεται $0,0001_{<2>} = 0,1000\times 2^{-3}$.

Ο αμέσως μεγαλύτερος αριθμός που μπορεί να παρασταθεί με 4 bits συντελεστή είναι ο $0,1001\times 2^{-3} = 0,0703125$. Αν θέλουμε να αναπαραστήσουμε κάποιο πραγματικό αριθμό μεταξύ των 0,0625 και 0,0703125, τότε, λόγω έλλειψης αρκετής ακρίβειας, ο πραγματικός αριθμός θα στρογγυλοποιηθεί στον πλησιέστερο εκ των δύο. Το σύνολο των διαφορετικών αριθμών που μπορούμε να αναπαραστήσουμε στην περιοχή $[2^{-4}, 2^{-3})$ είναι 2^{n1-1}=8. Το μέγιστο σφάλμα λόγω στρογγυλοποίησης είναι $2^{-n1-1}\times 2^{\varepsilon} = 2^{-8} = 0,00390625$.

Όμοια, το σύνολο των διαφορετικών αριθμών που μπορούμε να αναπαραστήσουμε στην περιοχή $[2^{2}, 2^{3})$ είναι 2^{n1-1}=8. Ο δεκαδικός αριθμός 4 γράφεται $0,1000\times 2^{3}$ και ο αμέσως επόμενος αριθμός που μπορεί να αναπαρασταθεί με 4 bits είναι ο $0,1001\times 2^{3}$=4,5.

Ο 4,2 γράφεται: $0,100000110011001\ldots 2^{3}=0,10002^{3}$, όμοια με τον 4 (σφάλμα λόγω έλλειψης αρκετής ακρίβειας). Στην περιοχή, λοιπόν, $[2^{2}, 2^{3})$ η ακρίβεια είναι διαφορετική

και το μέγιστο σφάλμα λόγω στρογγυλοποίησης είναι πλέον $2^{-n1-1} \times 2^{\varepsilon} = 0,25$, πολύ μεγαλύτερο από πριν.

6.2.2. Παράσταση κινητής υποδιαστολής σύμφωνα με το πρότυπο IEEE 754

Η μορφή που παρουσιάσαμε παραπάνω για την παράσταση των αριθμών κινητής υποδιαστολής στον ΕΚΥ είναι αρκετά απλή και καλή για εκπαιδευτικούς σκοπούς. Η απλότητα αυτή, όμως, δεν την καθιστά κατάλληλη για χρήση σε πραγματικούς υπολογιστές.

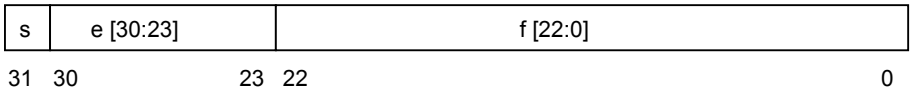
Για παράδειγμα, σύμφωνα με όσα είπαμε παραπάνω, το πρώτο bit της παράστασης του συντελεστή πρέπει πάντα να ισούται με 1. Αυτό σημαίνει ότι το συγκεκριμένο bit ουσιαστικά δεν παρέχει καμία απολύτως πληροφορία, ή ότι 1 bit της λέξης του υπολογιστή παραμένει αναξιοποίητο.

Επίσης, ο αριθμός 0, αφού η δυαδική αναπαράστασή του δεν περιέχει κανένα 1, δεν είναι δυνατόν να μετατραπεί σε κανονική μορφή, ούτε να παρασταθεί σύμφωνα με το πρότυπο κινητής υποδιαστολής που παρουσιάσαμε προηγουμένως.

Στη συνέχεια, ως παράδειγμα πραγματικών αναπαραστάσεων κινητής υποδιαστολής, θα δούμε το πρότυπο IEEE 754, το οποίο έχει χρησιμοποιηθεί ευρέως σε πραγματικούς υπολογιστές. Το πρότυπο αυτό καθορίζει δύο βασικές μορφές κινητής υποδιαστολής: *απλής* και *διπλής ακρίβειας*.

6.2.2.1. Αναπαράσταση απλής ακρίβειας

Η αναπαράσταση ενός αριθμού, σύμφωνα με τη μορφή απλής ακρίβειας του προτύπου IEEE, αποτελείται από τρία πεδία: το συντελεστή f (23 bits), τον πολωμένο εκθέτη e (8 bits) και το πρόσημο s (1 bit). Τα πεδία αυτά αποθηκεύονται συνεχόμενα σε μία λέξη 32 bits του υπολογιστή, όπως φαίνεται στο παρακάτω σχήμα. Τα bits 0-22 περιέχουν το συντελεστή f. Το bit 0 είναι το λιγότερο σημαντικό, ενώ το bit 22 είναι το πιο σημαντικό bit του συντελεστή. Τα bits 23-30 περιέχουν τον πολωμένο εκθέτη. Το bit 23 είναι το λιγότερο σημαντικό, ενώ το bit 30 είναι το πιο σημαντικό bit του πολωμένου εκθέτη. Το bit 31 αναπαριστά το πρόσημο του αριθμού.



