Κεφάλαιο 2

Σύνοψη
Στο κεφάλαιο αυτό θα παρουσιαστεί αναλυτικά η κωδικοποίηση των αριθμών μέσω του δυαδικού συστήματος αρίθμησης. Αρχικά περιλαμβάνονται τα ψηφία του δεκαδικού συστήματος κωδικοποιημένα στο δυαδικό. Ακολουθούν τα ψηφία του οκταδικού και του δεκαεξαδικού συστήματος κωδικοποιημένα επίσης στο δυαδικό. Κατόπιν ακολουθεί η παρουσίαση των κωδίκων που χρησιμοποιούνται συχνά. Έτσι αναλύονται οι: δυαδικός κώδικας (Binary code), οι κώδικες βαρών 5421, 5311, ο κώδικας Aiken ή 2421, ο XS3 (Υπερβολής κατά τρία), ο δυπεντadικός, ο δυδεκαδικός κώδικας. Τέλος αναφορά γίνεται στους κώδικες 2-από-5 και στον κώδικα Gray. Η κωδικοποίηση των χαρακτήρων στο δυαδικό σύστημα γίνεται με τους αλφαριθμητικοί κώδικες. Ακολουθεί η ασφάλεια κωδικοποίησης των χαρακτήρων. Οι κώδικες διόρθωσης σφαλμάτων, με κύρια αναφορά στον κώδικα Hamming. Βασική για τους υπολογιστές είναι και η αριθμητική κινητής υποδιαστολής και το πρότυπο IEEE-754. Δίνεται η αναπαράσταση των αριθμών κινητής υποδιαστολής. απλής/διπλής ακριβείας. Η αριθμητική κινητής υποδιαστολής καθώς και η μετατροπή αριθμών μεταξύ της IEEE-754 και του δεκαδικού συστήματος με τις βασικές πράξεις αριθμών κινητής υποδιαστολής ολοκληρώνουν το κεφάλαιο.

Προαπαιτούμενη γνώση
Βασικές γνώσεις των αριθμητικών συστημάτων καθώς και οι αριθμητικές πράξεις.

2. Κωδικοποίηση των ψηφίων του δεκαδικού συστήματος

2.1. Κωδικοποίηση των ψηφίων του δεκαδικού μέσω του δυαδικού συστήματος

Όπως αναφέραμε, η λειτουργία των υπολογιστών στηρίζεται στο δυαδικό σύστημα αρίθμησης. Οι οποιεσδήποτε πράξεις πρέπει να εκτελούνται στο σύστημα αυτό. Κατά συνέπεια, τα προς εισαγωγή δεδομένα, ανεξάρτητα από τον τρόπο εγγραφής τους, πρέπει να κωδικοποιούνται αυτόματα από τον υπολογιστή στο δυαδικό σύστημα. Το ίδιο ισχύει και για την εξαγωγή τους. Τα αποτελέσματα μετά την επιτυχή, ή όχι, εκτέλεση των πράξεων πρέπει να αποκωδικοποιούνται σε σύστημα αρίθμησης κατανοητό από τον άνθρωπο. Για την κωδικοποίηση των ψηφίων των διαφόρων συστημάτων αρίθμησης μέσω του δυαδικού συστήματος απαιτούνται τόσα δυαδικά ψηφία έτσι, ώστε ο συνδυασμός αυτών να μπορεί να καλύψει τα κωδικοποιούμενα ψηφία. Στις περιπτώσεις στις οποίες το μεγαλύτερο ψηφίο του συστήματος που κωδικοποιείται είναι δύναμη του 2, τότε κατά την κωδικοποίηση έχουμε μονοσήμαντο κώδικα. Σε αντίθετη περίπτωση, όταν το μεγαλύτερο ψηφίο του προς κωδικοποίηση συστήματος δεν είναι δύναμη του 2, τότε η κωδικοποίηση είναι πλεονάζουσα, και μπορούν να δημιουργηθούν περισσότεροι του ενός τρόποι με τους οποίους μπορούμε να παραστήσουμε τα δεκαδικά ψηφία, και, συνεπώς, έχουμε περισσότερους κώδικες.

2.2. Οκταδικός κωδικοποιημένος στο δυαδικό

Τα οκτώ ψηφία του οκταδικού συστήματος αρίθμησης για την κωδικοποίησή τους απαιτούν 3 δυαδικά ψηφία. Η κωδικοποίηση είναι μονοσήμαντη, αφού $2^3 = 8$. Στον πίνακα 2.1, που ακολουθεί, δίνεται η αντιστοιχία μεταξύ οκταδικών ψηφίων και των αντίστοιχων τριάδων δυαδικών ψηφίων.

Ακολουθούν μερικά παραδείγματα για την καλύτερη κατανόηση της μετατροπής: Ο αριθμός $\mathbf{N_8 = 254}$ κωδικοποιημένος στο δυαδικό είναι, επομένως, ίσος με τον **010-101-100** δηλαδή τον **010101100**.

Οκταδικό ψηφίο	Δυαδική κωδικοποίηση	Οκταδικό ψηφίο	Δυαδική κωδικοποίηση
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

Πίνακας 2.1. *Δυαδική παράσταση των οκταδικών ψηφίων*

Εάν έχουμε έναν οκταδικό αριθμό κωδικοποιημένο στο δυαδικό σύστημα, για να βρούμε τα ψηφία του στο οκταδικό, το χωρίζουμε σε τριάδες δυαδικών ψηφιών, ξεκινώντας από το δεξιότερο ψηφίο του. π.χ. ο οκταδικός κωδικοποιημένος στο δυαδικό αριθμός **001100110**, στο οκταδικό είναι ίσος με τον **N_8 = 146**

2.3. Δεκαεξαδικός κωδικοποιημένος στο δυαδικό

Και στο δεκαεξαδικό σύστημα αρίθμησης έχουμε μονοσήμαντη αντιστοιχία των ψηφίων. Πράγματι η βάση του δυαδικού υψωμένη στην τέταρτη δύναμη δίνει τη βάση του δεκαεξαδικού, δηλαδή $2^4 = 16$. Απαιτούνται, επομένως, τέσσερα δυαδικά ψηφία για τον καθορισμό ενός δεκαεξαδικού ψηφίου. (Wakerly, 2006; Mano, & Ciletti, 2014)

Ο αριθμός **N_{16} = 359_{16}** κωδικοποιημένος στο δυαδικό γράφεται: **001101011001**.

Ο δεκαεξαδικός κωδικοποιημένος στο δυαδικό **0110010010000111** παριστάνει το δεκαεξαδικό αριθμό: **N_{16} = 6487** (ο διαχωρισμός γίνεται πάντα από δεξιά)**.**

0110	**0100**	**1000**	**0111**
↓	↓	↓	↓
6	**4**	**8**	**7**

Στον *πίνακα 2.2* που ακολουθεί αποδίδεται η αντιστοιχία των δεκαεξαδικών αριθμών με τους αντίστοιχους τετραψήφιους δυαδικούς αριθμούς.

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

Πίνακας 2.2. *Δυαδική παράσταση δεκαεξαδικού ψηφίου*

2.4. Δεκαδικός κωδικοποιημένος στο δυαδικό

Δεδομένου ότι ο αριθμός των ψηφίων του δεκαδικού συστήματος αρίθμησης δεν είναι δύναμη του 2, έχουμε πλεονασμό κωδικοποίησης με αποτέλεσμα τη δημιουργία πάρα πολλών κωδικών. Πράγματι για την κωδικοποίηση των δέκα ψηφίων απαιτούνται 4 δυαδικά ψηφία. (Φραγκάκης, 1975; Nashelsky, 1994; Κοσσίδας, 1996; Blahut, 2003; Givone, 2002; Balch, 2003; Predko, 2005; Balabanian, & Carlson, 2007; Floyd, 2013; Mano, &Ciletti, 2014; Pritchard, 2015; Rajaraman, & Adabala, 2015)

Δεκαδικό ψηφίο	**Δυαδικός Κώδικας BCD 8421**	**Κώδικας Aiken 2421**	**Κώδικας Aiken 84-2-1**	**Κώδικας Υπερβολής κατά 3 (Excess -3)**
0	0000	0000	0000	0011
1	0001	0001	0111	0100
2	0010	0010	0110	0101
3	0011	0011	0101	0110
4	0100	0100	0100	0111
5	0101	1011	1011	1000
6	0110	1100	1010	1001
7	0111	1101	1001	1010
8	1000	1110	1000	1011
9	1001	1111	1111	1100

Πίνακας 2.3α. *Αντιστοιχία των δεκαδικών ψηφίων και των κυρίων κωδίκων*

Όλοι οι κώδικες χωρίζονται σε τρείς κατηγορίες α) Στους κώδικες βαρών, που είναι και η πλειονότητα, β) στους δυαδικούς κώδικες των ψηφίων του δεκαδικού, που είναι χωρίς βάρη και γ) στο κώδικα Gray. Στους πίνακες 2.3α, 2.3β και 2.3γ παραθέτουμε τους κυριότερους κώδικες του δεκαδικού συστήματος αρίθμησης.

Στον πίνακα 2.3α εκτός από τους κώδικες με τέσσερα δυαδικά ψηφία δίνονται και άλλοι που αποτελούνται από περισσότερα από τέσσερα δυαδικά ψηφία για κάθε δεκαδικό. Οι λόγοι δημιουργίας δεκαδικών κωδίκων με περισσότερα από τέσσερα δυαδικά ψηφία σχετίζονται κυρίως με την απλοποίηση των κυκλωμάτων κωδικοποίησης- αποκωδικοποίησης. Το τελευταίο γίνεται εμφανές στις δύο τελευταίες στήλες του *πίνακα 2.3β*, όπου άξιο παρατήρησης είναι το γεγονός ότι στο δυπενταδικό κώδικα (Biquinary) κάθε δεκαδικό ψηφίο έχει δύο δυαδικές μονάδες, ενώ στο δυδεκαδικό (Bin-decimal) έχει μόνο μια μονάδα, καθώς περιλαμβάνονται όλα τα ψηφία από το 0 μέχρι και το 9.

Άλλοι κώδικες με βάρη, που μπορούμε να χρησιμοποιήσουμε για την αναπαράσταση των δεκαδικών ψηφίων, είναι ο κώδικας 7421, ο κώδικας $75\overline{3}\overline{6}$, ο οποίος χρησιμοποιεί αρνητικά βάρη, ο 5421, ο κώδικας 5211 (κατοπτρικός κώδικας- αυτοσυμπληρούμενος κώδικας), και τέλος όλοι η σειρά των κωδίκων που ονομάζεται p από q με πιο κοινό τον κώδικα 2-από-5 κ.λπ. (να σχηματίσετε τα ψηφία των ανωτέρω κωδίκων στο δυαδικό σύστημα αρίθμησης).

Αναφέρεται επικουρικά ότι οι κώδικες, των οποίων τα βάρη έχουν άθροισμα 9, ονομάζονται αυτοσυμπληρούμενοι κώδικες, και είναι οι πλέον κατάλληλοι για την εκτέλεση αριθμητικών πράξεων. (Ποιοι από τους κώδικες των *πινάκων 2.3α, β, γ, δ* είναι αυτοσυμπληρούμενοι;)

Βασικοί δυαδικοί κώδικες κωδικοποίησης αριθμών του δεκαδικού συστήματος (και συνεπώς τετραδικοί κώδικες είναι οι: 8421(BCD κώδικας), 5421, 2421 Κώδικας Aiken (αυτοσυμπληρούμενος), 4221 Κώδικας Aiken,5311, 7421(κώδικας ελαχιστοποίησης των άσσων), 74-2-1.

2.4.1. Δυαδικός κώδικας (Binary code)

Ο *δυαδικός κώδικας BCD* χρησιμοποιεί τα βάρη του δυαδικού συστήματος αρίθμησης για την κωδικοποίηση των δεκαδικών ψηφίων. Στον *πίνακα 1.12α* φαίνεται ο τρόπος κωδικοποίησης. Π.χ. ο αριθμός 23 κωδικοποιείται σαν 0010 0011. Είναι προφανές ότι κάθε ψηφίο του δεκαδικού κωδικοποιείται ξεχωριστά στο δυαδικό σύστημα (γράφουμε την αντίστοιχη ισοδύναμη τετράδα).

2.4.2. Οι κώδικες 5421 και 5311

Στον κώδικα 5421 εάν ένας αριθμός έχει περισσότερες από μια αναπαραστάσεις, χρησιμοποιούμε αυτή η οποία έχει το λιγότερο σημαντικό ψηφίο βάρους, πχ. για την τιμή 5 δε θα χρησιμοποιήσουμε την 1000 αλλά την αναπαράσταση 0101. (Pritchard, 2015)

Αντίστοιχα στον κώδικα 5311 στην αντίστοιχη περίπτωση χρησιμοποιούμε την κωδικοποίηση με τον μικρότερο αριθμό άσσων, χρησιμοποιώντας, επίσης, τα λιγότερο σημαντικά ψηφία.

Δεκαδικό ψηφίο	Κώδικας 5421	Κώδικας 5311	Κώδικας 4221	Κώδικας 7421
0	0000	0000	0000	0000
1	0001	0001	0001	0111
2	0010	0011	0010	0010
3	0011	0100	0011	0011
4	0100	0101	0110	0100
5	**0101**	**0111**	1001	0101
6	0110	1001	1100	0110
7	0111	1011	1101	1000
8	1011	1100	1110	1001
9	1100	1101	1111	1010

Πίνακας 2.3β. *Αντιστοιχία των δεκαδικών ψηφίων και κυρίων κωδίκων*

2.4.3. Κώδικας Aiken ή κώδικας 2421 ή κώδικας 8 4 -2-1

Ο *κώδικας Aiken*, όπως και ο δυαδικός, μας δίνει τη δυνατότητα να προσδιορίσουμε αν ο κωδικοποιούμενος αριθμός είναι άρτιος ή περιττός. Η πληροφορία αυτή αναδεικνύεται από την τιμή του δεξιότερου ψηφίου του. Είναι ένας κώδικας με βάρη και μέγιστο αριθμό τον μέγιστο δεκαδικό. Παρέχει τη δυνατότητα εύκολης συμπλήρωσης,, σε περίπτωση που χρειαστεί να εκτελεστούν πράξεις αφαίρεσης, και διατηρεί την ιδιότητα των κωδικών που δίνουν βάρος στα δυαδικά ψηφία, από τα οποία δημιουργούνται τα ψηφία του κώδικα. Παρατηρήστε στον κώδικα αυτό ότι μετά την αναπαράσταση του αριθμού 4 του δεκαδικού συστήματος (πέμπτο στοιχείο του κώδικα) τα υπόλοιπα στοιχεία του προκύπτουν ως συμπληρωματικά κατοπτρικά των προηγουμένων. (Ψηφία 5 έως και 9).

