IEEE Standard 754 [IEEE 85]

ΠΑΡΑΣΤΑΣΗ ΑΡΙΘΜΩΝ ΚΙΝΗΤΗΣ ΥΠΟΔΙΑΣΤΟΛΗΣ

Αρχές

Με ένα σύστημα αριθμών κινητής υποδιαστολής είναι δυνατό να παραστήσουμε ένα διάστημα θετικών και αρνητικών ακεραίων με κέντρο το μηδέν . Ένας αριθμός σταθερής υποδιαστολής με βάση το **2** ή οποιαδήποτε άλλη , μπορεί με αυτό τον τρόπο να έχει επίσης και κλασματικό μέρος**.**

Αυτή η προσέγγιση έχει περιορισμούς , δηλαδή πολύ μεγάλοι αριθμοί και πολύ μικρά κλασματικά μέρη δεν μπορούν να παρασταθούν . Επίσης το κλασματικό μέρος του πηλίκου σε μια διαίρεση δυο μεγάλων αριθμών μπορεί να χαθεί .Για τον λόγο αυτό , αριθμοί όπως ο **976.000.000.000.000** και ο **0,000000000000976** μπορούν να παρασταθούν ως $\mathbf{9{,}76 * 10^{14}}$ και $\mathbf{9{,}76 * 10^{14}}$ αντίστοιχα . Αυτό που πετύχαμε , είναι ότι η υποδιαστολή μπορεί να κινηθεί "δυναμικά ",σε κατάλληλη θέση,χρησιμοποιώντας τον εκθέτη της βάσης **10** , για να εντοπίσουμε τη θέση της . Το παραπάνω , επιτρέπει να παρασταθεί στον υπολογιστή ένα εύρος πολύ μεγάλων και πολύ μικρών αριθμών , με πολύ λίγα ψηφία .

Η γενική μορφή παράστασης των αριθμών με τον παραπάνω τρόπο είναι :

$$\pm \boldsymbol{S} * \boldsymbol{B}^{\pm \boldsymbol{E}}$$

- **πρόσημο : συν ή μείον**
- $\boldsymbol{S}$ **: significand (σημαντικό μέρος) ή mantissa**
- $\boldsymbol{E}$ **: εκθέτης**

Η βάση $\boldsymbol{B}$ είναι η ίδια για όλους τους αριθμούς και δεν χρειάζεται να αποθηκευθεί.

Στο **σχήμα 1(a)** παριστάνεται ένα **32-bit κινητής υποδιαστολής format** , με βάση το **2**.Στο αριστερότερο **bit**,αποθηκεύεται το πρόσημο του αριθμού (**0 για θετικό , 1 για αρνητικό**) . Η τιμή του εκθέτη αποθηκεύεται στα bit **1** έως **8** . Η παράσταση που χρησιμοποιείται για τον εκθέτη , είναι γνωστή και ως **biased αναπαράσταση** : μια σταθερή τιμή , ορισμένη ως **bias σταθερά**, αφαιρείται από το πεδίο τιμών του εκθέτη , για να πάρει ο εκθέτης την πραγματική του τιμή . Σ 'αυτήν την περίπτωση , το πεδίο των **8 bits** αποδίδει τους αριθμούς μεταξύ **0** και **255** . Με **bias σταθερά** ίση με **128** , οι τιμές του εκθέτη βρίσκονται στο διάστημα από **–128** έως **+127** .

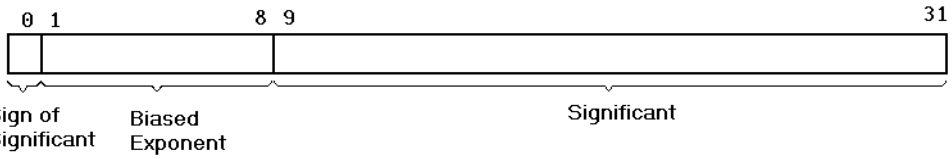
Το τελευταίο τμήμα της λέξης , είναι το **significand** , το οποίο συνήθως καλείται **mantissa** . Οποιοσδήποτε αριθμός κινητής υποδιαστολής μπορεί να εκφρασθεί με πολλούς τρόπους . Έτσι , τα ακόλουθα είναι ισοδύναμα :

$$\mathbf{0{,}110 * 2^5}$$
$$\mathbf{110 * 2^2}$$
$$\mathbf{0{,}0110 * 2^5}$$

Για να απλοποιήσουμε τις πράξεις με αριθμούς κινητής υποδιαστολής ,τυπικά απαιτείται να **κανονικοποιηθούν** (**normalize**) . Για παράδειγμα ένας **κανονικοποιημένος** αριθμός είναι της μορφής :

$$\pm \quad \mathbf{0,1bbbb\ldots b} * \mathbf{2}^{\pm \mathbf{E}}$$

όπου **b** είναι δυαδικά ψηφία (**0** ή **1**) . Αυτό συνεπάγεται πως το πιο αριστερό ψηφίο του **significand** θα πρέπει να είναι το **1** .Επομένως δεν χρειάζεται να το αποθηκεύσουμε στον υπολογιστή . Γι 'αυτό τον λόγο το πεδίο των **23 bit** αποθηκεύει ένα **24-bit significand** με μια τιμή μεταξύ του **0,5** και του **1,0** .



(a) Format

$0.11010001 \times 2^{10100}$	=	0	10010100 10100010000000000000000
$-0.11010001 \times 2^{10100}$	=	1	10010100 10100010000000000000000
$0.11010001 \times 2^{-10100}$	=	0	01101100 10100010000000000000000
$-0.11010001 \times 2^{-10100}$	=	1	01101100 10100010000000000000000

(b) Examples

Σχημα 1

Το **σχήμα 1(b)** δίνει μερικά παραδείγματα αριθμών μετασχηματισμένων σ 'αυτή την μορφή . Παρατηρούμε τα παρακάτω στοιχεία :

- Το **πρόσημο** αποθηκεύεται στο **πρώτο ψηφίο** της λέξης .
- Το πρώτο ψηφίο του πραγματικού **significand** είναι πάντα **1** και δεν χρειάζεται να αποθηκεύεται στο πεδίο του **significand** .
- Η **bias σταθερά** προστίθεται στον πραγματικό εκθέτη για να αποθηκευθεί στο πεδίο του.
- Η **βάση** είναι το **2** .

Το **σχήμα 2** δείχνει το εύρος των αριθμών που μπορούν να αναπαρασταθούν σε μια **32 bit** λέξη . Χρησιμοποιώντας **αναπαράσταση ακεραίου συμπληρώματος ως προς 2** ,όλοι οι ακέραιοι από το -2^{31} έως το $2^{31}-1$ μπορούν να αναπαρασταθούν , σε ένα σύνολο 2^{32} διαφορετικών αριθμών . Με το παράδειγμα συστήματος κινητής υποδιαστολής του **σχήματος 1** , τα ακόλουθα διαστήματα αριθμών , είναι δυνατόν να προκύψουν :

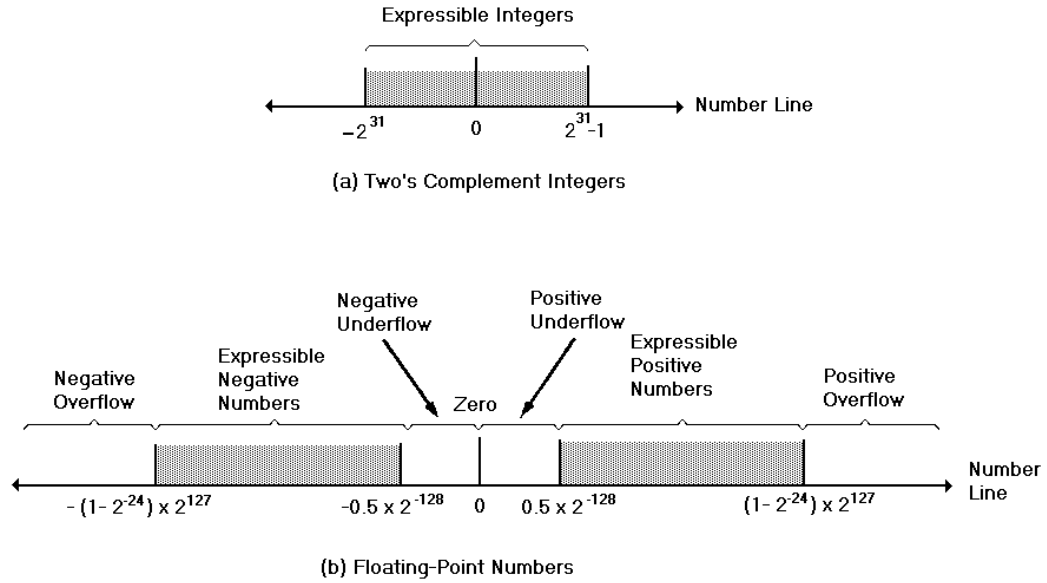
- Αρνητικοί μεταξύ $(-1-2^{-24}) * 2^{127}$ και $-0,5 * 2^{-128}$
- Θετικοί μεταξύ $0,5 * 2^{-128}$ και $(1-2^{-24}) * 2^{127}$

