Στο Κεφάλαιο 3 περιγράψαμε τον τρόπο αποθήκευσης διάφορων τύπων δεδομένων στον υπολογιστή. Σε αυτό το κεφάλαιο θα μελετήσουμε τις πράξεις με δεδομένα τα οποία είναι αποθηκευμένα στον υπολογιστή. Οι πράξεις με δεδομένα μπορούν να ταξινομηθούν σε τρεις μεγάλες κατηγορίες: τις λογικές πράξεις, τις πράξεις μετατόπισης (ή ολίσθησης), και τις αριθμητικές πράξεις.

Στόχοι του κεφαλαίου

Μετά την ολοκλήρωση αυτού του κεφαλαίου, ο σπουδαστής θα είναι σε θέση:

- ❑ Να προσδιορίζει τις τρεις κατηγορίες πράξεων που εκτελούνται σε δεδομένα.
- ❑ Να εκτελεί μονομελείς και διμελείς λογικές πράξεις σε σχήματα bit.
- ❑ Να ξεχωρίζει τις λογικές πράξεις μετατόπισης από τις αριθμητικές πράξεις μετατόπισης.
- ❑ Να εκτελεί λογικές πράξεις μετατόπισης σε σχήματα bit.
- ❑ Να εκτελεί αριθμητικές πράξεις μετατόπισης σε ακεραίους που είναι αποθηκευμένοι σε μορφή συμπληρώματος ως προς δύο.
- ❑ Να εκτελεί πράξεις πρόσθεσης και αφαίρεσης σε ακεραίους που είναι αποθηκευμένοι σε μορφή συμπληρώματος ως προς δύο.
- ❑ Να εκτελεί πράξεις πρόσθεσης και αφαίρεσης σε ακεραίους που είναι αποθηκευμένοι σε μορφή προσήμου και μεγέθους.
- ❑ Να εκτελεί πράξεις πρόσθεσης και αφαίρεσης σε πραγματικούς αριθμούς που είναι αποθηκευμένοι σε μορφή κινητής υποδιαστολής.
- ❑ Να κατανοεί ορισμένες εφαρμογές των λογικών πράξεων και των πράξεων μετατόπισης όπως η ενεργοποίηση, η απενεργοποίηση, και η αντιστροφή συγκεκριμένων bit.

Παράδειγμα 4.2

Στην πραγματικότητα, ο τελεστής XOR δεν είναι καινούργιος. Σε κάθε περίπτωση μπορούμε να τον προσομοιώνουμε χρησιμοποιώντας τους άλλους τρεις τελεστές. Για παράδειγμα, οι παρακάτω δύο εκφράσεις είναι ισοδύναμες

$$x \text{ XOR } y \quad \leftrightarrow \quad [x \text{ AND } (\text{NOT } y)] \text{ OR } [(\text{NOT } x) \text{ AND } y]$$

Η ισοδυναμία αποδεικνύεται αν δημιουργήσουμε τους πίνακες αληθείας και για τις δύο.

Μια ιδιότητα

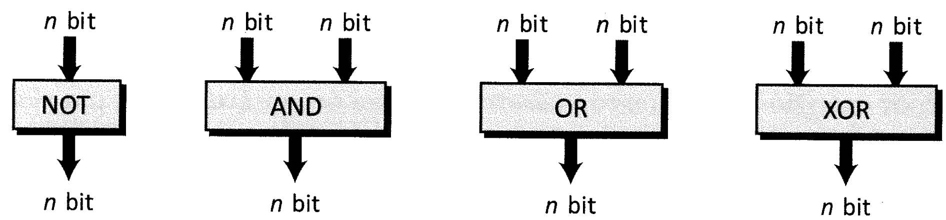
Μια ιδιότητα του XOR είναι ότι, αν ένα bit σε μία είσοδο είναι 1, το αποτέλεσμα είναι το συμπλήρωμα του αντίστοιχου bit στην άλλη είσοδο. Θα χρησιμοποιήσουμε αυτή την ιδιότητα όταν θα περιγράψουμε την εφαρμογή αυτής της πράξης σε σχήματα bit.

Για $x = 0$ ή 1 $\quad$ 1 XOR $x \rightarrow$ NOT x $\quad$ και $\quad$ x XOR 1 $\rightarrow$ NOT x

4.1.2 Λογικές πράξεις σε επίπεδο σχήματος

Και οι τέσσερις τελεστές (NOT, AND, OR, και XOR) μπορούν να εφαρμοστούν σε σχήματα με n bit. Το αποτέλεσμα είναι το ίδιο όπως αν εφαρμόζαμε κάθε τελεστή σε κάθε μεμονωμένο bit για τον τελεστή NOT, και σε κάθε αντίστοιχο ζεύγος bit για τους άλλους τρεις τελεστές. Στην Εικόνα 4.2 φαίνονται αυτοί οι τέσσερις τελεστές με σχήματα εισόδου και εξόδου.

Εικόνα 4.2 *Εφαρμογή λογικών τελεστών σε σχήματα bit*



Παράδειγμα 4.3

Να χρησιμοποιήσετε τον τελεστή NOT στο σχήμα bit 10011000.

Λύση

Η λύση φαίνεται παρακάτω. Ο τελεστής NOT αλλάζει κάθε 0 σε 1 και κάθε 1 σε 0.

NOT	1	0	0	1	1	0	0	0	Είσοδος
	0	1	1	0	0	1	1	1	Έξοδος

Παράδειγμα 4.4

Να χρησιμοποιήσετε τον τελεστή AND στα σχήματα bit 10011000 και 00101010.

Λύση

Η λύση φαίνεται παρακάτω. Παρατηρήστε ότι μόνο ένα bit της εξόδου είναι 1, ενώ και οι δύο αντίστοιχες είσοδοι είναι 1.

	1	0	0	1	1	0	0	0	Είσοδος 1
AND	0	0	1	0	1	0	1	0	Είσοδος 2
	0	**0**	**0**	**0**	**1**	**0**	**0**	**0**	Έξοδος

Παράδειγμα 4.5

Να χρησιμοποιήσετε τον τελεστή OR στα σχήματα bit 10011001 και 00101110.

Λύση

Η λύση φαίνεται παρακάτω. Μόνο ένα bit της εξόδου είναι 0, ενώ και οι δύο αντίστοιχες είσοδοι είναι 0.

	1	0	0	1	1	0	0	1	Είσοδος 1
OR	0	0	1	0	1	1	1	0	Είσοδος 2
	1	**0**	**1**	**1**	**1**	**1**	**1**	**1**	Έξοδος

Παράδειγμα 4.6

Να χρησιμοποιήσετε τον τελεστή XOR στα σχήματα bit 10011001 και 00101110.

Λύση

Η λύση φαίνεται παρακάτω. Μπορείτε να συγκρίνετε την έξοδο αυτού του παραδείγματος με εκείνη του Παραδείγματος 4.5. Η μοναδική διαφορά είναι ότι, όταν και οι δύο είσοδοι είναι 1, το αποτέλεσμα είναι 0 (λόγω του αποκλεισμού).

	1	0	0	1	1	0	0	1	Είσοδος 1
XOR	0	0	1	0	1	1	1	0	Είσοδος 2
	1	**0**	**1**	**1**	**0**	**1**	**1**	**1**	Έξοδος

Εφαρμογές

Για την τροποποίηση ενός σχήματος bit μπορούν να εφαρμοστούν τέσσερις λογικές πράξεις.

Συμπλήρωμα

Η μοναδική εφαρμογή του τελεστή NOT είναι για την εύρεση του συμπληρώματος ολόκληρου του σχήματος. Η εφαρμογή αυτού του τελεστή σε ένα σχήμα έχει αποτέλεσμα την αλλαγή κάθε 0 σε 1 και κάθε 1 σε 0. Μερικές φορές, αυτό αναφέρεται ως πράξη συμπληρώματος ως προς ένα. Στο Παράδειγμα 4.3 παρουσιάζεται η πράξη συμπληρώματος ως προς ένα.

Απενεργοποίηση συγκεκριμένων bit

Μία από τις εφαρμογές του τελεστή AND είναι η απενεργοποίηση συγκεκριμένων bit σε ένα σχήμα bit (δηλαδή η επιβολή του 0 σε αυτά). Σε αυτή την περίπτωση, η δεύτερη είσοδος ονομάζεται **μάσκα**. Τα bit 0 της μάσκας απενεργοποιούν τα αντίστοιχα bit στην πρώτη είσοδο, ενώ τα bit 1 της μάσκας αφήνουν αμετάβλητα τα αντίστοιχα bit στην πρώτη είσοδο. Αυτό οφείλεται στην ιδιότητα που αναφέραμε στην ενότητα του τελεστή AND: Όταν μία από τις εισόδους είναι 0, η έξοδος είναι 0 ανεξάρτητα από την τιμή της άλλης εισόδου. Η απενεργοποίηση bit σε ένα σχήμα έχει πολλές εφαρμογές. Για παράδειγμα, αν σε μια εικόνα χρησιμοποιείται μόνο ένα bit ανά πίξελ (δηλαδή πρόκειται για ασπρόμαυρη εικόνα), τότε μπορούμε να μετατρέψουμε ένα συγκεκριμένο πίξελ σε μαύρο χρησιμοποιώντας μια μάσκα και τον τελεστή AND.

Παράδειγμα 4.7

Να χρησιμοποιήσετε μια μάσκα για να απενεργοποιήσετε (ακυρώσετε) τα πέντε αριστερότερα bit ενός σχήματος. Να ελέγξετε τη μάσκα με το σχήμα 10100110.

Λύση

Η μάσκα είναι 00000111. Το αποτέλεσμα της εφαρμογής της μάσκας είναι το εξής:

	1	0	1	0	0	1	1	0	Είσοδος
AND	0	0	0	0	0	1	1	1	Μάσκα
	0	**0**	**0**	**0**	**0**	1	1	0	Έξοδος

Παρατηρούμε ότι τα τρία δεξιότερα bit παραμένουν αμετάβλητα, ενώ τα πέντε αριστερότερα bit απενεργοποιούνται (δηλαδή μετατρέπονται σε 0) ανεξάρτητα από τις προηγούμενες τιμές τους.