2.4.4. Κώδικας Υπερβολής κατά τρία

Ο κώδικας υπερβολής κατά τρία (Excess three, XS3), όπως και ο προηγούμενος, διαθέτει ευκολία συμπλήρωσης και αποτελεί μια παραλλαγή του Δυαδικού κώδικα. Ονομάζεται και κώδικας Stibitz. Προκύπτει από το δυαδικό κώδικα, αν προσθέτουμε το τρία στη δυαδική μορφή του καθενός κωδικοποιημένου ψηφίου. Και αυτός ο κώδικας είναι κατοπτρικός συμπληρωματικός κώδικας. Τέλος αναφέρουμε ότι είναι ένας κώδικας ο οποίος δεν έχει βάρη.

Είναι προφανές ότι, αν θέλουμε μπορούμε να «φτιάξουμε» οποιονδήποτε κώδικα υπερβολής, αρκεί η τελευταία δυαδική αναπαράσταση του κώδικα να μην υπερβαίνει τον αριθμό 1111. (Αυτοσυμπληρούμενοι κώδικες είναι οι XS3, 5211, 4221, 84-2-1). (Ποιος είναι ο μέγιστος δυνατός κώδικας υπερβολής για την αναπαράσταση των ψηφίων του δεκαδικού συστήματος;).

2.4.5. Δυπενταδικός και Δυδεκαδικός κώδικας

Ο Δυπενταδικός κώδικας (Biquinary code) κωδικοποιεί κάθε δεκαδικό ψηφίο με τη βοήθεια 7 bit χωρισμένων σε δύο ομάδες. Η πρώτη περιλαμβάνει 2 bit και η δεύτερη 5. Κάθε κωδικοποιούμενο ψηφίο περιλαμβάνει πάντοτε δύο δυαδικές μονάδες, μία σε κάθε τμήμα. Το μεγάλο πλεονέκτημα του Δυπενταδικού κώδικα είναι ο αυτοέλεγχος όσο αφορά στην ασφάλεια μεταβίβασης πληροφοριών. Χρησιμοποιείται συνήθως για την ανίχνευση λαθών και η ύπαρξη λιγότερων ή περισσότερων άσσων των δύο προσδιορίζει την ύπαρξη λάθους.

Ο Δυδεκαδικός κώδικας (Bindecimal code) περιλαμβάνει 10 bit για την κωδικοποίηση κάθε ψηφίου. Τα πλεονεκτήματά του, όπως και του δυπενταδικού, είναι η απλότητα κωδικοποίησης και η ασφάλεια μεταφοράς πληροφοριών. Μειονεκτήματα των κωδικών αυτών είναι το μεγάλο μήκος του κώδικα και, κατά συνέπεια, η μικρότερη ταχύτητα μετάδοσης πληροφοριών. (Πίνακας 2.3γ).

Δεκαδικό ψηφίο	**Κώδικας Gray**	**Κώδικας Biquinary 50,43210**	**Κώδικας Bindecimal 9876543210**
0	0000	01,00001	0000000001
1	0001	01,00010	0000000010
2	0011	01,00100	0000000100
3	0010	01,01000	0000001000
4	0110	01,10000	0000010000
5	0111	10,00001	0000100000
6	0101	10,00010	0001000000
7	0100	10,00100	0010000000
8	1100	10,01000	0100000000
9	1101	10,10000	1000000000

Πίνακας 2.3γ. *Αντιστοιχία κωδίκων στο δεκαδικό σύστημα*

Δεκαδικό ψηφίο	**Κώδικας 74210**	**Κώδικας 5211**	**Κώδικας $753\overline{6}$**
0	11000	0000	0000
1	00011	0001	1001
2	00101	0100	0111
3	00110	0101	0010
4	01001	0111	1011
5	01010	1000	0100
6	01100	1001	1101
7	10001	1100	1000
8	10010	1101	0110
9	10100	1111	1111

Πίνακας 2.3δ. *Κώδικες αντιστοίχησης ψηφίων του δεκαδικού συστήματος*

2.4.6. *Κώδικας 2-από-5*

Ο *κώδικας 2 -από -5 (2-out of-5)* Είναι και αυτός, όπως και ο κώδικας υπερβολής κατά 3, ένας κώδικας ο οποίος δεν έχει βάρη, έχει όμως την ιδιότητα, όπως και ο προηγούμενος κώδικας, να περιέχει μόνο δύο άσσους σε κάθε κωδικοποιημένη του λέξη.

Παρόλο που, όπως αναφέραμε, δεν έχει βάρη μπορούμε να πούμε ότι είναι κώδικας με βάρη 74210 εκτός της τιμής που αντιστοιχεί στο μηδέν (11000). Χρησιμοποιείται συνήθως για την ανίχνευση ενός λάθους και η ύπαρξη άσσων λιγότερων ή περισσότερων των δύο προσδιορίζει την ύπαρξη λάθους. (πίνακας 2.3δ).

2.4.7. *Κώδικας Gray (Κυκλικός κώδικας)*

Ο κώδικας Gray βρίσκει μεγάλη χρησιμότητα στις διατάξεις εισόδου- εξόδου πληροφοριών, αφού βελτιώνει τη δυνατότητα ανάγνωσης- εκτύπωσης πληροφοριών και απλουστεύει την κατασκευή των αντιστοίχων διατάξεων. Βασική ιδιότητα του κώδικα αυτού είναι ότι, μεταξύ διαδοχικών αριθμών του δεκαδικού συστήματος αρίθμησης, ο αντίστοιχος κώδικας Gray διαφέρει μόνο κατά ένα δυαδικό ψηφίο, όπως φαίνεται και στον *πίνακα 2.3γ*. (Nashelsky, 1994; Predko, 2005; Balabanian, & Carlson, 2007)

Για την κωδικοποίηση αριθμών με περισσότερα από ένα δεκαδικά ψηφία δεν κωδικοποιείται κάθε ψηφίο χωριστά, για να τοποθετηθούν κατόπιν οι αντίστοιχες τετράδες η μία δίπλα στην άλλη, αλλά κωδικοποιείται όλος ο αριθμός μαζί, αφού υπολογισθεί η δυαδική τιμή του.

Παρακάτω, με τη βοήθεια μερικών παραδειγμάτων, θα εξηγήσουμε τον τρόπο κωδικοποίησης του συγκεκριμένου κώδικα. Έστω ο αριθμός $\mathbf{N_{10} = 135}$ για τον οποίο θέλουμε να βρούμε τον αντίστοιχό του σε κώδικα Gray. Έχουμε ότι:

$\mathbf{N_{10} = 135} \longrightarrow$ **10000111** **Δυαδικός**

11000100 **Gray**

Αρχικά βρίσκουμε τον αντίστοιχο δυαδικό του δεκαδικού και στη συνέχεια εφαρμόζουμε τον παρακάτω κανόνα:

Αρχίζουμε από το ψηφίο μεγαλύτερης τάξης του δυαδικού και το συγκρίνουμε με το αμέσως προηγούμενο (για το πρώτο ψηφίο το προηγούμενο είναι πάντοτε το μηδέν). Εάν κατά τη σύγκριση έχουμε τα ίδια ψηφία, θέτουμε «0». Εάν έχουμε διαφορετικά θέτουμε «1» κ.ο.κ., έως ότου εξαντλήσουμε όλα τα δυαδικά ψηφία. Η μετατροπή από τον δυαδικό αριθμό στον αντίστοιχο του κώδικα Gray μπορεί να φανεί και από το παρακάτω παράδειγμα:

Δυαδικός **0100101011001**

Gray **0110111110101**

Ένας δεύτερος τρόπος που μπορούμε να χρησιμοποιήσουμε είναι η πρόσθεση των ψηφίων του αρχικού αριθμού ανά δύο (στη θέση που θέλουμε να τοποθετήσουμε το αποτέλεσμα και στην αμέσως προηγούμενη). (Να ελέγξετε το αποτέλεσμα αυτής της μεθόδου).

Για τη μετατροπή ενός αριθμού από μορφή Gray σε δυαδική μορφή, η διαδικασία είναι περισσότερο πολύπλοκη.

Δεκαδικός	Gray	Δεκαδικός	Gray
0	0000	10	1111
1	0001	11	1110
2	0011	12	1010
3	0010	13	1011
4	0110	14	1001
5	0111	15	1000
6	0101	16	11000
7	0100	17	11001
8	1100	18	11011
9	1101	19	11010

Πίνακας 2.4. *Κώδικας Gray*

Σ' αυτήν την περίπτωση αρχίζουμε από τα αριστερά του αριθμού και επαναλαμβάνουμε τα δυαδικά ψηφία, μέχρις ότου βρούμε ένα ψηφίο ίσο με «1». Μετά την πρώτη δυαδική μονάδα του Gray, εάν ακολουθούν μηδενικά, ένα ή περισσότερα, επαναλαμβάνουμε στο δυαδικό αριθμό την τιμή του προηγουμένου δυαδικού του ψηφίου. Στη συνέχεια, εάν στον Gray έχουμε «1», βάζουμε στο δυαδικό την αντίθετη τιμή από αυτή που είχε το προηγούμενο δυαδικό του ψηφίο, ενώ, εάν έχουμε «0», επαναλαμβάνουμε την τιμή που είχε το προηγούμενο δυαδικό του ψηφίο κ.ο.κ. Στον *πίνακα 2.4* δίδονται οι αριθμοί από το 0 έως το 19 μαζί με τους ισοδύναμούς τους σε κώδικα Gray. Ακολουθούν δύο παραδείγματα μετατροπής από τον κώδικα Gray στο δυαδικό, όπου εφαρμόζουμε τον παραπάνω αλγόριθμο.

Gray	**010100001001**	**Gray**	**11001010100101**
Δυαδικός	**011000001110**	**Δυαδικός**	**10001100111001**

2.5. Αλφαριθμητικοί κώδικες–Κωδικοποίηση χαρακτήρων μέσω του δυαδικού συστήματος αρίθμησης

Με τον όρο «χαρακτήρα» εννοούμε αριθμούς, γράμματα της αλφαβήτου, σημεία στίξης και γενικά κάθε σύμβολο που χρησιμοποιείται κατά την εγγραφή ενός προγράμματος ή την εκτύπωση ενός αποτελέσματος.

Εάν λάβουμε υπόψη μας ότι τα ψηφία του δεκαδικού συστήματος είναι 10 και τα γράμματα του Λατινικού αλφαβήτου 26, απαιτούνται τουλάχιστον 6 δυαδικά ψηφία, για να κωδικοποιηθούν. Με 6 δυαδικά ψηφία έχουμε όμως 64 συνδυασμούς και, κατά συνέπεια, μας περισσεύουν 28 για τα ειδικά σύμβολα και τα σημεία στίξης.

Για την κωδικοποίηση, επομένως όλων των χαρακτήρων, απαιτούνται τουλάχιστον 6 δυαδικά ψηφία. Με βάση το γεγονός αυτό έχουν αναπτυχθεί αρκετοί κώδικες, οι κυριότεροι των οποίων είναι ο Αλφαβητικός Εξαψήφιος Κώδικας που παρατίθεται στον *πίνακα 1.1* του Παραρτήματος, ο ASCII (American Standards Code for Information Interchange) που παρατίθεται στους *πίνακες* του παραρτήματος και ο EBCDIC (Extended Binary Coded Decimal Interchange Code) που παρατίθεται στο παράρτημα.

Ο κώδικας ASCII έχει 8 δυαδικά ψηφία και χρησιμοποιείται διεθνώς σε εφαρμογές με μικροεπεξεργαστές. Οι πρώτοι 128 χαρακτήρες του περιλαμβάνουν τα αλφαβητικά και αριθμητικά σύμβολα, ενώ οι δεύτεροι 128 περιλαμβάνουν, κατά κανόνα, σύμβολα που έχουν σχέση με την τοπική γλώσσα του κράτους που χρησιμοποιείται. Για το λόγο αυτό τον έχουμε χωρίσει σε δύο μέρη. Το πρώτο από αυτά (bit7=0) είναι αυτό με το διεθνές σετ συμβόλων, ενώ το δεύτερο (bit7=1) περιλαμβάνει τους χαρακτήρες της Ελληνικής γλώσσας. (Wakerly, 2006; Balabanian, & Carlson, 2007; Tocci, et al., 2010; Mano, &Ciletti, 2014)

Οι πρώτοι 32 χαρακτήρες μεταφέρουν μη εκτυπώσιμα σύμβολα, αυτά που είναι περισσότερο γνωστά σαν χαρακτήρες ελέγχου. Οι χαρακτήρες αυτοί έχουν ειδικές σημασίες και αποτείνονται προς τη μηχανή. Μιλώντας πιο συγκεκριμένα, μπορούμε να πούμε πως οι χαρακτήρες αυτοί αναγκάζουν, π.χ. την κεφαλή του εκτυπωτή να μεταφερθεί στο αριστερό μέρος της σελίδας και να αλλάξει γραμμή, να προχωρήσει στην αρχή της επόμενης σελίδας κ.λπ. Ακόμα ένας από αυτούς τους χαρακτήρες προκαλεί την εκπομπή ενός σύντομου ακουστικού σήματος, χρήσιμου σαν σήματος ειδοποίησης προς το χειριστή της μηχανής.