Ο παρακάτω πίνακας δείχνει την αντιστοιχία μεταξύ των τιμών των τριών πεδίων s, e, f, και της τιμής του πραγματικού αριθμού που αναπαρίσταται. Το σύμβολο u σημαίνει «αδιάφορο», δηλαδή η τιμή του συγκεκριμένου πεδίου δεν επηρεάζει τον υπολογισμό της τιμής του αντίστοιχου πραγματικού αριθμού.

Αναπαράσταση απλής ακρίβειας	*Τιμή πραγματικού αριθμού*
$0 < e < 255$	$(-1)^s \times 2^{e-127} \times 1,f$ (κανονικοί αριθμοί)
$e = 0, f \neq 0$ (τουλάχιστον ένα bit του f είναι μη μηδενικό)	$(-1)^s \times 2^{-126} \times 0,f$ (υπο-κανονικοί αριθμοί)
$e = 0, f = 0$ (όλα τα bits του f είναι μηδενικά)	$(-1)^s \times 0,0$ (προσημασμένο μηδέν)
$s = 0, e = 255, f = 0$ (όλα τα bits του f είναι μηδενικά)	+INF (συν άπειρο)
$s = 1, e = 255, f = 0$ (όλα τα bits του f είναι μηδενικά)	-INF (μείον άπειρο)
$s = u, e = 255, f \neq 0$ (τουλάχιστον ένα bit του f είναι μη μηδενικό)	NaN (Not a Number)

Παρατηρούμε ότι, όταν $e < 255$, η τιμή του αντίστοιχου πραγματικού αριθμού υπολογίζεται τοποθετώντας την υποδιαστολή αμέσως πριν το πιο σημαντικό bit του συντελεστή και ένα bit (0 ή 1) πριν την υποδιαστολή. Το bit αυτό, που τοποθετούμε μπροστά από την υποδιαστολή, ονομάζεται «έμμεσο» bit, επειδή η τιμή του δεν δίδεται απευθείας από την αναπαράσταση του συντελεστή σε δυαδική μορφή, αλλά υπονοείται από την τιμή του πολωμένου εκθέτη.

Η διαφορά μεταξύ κανονικών και υπο-κανονικών αριθμών έγκειται στο ότι για τους μεν κανονικούς αριθμούς η τιμή του bit που τοποθετείται πριν την υποδιαστολή είναι το 1, ενώ για τους δε υπο-κανονικούς τοποθετείται το 0.

Επομένως, το δεκαδικό τμήμα των 23 bits, σε συνδυασμό με το έμμεσο πιο σημαντικό bit, παρέχει ακρίβεια 24 bits στους κανονικούς αριθμούς απλής ακρίβειας.

Στον παρακάτω πίνακα δίδουμε παραδείγματα αναπαραστάσεων σε μορφή απλής ακρίβειας. Ο μέγιστος θετικός κανονικός αριθμός είναι ο μέγιστος πεπερασμένος αριθμός που μπορεί να αναπαρασταθεί σε μορφή απλής ακρίβειας κατά το πρότυπο της IEEE.

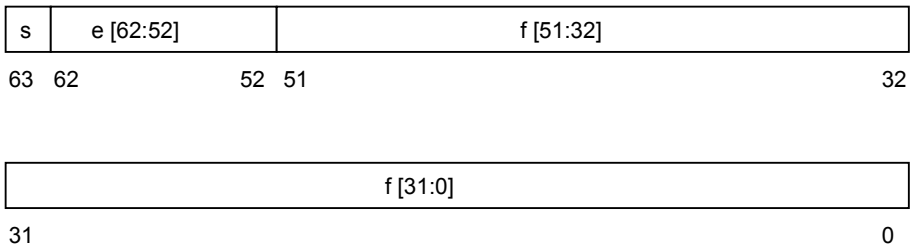
Ονομασία	*Δυαδική αναπαράσταση (σε δεκαεξαδικό)*	*Δεκαδική τιμή*
+0	00000000	0,0
-0	80000000	-0,0
1	3f800000	1,0
2	40000000	2,0
μέγιστος κανονικός αριθμός	7f7fffff	3,40282347e+38
ελάχιστος θετικός κανονικός αριθμός	00800000	1,17549435e-38
μέγιστος υπο-κανονικός αριθμός	007fffff	1,17549421e-38
ελάχιστος θετικός υπο-κανονικός αριθμός	00000001	1,40129846e-45
+∞	7f800000	άπειρο
-∞	ff800000	μείον άπειρο
Not-a-Number	7fc00000	NaN