Πέντε περιοχές στον άξονα των αριθμών δεν συμπεριλαμβάνονται με τα παραπάνω διαστήματα :

- Αρνητικοί μικρότεροι από $(1-2^{-24}) * 2^{127}$ (**αρνητική υπερχείλιση:negative overflow**)
- Αρνητικοί μεγαλύτεροι από $-0,5 * 2^{-128}$ (**αρνητική υποχείλιση:negative underflow**)

- Μηδέν
- Θετικοί μικρότεροι από $\mathbf{0,5 * 2^{-128}}$ (**θετική υποχείλιση:positive underflow**)
- Θετικοί μεγαλύτεροι από $\mathbf{(1\text{-}2^{-24}) * 2^{127}}$ (**θετική υπερχείλιση:positive overflow**)

Σχημα 2

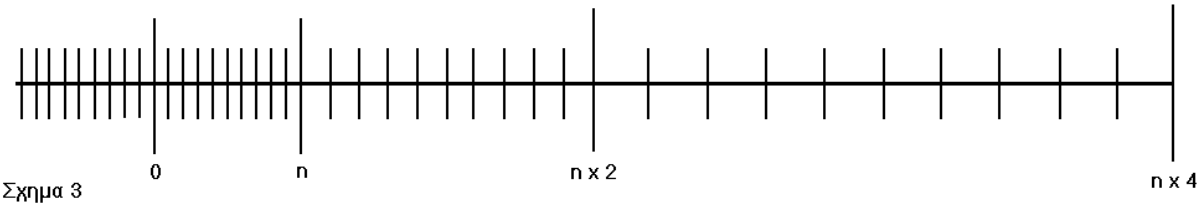


Η αναπαράσταση αυτή ,όπως παρουσιάζεται ,δεν θα ορίζει ποτέ τιμή για το μηδέν . Ωστόσο , όπως θα δούμε ,οι πραγματικές αναπαραστάσεις κινητής υποδιαστολής περιλαμβάνουν μια ειδική παράσταση από **bit** για να παρασταθεί το μηδέν .

Υπερχείλιση συμβαίνει όταν ένα αριθμητικό αποτέλεσμα κάποιας πράξης , έχει μέγεθος μεγαλύτερο από αυτό που μπορεί να εκφρασθεί με εκθέτη το **127** (**π.χ.** $\mathbf{2^{120} * 2^{100} = 2^{220}}$) . **Υποχείλιση** συμβαίνει όταν το κλασματικό μέρος είναι πάρα πολύ μικρό (**π.χ.** $\mathbf{2^{-120} * 2^{-100} = 2^{-220}}$) . Η **υποχείλιση** είναι ένα λιγότερο σημαντικό πρόβλημα , επειδή το αποτέλεσμα μπορεί γενικά να προσεγγιστεί ικανοποιητικά από το μηδέν .

Είναι σημαντικό να παρατηρήσουμε ότι δεν αναπαριστούμε περισσότερες το πλήθος τιμές , με το σύστημα κινητής υποδιαστολής .Ο μέγιστος αριθμός που μπορεί να παρασταθεί με **32 bits** είναι ακόμα ο $\mathbf{2^{32}}$.Αυτό που καταφέραμε είναι να κατανείμουμε αυτούς τους αριθμούς σε δυο διαστήματα , ένα θετικό και ένα αρνητικό.

Επίσης , παρατηρούμε ότι οι αριθμοί που παριστάνονται με σύστημα κινητής υποδιαστολής δεν είναι τοποθετημένοι τακτικά κατά μήκος του άξονα των αριθμών , όπως οι αριθμοί σταθερής υποδιαστολής . Οι δυνατές τιμές πλησιάζουν μεταξύ τους , όσο βρισκόμαστε κοντά στο μηδέν και σταδιακά απομακρύνονται μεταξύ τους , όσο πηγαίνουμε προς μεγαλύτερες τιμές όπως φαίνεται στο **σχήμα 3** .



Σχημα 3

Αυτό είναι ένα από τα μειονεκτήματα των υπολογισμών κινητής υποδιαστολής : πολλοί υπολογισμοί παράγουν αποτελέσματα που δεν είναι ακριβή και πρέπει να στρογγυλοποιηθούν στην κοντινότερη δυνατή τιμή που μπορεί να παρασταθεί .

Σε **format** όπως του **σχήματος 1** ,το εύρος μεγαλώνει σε βάρος της ακρίβειας και αντίθετα .Το παράδειγμα δείχνει **8 bit** παραχωρημένα στον εκθέτη και **23** στο **significand** .Αν αυξήσουμε τον αριθμό των ψηφίων του εκθέτη , αυξάνουμε το εύρος των παραστάσιμων αριθμών .Όμως , παρόλο που μόνο ένας σταθερός αριθμός διαφορετικής τιμής μπορεί να εκφρασθεί , έχουμε μειώσει την πυκνότητα αυτών των αριθμών και ακόμα περισσότερο την ακρίβεια . Ο μόνος τρόπος να αυξήσουμε ταυτόχρονα εύρος και ακρίβεια , είναι να χρησιμοποιήσουμε περισσότερα **bit** . Γι 'αυτό τον λόγο,οι περισσότεροι υπολογιστές προσφέρουν τουλάχιστον απλής και διπλής ακρίβειας αριθμούς .Για παράδειγμα , μια απλής ακρίβειας μορφή μπορεί να έχει **32 bit** και μια διπλής ακρίβειας , **64 bit** .

Έτσι,υπάρχει αντιστρόφως ανάλογη σχέση,μεταξύ του αριθμού των ψηφίων του εκθέτη και του αριθμού των ψηφίων του **significand** .Όμως , είναι ακόμα πιο πολύπλοκο . Η ισχύουσα βάση του εκθέτη δεν χρειάζεται να είναι το **2** . Η **IBM S/370** δομή , για παράδειγμα ,χρησιμοποιεί για βάση το **16** . Η μορφή αυτή αποτελείται από ένα **7 bit** εκθέτη και ένα **24 bit significand** . Έτσι για παράδειγμα :

$$\mathbf{0{,}11010001 * 2^{10100}} = \mathbf{0{,}11010001 * 16^{101}}$$

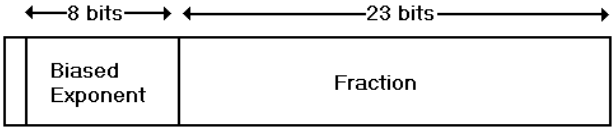
και ο εκθέτης αποθηκεύεται ώστε να παριστάνει **5** αντί **20**.

Το πλεονέκτημα της χρησιμοποίησης , ενός μεγαλύτερου εκθέτη είναι ότι ένα μεγαλύτερο εύρος μπορεί να επιτευχθεί , για τον ίδιο αριθμό ψηφίων του εκθέτη . Όμως , ας θυμηθούμε , ότι δεν αυξήσαμε τον αριθμό των διαφορετικών τιμών που μπορούν να παρασταθούν . Γι'αυτό τον λόγο , για μια σταθερή μορφή , μια μεγαλύτερη βάση εκθέτη δίνει ένα μεγαλύτερο εύρος εις βάρος της ακρίβειας .