Στον πίνακα αυτόν γίνεται εμφανής η παρουσία των Ελληνικών συμβόλων, τα οποία, όπως και στην περίπτωση του διεθνούς κώδικα, εμφανίζονται με διαφορετικό κωδικό ανάλογα με τον αν είναι πεζά ή κεφαλαία. Αξιοσημείωτο είναι ότι τα φωνήεντα εμφανίζονται και αυτά με δύο κωδικούς ανάλογα με το αν είναι τονισμένα ή όχι.

Ο κώδικας EBCDIC, αποτελεί μία τροποποίηση του ASCII, και χρησιμοποιείται ευρέως από την IBM. Δε χρησιμοποιείται σε εφαρμογές με μικροεπεξεργαστές.

2.5.1. Χρήση της ισοτιμίας στην ασφαλή κωδικοποίηση των χαρακτήρων

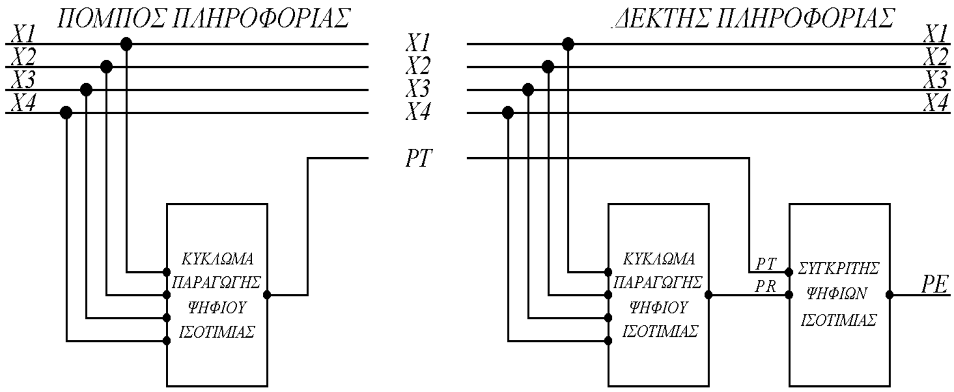
Κατά τη μεταφορά κωδικοποιημένων χαρακτήρων τόσο μέσα σε ένα υπολογιστικό σύστημα όσο και έξω από αυτό θα πρέπει να είμαστε βέβαιοι για το κατά πόσο οι μεταφερόμενες πληροφορίες είναι σωστές, δηλαδή για το κατά πόσο, στη διάρκεια της μεταφοράς των δυαδικών ψηφίων «0» ή «1», δε συνέβησαν μεταβολές ικανές να αλλοιώσουν τους κωδικοποιούμενους χαρακτήρες.

Για να είναι εύκολη η αναγνώριση ενός πιθανού σφάλματος, έχει προστεθεί στους ήδη κωδικοποιημένους χαρακτήρες ένα επιπλέον δυαδικό ψηφίο, το λεγόμενο ***ψηφίο ισοτιμίας*** (parity bit), με το οποίο αναγνωρίζεται εύκολα η παρουσία του πιθανού σφάλματος. Το ψηφίο ισοτιμίας τίθεται στην αρχή ή στο τέλος του κωδικοποιημένου χαρακτήρα και η τιμή του («0» ή «1») καθορίζεται από τον αριθμό των δυαδικών μονάδων του κωδικοποιημένου χαρακτήρα και από τον τρόπο ισοτιμίας που θέλουμε να εφαρμόσουμε (περιττή ή άρτια ισοτιμία). Εάν, για παράδειγμα, ο υπό μεταφορά κωδικοποιημένος χαρακτήρας έχει τέσσερις μονάδες και έχουμε περιττή ισοτιμία, το parity bit που προστίθεται έχει τιμή «1» έτσι, ώστε το σύνολο των μονάδων να είναι περιττό. Αντίθετα, εάν ο υπό μεταφορά κωδικοποιημένος χαρακτήρας έχει τέσσερις μονάδες και έχουμε άρτια ισοτιμία, τότε το parity bit που προστίθεται έχει τιμή «0» έτσι, ώστε το σύνολο των μονάδων να παραμείνει άρτιο. Είναι λογικό ότι τυχόν λάθος ανακαλύπτεται εύκολα, γιατί θα έχει σαν αποτέλεσμα την αλλαγή του αριθμού των μονάδων από περιττό σε άρτιο αριθμό. Στον πίνακα *2.5* δίνονται οι αριθμοί από 1 έως 10 κωδικοποιημένοι στο δυαδικό με ψηφία άρτιας και περιττής ισοτιμίας.

Δεκαδικός	**Δυαδικός με άρτια ισοτιμία**	**Δυαδικός με περιττή ισοτιμία**
0	0 0000	1 0000
1	1 0001	0 0001
2	1 0010	0 0010
3	0 0011	1 0011
4	1 0100	0 0100
5	0 0101	1 0101
6	0 0110	1 0110
7	1 0111	0 0111
8	1 1000	0 1000
9	0 1001	1 1001

Πίνακας 2.5. *Παράσταση δυαδικών αριθμών με ψηφία ισοτιμίας*

Στο *Σχήμα 2.1* αποδίδεται ένα τυπικό διάγραμμα που δείχνει τον τρόπο ελέγχου της ύπαρξης πιθανών σφαλμάτων σε μια μεταδιδόμενη λέξη.



Σχήμα 2.1. *Τυπικά διαγράμματα πομπού και δέκτη πληροφορίας 4 bit, κωδικοποιημένης σύμφωνα με τους κανόνες περιττής ή άρτιας ισοτιμίας*

Τα ψηφία $X_1X_2X_3X_4$ της λέξης μεταδίδονται μαζί με ένα ακόμα ψηφίο ισοτιμίας, το PT (Parity Transmitter), το οποίο παράγεται από την επεξεργασία των αρχικών ψηφίων. Στη μεριά του δέκτη της πληροφορίας παράγεται πάλι με τον ίδιο τρόπο το ψηφίο ισοτιμίας PR (Parity Receiver), το οποίο συγκρίνεται με το PT. Αν η βαθμίδα της σύγκρισης βρει τα ψηφία αυτά διαφορετικά, ενεργοποιεί το σήμα PE (Parity Error) σαν ενδεικτικό της λανθασμένης λήψης.

Αξίζει να σημειώσουμε ότι η παραπάνω μέθοδος ανίχνευσης σφαλμάτων είναι σε θέση να σηματοδοτήσει εσφαλμένη λήψη, μόνο όταν το σύνολο των ψηφίων που έχουν υποστεί αλλοίωση εκφράζεται από έναν περιττό αριθμό.

2.5.2. Κώδικες διόρθωσης σφαλμάτων

Η κωδικοποίηση για τη διόρθωση σφαλμάτων είναι πιο ενδιαφέρουσα από την κωδικοποίηση για την ανίχνευση σφαλμάτων. Ο στόχος της διόρθωσης σφαλμάτων είναι η ανίχνευση και ο εντοπισμός σφαλμάτων στη μετάδοση της πληροφορίας. Όταν γίνει ο εντοπισμός η διόρθωση είναι τετριμμένη διαδικασία, καθώς απλά γίνεται αντιστροφή του ψηφίου.

Στις επικοινωνίες συχνά προτιμάται η ανίχνευση σφάλματος και η επανεκπομπή. Στις τηλεπικοινωνίες εντούτοις, λόγω της καθυστέρησης της μετάδοσης της πληροφορίας, αποστέλλονται πολλά πακέτα από τον εκπομπό, πριν φθάσει η πληροφορία για επανεκπομπή. Αυτό κάνει το χειρισμό των δεδομένων αρκετά πολύπλοκο. Επίσης η μετάδοση σε πραγματικό χρόνο αποκλείει την επανεκπομπή. Είναι, συνεπώς, απαραίτητο να μην υπάρχει σφάλμα. Έτσι απαιτείται ένα μεγαλύτερο εύρος ζώνης το κόστος του οποίου δεν είναι υψηλό για τα πλεονάζοντα ψηφία ελέγχου (redundant check-bits). Υπάρχουν δύο κατηγορίες κωδίκων: Οι κώδικες Μπλοκ (π.χ. κώδικας Hamming) και οι συγκεραστικοί κώδικες (convolutional codes). (Nelson, et al., 1995; Klove, & Korzhik, 1995; Blahut, 2003; Lee, 2006; Wakerly, 2006; Rajaraman, & Adabala, 2015)

Κυκλικός Πλεονάζων Κώδικας Ελέγχου (CRC)

Οι *Κυκλικοί Πλεονάζοντες Κώδικες Ελέγχου* **(**cyclic redundancy check) περιλαμβάνουν τη διαίρεση της ακολουθίας των bytes των δεδομένων με μια σταθερά. Κάθε υπόλοιπο από αυτή τη διαίρεση είναι γραμμένο σε μορφή 2 CRC bytes, ή χαρακτήρων. Όταν τα δεδομένα διαβάζονται, αυτά τα 2 bytes αφαιρούνται από την ακολουθία των δεδομένων. Το αποτέλεσμα διαιρείται με την αρχική σταθερά. Αυτή η διαίρεση δίνει μηδενικό υπόλοιπο, αν τα δεδομένα δεν περιέχουν λάθη. Η μέθοδος CRC χρησιμοποιείται συνήθως για έλεγχο λαθών του δίσκου αποθήκευσης και των δικτύων δεδομένων. Σε ένα οδηγό δίσκου, κάθε μπλοκ δεδομένων (τυπικά 512 bytes) προστατεύεται από ένα CRC κώδικα, έτσι, ώστε σφάλμα σε ένα μπλοκ να μπορεί να ανιχνευθεί και, σε ορισμένους οδηγούς, να μπορεί να διορθωθεί. Σε ένα δίκτυο δεδομένων κάθε πακέτο δεδομένων τελειώνει με ένα CRC byte. Οι κώδικες CRC επιλέγονται και για τις δύο εφαρμογές λόγω των ιδιοτήτων ανίχνευσης καταιγιστικών σφαλμάτων. Επιπλέον, από το να μπορούν να ανιχνεύσουν ένα σφάλμα, μπορούν να ανιχνεύσουν σφάλματα πολλαπλών bits, τα οποία είναι ομαδοποιημένα στο μπλοκ του δίσκου ή του πακέτου.

Κώδικες δύο διαστάσεων

Όταν μερικές δυαδικές λέξεις μεταδίδονται ή λαμβάνονται διαδοχικά, η προκύπτουσα συλλογή των bits μπορεί να θεωρηθεί ως ένα μπλοκ δεδομένων, το οποίο έχει σειρές και στήλες. Για παράδειγμα, έστω τέσσερις διαδοχικές οκτάμπιτες λέξεις δημιουργούν μια μορφή ενός μπλοκ 4x8. Έτσι μπορούμε να δημιουργήσουμε τα bits ισοτιμίας και ανά γραμμή και ανά στήλη. Αυτός ο κώδικας αναφέρεται σαν δισδιάστατος κώδικας ή κώδικας του μπλόκ ισοτιμίας ή κώδικας γινομένου. Είναι δυνατόν να ανιχνευθεί οποιοδήποτε απλό σφάλμα ανά λέξη δεδομένων ή διπλό σφάλμα ανά λέξη. (Σχήμα 2.2α.)

1	0	1	1	0	1	0	1	**1**
0	1	0	1	0	1	0	1	**0**
1	1	0	0	1	1	0	1	**1**
1	0	0	1	1	0	1	1	**1**
1	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**

Σχήμα 2.2α. *Διόρθωση απλού σφάλματος*

Από τον παραπάνω πίνακα διαπιστώνουμε σφάλμα στη δεύτερη γραμμή και στην τρίτη κολώνα που δείχνει ότι το τρίτο bit της δεύτερης στήλης είναι εσφαλμένο, οπότε το διορθώνουμε σε 1.

1	0	0	0	0	1	0	1	**1**
0	1	0	1	0	1	0	1	**0**
1	1	0	0	1	1	0	1	**1**
1	0	0	1	1	0	1	1	**1**
1	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**
1	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**

Σχήμα 2.2β. *Αδυναμία διόρθωσης διπλού σφάλματος*

Στο σχήμα 2.2β απεικονίζεται η περίπτωση, όπου έχουμε δύο σφάλματα, π.χ. στην πρώτη γραμμή. Σε αυτή την περίπτωση το ψηφίο ισοτιμίας της πρώτης γραμμής δεν αναγνωρίζει κάποιο λάθος. Παράλληλα τα ψηφία ισοτιμίας των στηλών 3 και 4 αναγνωρίζουν λάθος. Σε αυτή την περίπτωση δεν μπορούμε να διορθώσουμε κάποιο λάθος, καθώς δε γνωρίζουμε τη γραμμή από την οποία προέρχεται το σφάλμα.

Κώδικες Αθροίσματος Ελέγχου (Check sum codes)
Οι κώδικες αθροίσματος ελέγχου χρησιμοποιούνται, για να ανιχνεύσουν λάθος σε μπλοκ δεδομένων. Το άθροισμα ελέγχου δημιουργείται από την άθροιση όλων των bytes των δεδομένων σε ένα μπλόκ, ***αγνοώντας οποιοδήποτε κρατούμενο***. Είναι το αριθμητικό άθροισμα όλων των bytes δεδομένων. Το νούμερο που προκύπτει αποθηκεύεται μετά το μπλοκ δεδομένων. Όταν τα δεδομένα διαβάζονται, αυτό προστίθεται εκ νέου, και το άθροισμα συγκρίνεται με την αποθηκευμένη τιμή του ελέγχου αθροίσματος. Αν δεν υπάρχει σφάλμα, τότε τα δύο αθροίσματα είναι ταυτόσημα. Σε κάθε άλλη περίπτωση, στην περίπτωση λάθους, τα υπό μετάδοση δεδομένα αναμεταδίδονται.