Ενεργοποίηση συγκεκριμένων bit

Μία από τις εφαρμογές του τελεστή OR είναι η ενεργοποίηση συγκεκριμένων bit σε ένα σχήμα bit (δηλαδή η επιβολή του 1 σε αυτά). Και εδώ μπορούμε να χρησιμοποιήσουμε μια μάσκα, η οποία όμως είναι διαφορετική. Τα bit 1 της μάσκας ενεργοποιούν τα αντίστοιχα bit στην πρώτη είσοδο, ενώ τα bit 0 της μάσκας αφήνουν αμετάβλητα τα αντίστοιχα bit στην πρώτη είσοδο. Αυτό οφείλεται στην ιδιότητα που αναφέραμε στην ενότητα του τελεστή OR: Όταν μία από τις εισόδους είναι 1, τότε και η έξοδος είναι 1 ανεξάρτητα από την τιμή της άλλης εισόδου. Η ενεργοποίηση bit σε ένα σχήμα έχει πολλές εφαρμογές. Για παράδειγμα, αν σε μια εικόνα χρησιμοποιείται μόνο ένα bit ανά πίξελ (δηλαδή πρόκειται για ασπρόμαυρη εικόνα), τότε μπορούμε να μετατρέψουμε ένα συγκεκριμένο πίξελ σε λευκό χρησιμοποιώντας μια μάσκα και τον τελεστή OR.

Παράδειγμα 4.8

Να χρησιμοποιήσετε μια μάσκα για να ενεργοποιήσετε τα πέντε αριστερότερα bit ενός σχήματος. Να ελέγξετε τη μάσκα με το σχήμα 10100110.

Λύση

Η μάσκα είναι 11111000. Το αποτέλεσμα της εφαρμογής της μάσκας είναι το εξής:

	1	0	1	0	0	1	1	0	Είσοδος
OR	1	1	1	1	1	0	0	0	Μάσκα
	1	**1**	**1**	**1**	**1**	1	1	0	Έξοδος

Αντιστροφή συγκεκριμένων bit

Μία από τις εφαρμογές του τελεστή XOR είναι η **αντιστροφή** συγκεκριμένων bit σε ένα σχήμα bit (δηλαδή η επιβολή του συμπληρώματός τους σε αυτά). Και εδώ μπορούμε να χρησιμοποιήσουμε μια μάσκα, η οποία όμως είναι διαφορετική. Τα bit 1 της μάσκας αντιστρέφουν τα αντίστοιχα bit στην πρώτη είσοδο, ενώ τα bit 0 της μάσκας αφήνουν αμετάβλητα τα αντίστοιχα bit της πρώτης εισόδου. Αυτό οφείλεται στην ιδιότητα που αναφέραμε στην ενότητα του τελεστή XOR: Όταν μία από τις εισόδους είναι 1, τότε και η έξοδος είναι το συμπλήρωμα του αντίστοιχου bit. Εδώ πρέπει να επισημάνουμε τη διαφορά μεταξύ του τελεστή NOT και του τελεστή XOR. Ο τελεστής NOT συμπληρώνει όλα τα bit της εισόδου, ενώ ο τελεστής XOR συμπληρώνει μόνο τα συγκεκριμένα bit της πρώτης εισόδου που καθορίζονται από τη μάσκα.

Παράδειγμα 4.9

Να χρησιμοποιήσετε μια μάσκα για να αντιστρέψετε τα πέντε αριστερότερα bit ενός σχήματος. Να ελέγξετε τη μάσκα με το σχήμα 10100110.

Λύση

Η μάσκα είναι 11111000. Το αποτέλεσμα της εφαρμογής της μάσκας είναι το εξής:

	1	0	1	0	0	1	1	0	Είσοδος 1
XOR	1	1	1	1	1	0	0	0	Μάσκα
	0	**1**	**0**	**1**	**1**	1	1	0	Έξοδος

4.2 ΠΡΑΞΕΙΣ ΜΕΤΑΤΟΠΙΣΗΣ

Οι πράξεις μετατόπισης (ή ολίσθησης, shift) μετακινούν τα bit σε ένα σχήμα, αλλάζοντας τις θέσεις τους. Τα bit μπορούν να μετακινηθούν προς τα αριστερά ή προς τα δεξιά. Οι πράξεις μετατόπισης χωρίζονται σε δύο κατηγορίες: στις λογικές πράξεις μετατόπισης και στις αριθμητικές πράξεις μετατόπισης.

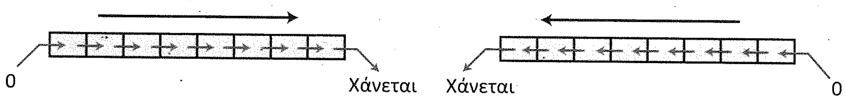
4.2.1 Λογικές πράξεις μετατόπισης

Λογικές πράξεις μετατόπισης εφαρμόζονται σε σχήματα τα οποία δεν αναπαριστούν προσημασμένους αριθμούς. Αυτό συμβαίνει επειδή αυτές οι πράξεις μετατόπισης ενδέχεται να αλλάξουν το πρόσημο ενός αριθμού το οποίο καθορίζεται από το αριστερότερο bit του σχήματος. Υπάρχουν δύο τύποι λογικών πράξεων μετατόπισης, τους οποίους περιγράφουμε στη συνέχεια.

Λογική μετατόπιση

Μια λογική πράξη δεξιάς μετατόπισης μετατοπίζει κάθε bit μία θέση προς τα δεξιά. Σε ένα σχήμα n bit, το δεξιότερο bit χάνεται και το αριστερότερο bit συμπληρώνεται με ένα 0. Μια λογική πράξη αριστερής μετατόπισης μετατοπίζει κάθε bit μία θέση προς τα αριστερά. Σε ένα σχήμα n bit, το αριστερότερο bit χάνεται και το δεξιότερο bit συμπληρώνεται με ένα 0. Στην Εικόνα 4.3 φαίνονται οι λογικές πράξεις δεξιάς και αριστερής μετατόπισης για ένα σχήμα 8 bit.

Εικόνα 4.3 *Λογικές πράξεις μετατόπισης*



α. Λογική δεξιά μετατόπιση β. Λογική αριστερή μετατόπιση

Παράδειγμα 4.10

Να χρησιμοποιήσετε μια λογική πράξη αριστερής μετατόπισης στο σχήμα bit 10011000.

Λύση

Η λύση φαίνεται παρακάτω. Το αριστερότερο bit (λευκό σε μαύρο φόντο) χάνεται και προστίθεται ένα 0 ως το δεξιότερο bit (το «έγχρωμο» bit).

←	**1**	0	0	1	1	0	0	0	Αρχικό
	0	0	1	1	0	0	0	0	Μετά τη μετατόπιση

Κυκλική μετατόπιση

Η **πράξη κυκλικής μετατόπισης** (ή πράξη περιστροφής) μετατοπίζει bit χωρίς όμως να χάνονται ή να προστίθενται bit. Μια δεξιά κυκλική μετατόπιση (ή *δεξιά περιστροφή*) μετατοπίζει κάθε bit μία θέση προς τα δεξιά. Το δεξιότερο bit μετατοπίζεται κυκλικά και γίνεται το αριστερότερο bit.

Μια αριστερή κυκλική μετατόπιση (ή *αριστερή περιστροφή*) μετατοπίζει κάθε bit μία θέση προς τα αριστερά. Το αριστερότερο bit μετατοπίζεται κυκλικά και γίνεται το δεξιότερο bit. Στην Εικόνα 4.4 φαίνεται η πράξη της κυκλικής αριστερής μετατόπισης και της κυκλικής δεξιάς μετατόπισης.

Εικόνα 4.4 *Πράξεις κυκλικής μετατόπισης*

α. Κυκλική δεξιά μετατόπιση β. Κυκλική αριστερή μετατόπιση

Παράδειγμα 4.11

Να χρησιμοποιήσετε μια πράξη αριστερής κυκλικής μετατόπισης στο σχήμα bit 10011000.

Λύση

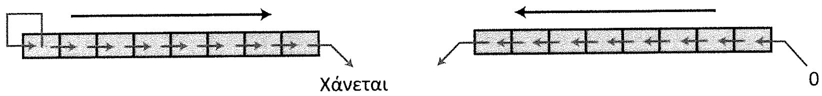
Η λύση φαίνεται παρακάτω. Το αριστερότερο bit (το λευκό bit σε μαύρο φόντο) μετατοπίζεται κυκλικά και γίνεται το δεξιότερο bit.

Αρχικό	1	0	0	1	1	0	0	0	Κυκλική μετατόπιση
Μετά τη μετατόπιση	0	0	1	1	0	0	0	1	

Αριθμητικές πράξεις μετατόπισης

Οι **αριθμητικές πράξεις μετατόπισης** προϋποθέτουν ότι το σχήμα bit είναι ένας προσημασμένος ακέραιος σε μορφή συμπληρώματος ως προς δύο. Η αριθμητική πράξη δεξιάς μετατόπισης χρησιμοποιείται για τη διαίρεση ενός ακεραίου με το δύο, ενώ η αριθμητική πράξη αριστερής μετατόπισης χρησιμοποιείται για τον πολλαπλασιασμό ενός ακεραίου με το δύο (περιγράφεται παρακάτω). Κατά την εφαρμογή αυτών των πράξεων, δεν θα πρέπει να αλλάζει το bit προσήμου (δηλαδή το αριστερότερο bit). Με μια αριθμητική πράξη δεξιάς μετατόπισης το bit προσήμου δεν μεταβάλλεται, όμως αντιγράφεται επίσης στο δεξιότερο bit ώστε το πρόσημο να διατηρείται. Με μια αριθμητική πράξη αριστερής μετατόπισης το bit προσήμου απορρίπτεται και ως πρόσημο καθορίζεται το bit που βρίσκεται αριστερά από αυτό. Αν το νέο bit προσήμου είναι το ίδιο με το προηγούμενο, η πράξη εκτελείται με επιτυχία, διαφορετικά προκύπτει υπερχείλιση ή υποχείλιση και το αποτέλεσμα δεν είναι έγκυρο. Στην Εικόνα 4.5 φαίνονται αυτές οι δύο πράξεις.

