

Μάθημα: Εισαγωγή στην Πληροφορική και την Ανοικτότητα

Κεφάλαιο 1.3 - Αναπαράσταση δεδομένων

Περιεχόμενα

1.3.1 Τύποι δεδομένων

1.3.2 Ακέραιοι αριθμοί (integers)

Δυαδική πρόσθεση

Δυαδική αφαίρεση

Υπερχείλιση

1.3.3 Προσημασμένοι ακέραιοι

Προσημασμένο μέτρο (signed magnitude)

Συμπλήρωμα ως προς 1 (1's complement)

Συμπλήρωμα ως προς 2 (2's complement)

1.3.4 Πραγματικοί αριθμοί (reals)

Μετατροπή πραγματικών αριθμών από το δεκαδικό στο δυαδικό σύστημα

Μετατροπή πραγματικών αριθμών από το δυαδικό στο δεκαδικό σύστημα

Μέθοδος σταθερής υποδιαστολής

Μέθοδος κινητής υποδιαστολής

1.3.5 Κωδικοποίηση αριθμητικών και αλφαριθμητικών δεδομένων

Κώδικας αριθμών BCD (Binary Coded Decimal)

Κώδικας αριθμών Gray

Κώδικες χαρακτήρων

Κώδικας χαρακτήρων ASCII

Κώδικας UNICODE

1.3.6 Κωδικοποίηση ήχου

Δειγματοληψία

Κβάντιση

Κωδικοποίηση

Συμπίεση

Πρότυπα κωδικοποίησης ήχου

1.3.7 Ψηφιακή παράσταση εικόνας

Ψηφιογραφικές εικόνες (bitmap)

Χρωματικό μοντέλο RGB

Χρωματικό μοντέλο CMY ή CMYK

Χρωματικό μοντέλο HSV

Διανυσματικές εικόνες

Τύποι αρχείων εικόνας - Συμπίεση

1.3.8 Ψηφιακή παράσταση βίντεο



Στόχοι

Μετά την ολοκλήρωση του κεφαλαίου οι σπουδαστές/τριες θα μπορούν να :

- ✓ περιγράφουν τους διαφορετικούς τύπους δεδομένων που χρησιμοποιούνται στους υπολογιστές
- ✓ περιγράφουν τους τρόπους αποθήκευσης των ακεραίων αριθμών στον υπολογιστή
- ✓ περιγράφουν τους τρόπους αποθήκευσης των πραγματικών αριθμών στον υπολογιστή
- ✓ κατανοούν τους κώδικες αποθήκευσης κειμένου στον υπολογιστή
- ✓ περιγράφουν τους τρόπους αποθήκευσης ήχου, εικόνων και βίντεο στον υπολογιστή

1.3.1 Τύποι δεδομένων

Τύπος Δεδομένων (Data Type) είναι ένα σύνολο δεδομένων με κοινά χαρακτηριστικά όπως οι πιθανές τιμές που μπορεί να έχουν (π.χ. ακέραιος αριθμός, πραγματικός αριθμός, χαρακτήρας, λογικός τύπος κ.λ.π.), οι πράξεις που μπορούν να εκτελεστούν στις τιμές τους (πρόσθεση, αφαίρεση, αληθής, ψευδής κ.λ.π.)

Για παράδειγμα ο τύπος δεδομένων των ακεραίων, δηλαδή οι αριθμοί ... -7, -5, -1, 0, 1, 2, 3 ... μαζί με τις πράξεις που ορίζονται πάνω σ' αυτούς (πρόσθεση, αφαίρεση, πολλαπλασιασμός, ακέραιο πηλίκο, ακέραιο υπόλοιπο, ύψωση σε δύναμη....)

Όλοι οι τύποι δεδομένων προκειμένου ν' αποθηκευτούν στον υπολογιστή πρέπει να μετατραπούν σε μια ακολουθία από δυαδικά ψηφία (bits), όπως είδαμε στο προηγούμενο κεφάλαιο, και να επανέλθουν στην αρχική τους μορφή όταν ανακτηθούν.

1.3.2 Ακέραιοι αριθμοί (integers)

Είναι ο τύπος δεδομένων που αποτελείται από αριθμούς χωρίς υποδιαστολή και μπορεί να είναι θετικός ή αρνητικός αριθμός ή μηδέν. Οι αριθμοί αυτοί αποθηκεύονται στον υπολογιστή σε δυαδική μορφή και μπορούμε να εκτελέσουμε σε αυτούς όλες τις πράξεις στο δυαδικό σύστημα αρίθμησης.

Δυαδική πρόσθεση

Γίνεται όπως και με την πρόσθεση στο δεκαδικό σύστημα. Η διαφορά είναι ότι έχουμε κρατούμενο όταν συμπληρωθεί μία δυάδα και όχι όταν συμπληρωθεί μία δεκάδα. Υπάρχουν όμως οι εξής κανόνες:

$$0+1=1$$

$$1+0=1$$

$$0+0=0$$

$$1+1=0 \text{ και κρατούμενο σε μεταφορά } 1$$

Παραδείγματα πρόσθεσης δυαδικών αριθμών

1 1 κρατούμενο

$$1\ 0\ 0\ 1\ 0 \qquad 18$$

$$0\ 0\ 1\ 1\ 0\ + \qquad 6+$$

$$1\ 1\ 0\ 0\ 0 \text{ αποτέλεσμα} \qquad 24$$

1 1 1 1 1 κρατούμενο

$$1\ 1\ 1\ 0\ 1$$

$$1\ 1\ 0\ 1\ 1\ +$$

$$1\ 1\ 1\ 0\ 0\ 0 \text{ αποτέλεσμα}$$

Στο προηγούμενο παράδειγμα παρατηρούμε ότι στην περίπτωση πρόσθεσης τριών μονάδων, το αποτέλεσμα λαμβάνεται προσθέτοντας τις δύο πρώτες και έπειτα προσθέτοντας το αποτέλεσμα που προκύπτει με την τρίτη.