Παράδειγμα: Να υπολογίσετε το άθροισμα ελέγχου των χαρακτήρων

1	**1**	**0**	**1**	**1**	**0**	**0**	**1**
0	**0**	**1**	**1**	**1**	**0**	**1**	**0**
1	1	1	0	0	0	1	1

Αθροίζοντας προκύπτει ο αριθμός 11100011 (αντιστοιχεί στον δεκαεξαδικό E3).

Κώδικες m από n
Σε αυτούς τους κώδικες η κάθε κωδική λέξη έχει m άσσους και όλα τα υπόλοιπα είναι μηδέν. Αυτός ο κωδικός έχει ελάχιστη απόσταση το 2, γιατί οποιαδήποτε αλλαγή σε ένα bit αλλάζει τον αριθμό των άσσων και, κατά συνέπεια, προκύπτει μια νέα λέξη που δεν είναι κωδική λέξη. Αυτός ο κώδικας είναι χρήσιμος για την ανίχνευση πολλαπλών σφαλμάτων μονής κατεύθυνσης. Σε αυτή την περίπτωση όλα τα σφάλματα αλλάζουν με την ίδια κατεύθυνση (τα 0 σε 1 και το 1 σε 0). Ως εκ τούτου, ο κώδικας χρησιμοποιείται για την ανίχνευση σφαλμάτων σε συστήματα στα οποία ο κυρίαρχος μηχανισμός σφαλμάτων τείνει να αλλάξει όλα τα bits προς την ίδια διεύθυνση.

2.5.3. Κώδικας Hamming
Δημιουργήθηκε το 1949 από τον Richard Hamming. Είναι ένας κώδικας ο οποίος μπορεί να διορθώσει ένα απλό λάθος και μπορεί να ανιχνεύσει οποιοδήποτε διπλό λάθος (δηλ. δύο απλά λάθη). Προσδιορίζεται με δύο αριθμούς N, K και αναφερόμαστε σε αυτόν ως *κώδικα Hamming* (N,K). Οι *κώδικες Hamming* χρησιμοποιούν αριθμητική modulo-2. Η πρόσθεση σε αριθμητική modulo-2 είναι η λογική λειτουργία της πύλης exclusive OR (XOR) ο πίνακας της οποίας παρατίθεται τον *πίνακα 2.6*. (*Shanmugam, 1979; Nashelsky, 1994; Klove, & Korzhik, 1995; Nelson, et al., 1995; Κοσσίδας, 1996; Holdsworth, & Woods, 2002; Givone, 2002; Blahut, 2003; Wakerly, 2006; Balabanian, & Carlson, 2007; Tervo, 2014; Tocci, et al., 2010*)

Ο κώδικας Hamming χρησιμοποιείται στη μνήμη RAM του υπολογιστή, και είναι μια καλή επιλογή για τυχαία εμφανιζόμενα σφάλματα. Στην περίπτωση που έχουμε καταιγισμό λαθών, χρησιμοποιούνται άλλοι κώδικες. Η απλότητα του κώδικα Hamming είναι αυτή που τον κάνει εκπαιδευτικά χρήσιμο.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Πίνακας 2.6. *Πίνακας αλήθειας της πύλης XOR*

Ο κώδικας χρησιμοποιεί επιπλέον ψηφία για τον έλεγχο σφαλμάτων και εκτελεί τους ελέγχους με ειδικές εξισώσεις ελέγχων. Η εξίσωση ελέγχου της ισοτιμίας μιας ακολουθίας δυαδικών ψηφίων απλά προσθέτει τα ψηφία της ακολουθίας και ορίζει ότι το άθροισμα είναι άρτιο (για άρτια ισοτιμία) ή περιττό (για περιττή ισοτιμία). Εναλλακτικά λέμε ότι το άθροισμα είναι MOD 2, ή ότι το άθροισμα λαμβάνεται από την πράξη MOD 2 επί των ακεραίων. Η περαιτέρω θεωρητική ανάλυση του κώδικα ξεφεύγει από τα όρια του μαθήματος.

Ο κώδικας Hamming διαφέρει από όλους τους άλλους κώδικες αναγνώρισης σφαλμάτων, επειδή είναι σε θέση όχι μόνο να παρέχει ένδειξη για τη παρουσία ενός εσφαλμένου δυαδικού ψηφίου αλλά και να προσδιορίζει με ακρίβεια τη θέση του μέσα στη μεταδιδόμενη λέξη. Για να πετύχει τον ακριβή προσδιορισμό, επαυξάνει το σύνολο των ψηφίων m της προς μετάδοση λέξης, με ένα συγκεκριμένο αριθμό ψηφίων ισοτιμίας P_i που το πλήθος τους είναι ίσο με k.

Το μέγεθος του κώδικα Hamming

Ο κώδικας Hamming μπορεί να εξυπηρετήσει οποιονδήποτε αριθμό ψηφίων δεδομένων, αλλά είναι ενδιαφέρον να παραθέσουμε το μέγιστο αριθμό των απαιτούμενων ψηφίων ελέγχου. Ο *πίνακας 2.7,* που ακολουθεί, περιλαμβάνει το συνολικό αριθμό ψηφίων ελέγχου έτσι, ώστε να δίνει το πλήθος των ψηφίων του κώδικα Hamming, ο οποίος περιλαμβάνει και την ανίχνευση διπλού σφάλματος, καθώς επίσης, και την απόδοση του κώδικα Hamming, η οποία προκύπτει από τη διαίρεση του αριθμού των ψηφίων της λέξεως προς την ποσότητα (ολικό μέγεθος -1).

Έτσι για 64 bits ή 8 bytes, απαιτούνται 7 bytes δεδομένων (συν 1 bit) και χρησιμοποιείται 1 byte για τον έλεγχο των ψηφίων (πραγματικά μόνο 7 bits). Συνεπώς σ' ένα σύστημα που είναι επιρρεπές στα σφάλματα απαιτείται μόνο 1 επιπλέον byte στα συνολικά 8, ήτοι 12.5% επιπλέον για την ανίχνευση και διόρθωση σφαλμάτων).

Η χρήση του κώδικα αυτού αφορά κυρίως στη μετάδοση πληροφορίας από έναν υπολογιστή-πομπό σ' έναν άλλο, στις περιπτώσεις εκείνες που ο υπολογιστής-δέκτης δεν είναι σε θέση να ζητήσει επανάληψη της πληροφορίας που έχει αποσταλεί, αλλά οφείλει μόνος του να τη διορθώσει. Ακόμα χρησιμοποιείται κατά την εγγραφή/ανάγνωση δεδομένων σε μνήμες RAM.

Απαιτούμενα Ψηφία ελέγχου	**Μέγιστος αριθμός Ψηφίων λέξεως**	**Ολικό Μέγεθος**	**Απόδοση (E)**
3	1	4	0,33
4	4	8	0,57
5	11	16	0,73
6	26	32	0,84
7	**57**	**64**	**0,90**
8	120	128	0,94

Πίνακας 2.7. *Απόδοση του κώδικα Hamming*

Συνδυασμός ψηφίων C_i	**Εσφαλμένο ψηφίο**
000	κανένα
001	1^o
010	2^o
011	3^o
100	4^o
101	5^o
110	6^o
111	7^o

Πίνακας 2.8. *Θέση του εσφαλμένου ψηφίου στη μεταδιδόμενη λέξη*

Κατά την εκπομπή της πληροφορίας, συγκεκριμένες ομάδες δυαδικών ψηφίων που αποτελούν την προς μετάδοση λέξη, οδηγούνται σε κυκλώματα παραγωγής ψηφίων ισοτιμίας. Τα παραγόμενα ψηφία αποστέλλονται, κατόπιν, μαζί με τα χρήσιμα, στον υπολογιστή - δέκτη. Ο δέκτης, έχοντας λάβει όλα τα ψηφία, παράγει και αυτός ψηφία ισοτιμίας, συγκρίνοντας τα ψηφία των ίδιων ομάδων που ελήφθησαν υπόψη κατά την εκπομπή. Κατόπιν συγκρίνει τα ψηφία ισοτιμίας που ο ίδιος παρήγαγε με εκείνα που ήδη έχει λάβει. Αν το αποτέλεσμα της σύγκρισης οδηγήσει σε μια ακολουθία δυαδικών ψηφίων που αποτελείται από μηδενικά, τότε η λήψη έχει γίνει με επιτυχία. Σε αντίθετη περίπτωση, ο δυαδικός αριθμός που εκφράζει η ακολουθία είναι ενδεικτικός της θέσης του ψηφίου που έχει αλλοιωθεί. Στον *πίνακα 2.8* φαίνεται η σημασία των ψηφίων C_i, που αποτελούν τα ψηφία εξόδου του συγκριτή. (*Απόδοση του κώδικα Hamming, 2015*)

Στο *Σχήμα 2.3* φαίνονται, σε απλοποιημένη μορφή, τα διαγράμματα ενός πομπού πληροφορίας κωδικοποιημένης σύμφωνα με τον κώδικα Hamming και ενός δέκτη, που την αποκωδικοποιεί και τη διορθώνει. Τα εισερχόμενα ψηφία οδηγούνται κατά ομάδες στα κυκλώματα παραγωγής ψηφίων ισοτιμίας, των οποίων οι έξοδοι αναδεικνύουν τα ομώνυμα bit. Η επαυξημένη σε πλήθος ψηφίων πληροφορία φθάνει στο δέκτη, όπου οι έξοδοι C_i του συγκριτή ψηφίων ισοτιμίας διορθώνουν, εφόσον χρειάζεται, τη δυαδική πληροφορία.

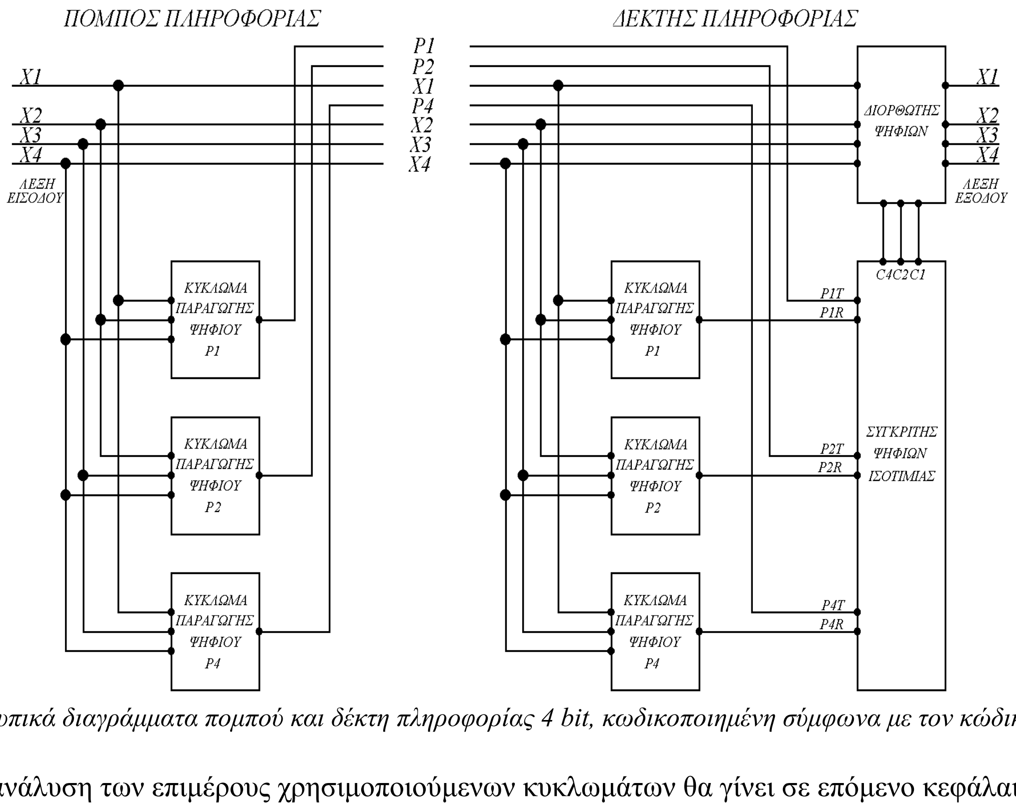
Επειδή το σύνολο k των ψηφίων ισοτιμίας P_i πρέπει να είναι σε θέση να προσδιορίζει με ακρίβεια τη θέση του λανθασμένου ψηφίου μέσα στη μεταδιδόμενη λέξη, που τώρα πια θα έχει μήκος m+k, θα πρέπει οι συνδυασμοί των k bit να σχηματίζουν αριθμούς που η τιμή τους θα μπορεί να προσδιορίζει με ακρίβεια τη θέση του λανθασμένου ψηφίου. Αν μάλιστα λάβουμε υπόψη πως ο συνδυασμός των k ψηφίων ισοτιμίας, που είναι ίσος με το 0, υποδηλώνει μια λέξη που μεταδόθηκε σωστά, τότε μπορούμε εύκολα να καταλήξουμε στη σχέση: $\mathbf{2^k - 1 \geq m + k}$.

Τροποποιώντας τη σχέση αυτή, λαμβάνουμε την τελική που μας δίνει το πλήθος των k ψηφίων ισοτιμίας P_i που συνοδεύουν μια μεταδιδόμενη λέξη. Θα έχουμε επομένως: $\mathbf{2^k \geq m + k + 1}$

m	k (ελάχιστο)
1	2
2 - 4	3
5 - 11	4
12 - 26	5

Πίνακας 2.9. *Ελάχιστα ψηφία ισοτιμίας για λέξεις μήκους m*

Στον παραπάνω *πίνακα 2.9* σημειώνεται ο ελάχιστος αριθμός των ψηφίων ισοτιμίας k, για μεταδιδόμενες λέξεις μήκους m. (εύκολα προκύπτει από τον γενικό *πίνακα 2.7*).



Σχήμα 2.3. *Τυπικά διαγράμματα πομπού και δέκτη πληροφορίας 4 bit, κωδικοποιημένη σύμφωνα με τον κώδικα Hamming*

(Η ανάλυση των επιμέρους χρησιμοποιούμενων κυκλωμάτων θα γίνει σε επόμενο κεφάλαιο).