Εικόνα 4.5 *Αριθμητικές πράξεις μετατόπισης*



α. Αριθμητική δεξιά μετατόπιση β. Αριθμητική αριστερή μετατόπιση

Παράδειγμα 4.12

Να χρησιμοποιήσετε μια αριθμητική πράξη δεξιάς μετατόπισης στο σχήμα bit 10011001. Το σχήμα είναι ένας ακέραιος σε μορφή συμπληρώματος ως προς δύο.

Λύση

Η λύση φαίνεται παρακάτω. Το αριστερότερο bit διατηρείται και αντιγράφεται επίσης στο δεξιό παρακείμενο bit (το λευκό bit σε μαύρο φόντο). Το «έγχρωμο» bit χάνεται.

Αριθμητική πράξη δεξιάς μετατόπισης									
	1	0	0	1	1	0	0	1	Αρχικό
	1	1	0	0	1	1	0	0	Μετά τη μετατόπιση

Ο αρχικός αριθμός ήταν το –103 και ο νέος αριθμός είναι το –52, το οποίο είναι το αποτέλεσμα της διαίρεσης του –103 με το 2 με περικοπή στον μικρότερο ακέραιο.

Παράδειγμα 4.13

Να χρησιμοποιήσετε μια αριθμητική πράξη αριστερής μετατόπισης στο σχήμα bit 11011001. Το σχήμα είναι ένας ακέραιος σε μορφή συμπληρώματος ως προς δύο.

Λύση

Η λύση φαίνεται παρακάτω. Το αριστερότερο bit (το «έγχρωμο») χάνεται και προστίθεται ένα 0 ως το δεξιότερο bit (λευκό σε μαύρο φόντο).

Αριθμητική πράξη αριστερής μετατόπισης									
	1	1	0	1	1	0	0	1	Αρχικό
	1	0	1	1	0	0	1	0	Μετά τη μετατόπιση

Ο αρχικός αριθμός ήταν το –39 και ο νέος αριθμός είναι το –78. Δηλαδή, ο αρχικός αριθμός πολλαπλασιάζεται με το δύο. Η πράξη είναι έγκυρη αφού δεν προέκυψε υποχείλιση.

Παράδειγμα 4.14

Να χρησιμοποιήσετε μια αριθμητική πράξη αριστερής μετατόπισης στο σχήμα bit 01111111. Το σχήμα είναι ένας ακέραιος σε μορφή συμπληρώματος ως προς δύο.

Λύση

Η λύση φαίνεται παρακάτω. Το αριστερότερο bit (το «έγχρωμο») χάνεται και προστίθεται ένα 0 ως το δεξιότερο bit (λευκό σε μαύρο φόντο).

Αριθμητική πράξη αριστερής μετατόπισης									
	0	1	1	1	1	1	1	1	Αρχικό
	1	1	1	1	1	1	1	0	Μετά τη μετατόπιση

Ο αρχικός αριθμός ήταν το 127 και ο νέος αριθμός είναι το –2. Εδώ το αποτέλεσμα δεν είναι έγκυρο επειδή προέκυψε υπερχείλιση. Αυτό συνέβη επειδή το αναμενόμενο αποτέλεσμα $127 \times 2 = 254$ δεν μπορεί να αναπαρασταθεί από ένα σχήμα μήκους 8 bit.

Παράδειγμα 4.15

Ο συνδυασμός λογικών πράξεων και λογικών πράξεων μετατόπισης μας παρέχει ορισμένα εργαλεία για τον χειρισμό σχημάτων bit. Ας υποθέσουμε ότι έχουμε ένα σχήμα και πρέπει να χρησιμοποιήσουμε το τρίτο bit (από τα δεξιά) του σχήματος σε μια διαδικασία λήψης αποφάσεων. Επομένως, πρέπει να γνωρίζουμε αν το συγκεκριμένο bit είναι 0 ή 1. Εδώ θα δούμε πώς μπορούμε να το διαπιστώσουμε αυτό.

	h	g	f	e	d	c	b	a	Αρχικό
	0	h	g	f	e	d	c	b	Μία δεξιά μετατόπιση
	0	0	h	g	f	e	d	c	Δύο δεξιές μετατοπίσεις
AND	0	0	0	0	0	0	0	1	Μάσκα
	0	0	0	0	0	0	0	c	Αποτέλεσμα

Μετατοπίζουμε το σχήμα δύο bit προς τα δεξιά ώστε το bit που μας ενδιαφέρει να μετακινηθεί στη δεξιότερη θέση. Στη συνέχεια, στο αποτέλεσμα εφαρμόζεται ο τελεστής AND με μια μάσκα στην οποία υπάρχει ένα 1 στην αριστερότερη θέση. Το αποτέλεσμα είναι ένα σχήμα με επτά 0 και το bit που μας ενδιαφέρει στη δεξιότερη θέση. Τώρα μπορούμε να ελέγξουμε το αποτέλεσμα: Αν είναι ένας μη προσημασμένος ακέραιος 1, το bit που μας ενδιαφέρει ήταν 1, ενώ αν το αποτέλεσμα είναι ένας μη προσημασμένος ακέραιος 0, τότε το bit ήταν 0.

4.3 ΑΡΙΘΜΗΤΙΚΕΣ ΠΡΑΞΕΙΣ

Στις **αριθμητικές πράξεις** περιλαμβάνονται η πρόσθεση, η αφαίρεση, ο πολλαπλασιασμός, και η διαίρεση. Αυτές οι πράξεις μπορούν να εφαρμοστούν σε ακεραίους και σε αριθμούς κινητής υποδιαστολής.

4.3.1 Αριθμητικές πράξεις σε ακεραίους

Στους ακέραιους αριθμούς μπορούν να εφαρμοστούν όλες οι αριθμητικές πράξεις, όπως η πρόσθεση, η αφαίρεση, ο πολλαπλασιασμός, και η διαίρεση. Παρόλο που ο πολλαπλασιασμός ή η διαίρεση ακεραίων μπορούν να υλοποιηθούν με επαναλαμβανόμενες προσθέσεις ή αφαιρέσεις, αντίστοιχα, η διαδικασία δεν είναι αποδοτική. Υπάρχουν πιο αποδοτικές διαδικασίες για τον πολλαπλασιασμό και τη διαίρεση, όπως ο αλγόριθμος Booth, όμως αυτές ξεπερνούν τη θεματολογία αυτού του βιβλίου. Γι' αυτό, σε αυτή την ενότητα θα περιγράψουμε μόνο την πρόσθεση και την αφαίρεση ακεραίων.

Πρόσθεση και αφαίρεση ακεραίων συμπληρώματος ως προς δύο

Πρώτα θα ασχοληθούμε με την πρόσθεση και την αφαίρεση ακεραίων στην αναπαράσταση συμπληρώματος ως προς δύο επειδή είναι ευκολότερες. Όπως περιγράψαμε στο Κεφάλαιο 3, οι ακέραιοι συνήθως αποθηκεύονται σε μορφή συμπληρώματος ως προς δύο. Ένα από τα πλεονεκτήματα της αναπαράστασης συμπληρώματος ως προς δύο είναι ότι δεν υπάρχει διαφορά ανάμεσα στην πρόσθεση και την αφαίρεση. Για να εκτελέσει την πράξη της αφαίρεσης, ο υπολογιστής απλώς τη μετατρέπει στην πράξη της πρόσθεσης, χρησιμοποιώντας όμως το συμπλήρωμα ως προς δύο του δεύτερου αριθμού. Με άλλα λόγια:

$$\mathbf{A - B \leftrightarrow A + (\overline{B} + 1)} \quad \textbf{όπου } ((\overline{B} + 1)) \textbf{ είναι το συμπλήρωμα ως προς δύο του B}$$

Αυτό σημαίνει ότι χρειάζεται να περιγράψουμε μόνο την πρόσθεση. Η πρόσθεση αριθμών στη μορφή συμπληρώματος ως προς δύο μοιάζει πολύ με την πρόσθεση αριθμών στο δεκαδικό σύστημα: Προσθέτουμε στήλη προς στήλη και, αν υπάρχει κρατούμενο, το προσθέτουμε στην επόμενη στήλη. Το κρατούμενο που προκύπτει από την τελευταία στήλη, όμως, απορρίπτεται.

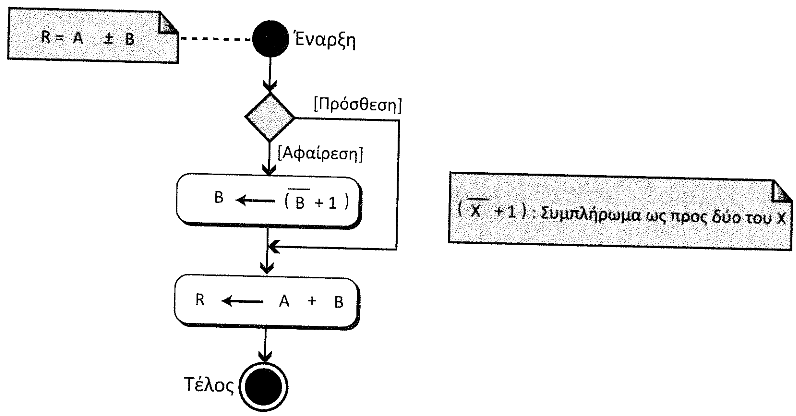
Θα πρέπει να θυμόμαστε ότι προσθέτουμε ακεραίους στήλη προς στήλη. Κάθε στήλη περιέχει είτε δύο bit που πρέπει να προστεθούν, εφόσον δεν υπάρχει κρατούμενο από την προηγούμενη στήλη, είτε τρία bit που πρέπει να προστεθούν, εφόσον υπάρχει κρατούμενο από την προηγούμενη στήλη. Το πλήθος των 1 σε κάθε στήλη μπορεί να είναι μηδέν, ένα, δύο, ή τρία. Στον Πίνακα 4.1 παρουσιάζονται τα αθροίσματα και τα κρατούμενα.