Ο Not-a-Number μπορεί να αναπαρασταθεί με οποιαδήποτε μορφή δίνει εκθέτη e=255 και συντελεστή f≠0. Η δυαδική αναπαράσταση του προηγούμενου πίνακα είναι απλώς μία από τις πολλές που μπορούν να χρησιμοποιηθούν για την αναπαράσταση του NaN.

6.2.2.2. Αναπαράσταση διπλής ακρίβειας

Η αναπαράσταση ενός αριθμού, σύμφωνα με τη μορφή διπλής ακρίβειας του προτύπου IEEE, αποτελείται από τρία πεδία: το συντελεστή f (52 bits), τον πολωμένο εκθέτη e (11 bits) και το πρόσημο s (1 bit). Τα πεδία αυτά αποθηκεύονται συνεχόμενα σε δύο λέξεις 32 bits του υπολογιστή, όπως φαίνεται στο παρακάτω σχήμα. Τα bits 0-51 περιέχουν το συντελεστή f. Το bit 0 είναι το λιγότερο σημαντικό, ενώ το bit 51 είναι το πιο σημαντικό bit του συντελεστή. Τα bits 52-62 περιέχουν τον πολωμένο εκθέτη. Το bit 52 είναι το λιγότερο σημαντικό, ενώ το bit 62 είναι το πιο σημαντικό bit του πολωμένου εκθέτη. Το bit 63 αναπαριστά το πρόσημο του αριθμού.

Στην αρχιτεκτονική του SPARC, η λέξη 32 bits με τη μεγαλύτερη διεύθυνση περιέχει τα 32 λιγότερο σημαντικά bits της δυαδικής αναπαράστασης, ενώ στην αρχιτεκτονική των Intel και PowerPC, η λέξη 32 bits με τη μικρότερη διεύθυνση περιέχει τα 32 λιγότερο σημαντικά bits της δυαδικής αναπαράστασης.



Ο παρακάτω πίνακας δείχνει την αντιστοιχία μεταξύ των τιμών των τριών πεδίων s, e, f, και της τιμής του πραγματικού αριθμού που αναπαρίσταται. Το σύμβολο u σημαίνει «αδιάφορο», δηλαδή η τιμή του συγκεκριμένου πεδίου δεν επηρεάζει τον υπολογισμό της τιμής του αντίστοιχου πραγματικού αριθμού.

Αναπαράσταση απλής ακρίβειας	***Τιμή πραγματικού αριθμού***
$0 < e < 2047$	$(-1)^s \times 2^{e-1023} \times 1,f$ (κανονικοί αριθμοί)
$e = 0, f \neq 0$ (τουλάχιστον ένα bit του f είναι μη μηδενικό)	$(-1)^s \times 2^{-1023} \times 0,f$ (υπο-κανονικοί αριθμοί)
$e = 0, f = 0$ (όλα τα bits του f είναι μηδενικά)	$(-1)^s \times 0,0$ (προσημασμένο μηδέν)
$s = 0, e = 2047, f = 0$ (όλα τα bits του f είναι μηδενικά)	+INF (συν άπειρο)
$s = 1, e = 2047, f = 0$ (όλα τα bits του f είναι μηδενικά)	-INF (μείον άπειρο)
$s = u, e = 2047, f \neq 0$ (τουλάχιστον ένα bit του f είναι μη μηδενικό)	NaN (Not a Number)

Παρατηρούμε ότι, όταν $e < 2047$, η τιμή του αντίστοιχου πραγματικού αριθμού υπολογίζεται τοποθετώντας την υποδιαστολή αμέσως πριν το πιο σημαντικό bit του συντελεστή και ένα bit (0 ή 1) πριν την υποδιαστολή. Το bit αυτό, που τοποθετούμε μπροστά από την υποδιαστολή, ονομάζεται «έμμεσο» bit, επειδή η τιμή του δεν δίδεται απευθείας από την αναπαράσταση του συντελεστή σε δυαδική μορφή, αλλά υπονοείται από την τιμή του πολωμένου εκθέτη.

Η διαφορά μεταξύ κανονικών και υπο-κανονικών αριθμών έγκειται στο ότι για τους μεν κανονικούς αριθμούς η τιμή του bit, που τοποθετείται πριν την υποδιαστολή, είναι το 1, ενώ για τους δε υπο-κανονικούς τοποθετείται το 0.