Τα ψηφία ισοτιμίας P_i παρεμβάλλονται στις θέσεις 1, 2, 4, 8,..., 2^{k-1}, δηλαδή στις θέσεις εκείνες που προσδιορίζονται από τις δυνάμεις του 2. Η τιμή τους εξαρτάται από το αν έχουμε επιλέξει άρτια ή περιττή ισοτιμία. Τα ψηφία της χρήσιμης πληροφορίας X_1, X_2, X_3,..., X_m καταλαμβάνουν τις υπόλοιπες θέσεις της μεταδιδόμενης πληροφορίας.

Στην περίπτωση όπου η πληροφορία που προορίζεται να μεταδοθεί έχει μήκος π.χ. m=8 bit, η μεταδιδόμενη λέξη θα έχει μήκος m+k=12 bit, αφού θα έχουν συμπεριληφθεί, σύμφωνα με τον *πίνακα 1.18,* τέσσερα ψηφία ισοτιμίας τα P_1, P_2, P_4 και P_8, θα έχει δηλαδή τη μορφή:

Θέση ψηφίων	1	2	3	4	5	6	7	8	9	10	11	12
Σημασία ψηφίων	$\mathbf{P_1}$	$\mathbf{P_2}$	$\mathbf{X_1}$	$\mathbf{P_4}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$	$\mathbf{P_8}$	$\mathbf{X_5}$	$\mathbf{X_6}$	$\mathbf{X_7}$	$\mathbf{X_8}$

Τρόπος προσδιορισμού των bits του κώδικα Hamming

Στον κώδικα Hamming χρησιμοποιούνται επιπλέον ψηφία ισοτιμίας, που μας δίνουν τη δυνατότητα προσδιορισμού ενός απλού λάθους.

Για τη δημιουργία της κωδικής λέξης εργαζόμαστε ως εξής: Σημειώνουμε όλες τις θέσεις που αντιστοιχούν σε δυνάμεις του 2, οι οποίες αποτελούν και τα ψηφία ισοτιμίας (parity bits). Τα ψηφία αυτά θα γεμίσουν με δυαδικά ψηφία, τα οποία είναι αποτέλεσμα υπολογισμών με τη χρήση των δυαδικών ψηφίων της αρχικής λέξης. (θέσεις 1, 2, 4, 8, 16, 32, 64, κ.λπ.)

Όλες οι υπόλοιπες θέσεις θα χρησιμοποιηθούν για τα δεδομένα που πρόκειται να κωδικοποιηθούν. (θέσεις 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17, κ.λπ.)

Θέση	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Ψηφίο 1	***x***		x		x		x		x		x		x		x		x
Ψηφίο 2		***x***	x			x	x			x	x			x	x		
Ψηφίο 4				***x***	x	x	x					x	x	x	x		
Ψηφίο 8								***x***	x	x	x	x	x	x	x		
Ψηφίο 16																***x***	x

Κάθε ψηφίο ισοτιμίας υπολογίζει την ισοτιμία κάποιων συγκεκριμένων δυαδικών ψηφίων της αρχικής, υπό κωδικοποίηση , λέξης. Η θέση του ψηφίου ισοτιμίας προσδιορίζει και την ακολουθία των υπό χρήση bits, ακολουθώντας τον παρακάτω αλγοριθμική διαδικασία. (για τις θέσεις 1 – 32).

Θέση 1: έλεγξε 1 bit, παράλειψε 1 bit, έλεγξε 1 bit, παράλειψε 1 bit κ.λπ. (1, 3, 5, 7 , 9 , 11, 13, 15,...)

Θέση 2: έλεγξε 2 bits, παράλειψε 2 bits, έλεγξε 2 bits, παράλειψε 2 bits, κ.λπ. (2, 3, 6, 7, 10, 11, 14, 15, ...)

Θέση 4: έλεγξε 4 bits, παράλειψε 4 bits, έλεγξε 4 bits, παράλειψε 4 bits, κ.λπ. (4, 5, 6, 7, 12, 13, 14, 15, 20, 21, 22, 23,...)

Θέση 8: έλεγξε 8 bits, παράλειψε 8 bits, έλεγξε 8 bits, παράλειψε 8 bits, κ.λπ. (8-15, 24-31, 40-47,...)

Θέση 16: έλεγξε 16 bits, παράλειψε 16 bits, έλεγξε 16 bits, παράλειψε 16 bits, κ.λπ. (16-31, 48-63, 80-95,...)

Θέση 32: έλεγξε 32 bits, παράλειψε 32 bits, έλεγξε 32 bits, παράλειψε 32 bits, κ.λπ. (32-63, 96-127, 160-191, ...)

Θέστε το parity bit σε 1, εάν το συνολικό άθροισμα των άσσων στις υπό έλεγχο θέσεις είναι περιττό.

Θέστε το parity bit σε 0, εάν το συνολικό άθροισμα των άσσων στις υπό έλεγχο θέσεις είναι άρτιο.

Παρατηρήστε λοιπόν ότι ξεκινάμε πάντα από το ψηφίο ισοτιμίας του οποίου θέλουμε να προσδιορίσουμε την τιμή. Η ομάδα ψηφίων που παράγει το ψηφίο ισοτιμίας P_1 αποτελείται από τα bit που βρίσκονται στις θέσεις 3, 5, 7, 9, 11 της λέξης. (Το ψηφίο ή τα ψηφία τα οποία ανήκουν στις θέσεις των ψηφίων ισοτιμίας δεν λαμβάνονται υπόψη). Η αντίστοιχη ομάδα που δίνει το P_2 αποτελείται από τα bit των θέσεων 3, 6, 7, 10, 11, ενώ οι ομάδες που δίνουν τα P_4 και P_8 είναι αυτές που περιλαμβάνουν τα bit των θέσεων 5, 6, 7, 12 και 9, 10, 11, 12 αντίστοιχα.

Παράδειγμα 1ο:

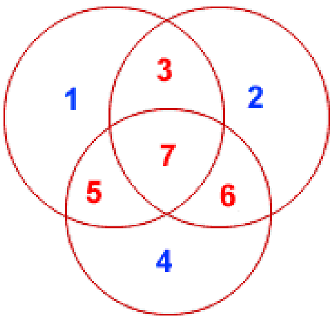
Ένα εύλογο ερώτημα είναι το γιατί επιλέγουμε τα ψηφία αυτά - Τα τρία πρώτα ψηφία ισοτιμίας (1, 2, 4) σχετίζονται με τα ψηφία δεδομένων (3,5,6,7). Στα σχήματα.2.3α και 2.3β κάθε αλληλεπικαλυπτόμενος κύκλος αντιστοιχεί σε ένα ψηφίο ισοτιμίας και ορίζει τα τέσσερα ψηφία, που συνεισφέρουν στον υπολογισμό της ι-σοτιμίας. Για παράδειγμα, το ψηφίο δεδομένων 3 συνεισφέρει στα ψηφία ισοτιμίας 1 και 2. Κάθε κύκλος (ψηφίο ισοτιμίας-parity bit) συμπεριλαμβάνει ένα σύνολο από τέσσερα ψηφία, και κάθε κύκλος πρέπει να έχει άρτια ισοτιμία. Λαμβάνοντας υπόψη μας τα τέσσερα ψηφία των δεδομένων, μπορούμε πολύ εύκολα να επιλέξουμε τα τρία ψηφία ισοτιμίας ώστε να εξασφαλισθεί η άρτια ισοτιμία.

Εύκολα διαπιστώνουμε ότι η αλλαγή ενός οποιουδήποτε δυαδικού ψηφίου από τα 7 επηρεάζει τα τρία ψηφία ισοτιμίας. Η τροποποίηση του 7^{ου} δυαδικού ψηφίου επηρεάζει όλα τα parity bits, ενώ σφάλμα στο 6^{ο} δυαδικό ψηφίο επηρεάζει μόνο τα ψηφία ισοτιμίας 2 και 4, και ένα σφάλμα ενός parity bit επηρεάζει μόνο αυτό το δυαδικό ψηφίο. Η θέση οποιοδήποτε απλού σφάλματος προσδιορίζεται από τον έλεγχο των τριών κύκλων ισοτιμίας.

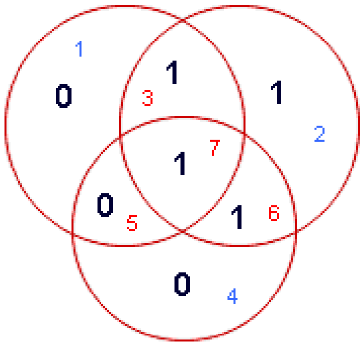
Το μήνυμα 1011 θα μπορούσε να σταλεί ως 0110011, καθώς:

1	2	3	4	5	6	7	
0	1	1	0	0	1	1	7-ψήφια κωδική λέξη
0	-	1	-	0	-	1	Άρτια ισοτιμία
-	1	1	-	-	1	1	Άρτια ισοτιμία
-	-	-	0	0	1	1	Άρτια ισοτιμία

Όταν τα τέσσερα αυτά ψηφία εισαχθούν στους κύκλους ισοτιμίας, επιβεβαιώνεται ότι η επιλογή των τριών αυτών ψηφίων ισοτιμίας βεβαιώνει ότι η ισοτιμία σε κάθε κύκλο είναι άρτια, όπως φαίνεται από τα σχήματα *2.4α* και *2.4β*.



Σχήμα 2.4α.



Σχήμα 2.4β.

Παράδειγμα 2ο:

Έστω η ακολουθία δεδομένων: 10011010. Να βρεθεί η προς αποστολή λέξη η οποία κωδικοποιείται με τον κώδικα Hamming.

Θέση	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Ισοτιμία**
Ψηφίο / (δεδομένα)	**P1**	**P2**	1	**P4**	0	0	1	**P8**	1	0	1	0	
Έλεγχος: 1	*x*		x		x		x		x		x		0
Έλεγχος: 2		*x*	x			x	x			x	x		1
Έλεγχος: 4				*x*	x	x	x					x	1
Έλεγχος: 8								*x*	x	x	x	x	0
Τελική Κωδική Λέξη	0	1	1	1	0	0	1	0	1	0	1	0	

Άρα η τελική κωδική λέξη που θα αποσταλεί είναι : **0111 0010 1010**

Παράδειγμα 3ο:
Ας θεωρήσουμε ότι θέλουμε να μεταδώσουμε μια πληροφορία 4 bit, έστω τον δυαδικό αριθμό **1010**. Θα θεωρήσουμε ακόμα ότι τα εμπλεκόμενα ψηφία ισοτιμίας αποκτούν τιμή σύμφωνα με τους κανόνες της άρτιας ισοτιμίας.

Τα ψηφία της πληροφορίας είναι 4, δηλαδή m=4, οπότε σύμφωνα με τον *πίνακα 1.18* έχουμε k=3. Η μεταδιδόμενη λέξη θα έχει, επομένως, τη μορφή:

Θέση ψηφίων	1	2	3	4	5	6	7
Σημασία ψηφίων	$\mathbf{P_1}$	$\mathbf{P_2}$	$\mathbf{X_1}$	$\mathbf{P_4}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$
Τιμή ψηφίων	-	-	1	-	0	1	0

Η τιμή των ψηφίων P_i υπολογίζεται αμέσως παρακάτω σύμφωνα με τους κανόνες της άρτιας ισοτιμίας. Για το ψηφίο ισοτιμίας P_1 λαμβάνουμε υπόψη τις τιμές των ψηφίων 3, 5 και 7. Έχουμε επομένως:

Θέση ψηφίων	3	5	7
Σημασία ψηφίων	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_4}$
Τιμή ψηφίων	1	0	0

από όπου προκύπτει ότι $\mathbf{P_1 = 1}$.

Για το ψηφίο ισοτιμίας P_2 λαμβάνουμε υπόψη τις τιμές των ψηφίων 3, 6 και 7. Έχουμε επομένως:

Θέση ψηφίων	3	6	7
Σημασία ψηφίων	$\mathbf{X_1}$	$\mathbf{X_3}$	$\mathbf{X_4}$
Τιμή ψηφίων	1	1	0

από όπου προκύπτει ότι $\mathbf{P_2 = 0}$.

Για το ψηφίο ισοτιμίας P_4 λαμβάνουμε υπόψη τις τιμές των ψηφίων 5, 6 και 7. Έχουμε επομένως:

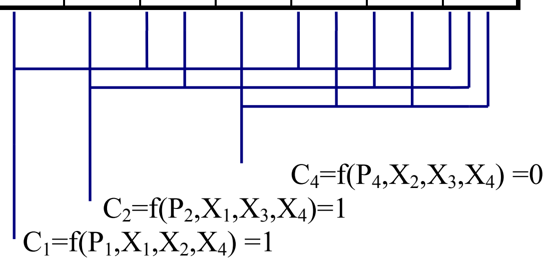
Θέση ψηφίων	5	6	7
Σημασία ψηφίων	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$
Τιμή ψηφίων	0	1	0

από όπου προκύπτει ότι $\mathbf{P_4 = 1}$.

Παρεμβάλλοντας στη λέξη τα ψηφία ισοτιμίας, που υπολογίσθηκαν με τον παραπάνω τρόπο, καταλήγουμε στο συμπέρασμα ότι η προς μετάδοση λέξη θα είναι η **1011010**.

Προσδιορισμός και διόρθωση λάθους
Αν, κατά τη μετάδοση, συμβεί σφάλμα στη θέση 3, η λέξη θα αποκτήσει τιμή ίση με 1001010. Η έξοδος του κυκλώματος λήψης θα αναδείξει την πληροφορία:

Θέση ψηφίων	1	2	3	4	5	6	7
Σημασία ψηφίων	$\mathbf{P_1}$	$\mathbf{P_2}$	$\mathbf{X_1}$	$\mathbf{P_4}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$
Τιμή ψηφίων	1	0	0	1	0	1	0



Υποβάλλοντας τη ληφθείσα ακολουθία δυαδικών ψηφίων σε διαδικασίες παραγωγής ψηφίων ισοτιμίας, σύμφωνα με τους ίδιους κανόνες που χρησιμοποιήθηκαν για την παραγωγή των ίδιων ψηφίων κατά την εκπομπή, και συγκρίνοντας κατόπιν τα ψηφία αυτά με τα ψηφία ισοτιμίας που έχουν ήδη ληφθεί, διαπιστώνουμε ότι οι έξοδοι του κυκλώματος σύγκρισης παράγουν τον συνδυασμό $\mathbf{C_4C_2C_1 = 011}$ που είναι διαφορετικός από τον $C_4C_2C_1$ **= 000,** ο οποίος εκφράζει τη λήψη χωρίς σφάλματα. Η τιμή του συνδυασμού υποδηλώνει και τη θέση του λανθασμένου ψηφίου, που, στην περίπτωσή μας, είναι το τρίτο ($011_2 = 3_{10}$).