Δυαδική αφαίρεση

Η αφαίρεση των δυαδικών αριθμών ακολουθεί τους παρακάτω κανόνες:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$10 - 1 = 1$ ← με δανεισμό από την ανώτερη τάξη.

Ο τέταρτος κανόνας μπορεί να γίνει κατανοητός αν σκεφτεί κανείς τη δεκαδική αφαίρεση. Αν ο μειωτέος είναι μικρότερος από τον αφαιρετέο, δανειζόμαστε μία δεκάδα και εκτελούμε την αφαίρεση. Όμοια, στο δυαδικό σύστημα, για να αφαιρέσουμε 1 από 0, δανειζόμαστε μία δυάδα (θυμηθείτε ότι η βάση του συστήματος είναι το 2) και αφαιρούμε 1 από 2. Το αποτέλεσμα είναι 1 και μία δανειζόμενη μονάδα μεταφέρεται στην επόμενη θέση αριστερά.

Για δυαδικό αριθμό με δύο ή περισσότερα ψηφία, η αφαίρεση πραγματοποιείται στήλη ανά στήλη όπως στην αφαίρεση δεκαδικών ψηφίων. Επίσης, μερικές φορές κάποιος πρέπει να δανειστεί από την επόμενη ανώτερη στήλη.

Παραδείγματα αφαίρεσης δυαδικών αριθμών

0 10 κρατούμενο

1 1 0 0

1 0 1 0 -

0 0 1 0 αποτέλεσμα

Η παραπάνω αφαίρεση πραγματοποιείται με τα ακόλουθα βήματα:

$$0 - 0 = 0$$

Για το $0 - 1 = 1$, λαμβάνοντας δανεισμό από την ανώτερη στήλη το 0 γίνεται 10 και στη συνέχεια $10 - 1 = 1$

Για το $1 - 0$, δεδομένου ότι έχει ήδη δοθεί το 1, γίνεται $0 - 0 = 0$

$$1 - 1 = 0$$

10 10 κρατούμενο

1 1 0 1

1 1 0 -

1 1 1 αποτέλεσμα

Στην αφαίρεση δύο θετικών αριθμών το αποτέλεσμα θα μπορούσε να ήταν αρνητικό αν ο μειωτέος ήταν μικρότερος από τον αφαιρετέο. Συνεπώς υπάρχει η ανάγκη ορισμού θετικών και αρνητικών δυαδικών αριθμών.

Υπερχείλιση

Έχουμε συνηθίσει στην καθημερινή μας ζωή να κάνουμε πράξεις με δεκαδικούς αριθμούς χωρίς ποτέ να σκεφτόμαστε τον αριθμό των ψηφίων που μας είναι διαθέσιμα. Ωστόσο σε ένα υπολογιστικό σύστημα ο αριθμός των διαθέσιμων δυαδικών ψηφίων που καταλαμβάνει ένας αριθμός είναι πεπερασμένος.

Ας υποθέσουμε ότι στο δεκαδικό σύστημα υπήρχε περιορισμός των παραστάσεων των αριθμών στο ένα δεκαδικό ψηφίο. Διαπιστώνουμε τότε ότι η πράξη $8+6=14$ δεν μπορεί να εκτελεστεί γιατί το αποτέλεσμα χρειάζεται δύο ψηφία για να παρασταθεί.

Εάν ο μέγιστος αριθμός ψηφίων είναι n και το αποτέλεσμα μιας πράξης χρειάζεται $n+1$ ψηφία για να παρασταθεί τότε έχουμε υπερχείλιση.

Έστω ότι θέλουμε να προσθέσουμε δύο δυαδικούς αριθμούς των τεσσάρων bits:

$$\begin{array}{r} +14_{10} \qquad \qquad 1\ 1\ 1\ 0_2 \\ +\ 3_{10} \qquad \qquad 0\ 0\ 1\ 1_2 \\ \hline +17_{10} \qquad \qquad 1\ 0\ 0\ 0\ 1_2 \end{array}$$

Το αποτέλεσμα $(10001)_2$ έχει μήκος πέντε bits αν και οι είσοδοι μας είχαν μόνο 4 bits. Παρόλο που η απάντηση είναι σωστή δεν μπορούμε να χρησιμοποιήσουμε το αποτέλεσμα αυτό. Άρα, όταν το κρατούμενο είναι 1, τότε έχουμε υπερχείλιση στην πρόσθεση. Σε αυτή την περίπτωση το bit που βρίσκεται στην 5η θέση αγνοείται και έτσι τα υπόλοιπα bits αναπαριστούν λάθος αποτέλεσμα, δηλαδή 0001 αντί του 10001.

Συνεπώς θα πρέπει πάντα να λαμβάνουμε υπόψη το εύρος των δυνατών παραστάσεων και να ελέγχουμε για τυχόν καταστάσεις υπερχείλισης.

Το πρόβλημα της υπερχείλισης παρατηρείται στην πρόσθεση όταν το κρατούμενο είναι 1. Στην αφαίρεση δεν υπάρχει υπερχείλιση γιατί το αποτέλεσμα είναι πάντα ίσο ή μικρότερο από τον πιο μεγάλο των 2 αριθμών.