Επομένως, το δεκαδικό τμήμα των 52 bits, σε συνδυασμό με το έμμεσο πιο σημαντικό bit, παρέχει ακρίβεια 53 bits στους κανονικούς αριθμούς διπλής ακρίβειας.

Στον παρακάτω πίνακα δίδουμε παραδείγματα αναπαραστάσεων σε μορφή διπλής ακρίβειας. Οι δυαδικές αναπαραστάσεις της δεύτερης στήλης δίδονται σαν δύο οκταψήφιοι δεκαεξαδικοί αριθμοί. Στην αρχιτεκτονική του SPARC, ο αριστερός αποτελεί την τιμή της λέξης με τη μικρότερη διεύθυνση, ενώ στην αρχιτεκτονική των Intel και PowerPC, αποτελεί

την τιμή της λέξης με τη μεγαλύτερη διεύθυνση. Ο μέγιστος θετικός κανονικός αριθμός είναι ο μέγιστος πεπερασμένος αριθμός που μπορεί να αναπαρασταθεί σε μορφή διπλής ακρίβειας κατά το πρότυπο της IEEE.

Ονομασία	***Δυαδική αναπαράσταση (σε δεκαεξαδικό)***	***Δεκαδική τιμή***
+0	`00000000 00000000`	0,0
-0	`80000000 00000000`	-0,0
1	`3f800000 00000000`	1,0
2	`40000000 00000000`	2,0
μέγιστος κανονικός αριθμός	`7f7fffff ffffffff`	1,7976931348623157e+308
ελάχιστος θετικός κανονικός αριθμός	`00100000 00000000`	2,2250738585072014e-308
μέγιστος υπο-κανονικός αριθμός	`000fffff ffffffff`	2,2250738585072009e-308
ελάχιστος θετικός υπο-κανονικός αριθμός	`00000000 00000001`	4,9406564584124654e-324
+∞	`7ff00000 00000000`	άπειρο
-∞	`fff00000 00000000`	μείον άπειρο
Not-a-Number	`7ff80000 00000000`	NaN

Ο Not-a-Number μπορεί να αναπαρασταθεί με οποιαδήποτε μορφή δίνει εκθέτη e=2047και συντελεστή f≠0. Η δυαδική αναπαράσταση του προηγούμενου πίνακα είναι απλώς μία από τις πολλές που μπορούν να χρησιμοποιηθούν για την αναπαράσταση του NaN.

7. Ασκήσεις

7.1. Να μετατραπούν οι δεκαδικοί αριθμοί 167,32 και 93,01 σε δεκαεξαδικούς.

Για κάθε αριθμό μετατρέπουμε ξεχωριστά το ακέραιο και το κλασματικό τμήμα στις αντίστοιχες δεκαεξαδικές παραστάσεις τους:

- Η μετατροπή του ακέραιου μέρους από δεκαδικό σε δεκαεξαδικό γίνεται με διαδοχικές διαιρέσεις του ακεραίου με το 16. Τα διαδοχικά υπόλοιπα των διαιρέσεων δίνουν τα δεκαεξαδικά ψηφία του αριθμού (με αύξουσα σειρά τάξης μεγέθους), ενώ τα πηλίκα που προκύπτουν αποτελούν τους διαιρετέους των επόμενων διαιρέσεων. Η διαδικασία τελειώνει όταν το πηλίκο είναι < 16.
- Η μετατροπή του κλασματικού μέρους από δεκαδικό σε δεκαεξαδικό γίνεται με διαδοχικούς πολλαπλασιασμούς με το 16. Τα διαδοχικά ακέραια μέρη που προκύπτουν από τους πολ/σμούς δίνουν (με φθίνουσα τάξη μεγέθους) τα δεκαεξικά ψηφία, όπως φαίνεται στη συνέχεια:

167,32 = 167 + 0,32

167	16
7	10 < 16
	A

0,32 × 16 = **5**,12
0,12 × 16 = **1**,92
0,92 × 16 = **14**,72 → **E**
0,72 × 16 = **11**,52 → **B**
0,52 × 16 = **8**,32
0,32 × 16 = **5**,12 (επαναλαμβάνονται στο εξής τα ίδια ψηφία)

167 → A7

0,32 → 0, $\overline{51EA8}$

Άρα: $167{,}32_{<10>} = A7{,}\overline{51EB8}_{<16>}$

93,01 = 93 + 0,01

93 | 16
13 | 5 < 16
D

0,01 × 16 = **0**,16
0,16 × 16 = **2**,56
0,56 × 16 = **8**,96
0,96 × 16 = **15**,36 → **F**
0,36 × 16 = **5**,76
0,76 × 16 = **12**,16 → **C**
0,16 × 16 = **2**,56 (επαναλαμβάνονται τα ίδια ψηφία)

93 → 5D

0,01 → 0,0$\overline{28F5C}$

Άρα: ***93,01$_{<10>}$ = 5D,$\overline{028F5C}$$_{<16>}$***

7.2. Να μετατραπούν οι δεκαδικοί αριθμοί A = -32, B = 64 σε δυαδικούς για υπολογιστή που έχει λέξη των 8 ψηφίων και χρησιμοποιεί

α)παράσταση συμπληρώματος ως προς ένα.