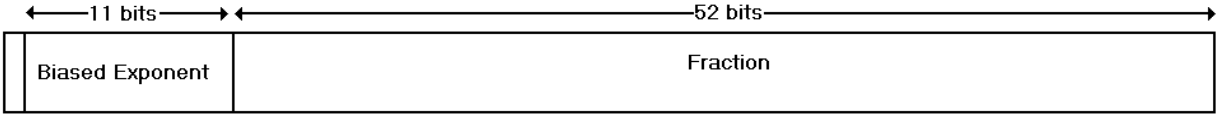
IEEE Standard για δυαδικούς αριθμούς κινητής υποδιαστολής

Η σημαντικότερη αναπαράσταση αριθμητικής κινητής υποδιαστολής ορίζεται στην **IEEE Standard 754 (IEEE 85)** .Αυτό το **standard** αναπτύχθηκε για να διευκολύνει τη μεταφερσιμότητα των προγραμμάτων από τον ένα επεξεργαστή στον άλλο και να ενθαρρύνει την ανάπτυξη πιο εκλεπτυσμένων αριθμητικά προσανατολισμένων προγραμμάτων .Το standard αυτό ,έχει ευρέως υιοθετηθεί και χρησιμοποιείται εικονικά σε όλους τους σύγχρονους επεξεργαστές και μαθηματικούς συνεπεξεργαστές .

Το **IEEE Standard** ορίζει ταυτόχρονα ένα **32 bit** απλό και ένα **64 bit** διπλό format,με **8 bit** και **11 bit** εκθέτες αντίστοιχα (Σχημα 4) .



(a) Single Format



Σχημα 4

Η βάση του συστήματος είναι **2** . Επιπροσθέτως , το **standard** ορίζει δυο διευρυμένα **format ,**απλό και διπλό, των οποίων το ακριβές **format** εξαρτάται από την εφαρμογή . Τα διευρυμένα **format** χρησιμοποιούνται σε ενδιάμεσους υπολογισμούς .Με την μεγαλύτερη ακρίβεια τους , τα διευρυμένα **format** μειώνουν την πιθανότητα ενός τελικού αποτελέσματος που έχει επηρεαστεί από μεγάλα σφάλματα στρογγύλευσης . Με το μεγαλύτερο εύρος τους ελαττώνουν επίσης την πιθανότητα μιας ενδιάμεσης υπερχείλισης , τερματίζοντας ένα υπολογισμό του οποίου το αποτέλεσμα θα μπορούσε να παρασταθεί ικανοποιητικά με την χρήση ενός μη διευρυμένου **format** . Ένα επιπλέον κίνητρο για την χρήση απλού **format** , είναι ότι έχει τα πλεονεκτήματα του διπλού **format** , χωρίς να υστερεί χρονικά όπως συμβαίνει σε υπολογισμούς υψηλής ακρίβειας .Το **πίνακας 1** συγκεντρώνει όλα τα χαρακτηριστικά των τεσσάρων **format** .

Πινακας 1 IEEE 754 Format Parameters

Parameter	Format			
	Single	Single Extended	Double	Double Extended
Word width (bits)	32	≥43	64	≥79
Exponent width (bits)	8	≥11	11	≥15
Exponent bias	127	unspecified	1023	unspecified
Maximum exponent	127	≥1023	1023	≥16,383
Minimum exponent	-126	≤1022	-1022	≤-16,382
Number range (base 10)	10^{-38}, 10^{+38}	unspecified	10^{-308}, 10^{+308}	unspecified
Significand width (bits)	23	≥31	52	≥63
Number of exponents	254	unspecified	2046	unspecified
Number of fractions	2^{23}	unspecified	2^{52}	unspecified
Number of values	1.98×2^{31}	unspecified	1.99×2^{63}	unspecified

Στα **IEEE format** , όλες οι παραστάσεις **bit** δεν μεταφράζονται με τον συνήθη τρόπο. Αντιθέτως ορισμένες παραστάσεις **bit** χρησιμοποιούνται για την παράσταση ειδικών τιμών . Ο **πίνακας 2** παρουσιάζει τις τιμές που αντιστοιχούν σε διάφορες παραστάσεις **bit** .

Οι ακραίες περιπτώσεις έκθετων που περιέχουν μόνο μηδενικά ή μόνο μονάδες (**255** στο απλό **format** , **2047** στο διπλό **format**) ορίζουν ειδικές τιμές . Οι παρακάτω κλάσεις αριθμών μπορούν να παρασταθούν :

- **Κανονικοποιημένοι** μη μηδενικοί αριθμοί κινητής υποδιαστολής , με τιμές εκθετών στο διά-στημα **1** έως **254** , για **single format** ,και στο διάστημα **1** έως **2046** ,για **double format** . Στον εκθέτη προστίθεται μια **bias σταθερά** ούτως ώστε το εύρος των τιμών του εκθέτη να κυμαί-νεται μεταξύ των τιμών **–126** και **+127** ,για το **single format** , και **-1022** έως **+1023** , για το **double format** . Για έναν **κανονικοποιημένο** αριθμό απαιτείται το πρώτο ψηφίο του **signifι-cand** εί-ναι **1** , επομένως αυτό μπορεί να μην αποθηκεύεται , με συνέπεια να έχουμε ένα **significand 24 bit** ή **53 bit** (το **significand** καλείται και **fraction** στο **standard**) .
- Ένας **έκθετης** από **μηδενικά** μαζί με ένα **fraction** από **μηδενικά** , αναπαριστά το **θετικό** ή το **αρνητικό μηδέν** ,ανάλογα με το πρόσημο . Όπως αναφέρθηκε είναι χρήσιμο να αναπαρί-σταται μια ακριβώς τιμή για το **μηδέν** .
- Ένας **έκθετης** αποτελούμενος από **μονάδες** με ένα **fraction** από **μηδενικά** , αναπαριστά το **θετικό** ή το **αρνητικό άπειρο** ,ανάλογα με το πρόσημο . Είναι επίσης χρήσιμο να έχουμε μια αναπαράσταση του **άπειρου** . Αφήνεται στον χρηστή να αποφασίσει εάν την υπερχείλιση την αντιμετωπίσει ως λάθος ή να συνεχίσει την εκτέλεση του προγράμματος με την τιμή **άπειρο** .
- Ένας **έκθετης** από **μηδενικά** με ένα **μη μηδενικό fraction** , αναπαριστά ένα μη κανονικο-ποιημένο αριθμό σ' αυτή την περίπτωση το πρώτο **bit** αριστερά της υποδιαστολής είναι μη-δέν και η πραγματική τιμή του έκθετη είναι **–126** ή **–1022** . Ο αριθμός είναι θετικός ή αρνη-τικός ανάλογα με το πρόσημο .
- Ένας **έκθετης** που αποτελείται μόνο από **μονάδες** μαζί με ένα **μη αρνητικό fraction** δίνει την τιμή **NaN** που σημαίνει **Not a Number** (όχι αριθμός) και χρησιμοποιείται να επισημαίνει μερικές ειδικές περιπτώσεις .

Η σημασία των **μη κανονικοποιημένων** αριθμών και των **NaN** , αναλύεται παρακάτω .

Πινακας 2 Interpretation of IEEE 754 Floating - Point Numbers

	Single Precision (32 bits)				Double Precision (64 bits)			
	Sing	**Biased Exponent**	**Fraction**	**Value**	**Sing**	**Biased Exponent**	**Fraction**	**Value**
Positive zero	0	0	0	0	0	0	0	0
Negative zero	1	0	0	-0	1	0	0	-0
Plus infinity	0	255(all 1s)	0	∞	0	2047 (all 1s)	0	∞
Minus infinity	1	255(all 1s)	0	-∞	1	2047 (all 1s)	0	-∞
Quiet NaN	1 or 0	255(all 1s)	≠ 0	NaN	1 or 0	2047 (all 1s)	≠ 0	NaN
Signaling NaN	1 or 0	255(all 1s)	≠ 0	NaN	1 or 0	2047 (all 1s)	≠ 0	NaN
Positive normalized nonzero	0	0<e<255	f	$2^{e-127}(1.f)$	0	0 < e < 2047	f	$2^{e-1023}(1.f)$
Negative normalized nonzero	1	0<e<255	f	$2^{e-127}(1.f)$	1	0 < e < 2047	f	$-2^{e-1023}(1.f)$
Positive denormalized	0	0	f ≠ 0	$2^{e-126}(0.f)$	0	0	f ≠ 0	$2^{e-1022}(0.f)$
Negative denormalized	1	0	f ≠ 0	$-2^{e-126}(0.f)$	1	0	f ≠ 0	$-2^{e-1022}(0.f)$