Πίνακας 4.1 Το κρατούμενο και το άθροισμα από την πρόσθεση δύο bit

Στήλη	***Κρατούμενο***	***Άθροισμα***	***Στήλη***	***Κρατούμενο***	***Άθροισμα***
Κανένα 1	0	0	Δύο 1	1	0
Ένα 1	0	1	Τρία 1	1	1

Τώρα μπορούμε να δείξουμε τη διαδικασία πρόσθεσης ή αφαίρεσης δύο ακεραίων σε μορφή συμπληρώματος ως προς δύο (Εικόνα 4.6). Προσέξτε ότι για το συμπλήρωμα ως προς δύο του X χρησιμοποιούμε τον συμβολισμό ($\overline{X}$ + 1). Ο συμβολισμός αυτός χρησιμοποιείται συχνά σε συγγράμματα επειδή το $\overline{X}$ υποδηλώνει το συμπλήρωμα ως προς ένα του X. Αν προσθέσουμε 1 στο συμπλήρωμα ως προς ένα για έναν ακέραιο, τότε παίρνουμε το συμπλήρωμά του ως προς δύο.

Εικόνα 4.6 *Πρόσθεση και αφαίρεση ακεραίων σε μορφή συμπληρώματος ως προς δύο*



Η διαδικασία είναι η εξής:

1. Αν η πράξη είναι αφαίρεση, παίρνουμε το συμπλήρωμα ως προς δύο του δεύτερου ακεραίου. Διαφορετικά, προχωρούμε στο επόμενο βήμα.
2. Προσθέτουμε τους δύο ακεραίους.

Παράδειγμα 4.16

Δύο ακέραιοι A και B είναι αποθηκευμένοι σε μορφή συμπληρώματος ως προς δύο. Να δείξετε τον τρόπο πρόσθεσης του B στον A.

$$A = (00010001)_2 \quad B = (00010110)_2$$

Λύση

Η πράξη που πρέπει να εκτελεστεί είναι πρόσθεση. Ο A προστίθεται στον B και το αποτέλεσμα αποθηκεύεται στο R.

			1						**Κρατούμενο**
	0	0	0	1	0	0	0	1	A
+	0	0	0	1	0	1	1	0	B
	0	0	1	0	0	1	1	1	R

Ελέγχουμε το αποτέλεσμα στο δεκαδικό σύστημα: (+17) + (+22) = (+39).

Παράδειγμα 4.17

Δύο ακέραιοι A και B είναι αποθηκευμένοι σε μορφή συμπληρώματος ως προς δύο. Να δείξετε τον τρόπο πρόσθεσης του B στον A.

$A = (00011000)_2$ $B = (11101111)_2$

Λύση

Η πράξη που πρέπει να εκτελεστεί είναι πρόσθεση. Ο A προστίθεται στον B και το αποτέλεσμα αποθηκεύεται στο R. Ας επισημάνουμε όμως ότι το τελευταίο κρατούμενο απορρίπτεται επειδή το μέγεθος της μνήμης είναι μόνο 8 bit.

1	1	1	1	1					Κρατούμενο
	0	0	0	1	1	0	0	0	A
+	1	1	1	0	1	1	1	1	B
	0	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**R**

Ελέγχουμε το αποτέλεσμα στο δεκαδικό σύστημα: (+24) + (–17) = (+7).

Παράδειγμα 4.18

Δύο ακέραιοι A και B είναι αποθηκευμένοι σε μορφή συμπληρώματος ως προς δύο. Να δείξετε τον τρόπο αφαίρεσης του B από τον A.

$A = (00011000)_2$ $B = (11101111)_2$

Λύση

Η πράξη που πρέπει να εκτελεστεί είναι αφαίρεση. Ο A προστίθεται στο ($\overline{B}$ + 1) και το αποτέλεσμα αποθηκεύεται στο R.

			1						Κρατούμενο
	0	0	0	1	1	0	0	0	A
+	0	0	0	1	0	0	0	1	($\overline{B}$ + 1)
	0	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**R**

Ελέγχουμε το αποτέλεσμα στο δεκαδικό σύστημα: (+24) – (–17) = (+41).

Παράδειγμα 4.19

Δύο ακέραιοι A και B είναι αποθηκευμένοι σε μορφή συμπληρώματος ως προς δύο. Να δείξετε τον τρόπο αφαίρεσης του B από τον A.

$A = (11011101)_2$ $B = (00010100)_2$

Λύση

Η πράξη που πρέπει να εκτελεστεί είναι αφαίρεση. Ο A προστίθεται στο ($\overline{B}$ + 1) και το αποτέλεσμα αποθηκεύεται στο R.

	1	1	1	1	1	1			Κρατούμενο
	1	1	0	1	1	1	0	1	A
+	1	1	1	0	1	1	0	0	($\overline{B}$ + 1)
	1	1	0	0	1	0	0	1	R

Ελέγχουμε το αποτέλεσμα στο δεκαδικό σύστημα: (–35) – (+20) = (–55). Να σημειωθεί ότι το τελευταίο κρατούμενο απορρίπτεται.

Παράδειγμα 4.20

Δύο ακέραιοι A και B είναι αποθηκευμένοι σε μορφή συμπληρώματος ως προς δύο. Να δείξετε τον τρόπο πρόσθεσης του B στον A.

$A = (01111111)_2 \quad B = (00000011)_2$

Λύση

Η πράξη που πρέπει να εκτελεστεί είναι πρόσθεση. Ο A προστίθεται στον B και το αποτέλεσμα αποθηκεύεται στο R.

	1	1	1	1	1	1	1		Κρατούμενο
	0	1	1	1	1	1	1	1	A
+	0	0	0	0	0	0	1	1	B
	1	0	0	0	0	0	1	0	R

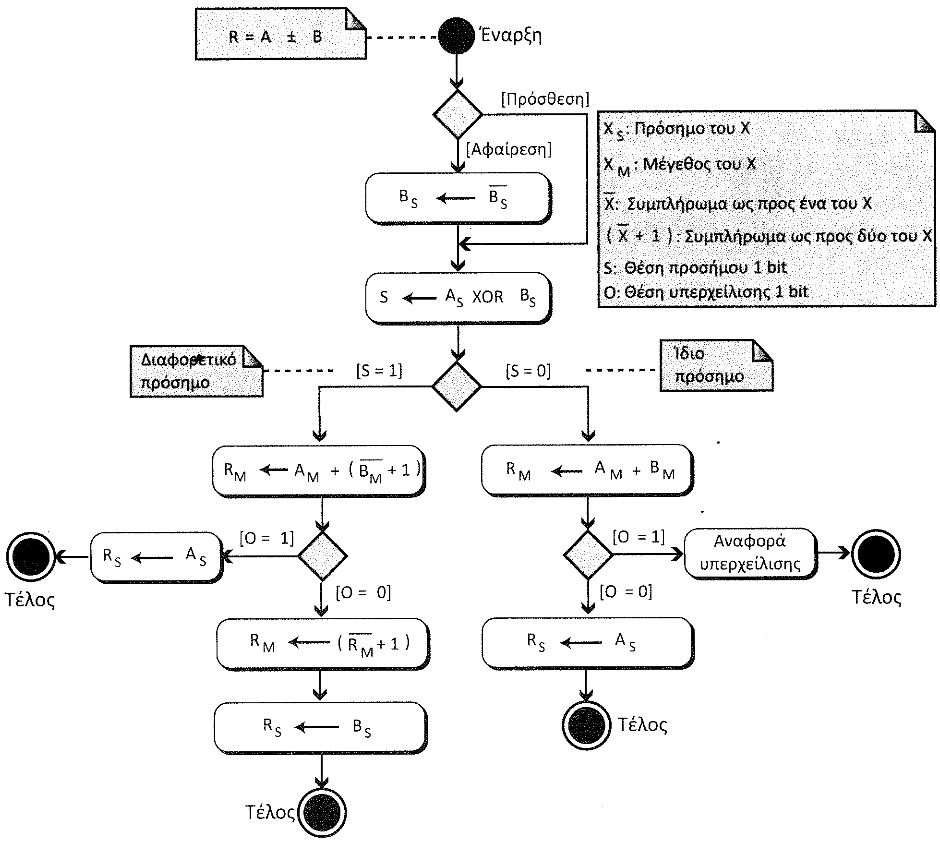
Παρόλο που θα περιμέναμε το αποτέλεσμα να είναι 127 + 3 = 130, η απάντηση είναι –126. Το σφάλμα αυτό οφείλεται σε υπερχείλιση επειδή η αναμενόμενη απάντηση (+130) δεν βρίσκεται στο διάστημα τιμών –128 έως +127.

Όταν εκτελούμε αριθμητικές πράξεις με αριθμούς στον υπολογιστή, θα πρέπει να θυμόμαστε ότι κάθε αριθμός, αλλά και το αποτέλεσμα, πρέπει να ανήκουν στο διάστημα τιμών που ορίζονται από τη δέσμευση bit.

Πρόσθεση ή αφαίρεση ακεραίων προσήμου και μεγέθους

Η πρόσθεση και η αφαίρεση ακεραίων στην αναπαράσταση προσήμου και μεγέθους φαίνονται ιδιαίτερα περίπλοκες. Υπάρχουν τέσσερις διαφορετικοί συνδυασμοί προσήμων (δύο πιθανά πρόσημα για καθεμιά από τις δύο τιμές) για την πρόσθεση, και τέσσερις διαφορετικοί συνδυασμοί για την αφαίρεση. Αυτό σημαίνει ότι πρέπει να λαμβάνουμε υπόψη μας οκτώ διαφορετικές καταστάσεις. Ωστόσο, αν ελέγξουμε πρώτα τα πρόσημα, τότε μπορούμε να μειώσουμε το πλήθος των περιπτώσεων, όπως φαίνεται στην Εικόνα 4.7.