Δραστηριότητες – Πράξεις στο δυαδικό σύστημα αρίθμησης

1. Να προσθέσετε τους παρακάτω δυαδικούς αριθμούς:

$$\begin{array}{r} 10010 \\ + 1100 \\ \hline \end{array} \qquad \begin{array}{r} 1011101 \\ + 1000000 \\ \hline \end{array} \qquad \begin{array}{r} 10011 \\ + 1111101 \\ \hline \end{array}$$

$$\begin{array}{r} 10011001 \\ + 100111 \\ \hline \end{array} \qquad \begin{array}{r} 11000011 \\ + 101111 \\ \hline \end{array} \qquad \begin{array}{r} 1001100 \\ + 1100101 \\ \hline \end{array}$$

2. Να αφαιρέσετε τους παρακάτω δυαδικούς αριθμούς:

$$\begin{array}{l} 1011_2 - 1001_2 \\ 1100_2 - 0110_2 \\ 1101_2 - 1011_2 \\ 11001010_2 - 10011010_2 \\ 11100001_2 - 10011101_2 \end{array}$$

1.3.3 Προσημασμένοι ακέραιοι

Μέχρι τώρα, δεν αναφέραμε το πρόσημο των αριθμών. Στους θετικούς αριθμούς, το πρόσημο συνήθως παραλείπεται όμως οι αρνητικοί αριθμοί πρέπει με κάποιον τρόπο να υποδηλώνουν το πρόσημο τους.

Υπάρχουν τρεις διαφορετικοί μέθοδοι για την παράσταση των προσημασμένων αριθμών, οι εξής:

- (i) Προσημασμένο μέτρο (signed magnitude),
- (ii) Συμπλήρωμα ως προς 1 (1's complement),
- (iii) Συμπλήρωμα ως προς 2 (2's complement).

Η μορφή συμπληρώματος ως προς 2 είναι εκείνη που χρησιμοποιείται στους υπολογιστές γιατί απλοποιεί την εκτέλεση των αριθμητικών πράξεων και για τους θετικούς και για τους αρνητικούς αριθμούς.

Προσημασμένο μέτρο (signed magnitude)

Ο πιο συνηθισμένος τρόπος συμβολισμού ενός αρνητικού αριθμού, στο δεκαδικό σύστημα, είναι με την προσθήκη του συμβόλου "μείον" στην αρχή του αριθμού, π.χ. -345. Ο συμβολισμός αυτός χρησιμοποιεί τα ψηφία για να σχηματίσουν το μέτρο ή την απόλυτη τιμή (value) του αριθμού, και ένα ξεχωριστό σύμβολο για να δηλώσει το πρόσημο του (sign). Αυτή η μέθοδος αναπαράστασης των αριθμών ονομάζεται μέθοδος του προσημασμένου μέτρου. Επειδή το πρόσημο μπορεί να πάρει μόνο δύο δυνατές τιμές (+ και -), μπορεί να παρασταθεί από ένα μοναδικό bit. Οι υπολογιστές χρησιμοποιούν το πρώτο bit από αριστερά (most significant bit-MSB) των δυαδικών αριθμών για την δήλωση του πρόσημου, και τα υπόλοιπα bits υποδηλώνουν το μέγεθος του αριθμού. Αν το MSB είναι μηδέν, ο αριθμός είναι θετικός και αν το MSB είναι μονάδα, ο αριθμός είναι αρνητικός. Σύμφωνα με αυτόν τον τρόπο παράστασης ο αριθμός $(+5)_{10}$ θα γράφεται στο δυαδικό ως $(101)_2$ με ακρίβεια τριών bits ή $(0101)_2$ με ακρίβεια τεσσάρων bits, ενώ ο αριθμός $(-5)_{10}$ θα γράφεται ως $(1101)_2$ με ακρίβεια τεσσάρων bits.

Αυτή η μέθοδος παράστασης είναι αρκετά βολική για εμάς, αφού οι αριθμοί, ανεξάρτητα από το πρόσημο τους, γράφονται με τον ίδιο τρόπο. Είναι όμως δύσχρηστη για τον υπολογιστή γιατί γίνεται σπατάλη δυαδικών ψηφίων αφού περιορίζει στο μισό το μέγεθος του αριθμού. Ένας μη προσημασμένος αριθμός των 8 bits παίρνει τιμές από το 0 έως το 255. Αν ο ίδιος αριθμός είναι προσημασμένος, τότε η μεγαλύτερη δυνατή τιμή του είναι το +127. Ένας δεύτερος λόγος που δεν χρησιμοποιείται αυτή η μέθοδος στους υπολογιστές είναι ότι η πρόσθεση δύο θετικών αριθμών ή δύο αρνητικών αριθμών είναι απλή, όμως όταν τα πρόσημα των δύο αριθμών δεν είναι τα ίδια τότε η πράξη γίνεται πολύπλοκη, επειδή τα υπολογιστικά συστήματα δε διαθέτουν ηλεκτρονικά κυκλώματα που να εκτελούν την πράξη της αφαίρεσης. Αυτό θα μπορούσε να λυθεί αν, σύμφωνα με τους νόμους της αριθμητικής, η αφαίρεση γινόταν μέσω της πρόσθεσης, δηλαδή $9-4=9+(-4)$. Όμως αν χρησιμοποιήσουμε τη μέθοδο του προσημασμένου μέτρου, το αποτέλεσμα που προκύπτει είναι εσφαλμένο όπως φαίνεται παρακάτω:

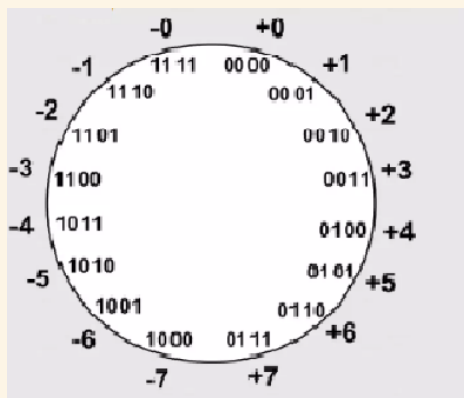
$$\begin{array}{r} 9 = 00001001 \\ + (-4) = 10000100 \\ \hline 10001101 = -13 \end{array}$$

Πρέπει να βρούμε εκείνη τη μέθοδο παράστασης των αρνητικών αριθμών, ώστε να καταλήγουμε στο επιθυμητό αποτέλεσμα.