ΠΡΑΞΕΙΣ ΚΙΝΗΤΗΣ ΥΠΟΔΙΑΣΤΟΛΗΣ

Ο **πίνακας 3** συγκεντρώνει τις βασικές πράξεις για την αριθμητική κινητής υποδιαστολής .Για την **πρόσθεση** και την **αφαίρεση** είναι αναγκαίο να διασφαλίσουμε ότι και οι δυο όροι της πράξης θα έχουν τον ίδιο έκθετη. Αυτό μπορεί να απαιτεί την μετατόπιση της υποδιαστολής σε ένα από τους δυο όρους για να πετύχουμε ευθυγράμμιση.Ο **πολλαπλασιασμός** και η **διαίρεση** είναι πιο ακριβείς . Προβλήματα μπορούν να εμφανιστούν στα αποτελέσματα αυτών των πράξεων . Αυτά είναι :

- **Υπερχείλιση έκθετη** : ένας θετικός έκθετης να υπερβαίνει την μέγιστη δυνατή τιμή έκθετη. Σε μερικά συστήματα αυτό μπορεί να ορίζεται σαν $\pm\infty$.
- **Υποχείλιση έκθετη** : ένας αρνητικός έκθετης να υπολείπεται της ελάχιστης δυνατής τιμής έκθετη . Αυτό σημαίνει ότι ο αριθμός είναι πολύ μικρός για να παρασταθεί και μπορεί να αναφέρεται σαν μηδέν .
- **Υποχείλιση significant** : στην διαδικασία ευθυγράμμισης των **significant** , ψηφία μπορεί να **'κυλήσουν'** εκτός του δεξιού άκρου του significant . Όπως θα δούμε,απαιτούνται μερικές μορφές στρογγύλευσης .
- **Υπερχείλιση significant** : η πρόσθεση δυο significant του ίδιου πρόσημου μπορεί να οδηγήσει σε απώλεια του **MSB** (**Most Significant Bit**) . Αυτό μπορεί να διορθωθεί με επανευθυγράμμιση όπως θα εξηγήσουμε .

Πίνακας 3 Floating - Point Numbers and Arithmetic Operations

Floating Point Numbers

$$x = x_s B^{x_E}$$

$$y = y_s B^{y_E}$$

Arithmetic Operations

$$\left.\begin{aligned} x + y &= (x_s B^{x_E - y_E} + y_s) \times B^{y_E} \\ x - y &= (x_s B^{x_E - y_E} - y_s) \times B^{y_E} \end{aligned}\right\} x_E \le y_E$$

$$x \times y = (x_s \times y_s) \times B^{x_E + y_E}$$

$$x \div y = (x_s \div y_s) \times B^{x_E - y_E}$$

Πρόσθεση και αφαίρεση

Σ την αριθμητική κινητής υποδιαστολής , η **πρόσθεση** και η **αφαίρεση** είναι πιο πολύπλοκες από ότι ο **πολλαπλασιασμός** και η **διαίρεση** .Αυτό οφείλεται στη αναγκαία ευθυγράμμιση εκθετών .Υπάρχουν τέσσερις βασικές φάσεις στον αλγόριθμο πρόσθεσης και αφαίρεσης :

- Έλεγχος για μηδενικά
- Ευθυγράμμιση των **significand**
- Πρόσθεση και αφαίρεση των **significand**

- Κανονικοποίηση του αποτελέσματος

Ένα τυπικό διάγραμμα ροής φαίνεται στο **σχήμα 5** . Ένας **βήμα προς βήμα** έλεγχος μας επισημαίνει τις κύριες διαδικασίες ,οι οποίες απαιτούνται για την **πρόσθεση** και την **αφαίρεση** κινητής υποδιαστολής .Θεωρούμε ένα **format** παρόμοιο με αυτό του **σχήματος 4** . Για τις πράξεις της πρόσθεσης και της αφαίρεσης , οι δύο όροι πρέπει να μεταφερθούν σε **καταχωρητές** (**μνήμης**) , οι οποίοι θα χρησιμοποιηθούν από την **ALU** (**Arithmetical Logical Unit**) .Αν το **format** κινητής υποδιαστολής περιλαμβάνει και το δεδομένο ψηφίο του **significand** που δεν έχει καταχωρηθεί ,αυτό το ψηφίο θα πρέπει να εισαχθεί στο **significand** ,για να εκτελεστεί η πράξη.Τυπικά ο εκθέτης και το **significand** θα αποθηκευθούν σε διαφορετικούς **καταχωρητές** (**μνήμης**) και επανενώνονται όταν παράγεται το αποτέλεσμα .

Επειδή η πρόσθεση και η αφαίρεση είναι παρόμοιες , εκτός του πρόσημου που αλλάζει , η διαδικασία ξεκινάει αλλάζοντας το πρόσημο του αφαιρετέου αν έχουμε πράξη αφαίρεσης . Εν συνεχεία , αν κάποιος από τους δύο όρους είναι μηδέν , ο άλλος αναφέρεται ως αποτέλεσμα .

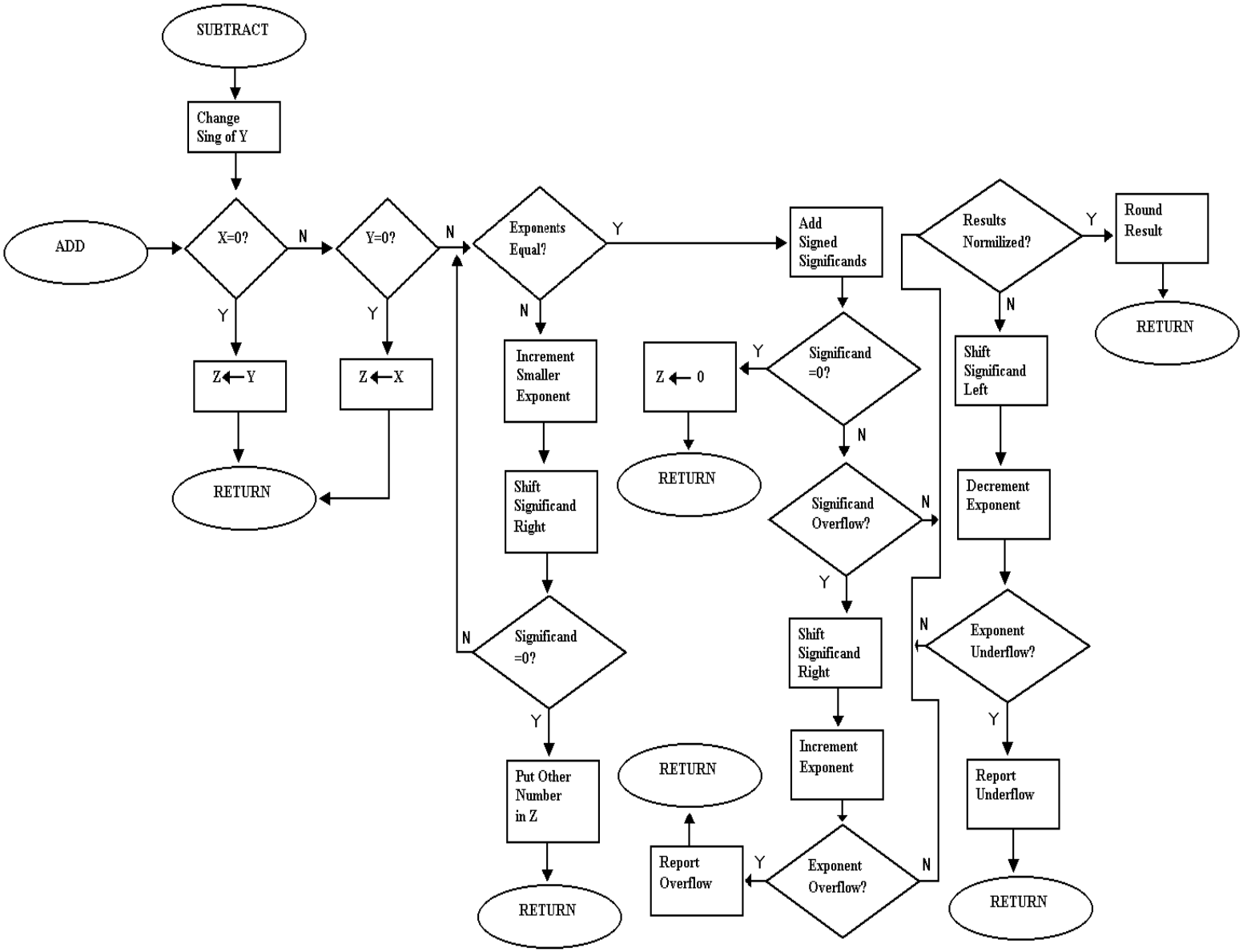
Η επόμενη φάση , είναι να τροποποιήσουμε τους αριθμούς έτσι ώστε οι εκθέτες να είναι ίσοι . Για να φανεί η ανάγκη γι ’αυτό , θεωρούμε την παρακάτω δεκαδική πρόσθεση :

$$\mathbf{123 * 10^{0} + 456 * 10^{-2}}$$

Όπως είναι φανερό ,δεν μπορούμε απλά να προσθέσουμε τα **significand** . Τα ψηφία πρέπει πρώτα να έρθουν σε κατάλληλες θέσεις :Το **4** του δεύτερου αριθμού θα πρέπει να ευθυγραμμιστεί με το **3** του πρώτου . Κάτω από αυτές τις προϋποθέσεις οι δύο εκθέτες θα είναι ίσοι , το οποίο αποτελεί αναγκαία συνθήκη , ώστε οι δύο αριθμοί σ ’ αυτή την μορφή να μπορούν να προστεθούν . Έτσι :

$$\mathbf{123 * 10^{0} + 456 * 10^{-2} = 123 * 10^{0} + 4.56 * 10^{0} = 127.56 * 10^{0}}$$