Εικόνα 4.7 *Πρόσθεση και αφαίρεση ακεραίων σε μορφή προσήμου και μεγέθους*



Ας εξηγήσουμε πρώτα το διάγραμμα:

1. Ελέγχουμε την πράξη. Αν η πράξη είναι αφαίρεση, αλλάζουμε το πρόσημο του δεύτερου ακεραίου (B). Αυτό σημαίνει ότι τώρα χρειάζεται μόνο να ασχοληθούμε με την πρόσθεση δύο προσημασμένων ακεραίων.
2. Εφαρμόζουμε την πράξη XOR στα δύο πρόσημα. Αν το αποτέλεσμα (το οποίο αποθηκεύεται στην προσωρινή θέση S) είναι 0, αυτό σημαίνει ότι τα πρόσημα είναι ίδια (δηλαδή, είτε είναι και τα δύο θετικά είτε είναι και τα δύο αρνητικά).
3. Αν τα πρόσημα είναι ίδια, τότε $R = \pm(A_M + B_M)$. Θα πρέπει να προσθέσουμε το μέγεθος, ενώ το πρόσημο του αποτελέσματος είναι το κοινό πρόσημο. Επομένως έχουμε:

   $$R_M = (A_M) + (B_M) \quad \text{και} \quad R_S = A_S$$

   όπου ο δείκτης M δηλώνει το μέγεθος και ο δείκτης S δηλώνει το πρόσημο. Σε αυτή την περίπτωση, όμως, θα πρέπει να προσέξουμε για υπερχείλιση. Κατά την πρόσθεση δύο μεγεθών, ενδέχεται να προκύψει υπερχείλιση η οποία πρέπει να αναφέρεται ώστε να διακόπτεται η διαδικασία.
4. Αν τα πρόσημα είναι διαφορετικά, τότε $R = \pm(A_M - B_M)$. Επομένως πρέπει να αφαιρέσουμε το B_M από το A_M και κατόπιν να λάβουμε μια απόφαση σχετικά με το πρόσημο. Αντί να εκτελέσουμε την αφαίρεση bit προς bit, παίρνουμε το συμπλήρωμα ως προς δύο του δεύτερου μεγέθους (B_M) και το προσθέτουμε. Το πρόσημο του αποτελέσματος είναι το πρόσημο του ακεραίου με το μεγαλύτερο μέγεθος.

α. Αποδεικνύεται ότι αν $A_M \geq B_M$, υπάρχει υπερχείλιση και το αποτέλεσμα είναι θετικός αριθμός. Επομένως, αν υπάρξει υπερχείλιση, την απορρίπτουμε και ορίζουμε ως πρόσημο του αποτελέσματος το πρόσημο του A.

β. Αποδεικνύεται ότι αν $A_M < B_M$, δεν υπάρχει υπερχείλιση, αλλά το αποτέλεσμα είναι αρνητικός αριθμός. Έτσι, όταν δεν υπάρχει υπερχείλιση, βρίσκουμε το συμπλήρωμα ως προς δύο του αποτελέσματος και ορίζουμε ως πρόσημο του αποτελέσματος το πρόσημο του B.

Παράδειγμα 4.21

Δύο ακέραιοι A και B είναι αποθηκευμένοι σε μορφή προσήμου και μεγέθους (έχουμε διαχωρίσει το πρόσημο από το μέγεθος για λόγους σαφήνειας). Να δείξετε τον τρόπο πρόσθεσης του B στον A.

$A = (0\ 0010001)_2 \quad B = (0\ 0010110)_2$

Λύση

Η πράξη που πρέπει να εκτελεστεί είναι πρόσθεση: Το πρόσημο του B δεν αλλάζει. Αφού $S = A_S$ XOR $B_S = 0$, $R_M = A_M + B_M$ και $R_S = A_S$. Δεν υπάρχει υπερχείλιση.

	Πρόσημο	Χωρίς υπερχείλιση			1						Κρατούμενο
A_S	0			0	0	1	0	0	0	1	A_M
B_S	0		+	0	0	1	0	1	1	0	B_M
R_S	0			0	1	0	0	1	1	1	R_M

Ελέγχουμε το αποτέλεσμα στο δεκαδικό σύστημα: (+17) + (+22) = (+39).

Παράδειγμα 4.22

Δύο ακέραιοι A και B είναι αποθηκευμένοι σε μορφή προσήμου και μεγέθους. Να δείξετε τον τρόπο πρόσθεσης του B στον A.

$A = (0\ 0010001)_2 \quad B = (1\ 0010110)_2$

Λύση

Η πράξη που πρέπει να εκτελεστεί είναι πρόσθεση: Το πρόσημο του B δεν αλλάζει. $S = A_S$ XOR $B_S = 1$ και $R_M = A_M + (\overline{B}_M + 1)$. Αφού δεν υπάρχει υπερχείλιση, πρέπει να πάρουμε το συμπλήρωμα ως προς δύο του R_M. Το πρόσημο του R είναι το πρόσημο του B.

	Πρόσημο	Χωρίς υπερχείλιση									Κρατούμενο
A_S	0			0	0	1	0	0	0	1	A_M
B_S	1		+	1	1	0	1	0	1	0	$(\overline{B}_M + 1)$
				1	1	1	1	0	1	1	R_M
R_S	1			0	0	0	0	1	0	1	$R_M = (\overline{R}_M + 1)$

Ελέγχουμε το αποτέλεσμα στο δεκαδικό σύστημα: (+17) + (−22) = (−5).

.γμα 4.23

.ραιοι A και B είναι αποθηκευμένοι σε μορφή προσήμου και μεγέθους. Να δείξετε τον .φαίρεσης του B από τον A.

$$A = (1\ 1010001)_2 \quad B = (1\ 0010110)_2$$

Λύση

Η πράξη που πρέπει να εκτελεστεί είναι αφαίρεση: $B_S = \overline{B}_S$. $S = A_S$ XOR $B_S = 1$, $R_M = A_M + (\overline{B}_M + 1)$. Αφού υπάρχει υπερχείλιση, η τιμή του R_M είναι η τελική. Το πρόσημο του R είναι το πρόσημο του A.

	Πρόσημο	Υπερχείλιση	1								Κρατούμενο
A_S	1			1	0	1	0	0	0	1	A_M
B_S	1		+	1	1	0	1	0	1	0	$(\overline{B}_M + 1)$
R_S	1			0	1	1	1	0	1	1	R_M

Ελέγχουμε το αποτέλεσμα στο δεκαδικό σύστημα: (–81) – (–22) = (–59).

4.3.2 Αριθμητικές πράξεις σε πραγματικούς αριθμούς

Στους πραγματικούς αριθμούς που είναι αποθηκευμένοι σε μορφή κινητής υποδιαστολής μπορούν να εφαρμοστούν όλες οι αριθμητικές πράξεις, όπως η πρόσθεση, η αφαίρεση, ο πολλαπλασιασμός, και η διαίρεση. Ο πολλαπλασιασμός δύο πραγματικών αριθμών απαιτεί τον πολλαπλασιασμό δύο ακεραίων στην αναπαράσταση προσήμου και μεγέθους. Παρόμοια, η διαίρεση δύο πραγματικών αριθμών απαιτεί τη διαίρεση δύο ακεραίων στην αναπαράσταση προσήμου και μεγέθους. Επειδή δεν περιγράψαμε τον πολλαπλασιασμό και τη διαίρεση ακεραίων στην αναπαράσταση προσήμου και μεγέθους, δεν θα περιγράψουμε ούτε τον πολλαπλασιασμό και τη διαίρεση πραγματικών αριθμών, αλλά θα δείξουμε μόνο τις πράξεις της πρόσθεσης και της αφαίρεσης για πραγματικούς αριθμούς.

Πρόσθεση και αφαίρεση πραγματικών αριθμών

Η πρόσθεση και η αφαίρεση πραγματικών αριθμών αποθηκευμένων σε μορφή κινητής υποδιαστολής περιορίζεται στην πρόσθεση και την αφαίρεση δύο ακεραίων αποθηκευμένων στη μορφή προσήμου και μεγέθους (συνδυασμός προσήμου και σημαινόμενου τμήματος) μετά τη στοίχιση των υποδιαστολών. Στην Εικόνα 4.8 παρουσιάζεται με απλουστευμένο τρόπο η διαδικασία (υπάρχουν ορισμένες ειδικές περιπτώσεις που έχουμε παραλείψει).

Η απλοποιημένη διαδικασία έχει ως εξής:

1. Αν οποιοσδήποτε από τους δύο αριθμούς (A ή B) είναι μηδέν, τότε το αποτέλεσμα είναι 0 και η διαδικασία διακόπτεται.
2. Αν η πράξη είναι αφαίρεση, αλλάζουμε το πρόσημο του δεύτερου αριθμού (B) ώστε να προσομοιώσουμε την πράξη της πρόσθεσης.
3. Αποκανονικοποιούμε και τους δύο αριθμούς με συμπερίληψη του κρυφού 1 στο σημαινόμενο τμήμα και την αύξηση των εκθετών. Τώρα μπορούμε να χειριστούμε το σημαινόμενο τμήμα ως ακέραιο.
4. Στη συνέχεια ευθυγραμμίζουμε τους εκθέτες, δηλαδή αυξάνουμε τον μικρότερο εκθέτη και μετατοπίζουμε το αντίστοιχο σημαινόμενο τμήμα μέχρι και τα δύο να έχουν τον ίδιο εκθέτη. Για παράδειγμα, αν έχουμε:

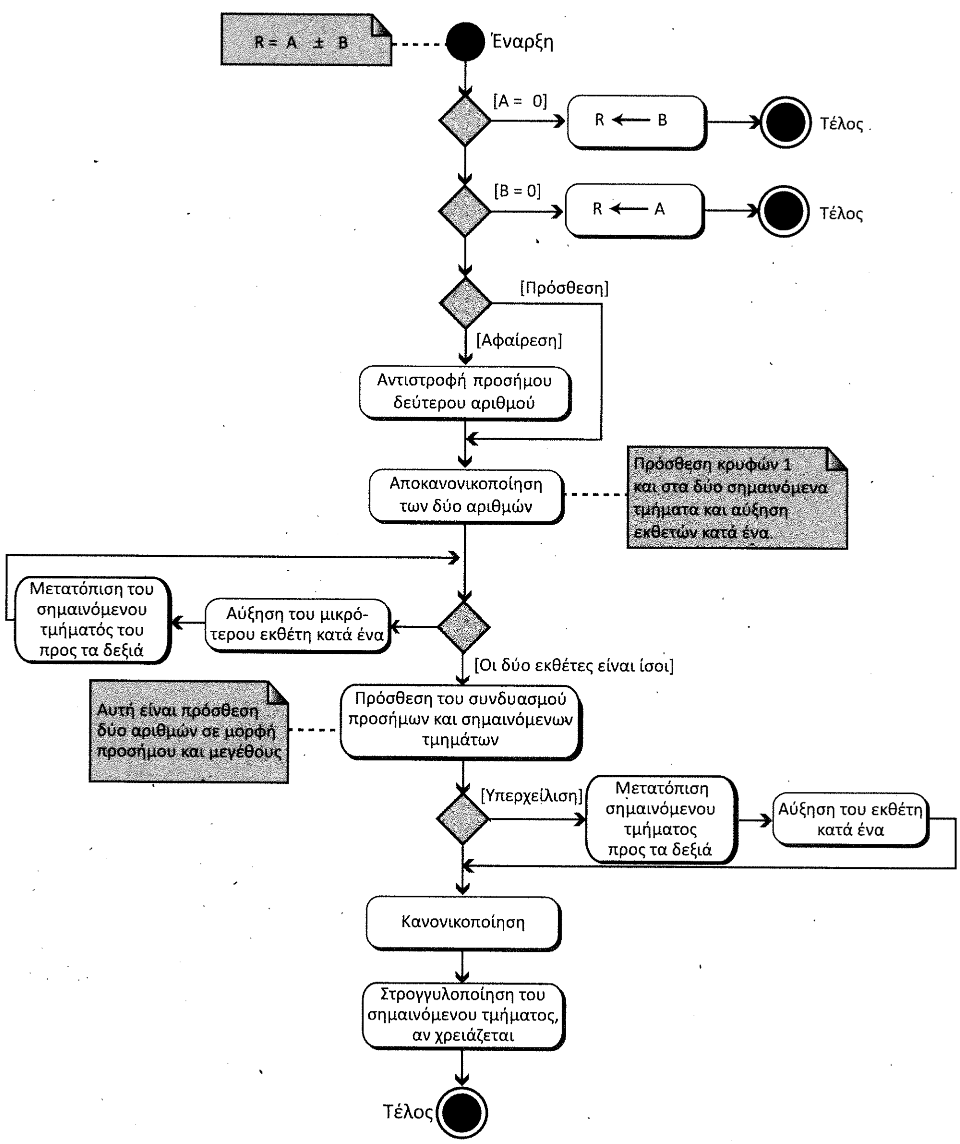
$$1{,}11101 \times 2^4 + 1{,}01 \times 2^2$$

θα πρέπει να κάνουμε και τους δύο εκθέτες 4:

$$1{,}11101 \times 2^4 + 0{,}0101 \times 2^4$$

5. Τώρα μπορούμε να χειριστούμε τον συνδυασμό του προσήμου και σημαινόμενου τμήματος κάθε αριθμού ως ακέραιο σε μορφή προσήμου και μεγέθους. Προσθέτουμε αυτούς τους δύο ακεραίους με τον τρόπο που περιγράψαμε νωρίτερα σε αυτό το κεφάλαιο.
6. Τέλος, κανονικοποιούμε πάλι τον αριθμό σε $1{,}000111 \times 2^5$.

Εικόνα 4.8 *Πρόσθεση και αφαίρεση πραγματικών αριθμών σε μορφή κινητής υποδιαστολής*



Παράδειγμα 4.24

Να δείξετε πώς ο υπολογιστής παράγει το αποτέλεσμα (+5,75) + (+161,875) = (+167,625).

Λύση

Όπως είδαμε στο Κεφάλαιο 3, αυτοί οι δύο αριθμοί αποθηκεύονται σε μορφή κινητής υποδιαστολής, όπως φαίνεται παρακάτω, όμως πρέπει να θυμόμαστε ότι καθένας τους έχει ένα κρυφό 1 (το οποίο δεν αποθηκεύεται, αλλά εννοείται). Θυμίζουμε ότι S είναι το πρόσημο, E ο εκθέτης, και M το σημαινόμενο τμήμα.

	S	E	M
A	0	10000001	01110000000000000000000
B	0	10000110	01000011110000000000000

Τα πρώτα βήματα του διαγράμματος UML (Εικόνα 4.8) είναι περιττά. Έτσι προχωρούμε στην αποκανονικοποίηση των αριθμών προσθέτοντας το κρυφό 1 στο σημαινόμενο τμήμα και αυξάνοντας τον εκθέτη. Τώρα και τα δύο αποκανονικοποιημένα σημαινόμενα τμήματα είναι 24 bit και περιλαμβάνουν το κρυφό 1. Αυτό σημαίνει ότι θα πρέπει να αποθηκευτούν σε μια θέση που μπορεί να δεχθεί και τα 24 bit. Αυξάνεται κάθε εκθέτης.

	S	E	Αποκανονικοποιημένο M
A	0	10000010	101110000000000000000000
B	0	10000111	101000011110000000000000

Τώρα πρέπει να ευθυγραμμίσουμε τα σημαινόμενα τμήματα. Για να το κάνουμε αυτό, πρέπει να αυξήσουμε τον πρώτο εκθέτη και να μετατοπίσουμε το σημαινόμενο τμήμα προς τα δεξιά. Επειδή αλλάζουμε τον πρώτο εκθέτη σε $(10000111)_2$, πρέπει να μετατοπίσουμε το πρώτο σημαινόμενο τμήμα πέντε θέσεις προς τα δεξιά.

	S	E	Αποκανονικοποιημένο M
A	0	10000111	000001011100000000000000
B	0	10000111	101000011110000000000000

Τώρα εκτελούμε πρόσθεση στην αναπαράσταση προσήμου και μεγέθους, αφού μπορούμε να χειριστούμε το πρόσημο και το σημαινόμενο τμήμα κάθε αριθμού ως ακέραιο αποθηκευμένο στην αναπαράσταση προσήμου και μεγέθους.

	S	E	Αποκανονικοποιημένο M
R	0	10000111	101001111010000000000000

Επειδή δεν υπάρχει υπερχείλιση, προχωρούμε στην κανονικοποίηση του σημαινόμενου τμήματος.

	S	E	M
R	0	10000110	01001111010000000000000

Το σημαινόμενο τμήμα είναι μόνο 23 bit, και έτσι δεν χρειάζεται να στρογγυλοποιήσουμε. E = $(10000110)_2$ = 134 M = 0100111101. Δηλαδή, το αποτέλεσμα είναι $(1{,}0100111101)_2 \times 2^{134-127}$ = $(10100111{,}101)_2$ = 167,625.

Παράδειγμα 4.25

Να δείξετε πώς ο υπολογιστής παράγει το αποτέλεσμα (+5,75) + (–7,0234375) = –1,2734375.

Λύση

Αυτοί οι δύο αριθμοί μπορούν να αποθηκευτούν σε μορφή κινητής υποδιαστολής, με τον τρόπο που παρουσιάζεται παρακάτω:

	S	E	M
A	0	10000001	01110000000000000000000
B	1	10000001	11000001100000000000000

Με την αποκανονικοποίηση παίρνουμε το εξής αποτέλεσμα:

	S	E	Αποκανονικοποιημένο M
A	0	10000010	101110000000000000000000
B	1	10000010	111000001100000000000000

Επειδή δεν απαιτείται ευθυγράμμιση (και οι δύο εκθέτες έχουν ίδια τιμή), εφαρμόζουμε την πράξη της πρόσθεσης στους συνδυασμούς προσήμου και σημαινόμενου τμήματος. Το πρόσημο του αποτελέσματος, που παρουσιάζεται παρακάτω, είναι αρνητικό:

	S	E	Αποκανονικοποιημένο M
R	1	10000010	001010001100000000000000

Τώρα πρέπει να κάνουμε κανονικοποίηση. Μειώνουμε τον εκθέτη τρεις φορές και μετατοπίζουμε το αποκανονικοποιημένο σημαινόμενο τμήμα τρεις θέσεις προς τα αριστερά:

	S	E	M
R	1	01111111	010001100000000000000000

Το σημαινόμενο τμήμα τώρα είναι 24 bit, γι' αυτό το στρογγυλοποιούμε στα 23 bit.

	S	E	M
R	1	01111111	01000110000000000000000

Το αποτέλεσμα είναι R = $-2^{127-127} \times 1{,}0100011 = -1{,}2734375$, δηλαδή το αναμενόμενο.