Συμπλήρωμα ως προς 1 (1's complement)

Μια διαφορετική μέθοδος παράστασης είναι το συμπλήρωμα ως προς 1, όπου αντιστρέφουμε όλα τα ψηφία ενός δυαδικού αριθμού αλλάζοντας το 0 σε 1 ή το 1 σε 0 αντίστοιχα. Όπως και στην περίπτωση των αριθμών προσημασμένου μέτρου, το πιο σημαντικό ψηφίο είναι το ψηφίο του πρόσημου. Έτσι το 0 αντιστοιχεί σε θετικούς, και το 1 σε αρνητικούς αριθμούς.

Για παράδειγμα, αν αντιστρέψουμε τα ψηφία του αριθμού 01101010_2 (106_{10}), προκύπτει ο αριθμός 10010101_2 (-106_{10}).



Εικόνα 1.3.3. Συμπλήρωμα ως προς 1 αριθμών μήκους 4 bit

Στην παραπάνω εικόνα φαίνεται η αναπαράσταση του συμπληρώματος ως προς 1 των αριθμών των 4 bit. Όλοι οι αρνητικοί αριθμοί έχουν 1 στο πρόσημο τους, ενώ οι θετικοί αριθμοί έχουν 0. Παρατηρούμε όμως ότι το μηδέν έχει δυο αναπαραστάσεις, την 0000 και την 1111. Λόγω της ύπαρξης δύο αναπαραστάσεων του μηδενός, αυτή η μέθοδος δε χρησιμοποιείται στους υπολογιστές.

Συμπλήρωμα ως προς 2 (2's complement)

Το συμπλήρωμα ως προς 2 (2s' complement) ενός δυαδικού αριθμού ορίζεται ως η τιμή που παίρνουμε όταν αντιστρέφουμε όλα τα ψηφία (bits) του δυαδικού αριθμού (αλλάζοντας τα 0 σε 1 και το αντίστροφο - το 0 είναι το συμπλήρωμα του 1 και το αντίθετο) και στην συνέχεια προσθέτουμε μια μονάδα.

Ισχύει ότι αν ο μειωτέος είναι \geq αφαιρετέου τότε παραλείπουμε το τελικό κρατούμενο στο αποτέλεσμα.

Ισχύει ότι αν ο μειωτέος είναι $<$ αφαιρετέου δεν προκύπτει κρατούμενο.

Αυτή η μέθοδος εξαλείφει το πρόβλημα της πολλαπλής παράστασης του μηδενός. Αφού, στην αριθμητική του συμπληρώματος ως προς 2 το κρατούμενο αγνοείται, το συμπλήρωμα του 00000000₂ ως προς 2 είναι πάλι το 00000000₂. Επομένως, η παράσταση του μηδενός είναι μοναδική.

Για παράδειγμα, η δυαδική μορφή του 4 είναι το 00000100₂.

Η αντιστροφή των ψηφίων του δίνει τον 11111011₂.

Το συμπλήρωμα του δύο είναι ο δυαδικός αριθμός, που προκύπτει από την πρόσθεση 11111011₂ + 1 = 11111100₂ και ισούται με τον αρνητικό δεκαδικό -4₁₀.

Η λύση του προβλήματος της αφαίρεσης 9₁₀-4₁₀ με τη μέθοδο αυτή είναι:

$$\begin{array}{r} 9 = 00001001 \\ + (-4) = 11111100 \\ \hline 10000101 = 5 \end{array}$$

Βλέπουμε ότι το κρατούμενο είναι μονάδα και τα υπόλοιπα 8 bits δίνουν τιμή ίση με 5. Ο μειωτέος είναι \geq αφαιρετέου (9 > 4) οπότε το κρατούμενο αγνοείται.

Παραδείγματα

1. Χρησιμοποιώντας το συμπλήρωμα ως προς 2, κάνετε την αφαίρεση: 1010100₂ - 1000011₂

1010100 (M) +

0111101 (συμπλήρωμα ως προς 2 του 1000011) (A)

10010001 (άθροισμα)

ισχύει ότι $M \geq A$ παραλείπουμε το τελικό κρατούμενο οπότε το αποτέλεσμα είναι: 0010001₂

2. Χρησιμοποιώντας το συμπλήρωμα ως προς 2, κάνετε την αφαίρεση: 1000011₂ - 1010100₂

1000011 (M) +

0101100 (συμπλήρωμα ως προς 2 του 1010100) (A)

1101111 (άθροισμα)

ισχύει ότι $M < A$ δεν προκύπτει κρατούμενο και το αποτέλεσμα είναι: 1101111_2

3. Προσθέσεις με τη μέθοδο Συμπληρώματος ως προς 2

Δεκαδικό σύστημα	Επίλυση στο Δυαδικό με χρήση Συμπληρώματος ως προς 2
3	0011
+ 3	+ 0011
	0110
7	0111
+ -5	+ 1011
	10010
-3	1101
+ -2	+ 1110
	11011

αγνοείται

Στον παρακάτω πίνακα φαίνεται ο υπολογισμός της αξίας των δυαδικών αριθμών με 8 bits σε μορφή συμπληρώματος ως προς 2.