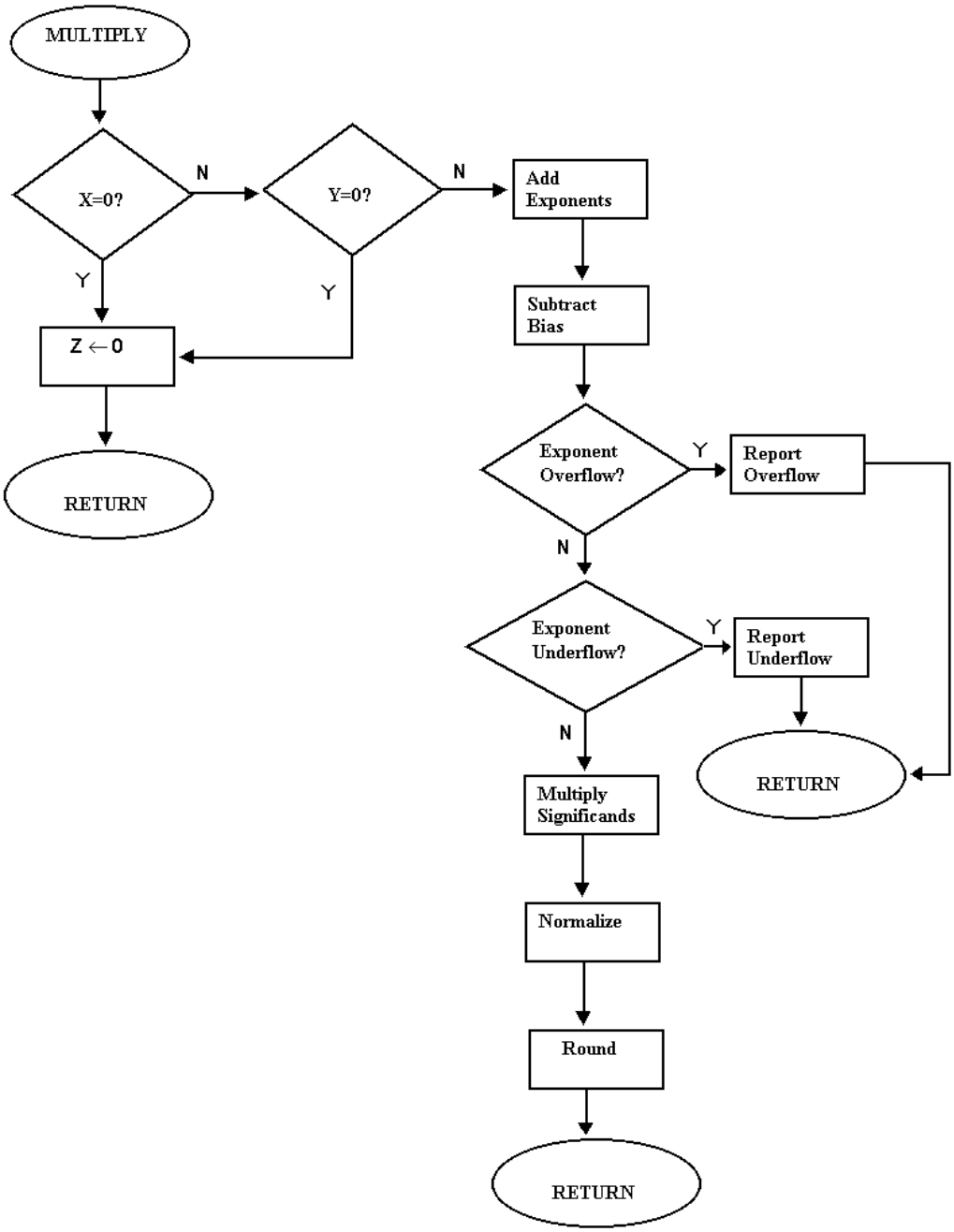
Η ευθυγράμμιση επιτυγχάνεται μεταφέροντας είτε τον μικρότερο αριθμό προς τα δεξιά (μεγαλώνοντας έτσι τον εκθέτη του) είτε τον μεγαλύτερο προς τα αριστερά . Αφού το αποτέλεσμα κάθε μίας από τις πράξεις μπορεί να χάσει ψηφία μετά την ολοκλήρωσή της , είναι ο μικρότερος από τους αριθμούς που πρέπει να μεταφερθεί . Επομένως κάθε ψηφίο που χάνεται είναι μικρότερης σημασίας . Η ευθυγράμμιση επιτυγχάνεται με διαδοχικές μεταφορές του τμήματος του αριθμού ,το οποίο μας δείχνει την τιμή του , κατά **1** ψηφίο προς τα δεξιά και αυξάνοντας κατά **1** τον έκθετη έως ότου οι δύο εκθέτες γίνουν ίσοι . (Παρατηρήστε ότι αν η βάση μας είναι το **16** μετακίνηση ενός ψηφίου είναι μετακίνηση **4** ψηφίων). Εάν αυτή η διαδικασία έχει ως αποτέλεσμα το **significand** να πάρει την τιμή μηδέν, τότε ο άλλος αριθμός αναφέρεται σαν αποτέλεσμα . Γι ’ αυτό τον λόγο, αν δύο αριθμοί έχουν εκθέτες που διαφέρουν σημαντικά, τότε ο μικρότερος αριθμός χάνεται . Στη συνέχεια οι δύο **significand** προστίθενται, λαμβάνοντας υπόψη τα πρόσημά τους . Σε περίπτωση που τα πρόσημα διαφέρουν το αποτέλεσμα μπορεί να είναι μηδέν . Υπάρχει επίσης πιθανότητα υπερχείλισης του **significand** για ένα ψηφίο . Σ ’αυτή την περίπτωση το **significand** του αποτελέσματος μετακινείται δεξιά και ο έκθετης μεγαλώνει . Μπορεί να επέλθει σαν αποτέλεσμα υπερχείλιση για τον εκθέτη . Αν συμβεί αυτό θα αναφερθεί και η πράξη θα σταματήσει . Η επόμενη φάση είναι η **κανονικοποίηση** του αποτελέσματος . Η **κανονικοποίηση** πραγματοποιείται με μεταφορά των ψηφίων του **significand** προς τα αριστερά , έως ότου το **MSB** (**Most Valuable Bit** , **1 bit** ή **4 bi**t για **δεκαεξαδικό** σύστημα) είναι μη μηδενικό . Κάθε μεταφορά επιφέρει μείωση στον έκθετη και γι’ αυτό τον λόγο μπορεί να προκληθεί υποχείλιση του έκθετη . Τέλος το αποτέλεσμα πρέπει να στρογγυλοποιηθεί και έπειτα να αναφερθεί .