4.4 ΣΥΝΟΔΕΥΤΙΚΟ ΥΛΙΚΟ ΚΕΦΑΛΑΙΟΥ

4.4.1 Προτεινόμενα βιβλία

Για περισσότερες λεπτομέρειες σχετικά με τα θέματα που περιγράφονται σε αυτό το κεφάλαιο σας προτείνουμε τα παρακάτω βιβλία:

- Mano, M. *Computer System Architecture*, Upper Saddle River, NJ: Prentice Hall, 1993
- Null, L. και Lobur, J. *Computer Organization and Architecture*, Sudbury, MA: Jones and Bartlett, 2003
- Stallings, W. *Computer Organization and Architecture,* Upper Saddle River, NJ: Prentice Hall, 2000

4.4.2 Βασικοί όροι

άλγεβρα Boole 100	αριθμητική πράξη 108
αριθμητική πράξη μετατόπισης 106	λογική πράξη 100
λογική πράξη μετατόπισης 105	μάσκα 103
πίνακας αληθείας 100	πράξη κυκλικής μετατόπισης 105
πράξη/τελεστής AND 101	πράξη/τελεστής NOT 100
πράξη/τελεστής OR 101	πράξη/τελεστής XOR 101

4.4.3 Περίληψη

- Οι πράξεις με δεδομένα μπορούν να ταξινομηθούν σε τρεις μεγάλες κατηγορίες: τις λογικές πράξεις, τις πράξεις μετατόπισης (ή ολίσθησης), και τις αριθμητικές πράξεις. Οι λογικές πράξεις είναι αυτές κατά τις οποίες εφαρμόζεται η ίδια βασική πράξη σε μεμονωμένα bit ενός σχήματος ή σε δύο αντίστοιχα bit δύο σχημάτων. Οι πράξεις μετατόπισης μετακινούν τα bit στο σχήμα. Στις αριθμητικές πράξεις περιλαμβάνονται η πρόσθεση, η αφαίρεση, ο πολλαπλασιασμός, και η διαίρεση.
- Οι τέσσερις λογικοί τελεστές που περιγράφηκαν σε αυτό το κεφάλαιο (NOT, AND, OR, και XOR) μπορούν να χρησιμοποιηθούν σε επίπεδο bit ή σε επίπεδο σχήματος. Ο τελεστής NOT είναι μονομελής, ενώ οι τελεστές AND, OR, και XOR είναι διμελείς.
- Η μοναδική εφαρμογή του τελεστή NOT είναι για την εύρεση του συμπληρώματος ολόκληρου του σχήματος. Μία από τις εφαρμογές του τελεστή AND είναι η απενεργοποίηση συγκεκριμένων bit σε ένα σχήμα bit (δηλαδή η επιβολή του 0 σε αυτά). Μία από τις εφαρμογές του τελεστή OR είναι η ενεργοποίηση συγκεκριμένων bit σε ένα σχήμα bit (δηλαδή η επιβολή του 1 σε αυτά). Μία από τις εφαρμογές του τελεστή XOR είναι η αντιστροφή συγκεκριμένων bit σε ένα σχήμα bit (δηλαδή η επιβολή του συμπληρώματός τους σε αυτά).
- Οι πράξεις μετατόπισης (ή ολίσθησης, shift) μετακινούν τα bit ενός σχήματος: αλλάζουν τις θέσεις των bit. Οι πράξεις μετατόπισης χωρίζονται σε δύο κατηγορίες: στις λογικές πράξεις μετατόπισης και στις αριθμητικές πράξεις μετατόπισης. Λογικές πράξεις μετατόπισης εφαρμόζονται σε σχήματα τα οποία δεν αναπαριστούν προσημασμένους αριθμούς. Οι αριθμητικές πράξεις μετατόπισης προϋποθέτουν ότι το σχήμα bit είναι ένας προσημασμένος ακέραιος σε μορφή συμπληρώματος ως προς δύο.
- Στους ακέραιους αριθμούς μπορούν να εφαρμοστούν όλες οι αριθμητικές πράξεις, όπως η πρόσθεση, η αφαίρεση, ο πολλαπλασιασμός, και η διαίρεση. Οι ακέραιοι συνήθως αποθηκεύονται σε μορφή συμπληρώματος ως προς δύο. Ένα από τα πλεονεκτήματα της αναπαράστασης συμπληρώματος ως προς δύο είναι ότι δεν υπάρχει διαφορά ανάμεσα στην πρόσθεση και την αφαίρεση. Για να εκτελέσει την πράξη της αφαίρεσης, ο υπολογιστής απλώς τη μετατρέπει στην πράξη της πρόσθεσης, παράγοντας όμως το συμπλήρωμα ως προς δύο του δεύτερου αριθμού. Η πρόσθεση και η αφαίρεση ακεραίων στην αναπαράσταση προσήμου και μεγέθους φαίνονται ιδιαίτερα περίπλοκες. Υπάρχουν οκτώ καταστάσεις που πρέπει να εξετάζουμε.

- ❑ Στους πραγματικούς αριθμούς που είναι αποθηκευμένοι σε μορφή κινητής υποδιαστολής μπορούν να εφαρμοστούν όλες οι αριθμητικές πράξεις, όπως η πρόσθεση, η αφαίρεση, ο πολλαπλασιασμός, και η διαίρεση. Η πρόσθεση και η αφαίρεση πραγματικών αριθμών αποθηκευμένων σε μορφή κινητής υποδιαστολής περιορίζεται στην πρόσθεση και την αφαίρεση δύο ακεραίων αποθηκευμένων στη μορφή προσήμου και μεγέθους μετά την ευθυγράμμιση των υποδιαστολών.

4.5 ΠΡΑΚΤΙΚΗ ΕΞΑΣΚΗΣΗ

4.5.1 Ερωτήσεις γνώσεων

Γ4-1. _______ είναι αριθμητική πράξη bit.

α. Το αποκλειστικό OR
β. Το μονομελές NOT
γ. Η αφαίρεση
δ. Το δυαδικό AND

Γ4-2. _______ είναι λογικός τελεστής bit.

α. Το αποκλειστικό OR
β. Το μονομελές NOT
γ. Το δυαδικό AND
δ. Όλα τα παραπάνω

Γ4-3. Η μέθοδος αναπαράστασης ακεραίων _______ είναι η πιο συνηθισμένη για την αποθήκευση ακεραίων στη μνήμη του υπολογιστή.

α. προσήμου και μεγέθους
β. συμπληρώματος ως προς ένα
γ. συμπληρώματος ως προς δύο
δ. μη προσημασμένων ακεραίων

Γ4-4. Αν στην πρόσθεση συμπληρώματος ως προς δύο υπάρχει κάποιο τελικό κρατούμενο μετά την πρόσθεση της αριστερότερης στήλης, _______.

α. το προσθέτουμε στη δεξιότερη στήλη
β. το προσθέτουμε στην αριστερότερη στήλη
γ. το απορρίπτουμε
δ. αυξάνουμε το πλήθος των bit

Γ4-5. Για δέσμευση 8 bit, ο μικρότερος δεκαδικός αριθμός που μπορεί να αναπαρασταθεί στη μορφή συμπληρώματος ως προς δύο είναι ο _______.

α. –8 γ. –128
β. –127 δ. –256

Γ4-6. Για δέσμευση 8 bit, ο μεγαλύτερος δεκαδικός αριθμός που μπορεί να αναπαρασταθεί στη μορφή συμπληρώματος ως προς δύο είναι ο _______.

α. 8 γ. 128
β. 127 δ. 256

Γ4-7. Σε αναπαράσταση συμπληρώματος ως προς δύο με δέσμευση 4 bit, η πρόσθεση του 1 με το 7 δίνει αποτέλεσμα _______.

α. 8
β. 1
γ. –7
δ. –8

Γ4-8. Σε αναπαράσταση συμπληρώματος ως προς δύο με δέσμευση 4 bit, η πρόσθεση του 5 με το 5 δίνει αποτέλεσμα _______.

α. –5
β. –6
γ. –7
δ. 10

Γ4-9. Αν ο εκθέτης σε μορφή πλεονάσματος του 127 είναι ο δυαδικός αριθμός 10000101, τότε ο εκθέτης στο δεκαδικό σύστημα είναι ο _______.

α. 6
β. 7
γ. 8
δ. 9

Γ4-10. Κατά την πρόσθεση δύο αριθμών, ο ένας από τους οποίους έχει τιμή εκθέτη 7 και ο άλλος τιμή εκθέτη 9, η υποδιαστολή του μικρότερου αριθμού πρέπει να μετατοπιστεί _______.

α. μία θέση προς τα αριστερά
β. μία θέση προς τα δεξιά
γ. δύο θέσεις προς τα αριστερά
δ. δύο θέσεις προς τα δεξιά

Γ4-11. Ο διμελής τελεστής _______ δέχεται δύο εισόδους και παράγει μία έξοδο.

α. AND
β. OR
γ. XOR
δ. όλοι οι παραπάνω

Γ4-12. Ο μονομελής τελεστής _______ αντιστρέφει τη μοναδική του είσοδο.

α. AND
β. OR
γ. NOT
δ. XOR

Γ4-13. Στον διμελή τελεστή _______, αν η είσοδος είναι δύο 0, τότε η έξοδος είναι 0.

α. AND
β. OR
γ. XOR
δ. σε όλους τους παραπάνω

Γ4-14. Στον διμελή τελεστή _______, αν η είσοδος είναι δύο 1, τότε η έξοδος είναι 0.

α. AND
β. OR
γ. XOR
δ. σε όλους τους παραπάνω

Γ4-15. Στον διμελή τελεστή AND, μόνο η είσοδος _______ δίνει έξοδο 1.

α. δύο 0
β. δύο 1
γ. ένα 0 και ένα 1
δ. δύο 2

Γ4-16. Στον διμελή τελεστή OR, μόνο η είσοδος _______ δίνει έξοδο 0.

α. δύο 0
β. δύο 1
γ. ένα 0 και ένα 1
δ. δύο 2

Γ4-17. Για τη μετατροπή ενός σχήματος bit χρησιμοποιείται ένα άλλο σχήμα bit που ονομάζεται _______.

α. μάσκα
β. κρατούμενο
γ. κινητή υποδιαστολή
δ. byte

Γ4-18. Για να αντιστρέψουμε όλα τα bit ενός σχήματος, κατασκευάζουμε μια μάσκα με όλα τα bit 1 και μετά εφαρμόζουμε τον τελεστή _______ στο σχήμα bit και τη μάσκα.

α. AND
β. OR
γ. XOR
δ. NOT

Γ4-19. Για να απενεργοποιήσουμε όλα τα bit ενός σχήματος (δηλαδή για να τους επιβάλουμε το 0), κατασκευάζουμε μια μάσκα με όλα τα bit 0 και μετά εφαρμόζουμε τον τελεστή _______ στο σχήμα bit και τη μάσκα.