-128	64	32	16	8	4	2	1	=		
0	1	1	1	1	1	1	1	=	$0+64+32+16+8+4+2+1$	$=+127=2^7-1$
....			
0	1	0	0	0	1	1	1	=	$0+64+ 0+ 0+0+4+2+1$	$= +71$
....			
0	0	0	1	0	1	1	1	=	$0+ 0+ 0+16+0+4+2+1$	$= +23$
....			
0	0	0	0	0	0	1	0	=	$0+ 0+ 0+ 0+0+0+2+0$	$= +2$
0	0	0	0	0	0	0	1	=	$0+64+ 0+ 0+0+0+0+0$	$= +1$
0	0	0	0	0	0	0	0	=	$0+ 0+ 0+ 0+0+0+0+0$	$= 0$
1	1	1	1	1	1	1	1	=	$-128+64+32+16+8+4+2+1$	$= -1$
1	1	1	1	1	1	1	0	=	$-128+64+32+16+8+4+2+0$	$= -2$
1	1	1	1	1	1	0	1	=	$-128+64+32+16+8+4+0+1$	$= -3$
....			
1	1	1	1	0	0	0	0	=	$-128+64+32+16+0+0+0+0$	$= -16$
....			
1	0	0	1	0	0	0	0	=	$-128+ 0+ 0+16+0+0+0+0$	$= -112$
....			
1	0	0	0	0	0	0	0	=	$-128+ 0+ 0+ 0+0+0+0+0$	$= -2^7 = -128$



Δραστηριότητα – Μετατροπή προσημασμένου αριθμού στο δεκαδικό σύστημα αρίθμησης

Να μετατρέψετε τον δυαδικό αριθμό 110101 σε δεκαδικό, θεωρώντας ότι ο αριθμός βρίσκεται σε μια από τις παρακάτω μορφές: (με τη χρήση 6 δυαδικών ψηφίων).

1. Μορφή προσημασμένου μέτρου
2. Συμπλήρωμα ως προς 1
3. Συμπλήρωμα ως προς 2

Λύση:

Καθώς το bit πρόσημου είναι 1, έχουμε αρνητικό αριθμό. Ο πιο απλός τρόπος να βρούμε το μέγεθός του είναι να τον μετατρέψουμε σε θετικό αριθμό.

1. Μορφή προσημασμένου μέτρου

Ο αρχικός δυαδικός αριθμός, 110101, αφαιρώντας το πρώτο δυαδικό ψηφίο που παριστάνει το πρόσημο δίνει τον 10101, ο οποίος είναι ο δεκαδικός +21. Άρα ο αριθμός 110101 αντιπροσωπεύει το -21_{10} .

2. Συμπλήρωμα ως προς 1

Η μετατροπή του 110101 σε συμπλήρωμα ως προς ένα δίνει 001010 ο οποίος είναι ο δεκαδικός $+10_{10}$, έτσι ο αρχικός αριθμός είναι ο -10_{10} .

3. Συμπλήρωμα ως προς 2

Η μετατροπή του 110101 σε συμπλήρωμα ως προς δύο δίνει 001011 ο οποίος είναι ο δεκαδικός $+11_{10}$, έτσι ο αρχικός αριθμός είναι το -11_{10} .

Αυτό που συμπεραίνουμε είναι, πως ο δυαδικός αριθμός έχει διαφορετικό νόημα ανάλογα με τον τρόπο αναπαράστασής του.

Κανόνες υπερχείλισης στην παράσταση συμπληρώματος ως προς δύο:

Αν προστεθούν δυο αριθμοί με το ίδιο πρόσημο, τότε προκύπτει υπερχείλιση αν και μόνο αν το αποτέλεσμα έχει το αντίθετο πρόσημο.

Οι υπολογιστές διαθέτουν τους κατάλληλους μηχανισμούς για να εντοπίζουν τις περιπτώσεις όπου σε μία αριθμητική πράξη προκαλείται υπερχείλιση.

Ο Ευρωπαϊκός πύραυλος Ariane 5, κόστισε 7 δισεκατομμύρια δολάρια και εκτοξεύτηκε στις 4 Ιουνίου 1996, απέκλινε από την πορεία του 40 δευτερόλεπτα μετά από την εκτόξευση του, κόπηκε στα δύο και εξερράγη.

Το ατύχημα προκλήθηκε όταν ο υπολογιστής που έλεγχε τον πύραυλο υπερχείλισε το εύρος (16 bit) των προσημασμένων τιμών του και κατέρρευσε. Το λογισμικό απέτυχε γιατί δεν περιλάμβανε διαχειριστή εξαιρέσεων.

1.3.4 Πραγματικοί αριθμοί (reals)

Είναι ο τύπος δεδομένων που αποτελείται από αριθμούς που έχουν ακέραιο και κλασματικό μέρος. Για παράδειγμα ο αριθμός 24,3 είναι πραγματικός αριθμός, με ακέραιο μέρος το 24 και κλασματικό το $3/10$.

Μετατροπή πραγματικών αριθμών από το δεκαδικό στο δυαδικό σύστημα

Η μετατροπή δεκαδικών αριθμών που έχουν ακέραιο και κλασματικό μέρος γίνεται χωριστά για το ακέραιο και το κλασματικό μέρος και στο τέλος τα “ενώνουμε” με το κόμμα μεταξύ τους. Το ακέραιο μέρος το διαχειριζόμαστε όπως μάθαμε πριν.

Βήμα 1: Πολλαπλασιάζουμε το κλασματικό μέρος με το 2.