Σχήμα 5 Floating-Point addition and subtraction $(z \leftarrow x \pm y)$

Πολλαπλασιασμός και διαίρεση

Ο πολλαπλασιασμός και η διαίρεση κινητής υποδιαστολής είναι πιο απλές πράξεις σε σχέση με την πρόσθεση και την αφαίρεση όπως θα δούμε παρακάτω . Για τον πολλαπλασιασμό (**σχήμα 6**) έχουμε :

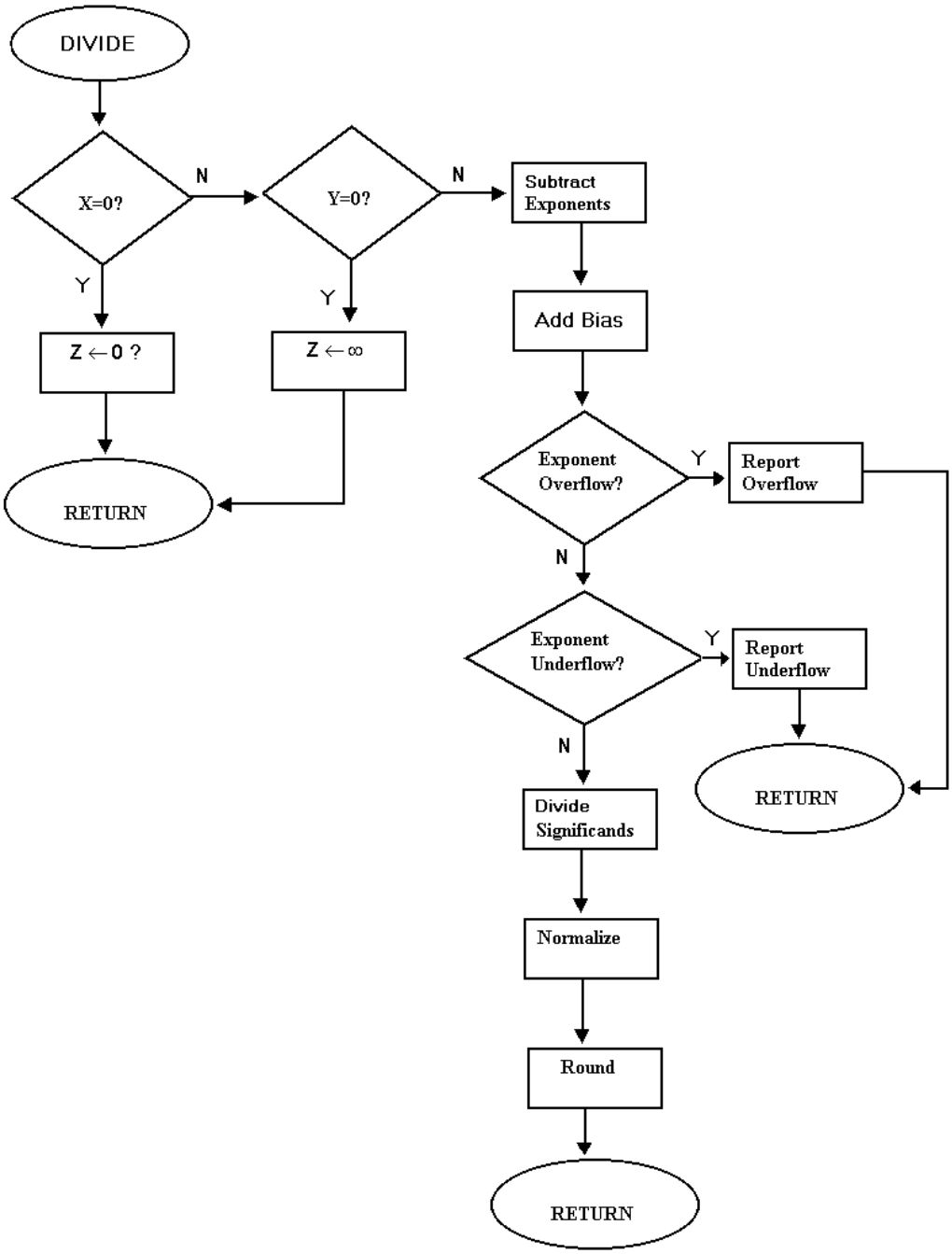


Σχήμα 6 Floating - Point multiplication ($z \leftarrow x \times y$)

Αρχικά αν ένας από τους δυο όρους είναι μηδέν , τότε το μηδέν αναφέρεται σαν αποτέλεσμα . Το επόμενο βήμα είναι να προσθέσουμε τους έκθετες . Εάν οι έκθετες είναι αποθηκευμένοι σε **biased** μορφή η πρόσθεση των έκθετων έχει ως αποτέλεσμα τον διπλασιασμό της **bias σταθεράς**.Γι' αυτό τον λόγο , η τιμή της **bias σταθεράς** αφαιρείται από το άθροισμα . Το αποτέλεσμα μπορεί να είναι είτε **υποχείλιση** είτε **υπερχείλιση** του έκθετη, γεγονός που αναφέρεται τερματίζοντας τον αλγόριθμο . Εάν ο έκθετης του αποτελέσματος βρίσκεται στο επιτρεπόμενο εύρος , το επόμενο βήμα είναι να πολλαπλασιάσουμε τα **significand** , λαμβάνοντας υπόψη τα πρόσημα τους . Ο πολλαπλασιασμός εκτελείται κατά τον ίδιο τρόπο όπως και στους ακέραιους . Το αποτέλεσμα έχει το διπλάσιο του μήκους των δυο όρων που πολλαπλασιάζονται . Τα επιπλέον **bit** θα χάνονται κατά την στογγύλευση .

Αφού το αποτέλεσμα έχει υπολογιστεί ,θα κανονικοποιείται και στρογγυλοποιείται όπως ακριβώς γίνεται στην πρόσθεση και την αφαίρεση . Σημειώνουμε ότι η κανονικοποίηση μπορεί να έχει σαν αποτέλεσμα την υποχειλιση του έκθετη .

Για την διαίρεση (**σχήμα 7**) έχουμε τα εξής : ξανά στο πρώτο βήμα κάνουμε έλεγχο για το μηδέν . Αν ο διαιρετής είναι μηδέν το γεγονός αναφέρεται ως λάθος ή το αποτέλεσμα είναι το άπειρο , ανάλογα με το πρόγραμμα . Ένας μηδενικός διαιρετέος επιστρέφει αποτέλεσμα μηδέν . Εν συνεχεία , ο έκθετης του διαιρέτη αφαιρείται από τον έκθετη του διαιρετέου . Αυτό έχει ως αποτέλεσμα οτι η **bias σταθερά** έχει αφαιρεθεί από τον **εκθέτη – αποτέλεσμα** και πρέπει να προστεθεί ξανά . Εξετάζουμε αν έχουμε υπερχείλιση ή υποχείλιση του εκθέτη . Το επόμενο βήμα είναι να διαιρέσουμε τα **significand** . Αυτό ακολουθείται από κανονικοποίηση και στρογγυλοποίηση του αποτελέσματος .



Σχήμα 7 Floating - Point division $(z \leftarrow x/y)$

ΜΕΛΕΤΗ ΑΚΡΙΒΕΙΑΣ

GUARD BITS

Αναφέραμε ότι πριν εκτελεστεί μια πράξη κινητής υποδιαστολής ο έκθετης και το **significand** του κάθε όρου της πράξης αποθηκεύονται σε **καταχωρητές** (**μνήμης**) της **ALU** . Στην περίπτωση του **significand** το μήκος του καταχωρητή είναι σχεδόν πάντα μεγαλύτερο από το μήκος του **significand** συν ένα επιπλέον ψηφίο το οποί-ο εννοείται όπως έχουμε δει στους **κανονικοποιημένους** αριθμούς . Ο καταχωρητής περιέχει επιπλέον ψηφία τα οποία ονομάζονται **guard bits** τα οποία χρησιμοποιούνται για να καλύψουν το δεξιό τμήμα του **significand με μηδενικά.**

Η αιτία της χρησιμοποίησης των **guard bits** φαίνεται στο **σχήμα 8** :

$$x = 1.000\ldots00 \times 2^{1}$$
$$-y = 0.111\ldots11 \times 2^{1}$$
$$z = 0.000\ldots01 \times 2^{1}$$
$$= 1.000\ldots00 \times 2^{-22}$$

(a) Binary Example , Without Guard Bits

$$x = 1.000\ldots00\ 0000 \times 2^{1}$$
$$-y = 0.111\ldots11\ 1000 \times 2^{1}$$
$$z = 0.000\ldots00\ 1000 \times 2^{1}$$
$$= 1.000\ldots00\ 0000 \times 2^{-23}$$

(b) Binary Example , With Guard Bits

$$x = .100000 \times 16^{1}$$
$$-y = .0FFFFF \times 16^{1}$$
$$z = .000001 \times 16^{1}$$
$$= .100000 \times 16^{-4}$$