Παράδειγμα 4ο:

Ας θεωρήσουμε ότι θέλουμε να μεταδώσουμε μια πληροφορία 8 bit, έστω το δυαδικό αριθμό **10101101**, χρησιμοποιώντας ψηφία άρτιας ισοτιμίας.

Τα ψηφία της πληροφορίας είναι 8, οπότε m=8 και σύμφωνα με τον *πίνακα 1.18*, k=4. Η μεταδιδόμενη λέξη θα έχει, επομένως, τη μορφή:

Θέση ψηφίων	1	2	3	4	5	6	7	8	9	10	11	12
Σημασία ψηφίων	P_1	P_2	X_1	P_4	X_2	X_3	X_4	P_8	X_5	X_6	X_7	X_8
Τιμή ψηφίων			1		0	1	0		1	1	0	1

Για το ψηφίο ισοτιμίας P_1 λαμβάνουμε υπόψη τις τιμές των ψηφίων 3, 5, 7, 9 και 11. Έχουμε επομένως:

Θέση ψηφίων	3	5	7	9	11
Σημασία ψηφίων	X_1	X_2	X_4	X_5	X_7
Τιμή ψηφίων	1	0	0	1	0

από όπου προκύπτει ότι $P_1 = 0$.

Για το ψηφίο ισοτιμίας P_2 λαμβάνουμε υπόψη τις τιμές των ψηφίων 3, 6, 7, 10 και 11. Έχουμε επομένως:

Θέση ψηφίων	3	6	7	10	11
Σημασία ψηφίων	X_1	X_3	X_4	X_6	X_7
Τιμή ψηφίων	1	1	0	1	0

από όπου προκύπτει ότι $P_2 = 1$.

Για το ψηφίο ισοτιμίας P_4 λαμβάνουμε υπόψη τις τιμές των ψηφίων 5, 6, 7 και 12. Έχουμε επομένως:

Θέση ψηφίων	5	6	7	12
Σημασία ψηφίων	X_2	X_3	X_4	X_8
Τιμή ψηφίων	0	1	0	1

από όπου προκύπτει ότι $P_4 = 0$.

Για το ψηφίο ισοτιμίας P_8 λαμβάνουμε υπόψη τις τιμές των ψηφίων 9, 10, 11 και 12. Έχουμε επομένως:

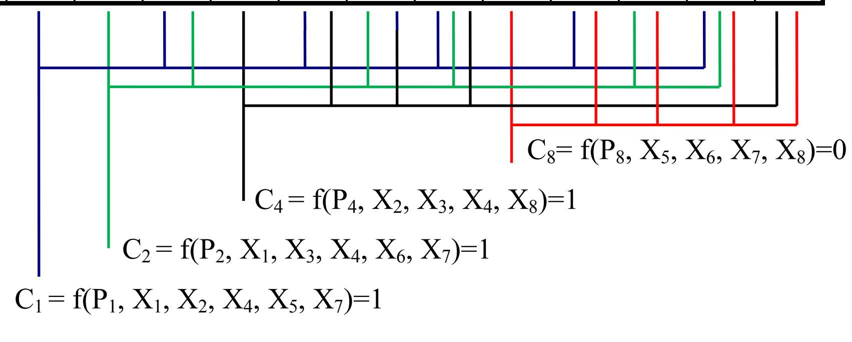
Θέση ψηφίων	9	10	11	12
Σημασία ψηφίων	X_5	X_6	X_7	X_8
Τιμή ψηφίων	1	1	0	1

από όπου προκύπτει ότι $P_8 = 1$.

Παρεμβάλλοντας στη λέξη τα ψηφία ισοτιμίας που υπολογίσθηκαν με τον παραπάνω τρόπο, καταλήγουμε στο συμπέρασμα ότι η προς μετάδοση λέξη θα είναι η **011001011101**.

Προσδιορισμός και διόρθωση λάθους

Θέση ψηφίων	1	2	3	4	5	6	7	8	9	10	11	12
Σημασία ψηφίων	P_1	P_2	X_1	P_4	X_2	X_3	X_4	P_8	X_5	X_6	X_7	X_8
Τιμή ψηφίων	0	1	1	0	0	1	1	1	1	1	0	1



Αν κατά τη μετάδοση συμβεί σφάλμα στη θέση 7, η λέξη θα αποκτήσει τιμή ίση με: 011001111101. Η έξοδος του κυκλώματος λήψης θα αναδείξει τότε την πληροφορία.

Υποβάλλοντας τα ληφθέντα δυαδικά ψηφία σε επεξεργασίες ίδιες μ' αυτές που αναφέρθηκαν στο προηγούμενο παράδειγμα, το κύκλωμα ελέγχου εγκυρότητας της πληροφορίας θα δώσει στις εξόδους του το συνδυασμό $\mathbf{C_8C_4C_2C_1 = 0111}$, ο οποίος είναι διαφορετικός από το συνδυασμό $\mathbf{C_8C_4C_2C_1 = 0000}$, που εκφράζει την ορθά ληφθείσα πληροφορία. Η τιμή του συνδυασμού υποδηλώνει και τη θέση του λανθασμένου ψηφίου, που στη συγκεκριμένη περίπτωση είναι το τρίτο ($0111_2 = 7_{10}$).

Ο κώδικας Hamming μας παρέχει: Διόρθωση ενός ψηφίου, ανίχνευση δύο εσφαλμένων ψηφίων, κόστος 3 ψηφίων, που πρέπει να προστεθούν στα 4 ψηφία του μηνύματος.
Η δυνατότητα της διόρθωσης ενός σφάλματος έχει κόστος, το οποίο είναι μικρότερο από το κόστος της επανεκπομπής του μηνύματος.

2.6. Αριθμητική κινητής υποδιαστολής και το πρότυπο IEEE-754

Κύριος στόχος μας είναι να μελετήσουμε τον τρόπο με τον οποίο γίνονται οι πράξεις κινητής υποδιαστολής. Αρχικά θα μελετήσουμε τους αριθμούς αυτούς και, στη συνέχεια, θα δούμε τα μοντέλα εκτέλεσης πράξεων. *(Predko, 2005; Kahan, 2008; IEEE, 2008; Pritchard, 2015; Rajaraman, & Adabala, 2015)*

2.6.1. Η ιστορία των πράξεων κινητής υποδιαστολής

Υπάρχουν διάφοροι τρόποι, για να αναπαρασταθούν οι πραγματικοί αριθμοί στους υπολογιστές. Ένας τρόπος είναι η κινητή υποδιαστολή. Η κινητή υποδιαστολή δεν είναι προκαθορισμένη, αλλά μεταβάλλεται και προσαρμόζεται αυτόματα στις μεταβαλλόμενες απαιτήσεις των υπολογισμών. Έτσι η θέση της υποδιαστολής μετακινείται (floats) και οι αριθμοί αυτοί ονομάζονται *αριθμοί κινητής υποδιαστολής*. Αυτό τους ξεχωρίζει από τους αριθμούς σταθερής υποδιαστολής, στους οποίους η υποδιαστολή βρίσκεται πάντα στην ίδια θέση.

Στις πρώτες μηχανές οι προγραμματιστές έπρεπε να γράψουν λογισμικό για την εκτέλεση των λειτουργιών της κινητής υποδιαστολής. Έτσι κάθε μηχανή είχε τον δικό της τρόπο υλοποίησης της αριθμητικής Κινητής Υποδιαστολής (Κ.Υ.), τόσο στο υλικό όσο και στο λογισμικό. Ήταν λοιπόν αδύνατο να γραφούν μεταβιβάσιμα προγράμματα, τα οποία θα οδηγούσαν στα ίδια αποτελέσματα σε διαφορετικά συστήματα.

Το πρότυπο IEEE-754 υιοθετήθηκε το 1985 και καθόριζε στους κατασκευαστές των Κ.Μ.Ε. τον τρόπο υλοποίησης των υπολογισμών της κινητής υποδιαστολής.

Έχοντας ένα πρότυπο, τουλάχιστον έχουμε τη διαβεβαίωση ότι όλες οι συμβατές μηχανές για το ίδιο πρόγραμμα θα παράξουν την ίδια έξοδο.

2.6.2. Υλικό Κινητής Υποδιαστολής

Όταν εισήχθη στους μικροεπεξεργαστές η κινητή υποδιαστολή, δεν υπήρχαν αρκετά τρανζίστορ στη ψηφίδα (on chip), ώστε να είναι δυνατή η υλοποίηση της. Έτσι οι κατασκευαστές των ΚΜΕ παρείχαν ξεχωριστούς συνεπεξεργαστές για τις απαραίτητες λειτουργίες της Κινητής Υποδιαστολής, π.χ., ο 8087 της Intel. Αποτέλεσμα ήταν πολλές ISA να χρησιμοποιούν ξεχωριστούς καταχωρητές για την Κινητή Υποδιαστολή.

Ο προϋπολογισμός του κόστους των νέων τρανζίστορ έδωσε τη δυνατότητα στη μονάδα Κ.Υ. να είναι πλέον στη ψηφίδα. Έτσι σήμερα οι ΚΜΕ έχουν ενσωματωμένες μονάδες κινητής υποδιαστολής. Ο 486 της Intel ήταν ο πρώτος x86 με ενσωματωμένη τη μονάδα Κ.Υ. (1989)

Ακόμα και οι νεότερες ISA έχουν ξεχωριστά αρχεία καταχωρητών για την κινητή υποδιαστολή.

2.6.3. Αριθμοί κινητής υποδιαστολής -Εισαγωγή

Μέχρι τώρα έχουμε ασχοληθεί, κατά κανόνα, με ακέραιες τιμές. Για τους πραγματικούς αριθμούς, δηλαδή για αριθμούς που περιέχουν και ακέραιο μέρος και μέρος μετά την υποδιαστολή, χρειαζόμαστε την αναπαράσταση της κινητής υποδιαστολής π.χ. οι αριθμοί: 3.14159, 1.563×10^{28}

[Σημείωση: Για να παραστήσουμε έναν αριθμό σε δεκαεξαδική μορφή, χρησιμοποιούμε συνήθως είτε το H είτε το x: Έτσι, π.χ., ο αριθμός 0 γράφεται είτε ως 00H είτε ως 0x00.

Για τους δυαδικούς αντίστοιχα χρησιμοποιούμε την παράσταση b '001' (binary), και για τους δεκαδικούς την d '17' (decimal)].

Έστω ο αριθμός 0.5467×2^{28}. Στον αριθμό αυτό η ποσότητα 0.5467 αντιστοιχεί στα ψηφία μετά την υποδιαστολή (Mantissa), ο αριθμός 28 είναι ο εκθέτης και τέλος το 2 αντιστοιχεί στη βάση. Οι αριθμοί κινητής υποδιαστολής ορίζονται συναρτήσει:

1.των ψηφίων μετά την υποδιαστολή (Mantissa),

2.του εκθέτη και τέλος
3.της βάσης – (πάντα 2 για δυαδικούς αριθμούς κινητής υποδιαστολής).

2.6.4. Το εύρος των αριθμών

Τι μπορούμε να παρουσιάσουμε με N δυαδικά ψηφία; Συνήθως:

Μη προσημασμένους	Από 0 μέχρι 2^{N-1}
Συμπληρώματα ως προς 2	Από -2^{N-1} μέχρι $2^{N-1}-1$

Αλλά τι γίνεται με τους:

Πολύ μεγάλους αριθμούς;	9,349,398,989,787,762,244,859,087,678
	1.23×10^{67}
Πολύ μικρούς;	0.0000000000000000000000045691
	2.98×10^{-32}
Δεκαδικούς αριθμούς;	0.35
Μικτούς αριθμούς;	10.28
Άρρητους;	π

2.6.5. Αναπαράσταση αριθμών κινητής υποδιαστολής

Οι αριθμοί κινητής υποδιαστολής, κατά IEEE-754, αποθηκεύονται με τη χρήση του τύπου: ± μέγεθος (mantissa) * $2^{\text{εκθέτης}}$

Μπορούμε να αναπαραστήσουμε τους αριθμούς Κ.Υ. με τρία δυαδικά πεδία: το ψηφίο προσήμου s, ένα εκθετικό πεδίο e και ένα κλασματικό πεδίο f.

s (sign)	***e (exponent)***	***f (fraction) (mantissa)***

Το Πρόσημο (sign)

Το ψηφίο προσήμου (s) είναι 0 για τους θετικούς αριθμούς και 1για τους αρνητικούς αριθμούς. Σε αντίθεση λοιπόν με τους ακεραίους, οι τιμές κατά IEEE-754 αποθηκεύονται σε μορφή προσημασμένου μεγέθους.

Τα ψηφία μετά την υποδιαστολή (Mantissa)

Το πεδίο f περιέχει μία δυαδική ακολουθία αξίας μικρότερης της μονάδας. Το πραγματικό πρόσημο και μέγεθος (mantissa) της τιμής του αριθμού κινητής υποδιαστολής είναι (1 + f).