Βήμα 2: Κρατάμε το ακέραιο μέρος του αποτελέσματος.

Βήμα 3: Πολλαπλασιάζουμε με το 2, μόνο το κλασματικό μέρος του προηγούμενου αποτελέσματος.

Βήμα 4: Επαναλαμβάνουμε τα βήματα 2 και 3 μέχρι να βρούμε κλασματικό μέρος = 0 ή να πετύχουμε την επιθυμητή ακρίβεια (συνήθως τα 8 bit).

Παράδειγμα 1: $(49.425)_{10} = (?)_2$ με ακρίβεια 8 bit.

Αριθμός	Αποτέλεσμα	Bit
$0.425 * 2$	0.85	0
$0.85 * 2$	1.7	1
$0.7 * 2$	1.4	1
$0.4 * 2$	0.8	0
$0.8 * 2$	1.6	1
$0.6 * 2$	1.2	1
$0.2 * 2$	0.4	0
$0.4 * 2$	0.8	0

Αποτέλεσμα: η στήλη "Bit", σε κανονική σειρά
 Αποτέλεσμα: $(49.425)_{10} = (110001.01101100)_2$

Παράδειγμα 2: $(24.25)_{10} = (?)_2$ με κλασματικό μέρος = 0

Ακέραιο μέρος			Κλασματικό μέρος		
Αριθμός	Πηλίκο	Υπόλοιπο	Αριθμός	Γινόμενο	Bit
$24 / 2$	12	0	$0.25 * 2$	0.50	0
$12 / 2$	6	0	$0.5 * 2$	1	1
$6 / 2$	3	0			
$3 / 2$	1	1			
$1 / 2$	0	1			

Τέλος Διαδικασίας:
 διότι: **Γινόμενο=1**

Αποτέλεσμα: $(24.25)_{10} = (11000.01)_2$

Μετατροπή πραγματικών αριθμών από το δυαδικό στο δεκαδικό σύστημα

Παράδειγμα 1: $(11001,101)_2$

1	1	0	0	1	1	0	1
2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	+ 8	+ 0	+ 0	+ 1=25	0,5	+ 0	+ 0,125= 0,625

Αποτέλεσμα: $(11001,101)_2 = (25,625)_{10}$

Για την παράστασή των πραγματικών αριθμών στο δυαδικό σύστημα χρειαζόμαστε:

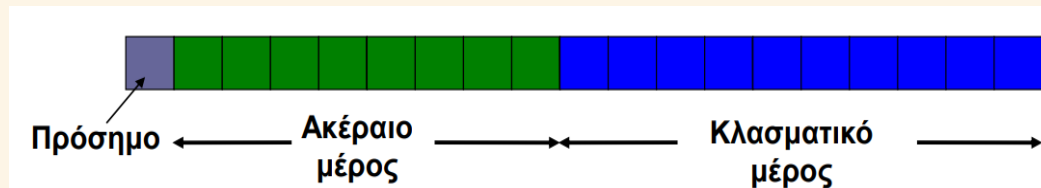
- το πρόσημο
- τα ψηφία του αριθμού
- τη θέση της υποδιαστολής

Υπάρχουν δύο μέθοδοι παράστασης των πραγματικών αριθμών

- Μέθοδος σταθερής υποδιαστολής
- Μέθοδος κινητής υποδιαστολής

Μέθοδος σταθερής υποδιαστολής

Για την παράσταση των πραγματικών αριθμών με μήκος n bits μοιράζουμε τα ψηφία του αριθμού στο ακέραιο και στο κλασματικό μέρος, δηλαδή τα n_1 bits χρησιμοποιούνται για το ακέραιο μέρος του αριθμού και τα υπόλοιπα n_2 bits για το κλασματικό μέρος του ($n_1 + n_2 = n$). Το αριστερότερο ψηφίο (MSB) διατίθεται για το πρόσημο του αριθμού.



Σε αυτή τη μέθοδο η θέση της υποδιαστολής είναι πάντα στην ίδια θέση για όλους τους αριθμούς. Αυτό όμως μας δημιουργεί πρόβλημα στην ακρίβεια και στο εύρος της αναπαράστασης των αριθμών. Για να καταλάβουμε το πρόβλημα ας θεωρήσουμε έναν υπολογιστή που διαθέτει 8 bits για την αποθήκευση των πραγματικών αριθμών. Το πλέον αριστερό bit διατίθεται για το πρόσημο και η υποδιαστολή βρίσκεται δεξιά του 6ου bit, οπότε τα 6 bits χρησιμοποιούνται για το ακέραιο μέρος και 2 bits για το κλασματικό μέρος. Η παράσταση των αρνητικών αριθμών γίνεται με το συμπλήρωμά ως προς 2. Ο μεγαλύτερος θετικός αριθμός που μπορούμε να αποθηκεύσουμε είναι ο $01111,11_2$ και είναι ο $31,75_{10}$. Ο μικρότερος αρνητικός αριθμός που μπορούμε να αποθηκεύσουμε είναι ο $10000,00_2$ και είναι ο -32_{10} . Ο αριθμός $31,875_{10}$ αναπαρίσταται στο δυαδικό σαν $01111,111$ και επειδή έχουμε μόνο 2 bits για το κλασματικό μέρος, αποθηκεύεται σαν $01111,11_2$ ($31,75_{10}$). Με τον περιορισμό των δύο ψηφίων στο κλασματικό μέρος έχουμε απώλεια της ακρίβειας του αριθμού. Αν θεωρούσαμε ότι η υποδιαστολή ήταν δεξιά του 5ου bit τότε θα είχαμε 5 bits για το ακέραιο μέρος και 3 bits για το κλασματικό. Ο αριθμός $31,875_{10}$ αναπαρίσταται στο δυαδικό σαν $01111,111$ και επειδή έχουμε μόνο 5 bits για το ακέραιο μέρος, αποθηκεύεται σαν $01111,111_2$ ($15,875_{10}$). Τώρα η απώλεια ακρίβειας είναι πάνω από 50%.