(c) Hexadecimal Example , Without Guard Bits

$$x = .100000\ 00 \times 16^{1}$$
$$-y = .0FFFFF\ F0 \times 16^{1}$$
$$z = .000000\ 10 \times 16^{1}$$
$$= .100000\ 00 \times 16^{-5}$$

(d) Hexadecimal Example , With Guard Bits

Σχημα 8 The use guard bits

Θεωρούμε αριθμούς , σ ' ένα **IEEE format** , το οποίο έχει ένα **24 bit significand** ,συμπεριλαμβανομένου ενός δεδομένου **bit** , στα αριστερά της υποδιαστολής . Έστω δύο αριθμοί που μοιάζουν πολύ, ο **X=1,00…0** $*2^{1}$ και ο **Y=1,11…1** $*2^{0}$. Αν ο μικρότερος είναι να αφαιρεθεί από τον μεγαλύτερο , πρέπει να μεταφερθεί δεξιά κατά **1 bit** ,για να ευθυγραμμιστούν οι εκθέτες .Αυτό φαίνεται στο σχήμα **8,24(a)** . Κατά τη διάρκεια αυτής της διαδικασίας , ο **Y** χάνει **1 bit** του **significand** .Το αποτέλεσμα είναι 2^{-23} . Η ίδια πράξη επαναλαμβάνεται στο **σχήμα 8,24(b)** με τη πρόσθεση κάποιων **guard bits** . Τώρα , το λιγότερο σημαντικό **bit** δεν χάνεται εξαιτίας της ευθυγράμμισης και το αποτέλεσμα είναι 2^{-24} , έχοντας μια διαφορά κατά **2** στο **factor** από ότι στη προηγούμενη απάντηση . Όταν η **βάση** είναι **16**,η απώλεια ακρίβειας μπορεί να είναι μεγαλύτερη . Στα **σχήματα 8,24 (c)** και **(d)** φαίνεται η διαφορά που μπορεί να υπάρξει σε ένα **δεκαεξαδικό**

ROUNDING

Άλλη μια λεπτομέρεια η οποία επηρεάζει την ακρίβεια του αποτελέσματος είναι η **στρογγύλευση** . Το αποτέλεσμα κάθε πράξης μεταξύ των **significand** γενικά αποθηκεύεται σε ένα μεγαλύτερου μήκους καταχωρητή (**μνήμης**) . Όταν το αποτέλεσμα **'επιστρέφει'** σε **format** κινητής υποδιαστολής θα πρέπει να απαλαγούμε από τα επιπλέον ψηφία .

Υπάρχει ένας μεγάλος αριθμός τεχνικών στρογγύλευσης . Όμως το **standard** της **IEEE** περιγράφει τέσσερις διαφορετικές προσεγγίσεις για το πρόβλημα :

- **Round to Nearest** : Το αποτέλεσμα σρογγυλοποιείται στο κοντινότερο παραστάσιμο αριθμο .
- **Round Toward** $-\infty$: Το αποτέλεσμα σρογγυλοποιείται προς το θετικό άπειρο.
- **Round Toward** $+\infty$: Το αποτέλεσμα σρογγυλοποιείται προς το αρνητικό άπειρο .
- **Round Toward** 0 : Το αποτέλεσμα σρογγυλοποιείται προς το μηδέν.

Ας τις εξετάσουμε μια προς μια :

Round to Nearest : είναι η **default** μέθοδος **στρογγύλευσης** από τις τέσσερις παραπάνω και περιγράφεται από το **standard** ως εξής : Σαν αποτέλεσμα θα παρουσιάζεται η παραστάσιμη τιμή που βρίσκεται εγγύτερα στην πραγματική τιμή του αποτελέσματος θα παρουσιάζεται . Εάν οι δυο εγγύτερες παραστάσιμες τιμές είναι το ίδιο κοντά στην πραγματική τιμή του αποτελέσματος της οποίας το LSB είναι μηδέν .

Για παράδειγμα εάν τα επιπλέον ψηφία πέρα από τα **23** που μπορούν να αποθηκευτούν είναι **10010** τότε σε αυτήν την περίπτωση το σωστό αποτέλεσμα θα προκύψει ,εάν προσθέσουμε το δυαδικό **1** στο τελευταίο παραστάσιμο ψηφίο ,**στρογγυλοποιώντας** προς τα πάνω , στον επόμενο αναπαραστάσιμο αριθμό . Εάν τώρα τα επιπλέον ψηφία είναι **01111** τότε η σωστή απάντηση είναι απλά να τα διαγράψουμε (**αποκοπή**) έχοντας σαν αποτέλεσμα να **στρογγυλοποιούμε** προς τα κάτω στον επόμενο παραστάσιμο αριθμό .

Το **standard** χειρίζεται και ειδικές περιπτώσεις επιπλέον ψηφίων της μορφής **10000…** . Εδώ το αποτέλεσμα είναι ακριβώς στην μέση ανάμεσα στις δυο πιθανές αναπαραστάσιμες τιμές . Μια πιθανή τεχνική σε αυτή την περίπτωση είναι απλά να εκτελέσουμε **αποκοπή** μιας και είναι η απλούστερη πράξη.Παρόλα αυτά η δυσκολία με αυτή την προσέγγιση είναι ότι εισάγεται ένα μικρό,αλλά με συσσωρευτική ιδιότητα σφάλμα όταν εκτελείται μια σειρά από πράξεις . Αυτό που χρειάζεται είναι μια μέθοδος **στρογγυλοποίησης** χωρίς να υπεισέρχεται κάποιο σφάλμα . Μια πιθανή προσέγγιση του προβλήματος είναι να **στρογγυλοποιούμε** πάνω ή κάτω με βάση ένα τυχαίο αριθμό έτσι ώστε κατά μέσο όρο το αποτέλεσμα να είναι χωρίς σφάλμα . Το επιχείρημα κατά αυτής της προσέγγισης είναι ότι δεν παράγει προβλέψιμα ή αιτιοκρατικά αποτελέσματα . Η μέθοδος που χρησιμοποιείται από το **standard** της **IEEE** είναι να απαιτούμε το αποτέλεσμα να είναι ίσο . Δηλαδή εάν το αποτέλεσμα ενός υπολογισμού ισαπέχει από δυο αναπαριστάσιμες τιμές , η τιμή να **στρογγυλοποιείται** πάνω εάν το τελευταίο παραστάσιμο ψηφίο είναι ένα και να μην αλλάζει εάν είναι μηδέν .

Round Toward $-\infty$ **και Round Toward** $+\infty$**:** Οι επόμενες δυο επιλογές είναι η **στρογγυλοποίηση** προς το $\pm\infty$. Είναι ιδιαιτέρως χρήσιμες για την κατανόηση μιας τεχνικής γνωστής και ως **interval arithmetic** . Η ιδέα πίσω από την **interval arithmetic** είναι η εξής : Στο τέλος κάθε ακολουθίας πράξεων κινητής υποδιαστολής δεν μπορούμε να γνωρίζουμε την ακριβή απάντηση λόγω των περιορισμών του **hardware** οι οποίοι επιβάλουν και την **στρογγύλευση** . Εάν εκτελέσουμε κάθε υπολογισμό της διαδικασίας δυο φορές **στρογγυλοποιώντας** την μια άνω και την άλλη κάτω τότε το αποτέλεσμα είναι να διατηρούμε ένα άνω και ένα κάτω φράγμα της σωστής απάντησης . Εάν το εύρος μεταξύ του άνω και του κάτω ορίου είναι ικανοποιητικά μικρό τότε έχουμε επιτύχει μια απάντηση αρκετά ακριβής . Εάν όχι τουλάχιστον γνωρίζουμε ότι το πρόβλημα απαιτεί περαιτέρω ανάλυση .

Round Toward 0: Η τελευταία τεχνική **στρογγύλευσης** που ορίζει το **standard** είναι η **στρογγυλοποίηση** προς το μηδέν . Ουσιαστικά είναι απλή αποκοπή και τα τελευταία ψηφία αγνοουνται . Είναι σίγουρα η πιο απλή τεχνική . Παρ ' όλα αυτά το αποτέλεσμα είναι ότι η μεγέθυνση της αποκομμένης τιμής είναι πάντα μικρότερη ή ίση με την ακριβή τιμή εισάγοντας ένα σφάλμα ,το ποιο είναι σύμφωνο με τα προηγούμενα , σε κάθε πράξη . Αυτό είναι πολύ πιο σημαντικό σφάλμα από τα προηγούμενα μιας και το σφάλμα επηρεάζει κάθε πράξη στην οποία υπάρχουν επιπλέον μη μηδενικά ψηφία .