α. AND
β. OR
γ. XOR
δ. NOT

Γ4-20. Για να ενεργοποιήσουμε όλα τα bit ενός σχήματος (δηλαδή για να τους επιβάλουμε το 1), κατασκευάζουμε μια μάσκα με όλα τα bit 1 και μετά εφαρμόζουμε τον τελεστή _______ στο σχήμα bit και τη μάσκα.

α. AND
β. OR
γ. XOR
δ. NOT

4.5.2 Ερωτήσεις επανάληψης

Ε4-1. Ποια είναι η διαφορά μεταξύ μιας αριθμητικής πράξης και μιας λογικής πράξης;

Ε4-2. Κατά την πρόσθεση ακεραίων στη μορφή συμπληρώματος ως προς δύο, τι συμβαίνει στο κρατούμενο από την αριστερότερη στήλη;

Ε4-3. Μπορεί το *n*, η δέσμευση bit, να ισούται με 1; Να δικαιολογήσετε την απάντησή σας.

Ε4-4. Να δώσετε τον ορισμό του όρου *υπερχείλιση*.

Ε4-5. Πώς γίνεται η ρύθμιση της αναπαράστασης των αριθμών με διαφορετικούς εκθέτες στην πρόσθεση αριθμών κινητής υποδιαστολής;

Ε4-6. Ποια είναι η διαφορά μεταξύ μιας μονομελούς και μιας διμελούς πράξης;

Ε4-7. Να αναφέρετε τις λογικές διμελείς πράξεις.

Ε4-8. Τι είναι ο πίνακας αλήθειας;

Ε4-9. Ποια είναι η λειτουργία του τελεστή NOT;

Ε4-10. Πότε η τιμή του αποτελέσματος ενός τελεστή AND είναι ίση με «αληθής»;

Ε4-11. Πότε η τιμή του αποτελέσματος ενός τελεστή OR είναι ίση με «αληθής»;

Ε4-12. Πότε η τιμή του αποτελέσματος ενός τελεστή XOR είναι ίση με «αληθής»;

Ε4-13. Να αναφέρετε μια σημαντική ιδιότητα του τελεστή AND που περιγράφηκε σε αυτό το κεφάλαιο.

Ε4-14. Να αναφέρετε μια σημαντική ιδιότητα του τελεστή OR που περιγράφηκε σε αυτό το κεφάλαιο.

Ε4-15. Να αναφέρετε μια σημαντική ιδιότητα του τελεστή XOR που περιγράφηκε σε αυτό το κεφάλαιο.

Ε4-16. Ποια διμελής πράξη μπορεί να χρησιμοποιηθεί για την απενεργοποίηση bit; Τι σχήμα bit πρέπει να έχει η μάσκα;

Ε4-17. Ποια διμελής πράξη μπορεί να χρησιμοποιηθεί για την απενεργοποίηση bit; Τι σχήμα bit πρέπει να έχει η μάσκα;

Ε4-18. Ποια διμελής πράξη μπορεί να χρησιμοποιηθεί για την αντιστροφή bit; Τι σχήμα bit πρέπει να έχει η μάσκα;

Ε4-19. Ποια είναι η διαφορά μεταξύ των λογικών και των αριθμητικών μετατοπίσεων;

4.5.3 Προβλήματα

Π4-1. Να υπολογίσετε το αποτέλεσμα των παρακάτω πράξεων:

α. NOT $(99)_{16}$
β. NOT $(FF)_{16}$
γ. NOT $(00)_{16}$
δ. NOT $(01)_{16}$

Π4-2. Να υπολογίσετε το αποτέλεσμα των παρακάτω πράξεων:

α. $(99)_{16}$ AND $(99)_{16}$
β. $(99)_{16}$ AND $(00)_{16}$
γ. $(99)_{16}$ AND $(FF)_{16}$
δ. $(FF)_{16}$ AND $(FF)_{16}$

Π4-3. Να υπολογίσετε το αποτέλεσμα των παρακάτω πράξεων:

α. $(99)_{16}$ OR $(99)_{16}$
β. $(99)_{16}$ OR $(00)_{16}$
γ. $(99)_{16}$ OR $(FF)_{16}$
δ. $(FF)_{16}$ OR $(FF)_{16}$

Π4-4. Να υπολογίσετε το αποτέλεσμα των παρακάτω πράξεων:

α. NOT [$(99)_{16}$ OR $(99)_{16}$]
β. $(99)_{16}$ OR [NOT $(00)_{16}$]
γ. [$(99)_{16}$ AND $(33)_{16}$] OR [$(00)_{16}$ AND $(FF)_{16}$]
δ. $(99)_{16}$ OR $(33)_{16}$ AND [$(00)_{16}$ OR $(FF)_{16}$]

Π4-5. Σε ένα σχήμα πρέπει να απενεργοποιηθούν τα τέσσερα αριστερότερα bit (δηλαδή να τους επιβληθεί το 0). Να δείξετε τη μάσκα και την πράξη.

Π4-6. Σε ένα σχήμα πρέπει να ενεργοποιηθούν τα τέσσερα δεξιότερα bit (δηλαδή να τους επιβληθεί το 1). Να δείξετε τη μάσκα και την πράξη.

Π4-7. Σε ένα σχήμα πρέπει να αντιστραφούν τα τρία δεξιότερα και τα δύο αριστερότερα bit. Να δείξετε τη μάσκα και την πράξη.

Π4-8. Σε ένα σχήμα πρέπει να απενεργοποιηθούν τα τρία αριστερότερα bit και να ενεργοποιηθούν τα δύο δεξιότερα. Να δείξετε τις μάσκες και τις πράξεις.

Π4-9. Να χρησιμοποιήσετε την πράξη της μετατόπισης για να διαιρέσετε έναν ακέραιο με το 4.

Π4-10. Να χρησιμοποιήσετε την πράξη της μετατόπισης για να πολλαπλασιάσετε έναν ακέραιο με το 8.

Π4-11. Να χρησιμοποιήσετε συνδυασμό λογικών πράξεων και πράξεων μετατόπισης για να εξαγάγετε το τέταρτο και το πέμπτο bit, από τα αριστερά, ενός μη προσημασμένου ακεραίου.

Π4-12. Χρησιμοποιώντας δέσμευση 8 bit, να μετατρέψετε καθέναν από τους παρακάτω ακεραίους σε μορφή συμπληρώματος ως προς δύο, να κάνετε την πράξη, και κατόπιν να μετατρέψετε το αποτέλεσμα στο δεκαδικό:

α. $19 + 23$
β. $19 - 23$
γ. $-19 + 23$
δ. $-19 - 23$

Π4-13. Χρησιμοποιώντας δέσμευση 16 bit, να μετατρέψετε καθέναν από τους παρακάτω αριθμούς σε μορφή συμπληρώματος ως προς δύο, να κάνετε την πράξη, και κατόπιν να μετατρέψετε το αποτέλεσμα στο δεκαδικό:

α. $161 + 1023$
β. $161 - 1023$
γ. $-161 + 1023$
δ. $-161 - 1023$

Π4-14. Ποια από τις παρακάτω πράξεις δημιουργεί υπερχείλιση αν οι αριθμοί και τα αποτελέσματα αναπαρίστανται σε μορφή συμπληρώματος ως προς δύο 8 bit;

α. 11000010 + 00111111
β. 00000010 + 00111111
γ. 11000010 + 11111111
δ. 00000010 + 11111111

Π4-15. Χωρίς να εκτελέσετε την πράξη, μπορείτε να καταλάβετε ποια από τις επόμενες πράξεις δημιουργεί υπερχείλιση αν οι αριθμοί και τα αποτελέσματα αναπαρίστανται σε μορφή συμπληρώματος ως προς δύο 8 bit;

α. $32 + 105$

β. $32 - 105$

γ. $-32 + 105$

δ. $-32 - 105$

Π4-16. Να υπολογίσετε το αποτέλεσμα των παρακάτω πράξεων θεωρώντας ότι οι αριθμοί είναι αποθηκευμένοι σε αναπαράσταση συμπληρώματος ως προς δύο 16 bit. Να δείξετε το αποτέλεσμα σε δεκαεξαδικό συμβολισμό:

α. $(012A)_{16} + (0E27)_{16}$

β. $(712A)_{16} + (9E00)_{16}$

γ. $(8011)_{16} + (0001)_{16}$

δ. $(E12A)_{16} + (9E27)_{16}$

Π4-17. Χρησιμοποιώντας δέσμευση 8 bit, να μετατρέψετε καθέναν από τους παρακάτω αριθμούς σε μορφή προσήμου και μεγέθους, να κάνετε την πράξη, και έπειτα να μετατρέψετε το αποτέλεσμα στο δεκαδικό:

α. $19 + 23$

β. $19 - 23$

γ. $-19 + 23$

δ. $-19 - 23$

Π4-18. Να δείξετε το αποτέλεσμα των παρακάτω πράξεων κινητής υποδιαστολής χρησιμοποιώντας τη μορφή κατά IEEE του πλεονάσματος του 127 –δείτε το Κεφάλαιο 3:

α. $34{,}75 + 23{,}125$

β. $-12{,}625 + 451{,}00$

γ. $33{,}1875 - 0{,}4375$

δ. $-344{,}3125 - 123{,}5625$

Π4-19. Σε ποιες από τις επόμενες περιπτώσεις δεν προκύπτει ποτέ υπερχείλιση; Να δικαιολογήσετε την απάντησή σας.

α. Κατά την πρόσθεση δύο θετικών ακεραίων.

β. Κατά την πρόσθεση ενός θετικού ακεραίου με έναν αρνητικό ακέραιο.

γ. Κατά την αφαίρεση ενός θετικού ακεραίου από έναν αρνητικό ακέραιο.

δ. Κατά την αφαίρεση δύο αρνητικών ακεραίων.

Π4-20. Ποιο είναι το αποτέλεσμα που προκύπτει από την πρόσθεση ενός ακεραίου με το συμπλήρωμά του ως προς ένα;

Π4-21. Ποιο είναι το αποτέλεσμα που προκύπτει από την πρόσθεση ενός ακεραίου με το συμπλήρωμά του ως προς δύο;