β) παράσταση συμπληρώματος ως προς δύο.

Να υπολογισθούν οι αριθμοί A+B, A-B, B-A για κάθε μία από τις παραπάνω περιπτώσεις.

Για την μετατροπή της απόλυτης τιμής των δεκαδικών αριθμών σε δυαδικούς ακολουθούμε την ίδια διαδικασία με την άσκηση 1, με τη διαφορά ότι στην περίπτωσή μας διαιρούμε (αφού οι αριθμοί Α και Β είναι ακέραιοι) με το 2 (και όχι με το 16)

|A| = 32

32 | 2 → 0
16 | 2 → 0
8 | 2 → 0
4 | 2 → 0
2 | 2 → 0
1

B = 64

64 | 2 → 0
32 | 2 → 0
16 | 2 → 0
8 | 2 → 0
4 | 2 → 0
2 | 2 → 0
1

Άρα: $\mathbf{32_{<10>} = 100000_{<2>} = 0100000_{<2>}}$ και $\mathbf{64_{<10>} = 1000000_{<2>}}$

α) Για το σχηματισμό της παράστασης συμπληρώματος ως προς ένα, όταν πρόκειται για αρνητικό αριθμό, αντιστρέφουμε όλα τα ψηφία του. Οι θετικοί αριθμοί παραμένουν αμετάβλητοι: (το πρώτο ψηφίο υποδηλώνει το πρόσημο του αριθμού)

Επομένως $A = -32_{<10>} = 11011111_{<2>}$

$-A = 32_{<10>} = 00100000_{<2>}$

και

$B = 64<10> = 01000000_{<2>}$

$-B = -64<10> = 10111111_{<2>}$

A + B :

$$\begin{array}{rr} & 11011111 \\ + & 01000000 \\ \hline 1 & 00011111 \end{array}$$

υπερχείλιση : 00011111 + 1 = 00100000

Άρα: $\mathbf{A + B = 00100000_{<2>}}$

A - B = A + (-B) :

$$\begin{array}{rr} & 11011111 \\ + & 10111111 \\ \hline 1 & 10011110 \end{array}$$

υπερχείλιση : 10011110 + 1 = 10011111

Άρα: $\mathbf{A - B = 10011111_{<2>}}$

B - A = B + (-A) :

$$\begin{array}{rr} & 01000000 \\ + & 00100000 \\ \hline & 01100000 \end{array}$$

Άρα: $\mathbf{B - A = 01100000_{<2>}}$

β) Για το σχηματισμό της παράστασης συμπληρώματος ως προς δύο, στους αρνητικούς αριθμούς προσθέτουμε 1 στην παράσταση του συμπληρώματος ως προς ένα.

Επομένως $A = -32_{<10>} = 11100000_{<2>}$

$-A = 32<10> = 00100000_{<2>}$

και

$B = 64<10> = 01000000_{<2>}$

$-B = -64<10> = 11000000_{<2>}$

A + B :

```
  11100000
+ 01000000
----------
1 00100000
```

υπερχείλιση : αγνοείται

Άρα: ***A + B = 00100000*** $_{<2>}$

A - B = A + (-B) :

```
  11100000
+ 11000000
----------
1 10100000
```

υπερχείλιση : αγνοείται

Άρα: ***A - B = 10100000*** $_{<2>}$

B - A = B + (-A) :

```
  01000000
+ 00100000
----------
  01100000
```

Άρα: ***B - A = 01100001*** $_{<2>}$

7.3. α) Να παρασταθούν στο δυαδικό σύστημα οι κλασματικοί αριθμοί: 0,1, 0,2, 0,3, 0,4, … 0,9. Τι παρατηρείτε;

β) Σε Η/Υ που διαθέτει λέξη με 3 κλασματικά δυαδικά ψηφία να παρασταθούν οι παραπάνω αριθμοί.

α) Η μετατροπή των κλασματικών αριθμών από δεκαδικοί σε δυαδικοί γίνεται με διαδοχικούς πολλαπλασιασμούς με το 2, όπως περιγράφτηκε στην άσκηση 1.