Συμπεραίνουμε ότι αν και η πρώτη περίπτωση μας δίνει καλύτερη ακρίβεια μας περιορίζει στο εύρος των αριθμών. Αν θέλαμε να έχουμε και μεγάλο εύρος αριθμών και μεγάλη ακρίβεια, τότε ο αριθμός των δυαδικών ψηφίων που θα χρησιμοποιούσαμε θα ήταν πολύ μεγάλος. Η αύξηση του συνολικού αριθμού των bit που διατίθενται για την παράσταση αριθμών αυξάνει το εύρος, ενώ η αύξηση του αριθμού των bits που διατίθενται για το κλασματικό μέρος αυξάνει την ακρίβεια. Αυτό μας οδηγεί σε τεράστιες ανάγκες μνήμης με συνέπεια σε πολύ υψηλό κόστους υπολογιστικά συστήματα.

Μέθοδος κινητής υποδιαστολής

Για να ξεπεραστούν τα μειονεκτήματα της μεθόδου σταθερής υποδιαστολής που είδαμε παραπάνω, έπρεπε να βρεθεί ένας τρόπος παράστασης των πολύ μεγάλων αριθμών όπως και των πολύ μικρών κλασματικών όπου η θέση της υποδιαστολής δεν θα ήταν σταθερή αλλά θα μετακινείται για να μπορεί να προσαρμόζεται στις υπολογιστικές ανάγκες. Ο τρόπος αυτός παράστασης ονομάζεται μέθοδος κινητής υποδιαστολής και είναι η μέθοδος που χρησιμοποιείται κυρίως στους υπολογιστές.

Στο δεκαδικό σύστημα, όταν θέλουμε να γράψουμε έναν ακέραιο αριθμό με πολλά μηδενικά στα δεξιά του ή ένα κλασματικό αριθμό με πολλά μηδενικά αμέσως μετά την υποδιαστολή χρησιμοποιούμε τη μορφή όπου κάθε αριθμός γράφεται ως το γινόμενο ενός αριθμού από το 1 ως το 10 υψωμένο στην κατάλληλη δύναμη του 10.

Για παράδειγμα:

Ο αριθμός 1.800.000 μπορεί να γραφεί σαν 180×10^4 ή 18×10^5 ή $1,8 \times 10^6$

Ο αριθμός 0,00045 μπορεί να γραφεί σαν 45×10^{-5} ή $4,5 \times 10^{-4}$ ή $0,45 \times 10^{-3}$

Όπως είδαμε ένας αριθμός μπορεί να έχει περισσότερες της μιας εκθετικές παραστάσεις στο ίδιο αριθμητικό σύστημα.

Το ίδιο συμβαίνει και στο δυαδικό σύστημα, κάθε αριθμός μπορεί να γραφεί με πολλές εκθετικές παραστάσεις.

Για παράδειγμα:

Ο αριθμός $101,011_2$ σε εκθετική μορφή μπορεί να γραφτεί:

$0,101011 \times 2^3$ ή

$1,01011 \times 2^2$ ή

$10,1011 \times 2^1$

Από τα παραπάνω συμπεραίνουμε ότι αν κάθε υπολογιστικό σύστημα υιοθετούσε διαφορετικό μοντέλο για τον τρόπο παράστασης του ίδιου αριθμού θα υπήρχε δυσκολία στη μεταφορά των δεδομένων μεταξύ τους. Συνήθως όταν υπάρχουν πολλές δυνατότητες επιλογών, τότε κάποια επιτροπή προτυποποίησης καλείται για να ορίσει κάποιο πρότυπο, το οποίο θα υιοθετηθεί από τους κατασκευαστές υπολογιστών για να λυθούν τα παραπάνω προβλήματα. Στην παράσταση κινητής υποδιαστολής τη λύση έδωσε το πρότυπο 754 της επιτροπής IEEE.

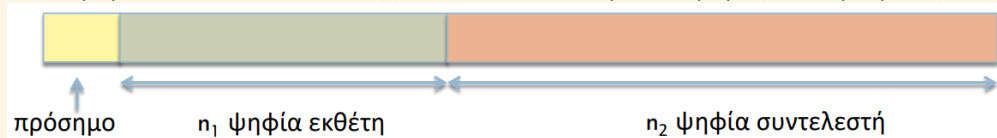
Το πρότυπο IEEE 754, καθορίζει δύο μορφές παράστασης των δυαδικών αριθμών κινητής υποδιαστολής, είτε με 32 (απλής ακρίβειας - single precision) είτε με 64 (διπλής ακρίβειας - double precision) δυαδικά ψηφία (bits).

Οι πραγματικοί αριθμοί αποτελούνται από τρία μέρη:

το 1ο μέρος είναι το πρόσημο που μπορεί να είναι είτε θετικό (τιμή 0) είτε αρνητικό (τιμή 1).

Το 2ο μέρος είναι ο εκθέτης που ορίζει πόσες θέσεις πρέπει να μετακινηθεί η υποδιαστολή προς τ' αριστερά ή τα δεξιά προκειμένου να σχηματιστεί ο αριθμός.

Το 3ο μέρος είναι ο συντελεστής και είναι το κλασματικό μέρος του αριθμού (mantissa).