IEEE Standard για δυαδική αριθμητική κινητής υποδιαστολής

Η **IEEE 754** εκτείνονται περισσότερο από τον απλό ορισμό ενός **format** για να στηρίξουμε απλές πρακτικές και διαδικασίες ώστε η αριθμητική κινητής υποδιαστολής να παράγει ομοιόμορφα και προβλέψιμα αποτελέσματα ανεξάρτητα από την **hardware** πλατφόρμα (υπολογιστή) που χρησιμοποιούμε . Μια πλευρά του παραπάνω η **στρογγύλευση** εξετάστηκε ήδη . Αυτή η παράγραφος εξετάζει άλλα τρία θέματα : το **άπειρο**, τους **NaN** αλλά και τους **denormalized** αριθμούς .

Άπειρο

Το **άπειρο** χρησιμοποιείται ως η οριακή τιμή όλων των αριθμών με τον ακόλουθο τρόπο :

$$-\infty < (\text{Κάθε πεπερασμένος αριθμός}) < +\infty$$

Τις ειδικές περιπτώσεις που αφορούν το **άπειρο** τις εξετάζουμε παρακάτω . Οι υπόλοιπες περιπτώσεις που αφορούν το **άπειρο** επιστρέφουν το φανερό αποτέλεσμα :

$5+(+\infty)=+\infty$
$5-(+\infty)=-\infty$
$5+(-\infty)=-\infty$
$5-(-\infty)=+\infty$
$(+\infty)+(+\infty)=+\infty$
$(-\infty)+(-\infty)=-\infty$
$(-\infty)-(+\infty)=-\infty$
$(+\infty)-(-\infty)=+\infty$

Quiet and Signaling NaNs

Ο **NaN** είναι ένας συμβολισμός που χρησιμοποιείται στο **format** κινητής υποδιαστολής και υπάρχουν δυο ειδών : **quiet** και **signaling** . Ένας **signaling NaN** μας προειδοποιεί για μια λάθος πράξη η οποία οφείλεται σε ένα από τους δυο όρους . Οι **signaling Nan** .Ένας **quiet NaN** διαδίδεται διαμέσου σχεδόν κάθε αριθμητικής πράξης χωρίς να προειδοποιεί για κάποιο ενδεχομενο λαθος .Ο **πίνακας 4** παριστά πράξεις οι οποίες παράγουν

ένα **quiet NaN** .

Πινακας 4 Operations that Produce a Quiet NaN

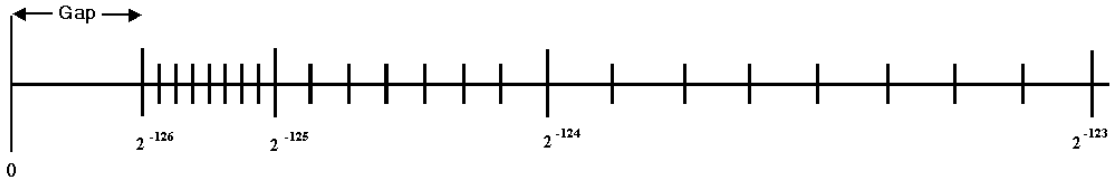
Operation	Quiet NaN Produced by
Any	Any operation on a signaling NaN
Add or Subtract	Magnitute subtraction of infinities $(+\infty)+(-\infty)$ $(-\infty)+(+\infty)$ $(+\infty)-(+\infty)$ $(-\infty)-(-\infty)$
Multiply	$0\times\infty$
Division	$\frac{0}{0}\text{or}\frac{\infty}{\infty}$
Remainder	$x\text{REM}0$ or $\infty\text{REM}y$
Square root	$\sqrt{x}\ \text{where}\ x<0$

Σημειώνουμε ότι και οι δύο τύποι των **NaN** έχουν το ίδιο γενικό **format** : ένας **εκθέτης** αποτελούμενος από **μονάδες** και ένα **fraction** μη μηδενικό . Η πραγματική παράσταση bit του **μη μηδενικού fraction** εξαρτάται από το πρόγραμμα . Οι τιμές του **fraction** μπορούν να χρησιμοποιηθούν για να διαχωρίζουμε το **quiet** από το **signaling NaN** και για να προσδιορίσουμε κάποιες ειδικές περιπτώσεις .

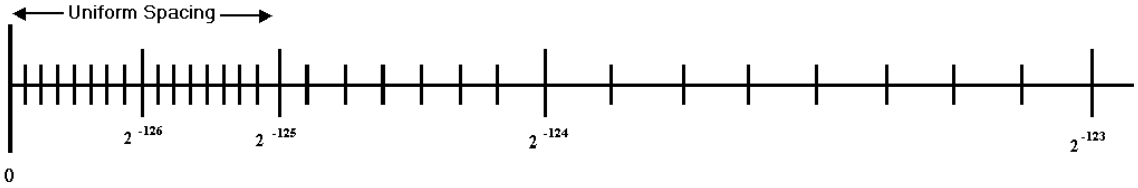
Denormalized Αριθμοί

Οι **denormalized** αριθμοί συμπεριλαμβάνονται στην **IEEE 754** για να χειριζόμαστε περιπτώσεις υποχείλισης εκθέτη .Όταν ο εκθέτης του αποτελέσματος είναι πολύ μικρός (ένας αρνητικός εκθέτης με πολύ μεγάλο μεγεθος) ,το αποτέλεσμα είναι **denormalised** . Με δεξιά μετακίνηση του **fraction** και αύξηση του εκθέτη για κάθε μετακίνηση , μέχρι ο εκθέτης να βρεθεί εντός ενός παραστάσιμου εύρους .

Το **σχήμα 9** απεικονίζει τις συνέπειες της πρόσθεσης μη κανονικοποιημένων αριθμών.Οι παραστάσιμοι αριθμοί μπορούν να ομαδοποιηθούν σε διαστήματα της μορφής $[2^n, 2^{n+1}]$. Εντός κάθε τέτοιου διαστήματος ,ο εκθέτης του αριθμού παραμένει σταθερός , ενώ το **fraction** ποικίλει , παράγοντας ένα ομοιογενές υποδιάστημα παραστάσιμων αριθμών εντός του διαστήματος .Όσο πλησιάζουμε προς το μηδέν,κάθε ικανοποιητικό διάστημα έχει το μισό μήκος του προηγούμενου διαστήματος ,αλλά περιέχοντας το ίδιο πλήθος παραστάσιμων αριθμών.



(a) 32-bit Format Without Denormalized Numbers



(b) 32-bit Format With Denormalized Numbers

Σχημα 9 The effect of IEEE 754 denormalized numbers

Γι ' αυτό το λόγο , η πυκνότητα των παραστάσιμων αριθμών αυξάνεται καθώς πλησιάζουμε το μηδέν . Παρόλα αυτά εάν χρησιμοποιούμε μονάχα **κανονικοποιημένους** αριθμούς υπάρχει ένα κενό ανάμεσα στο μικρότερο **κανονικοποιημένο** αριθμό και το μηδέν.Στην περίπτωση του **32 bit IEEE 754 format** υπάρχουν 2^{23} παραστάσιμοι αριθμοί σε κάθε διάστημα και ο μικρότερος παραστάσιμος θετικός αριθμός είναι το 2^{-126} .Με την προσθήκη των **denormalized** αριθμών ένα επιπρόσθετο πλήθος 2^{23} αριθμών προστίθενται ομοιόμορφα στο διάστημα **0** έως 2^{-126} .

Χωρίς **denormalized** αριθμούς το κενό μεταξύ του μικρότερου παραστάσιμου μη μηδενικού αριθμού και του μηδενός είναι πολύ μεγαλύτερο από το κενό μεταξύ του μικρότερου πατραστάσιμου μη μηδενικού αριθμού και του ακριβώς επόμενου παραστάσιμου αριθμού . Η σταδιακή υποχείλιση συμπληρώνει αυτό το κενό και μειώνει τις συνέπειες της υποχείλισης του εκθέτη σε ένα επίπεδο συγκρίσιμο με την **στρογγυλοποίηση** των **κανονικοποιημένων** αριθμών.