Έτσι υπάρχει ένας υπονοούμενος άσσος «1» αριστερά της υποδιαστολής του δυαδικού αριθμού. Για παράδειγμα, εάν το f είναι 01101…, τότε το πρόσημο και μέγεθος θα είναι 1.01101…

Υπάρχουν πολλοί τρόποι, για να γράψουμε ένα αριθμό με μαθηματικό συμβολισμό, αλλά πάντα υπάρχει μία μοναδική κανονικοποιημένη αναπαράσταση με ακριβώς ένα μη-μηδενικό ψηφίο στα αριστερά της υποδιαστολής. $0.232 * 10^3 = 23.2 * 10^1 = 2.32 * 10^2 = \dots$

Ένα παράπλευρο αποτέλεσμα είναι ότι έχουμε λίγο μεγαλύτερη ακρίβεια: Υπάρχουν 24 bits στο πραγματικό πρόσημο και μέγεθος, αλλά χρειάζεται να αποθηκεύσουμε μόνο τα 23 από αυτά.

Εκθέτης

Το πεδίο e αναπαριστά τον εκθέτη σαν έναν αριθμό χωρίς πρόσημο, ο οποίος περιέχει τον πραγματικό εκθέτη *συν* 127 για απλής ακριβείας αριθμούς, ή τον πραγματικό εκθέτη *συν* 1023 για διπλής ακριβείας αριθμούς.

Αυτό μετατρέπει όλους τους εκθέτες απλής ακριβείας από το -127 έως το +128, σε μη προσημασμένους αριθμούς από το 0 μέχρι το 255, και όλους τους αριθμούς διπλής ακριβείας από το -1023 έως +1024 σε μη προσημασμένους αριθμούς από το 0 μέχρι το 2047.

Παρατίθενται δύο παραδείγματα με αριθμούς απλής ακριβείας:

Εάν ο εκθέτης είναι 4, τότε το πεδίο e θα είναι 4 + 127 = 131 (10000011_2). Εάν το πεδίο e περιέχει τον αριθμό 01011101 (93_{10}), τότε ο εκθέτης είναι 93 - 127 = -34.

Αποθηκεύοντας ένα «μετατοπισμένο» εκθέτη *πριν* από την κανονικοποιημένη mantissa, σημαίνει ότι μπορούμε να συγκρίνουμε τις τιμές της IEEE-754 σαν να ήταν προσημασμένοι ακέραιοι.

Το πρότυπο IEEE-754 ορίζει διαφορετικές ακρίβειες.

2.6.6. Παράσταση αριθμών απλής / διπλής ακριβείας

Θα μελετήσουμε τους αριθμούς κινητής υποδιαστολής απλής και διπλής ακριβείας.

A) **Αριθμοί απλής ακριβείας**

Οι *αριθμοί απλής ακριβείας* έχουν ένα εκθετικό πεδίο με 8-bit και ένα κλασματικό πεδίο με 23-bit, σε ένα σύνολο 32 bits.

Πρόσημο 1-bit s(sign)	**Εκθέτης 8-bit** e (exponent)	**Μέγεθος (mantissa) 23-bit** f (fraction)

Ποιος είναι ο μικρότερος θετικός αριθμός απλής ακριβείας που μπορεί να παρασταθεί με την IEEE-754;

0	00000001	00000000000000000000000

+	1	0

(Όπως θα δούμε παρακάτω το 00000000 στον εκθέτη χρησιμοποιείται για ειδικό σκοπό.)

B) **Αριθμοί διπλής ακριβείας**

Οι *αριθμοί διπλής ακριβείας* έχουν ένα εκθετικό πεδίο 11-bit και ένα κλασματικό πεδίο 52-bit, σε ένα σύνολο από 64 bits.

Πρόσημο 1-bit s(sign)	**Εκθέτης 11-bit** e (exponent)	**Μέγεθος (mantissa) 52-bit** f (fraction) / κλάσμα

Είναι γνωστή η σχέση : $(1 - 2s) * (1 + f) * 2^{e-127}$.

- Ο μικρότερος θετικός μη μηδενικός αριθμός είναι ο $(1 - 0) * (1 + 0) * 2^{1-127} = 1 * 2^{-126} = 2^{-126}$.
- Ο μικρότερος εκθέτης e είναι ο είναι 00000001 (1).
- Το μικρότερο κλασματικό τμήμα f είναι 0000000000000000000000 (0).
- Ο μεγαλύτερος "κανονικός « αριθμός είναι $(1 - 2s) * (1 + f) * 2^{e-127} = (1 - 0)*(1+1-2^{-23})*2^{254-127} = (2- 2^{-23})*2^{127} = 2^{128} - 2^{104}$.
- Το μεγαλύτερο e είναι 11111110 (254). (η τιμή 11111111 είναι ειδική τιμή, άρα δε χρησιμοποιείται)
- Το μεγαλύτερο f είναι 11111111111111111111111 $(1 - 2^{-23})$.

Για να συγκρίνουμε, ο μικρότερος και ο μεγαλύτερος ακέραιος με 32-bit, σε συμπλήρωμα ως προς 2, είναι μόνο -2^{31} και $2^{31} – 1$ αντίστοιχα.

Πώς μπορούμε να αναπαραστήσουμε τόσες πολλές περισσότερες τιμές κατά IEEE-754, παρόλο που χρησιμοποιούμε τον ίδιο αριθμό ψηφίων. όπως και στους κανονικούς ακεραίους;

Περατότητα

Δεν υπάρχουν περισσότεροι αριθμοί στην αναπαράσταση κατά IEEE-754. Με 32 bits, υπάρχουν 2^{32}-1, ή περίπου 4 δισεκατομμύρια, διαφορετικές μορφές bit.

Αυτοί μπορούν να αναπαραστήσουν είτε 4 δισεκατομμύρια ακέραιους είτε 4 δισεκατομμύρια πραγματικούς αριθμούς. Αλλά υπάρχουν άπειροι πραγματικοί αριθμοί, και το μορφότυπο της IEEE-754 μπορεί να αναπαραστήσει μόνο αυτούς που κείνται στο εύρος -2^{128} έως $+2^{128}$.

Αναπαριστά τον ίδιο αριθμό τιμών μεταξύ των 2^n και 2^{n+1} όπως και 2^{n+1} και 2^{n+2} . Έτσι η αριθμητική της κινητής υποδιαστολής έχει «προβλήματα». Μικρά σφάλματα στρογγυλοποίησης μπορούν να συσσωρευθούν με τους πολλαπλασιασμούς ή τις εκθετικές συναρτήσεις, και να έχουν σαν αποτέλεσμα μεγάλα λάθη. Λάθη στρογγυλοποίησης μπορούν να καταστήσουν άκυρες πολλές βασικές αριθμητικές αρχές, όπως π.χ. την προσεταιριστική ιδιότητα, $(x + y) + z = x + (y + z)$.

Το πρότυπο IEEE-754 εγγυάται ότι όλα τα μηχανήματα θα παράγουν το ίδιο αποτέλεσμα αλλά αυτά τα αποτελέσματα ίσως να μην είναι μαθηματικώς σωστά!

2.6.7. Αριθμητική κινητής υποδιαστολής

Η κινητή υποδιαστολή (Κ.Υ.) απλοποιεί, κατά πολύ, την εργασία με μεγάλους (π.χ., 2^{70}) ή πολύ μικρούς (π.χ., 2^{-17}) αριθμούς.

Οι πρώτες μηχανές εκτελούσαν αυτή τη λειτουργία με λογισμικό με τους «παράγοντες κλιμακοποίησης». Θα επικεντρωθούμε στο πρότυπο IEEE-754 για την αριθμητική κινητής υποδιαστολής. Θα εξετάσουμε:

- Πώς αναπαρίστανται οι αριθμοί Κ.Υ.
- Ποια είναι τα όρια των αριθμών Κ.Υ.
- Την πρόσθεση και τον πολλαπλασιασμό αριθμών Κ.Υ.

2.6.8. Μετατροπή αριθμών μεταξύ της IEEE-754 και του δεκαδικού

s	*e*	*f*

Η δεκαδική τιμή ενός IEEE αριθμού δίνεται από την εξίσωση: $(1 - 2s) * (1 + f) * 2^{e\text{-}bias}$
Τα πεδία s, f και e θεωρούμε ότι είναι στο δεκαδικό σύστημα. Η ποσότητα (1 - 2s) είναι 1 ή -1, εξαρτώμενη από το εάν το ψηφίο του προσήμου είναι 0 ή 1.Τέλος στο κλασματικό πεδίο του f, προσθέτουμε έναν υπονοούμενο άσσο «1». Η *μετατόπιση*-offset(bias) είναι είτε 127 για αριθμούς απλής ακριβείας είτε 1023 για διπλής ακριβείας.

Παράδειγμα 5ο:
***1.* Μετατροπή από IEEE-754 σε δεκαδικό αριθμό**

a) (Πρώτη μέθοδος επίλυσης)
Έστω ότι θέλουμε να βρούμε τη δεκαδική τιμή του IEEE-754 αριθμού που ακολουθεί:
1 01111100 11000000000000000000000
Πρώτα μετατρέπουμε κάθε πεδίο στον αντίστοιχο δεκαδικό αριθμό.
Το ψηφίο προσήμου s είναι 1.
Το πεδίο e περιέχει την τιμή $01111100 = 124_{10}$.
Η mantissa είναι $0.11000\ldots = 0.75_{10}$. (½ + ¼)
Κατόπιν απλά τοποθετούμε αυτές τις δεκαδικές τιμές των s, e και f στην εξίσωσή μας:
$(1 - 2s) * (1 + f) * 2^{e\text{-}bias}$
Αυτό μας δίνει $(1 - 2) * (1 + 0.75) * 2^{124-127} = (-1) * (1.75 * 2^{-3}) = -0.21875$.

b) (Δεύτερη μέθοδος επίλυσης)
Έστω ότι θέλουμε να βρούμε τη δεκαδική τιμή του ακόλουθου IEEE-754 αριθμού απλής ακριβείας.
0 10001000 10010010000000000000000
Θα χρησιμοποιήσουμε την εξίσωση $X = (-1)^S \times 2^{E-127} \times (1.M)$
(Το .M συμβολίζει το μέρος μετά την υποδιαστολή και όχι κάποιο γινόμενο) όπου: S = 0
$E = 10001000_2 = 136_{10}$
$1.M = 1.10010010000000000000000 = 1 + 2^{-1} + 2^{-4} + 2^{-7} = 1.5703125$
Έτσι τελικά: $X = (-1)^0 \times 2^{136-127} \times 1.5703125 = 804 = 8.04 \times 10^2$.

2. Μετατροπή δεκαδικού αριθμού σε IEEE-754
Ποια είναι η αναπαράσταση του αριθμού 347.625 σε απλής ακριβείας;
Αρχικά μετατρέπουμε το δεκαδικό αριθμό σε δυαδικό: $347.625 = 101011011.101_2$.
Κανονικοποιούμε το δυαδικό αριθμό, ολισθαίνοντας τη δεκαδική τελεία, μέχρι να υπάρχει αριστερά της μόνο ένας 1: $101011011.101 \times 2^0 = 1.01011011101 \times 2^8$
Τα ψηφία στα δεξιά της υποδιαστολής αποτελούν το κλασματικό πεδίο f.
Ο αριθμός των ολισθήσεων που εκτελέσαμε μας δίνει τον εκθέτη. Το πεδίο e θα πρέπει να περιέχει την τιμή: εκθέτης + 127.
Ψηφίο προσήμου: 0, εάν είναι θετικός, και 1 εάν είναι αρνητικός.
(Να επαληθευθεί το αποτέλεσμα μετατρέποντάς το πάλι στο δεκαδικό)

Ειδικές περιπτώσεις
Ο μικρότερος και ο μεγαλύτερος εκθέτης είναι αντίστοιχα ο e=00000000 και ο e=11111111 (και τα διπλής ακριβείας αντίγραφά τους) και είναι δεσμευμένοι για ειδικές περιπτώσεις.
Εάν η mantissa είναι πάντα (1 + f), τότε πώς μπορεί να αναπαρασταθεί το 0;
Το κλασματικό πεδίο f θα πρέπει να είναι 0000...0000.
Το εκθετικό πεδίο e περιέχει την τιμή 00000000.

Με προσημασμένα μεγέθη, υπάρχουν δύο μηδενικά: +0.0 και -0.0.

Όσον αφορά στο άπειρο υπάρχουν αναπαραστάσεις του θετικού και του αρνητικού απείρου, οι ο-ποίες θα μπορούσαν να βοηθήσουν μερικές φορές στην περίπτωση της υπερχείλισης.

Το τμήμα f είναι τότε 0000...0000.

Το εκθετικό τμήμα e ορίζεται σαν 11111111.

Τέλος υπάρχει μία ειδική τιμή «όχι ένας αριθμός», η οποία μπορεί να χειρισθεί μερικές περιπτώσεις λαθών ή μη επιτρεπτών λειτουργιών όπως η πράξη 0.0/0.0.

Το κλασματικό πεδίο f τίθεται τότε σε οποιαδήποτε μη μηδενική τιμή.

Ο εκθέτης e θα περιέχει την τιμή 11111111.

2.6.9. Όρια της αναπαράστασης IEEE-754

Υπάρχουν επίσης μερικοί απλοί δεκαδικοί αριθμοί οι οποίοι δεν μπορούν να αναπαρασταθούν ακριβώς στο δυαδικό όπως π.χ. το 0.1010, όπου $0.1010 = 0.0001100110011..._{(2)}$

Προβλήματα της αναπαράστασης IEEE-754 - Ο δεκαδικός αριθμός 0.10

Μια μελέτη προσδιόρισε την ακριβή αναπαράσταση, του 0.10 σε δυαδική μορφή. Η 24-bit δυαδική αναπαρά-σταση του 0.10 στην πραγματικότητα αντιστοιχεί στον αριθμό 0.099999904632568359375, ο οποίος απέχει του 0.10 κατά 0.000000095367431640625. Αυτή η διαφορά δε φαίνεται μεγάλη, αλλά μετά από 100 ώρες ο χρόνος τελειώνει, και είναι μικρότερος κατά 0.34 sec. Η χρήση του λοιπόν σε ηλεκτρονικά συστήματα, π.χ. αναχαίτησης, δίνει σφάλμα στον προσδιορισμό μεγεθών που σχετίζονται με το χρόνο κίνησης του συστήμα-τος. (*GAO report 1992)*

2.6.10.Πράξεις αριθμών κινητής υποδιαστολής (πρόσθεση και πολλαπλασιασμός)

Α. Πρόσθεση αριθμών κινητής υποδιαστολής

1. **Εξισώνουμε τους εκθέτες**.