0,1

0,1 × 2 = **0**,2

0,2 × 2 = **0**,4

0,4 × 2 = **0**,8

0,8 × 2 = **1**,6

0,6 × 2 = **1**,2

0,2 × 2 = **0**,4 (επαναλαμβάνονται στο εξής τα ίδια ψηφία)

Άρα: $0{,}1_{<10>} = 0{,}\overline{00011}_{<2>}$

0,2

0,2 × 2 = **0**,4

0,4 × 2 = **0**,8

0,8 × 2 = **1**,6

0,6 × 2 = **1**,2

0,2 × 2 = **0**,4 (επαναλαμβάνονται στο εξής τα ίδια ψηφία)

Άρα: $0{,}2_{<10>} = 0{,}\overline{0011}_{<2>}$

0,3

0,3 × 2 = **0**,6

0,6 × 2 = **1**,2

0,2 × 2 = **0**,4

0,4 × 2 = **0**,8

0,8 × 2 = **1**,6

0,6 × 2 = **1**,2 (επαναλαμβάνονται στο εξής τα ίδια ψηφία)

Άρα: $0{,}3_{<10>} = 0{,}\overline{01001}_{<2>}$

0,4

0,4 × 2 = **0**,8

0,8 × 2 = **1**,6

0,6 × 2 = **1**,2

0,2 × 2 = **0**,4

0,4 × 2 = **0**,8 (επαναλαμβάνονται στο εξής τα ίδια ψηφία)

Άρα: $0{,}4_{<10>} = 0{,}\overline{0110}_{<2>}$

0,5

0,5 × 2 = **1**,0

Άρα: $0{,}5_{<10>} = 0{,}1_{<2>}$

0,6 0,6 × 2 = **1**,2

0,2 × 2 = **0**,4

0,4 × 2 = **0**,8

0,8 × 2 = **1**,6

0,6 × 2 = **1**,2 (επαναλαμβάνονται στο εξής τα ίδια ψηφία)

Άρα: $0,6_{<10>} = 0,\overline{1001}_{<2>}$

0,7 0,7 × 2 = **1**,4

0,4 × 2 = **0**,8

0,8 × 2 = **1**,6

0,6 × 2 = **1**,2

0,2 × 2 = **0**,4

0,4 × 2 = **0**,8 (επαναλαμβάνονται στο εξής τα ίδια ψηφία)

Άρα: $0,7_{<10>} = 0,1\overline{0110}_{<2>}$

0,8 0,8 × 2 = **1**,6

0,6 × 2 = **1**,2

0,2 × 2 = **0**,4

0,4 × 2 = **0**,8

0,8 × 2 = **1**,6 (επαναλαμβάνονται στο εξής τα ίδια ψηφία)

Άρα: $0,8_{<10>} = 0,\overline{1100}_{<2>}$

0,9 0,9 × 2 = **1**,8

0,8 × 2 = **1**,6

0,6 × 2 = **1**,2

0,2 × 2 = **0**,4

0,4 × 2 = **0**,8

0,8 × 2 = **1**,6 (επαναλαμβάνονται στο εξής τα ίδια ψηφία)

Άρα: $\underline{\boldsymbol{0,9_{<10>} = 0,1\overline{11100}_{<2>}}}$

Συνοψίζοντας έχουμε:

$0,1_{<10>} = 0,0001100110011\ldots_{<2>}$

$0,2_{<10>} = 0,001100110011\ldots_{<2>}$

$0,4_{<10>} = 0,011001100110\ldots_{<2>}$

$0,8_{<10>} = 0,110011001100\ldots_{<2>}$

Παρατηρούμε ότι ο πολ/μος με το 2 ισοδυναμεί με αριστερή ολίσθηση της δυαδικής παράστασης του αριθμού κατά 1 θέση.

$0,3_{<10>} = 0,0100110011001\ldots_{<2>}$

$0,6_{<10>} = 0,100110011001\ldots_{<2>}$

Προκύπτει το ίδιο συμπέρασμα, όπως παραπάνω. Επιπλέον, οι δυαδική παράσταση του 0,3 μπορεί να υπολογιστεί με πρόσθεση των δυαδικών 0,1 + 0,2

$0,5_{<10>} = 0,1_{<2>}$

$0,7_{<10>} = 0,1011001100110\ldots_{<2>} = 0,5 + 0,2$

$0,9_{<10>} = 0,1110011001100\ldots_{<2>} = 0,5 + 0,4$

β) Με στρογγυλοποίηση προκύπτει:

$0,1_{<10>} = 0,0001\ldots_{<2>} = 0,001_{<2>}$

$0,2_{<10>} = 0,0011\ldots_{<2>} = 0,010_{<2>}$

$0,3_{<10>} = 0,0100\ldots_{<2>} = 0,010_{<2>}$

$0,4_{<10>} = 0,0110\ldots_{<2>} = 0,011_{<2>}$

$0,5_{<10>} = 0,1_{<2>} = 0,100_{<2>}$

$0,6_{<10>} = 0,1001\ldots_{<2>} = 0,101_{<2>}$

$0,7_{<10>} = 0,1011\ldots_{<2>} = 0,110_{<2>}$

$0,8_{<10>} = 0,1100\ldots_{<2>} = 0,110_{<2>}$

$0,9_{<10>} = 0,1110\ldots_{<2>} = 0,111_{<2>}$

Οπότε, κάποιοι από τους δυαδικούς αριθμούς συμπίπτουν.

7.4. Στην μνήμη ενός Η/Υ με λέξη των 16 ψηφίων (1 bit: πρόσημο, 6 bits: εκθετικό μέρος και 9 bits: συντελεστής) είναι τοποθετημένοι σε κανονική μορφή οι αριθμοί:

i) **1 111111 010100000**

ii) **0 000111 100100000**

Να αναγνωρισθούν.

Εφόσον το εκθετικό μέρος παριστάνεται με 6 ψηφία, υπάρχουν $2^6 = 64$ διαφορετικοί εκθέτες που παριστάνονται ως εξής:

Δυαδική παράσταση		Εκθέτης
000000	→	-32
000001	→	-31
…	→	…
011111	→	-1
100000	→	0
100001	→	1
…	→	…
111111	→	31

i) Βρίσκουμε το συμπλήρωμα ως προς ένα ολόκληρου του αριθμού (αφού πρόκειται για αρνητικό αριθμό):

1 111111 010100000

→ **0 000000 101011111**

Υπολογίζουμε ξεχωριστά εκθέτη και κλασματικό μέρος:

Εκθέτης: **000000 = -32**

Κλασμ. μέρος: $\mathbf{101011111 = 2^{-1} + 2^{-3} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} \approx 0{,}6855}$

Επομένως πρόκειται για τον αριθμό $\boldsymbol{\underline{-2^{-32} \cdot 0{,}6855}}$

ii) Δε χρειάζεται να βρούμε το συμπλήρωμα ως προς ένα (ο αριθμός είναι θετικός):

0 000111 100100000

Υπολογίζουμε ξεχωριστά εκθέτη και κλασματικό μέρος:

Εκθέτης: **000111 = -25**

Κλασμ. μέρος: $\mathbf{100100000 = 2^{-1} + 2^{-4} \approx 0{,}5625}$

Επομένως πρόκειται για τον αριθμό ***– $2^{-25} \cdot 0{,}5625$***

7.5. Για τον Η/Υ της άσκησης 7.4 να προσδιορισθούν :

α) Ο μεγαλύτερος και ο μικρότερος αριθμός που μπορεί να παρασταθεί.

β) Η ακρίβεια.

α) Ο μεγαλύτερος αριθμός που μπορεί να παρασταθεί είναι

$+ 2^{+31} \cdot (1-2^{-9})$: 0 111111 111111111

Ο μικρότερος αριθμός που μπορεί να παρασταθεί είναι

$- 2^{+31} \cdot (1-2^{-9})$: 1 000000 000000000

β) Στους μεγάλους αριθμούς η ακρίβεια είναι μικρή, η απόσταση μεταξύ δύο διαδοχικών αριθμών είναι:

$+ 2^{+31} \cdot [(2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9}) -$

$- (2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8})] = + 2^{+31} \cdot 2^{-9} = \mathbf{2^{+22}}$:

(0 111111 111111111) – (0 111111 111111110) = **0 111111 000000001**

Το αντίστοιχο μέγιστο απόλυτο σφάλμα λόγω στρογγυλοποίησης στην περιοχή αυτή είναι: $2^{+22} / 2 = 2^{+21}$.

Την καλύτερη ακρίβεια την έχουμε στους αριθμούς που βρίσκονται κοντά στο 0 (αρκεί να εξετάσουμε μόνο τους θετικούς αριθμούς):

$2^{-32} \cdot [(2^{-1} + 2^{-9}) – (2^{-1})] = 2^{-32} \cdot 2^{-9} = \mathbf{2^{-41}}$:

(0 000000 100000001) – (0 000000 100000000) = **0 000000 000000001**

Το αντίστοιχο μέγιστο απόλυτο σφάλμα λόγω στρογγυλοποίησης στην περιοχή αυτή είναι: $2^{-41} / 2 = 2^{-42}$.