Ο τελεστέος με το μικρότερο εκθέτη πρέπει να ξαναγραφεί, αυξάνοντας τον εκθέτη του και ολισθαί-νοντας την υποδιαστολή προς τα αριστερά:

$2.710 * 10^{-1} = 0.0271x\ 10^{1}$ Με τέσσερα σημαντικά ψηφία, ο αριθμός στρογγυλοποιείται στον: $0.027x\ 10^{1}$.

Αυτό έχει σαν αποτέλεσμα την απώλεια των λιγότερο σημαντικών ψηφίων, στην περίπτωσή μας το τελευταίο 1. Αλλά, ξαναγράφοντας τον αριθμό με μεγαλύτερο εκθέτη, αυτό μπορεί να έχει σαν αποτέλεσμα την απώλεια των περισσότερο σημαντικών ψηφίων, το οποίο είναι χειρότερο.

2. **Προσθέτουμε τις mantissa.**

$$\begin{array}{rrcl} & 9.989 & * & 10^{1} \\ + & 0.027 & * & 10^{1} \\ \hline & 10.016 & * & 10^{1} \end{array}$$

Αναπαράσταση του αποτελέσματος (Βήματα 3-5)

3. **Κανονικοποιούμε το αποτέλεσμα, εάν είναι απαραίτητο.**

$10.016 * 10^{1} = 1.0016 * 10^{2}$

Αυτό το βήμα μπορεί να έχει σαν αποτέλεσμα η υποδιαστολή να ολισθήσει είτε προς τα αριστερά είτε προς τα δεξιά, ο δε εκθέτης είτε να αυξηθεί είτε να ελαττωθεί.

4. **Στρογγυλοποιούμε τον αριθμό, εάν είναι απαραίτητο.**

Ο αριθμός $1.0016 * 10^{2}$ στρογγυλοποιείται στον $1.002 * 10^{2}$

5. **Επανάληψη του βήματος 3 εάν το αποτέλεσμα δεν είναι κανονικοποιημένο**.

Στο παράδειγμά μας, το βήμα αυτό δεν απαιτείται, αλλά είναι πιθανόν να χρειάζεται στρογγυλοποίη-ση σε όλα τα ψηφία π.χ., η στρογγυλοποίηση του 9.9995 οδηγεί στον αριθμό 10.000.

Το αποτέλεσμά μας είναι $1.002 * 10^{2}$ ή 100.2. Το σωστό αποτέλεσμα είναι 100.16, και έτσι έχουμε την απάντηση με τέσσερα σημαντικά ψηφία, αλλά υπάρχει ένα μικρό λάθος.

Παράδειγμα 6ο:

Α. Πρόσθεση

Για να αντιληφθούμε τις λειτουργίες της κινητής υποδιαστολής, θα παραθέσουμε ένα παράδειγμα πρόσθεσης. Για λόγους απλότητας, θα χρησιμοποιήσουμε την αναπαράσταση με βάση το 10.

Θα κάνουμε την υπόθεση ότι η mantissa έχει τέσσερα ψηφία, και ο εκθέτης ένα ψηφίο. Έστω λοιπόν η πρόσθεση στο δεκαδικό σύστημα: 99.89 + 0.272 = 100.162

Οι τελεστέοι σαν κανονικοποιημένοι αριθμοί γράφονται ως εξής: $9.989 * 10^1$ και $2.720 * 10^{-1}$

Β. Πολλαπλασιασμός αριθμών κινητής υποδιαστολής

Έστω ότι θέλουμε να πολλαπλασιάσουμε τους δεκαδικούς αριθμούς: 99.89 και 0.272, δηλαδή τους κανονικο-ποιημένους αριθμούς, $9.989 * 10^1$ και $2.720 * 10^{-1}$.

Για να πολλαπλασιάσουμε δύο τιμές κινητής υποδιαστολής, πρώτα πολλαπλασιάζουμε τα μεγέθη και κατόπιν προσθέτουμε τους εκθέτες.

$$\begin{array}{rrcl} & 9.989 & * & 10^{1} \\ * & 2.720 & * & 10^{-1} \\ \hline & 27.170 & * & 10^{0} \end{array}$$

Κατόπιν μπορούμε να στρογγυλοποιήσουμε και να κανονικοποιήσουμε το αποτέλεσμα, παίρνοντας $2.717 * 10^1$. Το πρόσημο του γινομένου είναι η πράξη XOR των προσήμων των τελεστέων.

Εάν οι δύο αριθμοί έχουν το ίδιο πρόσημο, τότε το γινόμενό τους είναι θετικό, ενώ, εάν οι δύο αριθ-μοί έχουν διαφορετικό πρόσημο, τότε το γινόμενό τους είναι αρνητικό.

$$0 \oplus 0 = 0 \qquad 0 \oplus 1 = 1 \qquad 1 \oplus 0 = 1 \qquad 1 \oplus 1 = 0$$

Αυτό είναι ένα από τα κύρια πλεονεκτήματα της χρήσης προσημασμένων μεγεθών.

2.7. Ασκήσεις – Ερωτήσεις

ΕΡΩΤΗΣΕΙΣ ΠΟΛΛΑΠΛΗΣ ΕΠΙΛΟΓΗΣ

1. Στον κώδικα Gray, κάθε αριθμός είναι 3 φορές μεγαλύτερος από τη δυαδική αναπαράσταση του α-ριθμού.
   Α. Σωστό   Β. Λάθος

2. Χρησιμοποιήστε τους συντελεστές βαρύτητας, για να μετατρέψετε τους ακόλουθους αριθμούς του BCD σε δυαδικούς α)0101 0011 και β) 0010 0110 1000
   Α. 01010011 001001101000   Β. 11010100 100001100000
   Γ. 110101 100001100   Δ. 101011 001100001

3. Ένας δυαδικός κώδικας που εξελίσσεται έτσι ώστε να αλλάζει μόνο ένα bit μεταξύ δύο διαδοχικών κωδίκων είναι ο κώδικας:
   Α. Συμπληρώματος ως προς 9   Β. 8421
   Γ. Excess-3 (XS3)   Δ. Gray

4. Πόσες εξόδους έχει ένας αποκωδικοποιητής BCD;
   Α. 4   Β. 16   Γ. 8   Δ. 10

ΑΣΚΗΣΕΙΣ

1. Ο αριθμός N_2 = 101001100101 είναι στον κώδικα BCD 2421. Να μετατραπεί στον ισοδύναμο δε-καεξαδικό, δεκαδικό και οκταδικό αριθμό.

2. Να γράψετε έναν BCD κώδικα. Πως είναι δυνατόν να εφοδιασθεί αυτός με την ιδιότητα διαπί-στωσης ενός σφάλματος.

3. Καθεμία από τις ακόλουθες δυαδικές παραστάσεις είναι ένα BCD ψηφίο εφοδιασμένο με bit πε-ριττής ισοτιμίας. Σε ποιες παραστάσεις υπάρχει σφάλμα;
α) 00010   β) 00011   γ) 00010   δ) 0111 ε) 00101   στ) 11010

4. Να μετατρέψετε τους παρακάτω δυαδικούς αριθμούς στον κώδικα Gray.
α) 0111010010011101011   β) 100101110101011
γ) 1111100001010   δ) 0010101110001   ε) 1101101101

5. Να μετατρέψετε τους παρακάτω αριθμούς που είναι κωδικοποιημένοι με τον κώδικα Gray, στους αντίστοιχους δυαδικούς.
α) 10011100110101   β) 10101010101010011   γ) 1111111111
δ) 000111000111010011   ε) 10000001000111111

6. Να αναπαρασταθεί ο δεκαδικός αριθμός 468.625 κατά IEEE-754.

7. Να γίνουν οι πράξεις των δυαδικών αριθμών που ακολουθούν κατά IEEE-754: α) 53,97 + 0,15   β) 7,93 * 2,51

8. Δίνεται ο δεκαδικός αριθμός 192,21875. Να αναπαρασταθεί κατά IEEE-754.

9. Δίνεται ο κατά IEEE-754 αριθμός 1 01010111 1100............000. Να αναπαρασταθεί στο δεκαδικό σύστημα.

10. Να εκτελέσετε τον πολλαπλασιασμό των δεκαδικών αριθμών 10 επί -7. Να επιβεβαιώσετε το αποτέλεσμα ως προς την ορθότητά του. Τι παρατηρείτε συγκρίνοντας το αποτέλεσμα που παίρ-νετε με το αποτέλεσμα της προηγούμενης άσκησης;

11. Να εκτελέσετε στο δυαδικό σύστημα τη διαίρεση των δεκαδικών αριθμών 189 δια 7 με ανάκτη-ση και χωρίς ανάκτηση. Να σχολιάσετε τα τελικά αποτελέσματα καθώς και τον τρόπο υλοποίη-σης των διαιρέσεων.

12. Έστω ότι θέλουμε να μεταδώσουμε πληροφορία αποτελούμενη από 4 bit, έστω το δυαδικό α-ριθμό 1101, χρησιμοποιώντας περιττή ισοτιμία. Να προσδιορίσετε την υπό μετάδοση λέξη χρη-σιμοποιώντας τον κώδικα Hamming.

13. Έστω ότι θέλουμε να μεταδώσουμε πληροφορία αποτελούμενη από 8 bit, έστω το δυαδικό α-ριθμό 10110011, χρησιμοποιώντας άρτια ισοτιμία. Να προσδιορίσετε την υπό μετάδοση λέξη, χρησιμοποιώντας τον κώδικα Hamming.

14. **Έστω ότι θέλουμε να μεταδώσουμε πληροφορία αποτελούμενη από 8 bit, έστω το δυαδικό αριθμό 10101101. Να προσδιορίσετε την υπό μετάδοση λέξη, χρησιμοποιώντας τον κώδικα Hamming, είτε με άρτια είτε με περιττή ισοτιμία.**

15. **Να αντιστοιχίσετε τα ψηφία του δεκαδικού συστήματος στο δυπενταδικό κώδικα και στον κώδικα 2 από 5.**

16. **Αν η καμινάδα ενός σπιτιού λειτουργεί κανονικά, το σπίτι είναι κρύο και η ενδεικτική λυχνία είναι αναμμένη, να γράψετε τη λογική συνάρτηση που επιτρέπει να ανοίξει η κεντρική βαλβίδα καυσίμου, για να λειτουργήσει η θέρμανση. Δίνονται οι λογικές εκφράσεις b= καμινάδα σε κανονική λειτουργία, c= το σπίτι είναι κρύο, p= η ενδεικτική λυχνία είναι αναμμένη, v= άνοιξε την κεντρική βαλβίδα καυσίμουυ**

17. **Να συμπληρώσετε στον παρακάτω πίνακα στα κενά τον αριθμό που αντιστοιχεί στον κάθε κώδικα.**

Δεκαδικό	BCD	Κώδικας 84-2-1	Κώδικας 2421	Κώδικας 5421	Κώδικας 5311	Κώδικας XS3	Κώδικας Gray
					1001		
	0111						
			1111				
						1001	
				0010			

Αναφορές-Βιβλιογραφία

Κοσσίδας Α.Θ. (1996). *Σχεδίαση Ψηφιακών Κυκλωμάτων*, Εκδόσεις Μπένος

Φραγκάκης Γ. (1975). *Λογικά Κυκλώματα*, Αθήνα

Balabanian N., Carlson B. (2007). *Digital Logic Design Principles*, John Wiley

Blahut R. E. (2003). *Algebraic Codes for Data Transmission*, Cambridge University Press

Givone D, (2003). *Digital Principles and Des*ign,. Mc Graw Hill

Holdsworth Brian, Woods Clive (2002). *Digital Logic Design,* 4th Edition*,* Newnes

IEEE, (2008). *IEEE-754 Standard for Floating Point Arithmetic*

Klove T., Korzhik V. (1995). *Error detecting Codes*, Kluwer Academic Publishers, Boston

Lee S., (2006). *Advanced Digital Logic Design*, Thomson-Nelson

Mano M., Ciletti M., (2014). *Ψηφιακή Σχεδίαση*, 5η έκδοση, Παπασωτηρίου

Nashelsky Louis (1994). *Introduction to Digital Technology*, 4th Ed., Prentice Hall

Nelson V., Nagle H., Carroll B., Irwin J. (1995). *Digital Logic Circuit Analysis and Design*, Prentice-Hall

Predko Myke (2005). *Digital Electronics Demystified*, Mc Graw Hill

Pritchard N. (2015). *Fundamentals of Digital Electronics*, CreateSpace Independent Publishing Platform

Rajaraman V., Adabala N. (2015). *Fundamental of Computers*, 6th, PHI Learning

Shanmugam K. (1979*).* *Digital and Analog Communication Systems*, John Wiley ans Sons

Tocci R. J., Widmer N. S, and Moss Gr. L. (2010). *Digital Systems: Principles and Applications*, 11th, Boston, Addison-Wesley

Wakerly J. (2006). *Digital Design Principles and Practices*, 4/e, Prentice Hall

Διαδικτυακοί τόποι

Απόδοση του κώδικα Hamming, Ανακτήθηκε 10/3/2015 από το http://cnx.org/contents/uVzho1JX@27/Error-Correcting-Codes-Hamming

GAO Report (1992). GAO/IMTEC 92-26 Report http://www.fas.org/spp/starwars/gao/im92026.htm Patriot Missile Defense -Software Problem Led to System Failure at Dhahran, Saudi Arabia, USA General Accounting Office, B-247094.

Kahan W. (2008). http://http.cs.berkeley.edu/ ~demmel/cs267/lecture21/lecture21.html

Tervo R. (2014). http://www.ece.unb.ca/tervo/ee4253/hamming.shtml