Μεταξύ του 0 και του αμέσως μεγαλύτερου αριθμού (2^{-32} 2^{-1}) η ακρίβεια είναι: $\mathbf{2^{-33}}$:

0 000000 100000000

Στην πραγματικότητα, όμως, σύμφωνα με το πρότυπο παράστασης πραγματικών αριθμών που ακολουθεί ο ΕΚΥ, το 0 δεν μπορεί να παρασταθεί.

7.6. Σε Η/Υ που διαθέτει "λέξη" των 16 ψηφίων για την παράσταση των αριθμών κινητής υποδιαστολής (1 ψηφίο για το πρόσημο, 6 ψηφία για τον εκθέτη και 9 για τον συντελεστή) να παρασταθούν σε κανονική μορφή οι αριθμοί:

(i) $3 \times 10^{10}_{<10>}$ **(ii)** $-12{,}47_{<10>}$

Μετατρέπουμε τους αριθμούς σε δυαδικούς, στην κανονική μορφή και στη συνέχεια τους παριστάνουμε σε δυαδικό στη μορφή κινητής υποδιαστολής

(i) Για ευκολία πράξεων μετατρέπω αρχικά τον αριθμό σε δεκαεξαδικό:

	υπόλοιπο
30000000000 / 16 = 1875000000	0
1875000000 / 16 = 117187500	0
117187500 / 16 = 7324218	12 → C
7324218 / 16 = 457763	10 → A
457763 / 16 = 28610	3
28610 / 16 = 1788	12 → C
1788 / 16 = 111	15 → F
111 / 16 = 6 < 16	

Άρα $30000000000_{<10>} = 6FC3AC00_{<16>}$

$= 0110\ 1111\ 1100\ 0011\ 1010\ 1100\ 0000\ 0000_{<2>}$

6 F C 3 A C 0 0

$= +\ 2^{+31} \cdot 0{,}110\ 1111\ 1100\ 0011\ 1010\ 1100\ 0000\ 0000_{<2>}$

$=$ **0 111111 110111111**

(ii) Μετατρέπω ξεχωριστά το ακέραιο και το κλασματικό μέρος στη δυαδική μορφή:

$12_{<10>} = 1100_{<2>}$

και

0,47 × 2 = 0,94
0,94 × 2 = 1,88
0,88 × 2 = 1,76
0,76 × 2 = 1,52
0,52 × 2 = 1,04
0,04 × 2 = 0,08

Οπότε $0{,}47_{<10>} = 0{,}011110\ldots_{<2>}$

Άρα: $-12{,}47_{<10>} = -\ 1100{,}011110\ldots_{<2>} = -\ 2^{+4}\ 0{,}1100011110\ldots_{<2>}$

$= (-)\ 100100\ \ 110001111$

$= \mathbf{\ \ 1\ \ 011011\ \ 001110000}$

7.7. Σε ένα Η/Υ η μορφή της παράστασης πραγματικών αριθμών είναι 27 bits για τον συντελεστή, 8 bits για τον εκθέτη και 1 bit για το πρόσημο του αριθμού. Να βρεθεί ποιοι από τους αριθμούς 9000000,1 … 9999999,9 έχουν την ίδια εσωτερική παράσταση.

Πρέπει να βρούμε την ακρίβεια της παράστασης των αριθμών, στην περιοχή [9000000,1 έως 9999999,9]. Ισχύει:

$$9000000 = 9 \cdot 10^6 \approx 9 \cdot 2^{20} = 1001 \cdot 2^{20} = 0{,}1001 \cdot 2^{24}$$

Δηλαδή ενδιαφερόμαστε για την ακρίβεια στην τάξη μεγέθους 2^{24}.

Συγκεκριμένα είναι $2^{23} = 8388608 < 9000000{,}1$ και $9999999{,}9 < 16777217 = 2^{24}$

Επομένως η κανονική μορφή των αριθμών είναι $\boldsymbol{\alpha \times 2^{24}}$, όπου *α* έχει 27 δυαδικά ψηφία.

Στην περιοχή αυτή, η απόσταση μεταξύ δύο διαδοχικών αριθμών είναι $2^{24} \cdot 2^{-27} = 2^{-3}$. Άρα το κλασματικό τμήμα του αριθμού παριστάνεται με 3 δυαδικά ψηφία. Όπως έχει υπολογιστεί ήδη από την άσκηση 3β, οι αριθμοί με ίσα ακέραια μέρη θα ταυτίζονται όταν λήγούν σε 0,2 ή 0,3 και 0,7 ή 0,8.