Ένας δυαδικός αριθμός κινητής υποδιαστολής απλής ακρίβειας που υπακούει στο πρότυπο της I.E.E.E. διαθέτει ένα (1) bit για το πρόσημο, οκτώ (8) bits για τον εκθέτη και είκοσι τρία (23) bits για το συντελεστή.

Ένας δυαδικός αριθμός κινητής υποδιαστολής διπλής ακρίβειας που υπακούει στο πρότυπο της I.E.E.E. διαθέτει ένα (1) bit για το πρόσημο, έντεκα (11) bits για τον εκθέτη και πενήντα δύο (52) bits για το συντελεστή.

Το IEEE 754 ορίζει επίσης ότι, οι δυαδικοί αριθμοί θα πρέπει να διαμορφώνονται με τέτοιο τρόπο ώστε το ψηφίο που βρίσκεται αριστερά της υποδιαστολής (ακέραιο μέρος) πρέπει να είναι πάντοτε 1. Δηλαδή η μορφή των αριθμών που υποστηρίζει το πρότυπο είναι η $1,XXX..._2$. Η διαδικασία αυτή αναφέρεται ως κανονικοποίηση.

Αφού πάντα αριστερά της υποδιαστολής υπάρχει το 1, συμπεραίνουμε ότι δε χρειάζεται να αποθηκεύεται αυτό το ψηφίο αλλά μπορεί να υπονοείται.

Στους αριθμούς απλής ακρίβειας για την αποθήκευση των 8 bit του εκθέτη προστίθεται στην τιμή του εκθέτη το +127.

Στους αριθμούς διπλής ακρίβειας για την αποθήκευση των 11 bit του εκθέτη προστίθεται στην τιμή του εκθέτη το +1023.

Παραδείγματα

1. Έστω ότι θέλουμε να παραστήσουμε τον αριθμό $18,75_{10}$ σε δυαδικό με απλή ακρίβεια.

Μετατρέπουμε το ακέραιο μέρος του αριθμού το 18_{10} με τη μέθοδο των διαδοχικών διαιρέσεων σε δυαδικό και έχουμε

$$18_{10} = 10010_2.$$

Με τη μέθοδο των διαδοχικών πολλαπλασιασμών βρίσκουμε ότι το κλασματικό μέρος του αριθμού το $0,75_{10} = 0,11_2$.

Άρα $18,75_{10} = 10010,11_2$ και σε κανονικοποίηση $1,001011 \times 2^4$.

Μπορούμε τώρα να χωρίσουμε τον αριθμό στα 3 μέρη:

- Πρόσημο θετικό άρα το αριστερότερο δυαδικό ψηφίο είναι 0.
- Εκθέτης = $4_{10} + 127_{10} = 131_{10}$, δηλαδή 10000011_2 .
- Συντελεστής = $1,001011$. Το 1 πριν την υποδιαστολή δεν αποθηκεύεται οπότε ο συντελεστής σε 23 δυαδικά ψηφία θα έχει την παράσταση 0010110000000000000000_2 (προσθέτουμε μηδενικά στο τέλος ώστε να έχουμε συνολικά 23 bits).

Η συνολική παράσταση του αριθμού είναι : 0 1000011 0010110000000000000000₂

2. Έστω ότι θέλουμε να παραστήσουμε τον αριθμό $-161,875_{10}$ σε δυαδικό με απλή ακρίβεια.

Μετατρέπουμε το ακέραιο μέρος του αριθμού το 161_{10} με τη μέθοδο των διαδοχικών διαιρέσεων σε δυαδικό και έχουμε

$$161_{10} = 1010001_2.$$

Με τη μέθοδο των διαδοχικών πολλαπλασιασμών βρίσκουμε ότι το κλασματικό μέρος του αριθμού το $0,875_{10} = 0,111_2$.

Άρα $161,875_{10} = 1010001,111_2$ και σε κανονικοποίηση $1,010001111 \times 2^7$.

Μπορούμε τώρα να χωρίσουμε τον αριθμό στα 3 μέρη:

- Πρόσημο αρνητικό άρα το αριστερότερο δυαδικό ψηφίο είναι 1.
- Εκθέτης = $7_{10} + 127_{10} = 134_{10}$, δηλαδή 10000110_2 .
- Συντελεστής = $1,010001111$. Το 1 πριν την υποδιαστολή δεν αποθηκεύεται οπότε ο συντελεστής σε 23 δυαδικά ψηφία θα έχει την παράσταση 010001111000000000000_2 .

Η συνολική παράσταση του αριθμού είναι : 1 1000110 0100011110000000000000₂

3. Ας εξετάσουμε τώρα το αντίστροφο του παραπάνω παραδείγματος. Έστω ότι στη μνήμη του υπολογιστή υπάρχει αποθηκευμένη η παράσταση $1\ 1000110\ 0100011110000000000000_2$

- Πρόσημο 1 άρα ο αριθμός είναι αρνητικός.
- Εκθέτης = $1000110_2 = 134_{10}$. Αφαιρούμε $134 - 127 = 7$ (σύμφωνα με το πρότυπο ξέρουμε ότι στον αρχικό εκθέτη έχει προστεθεί το 127).
- Συντελεστής = 0100011110000000000000_2 .

Η αποκανονικοποίηση του μας δίνει $1,010001111 \times 2^7$

Άρα η υποδιαστολή πρέπει να μεταφερθεί 7 θέσεις δεξιά: $1010001,111_2$

Ο ζητούμενος δεκαδικός αριθμός είναι $-161,875_{10}$

1.3.5 Κωδικοποίηση αριθμητικών και αλφαριθμητικών δεδομένων

Οι υπολογιστές, όπως και όλα τα ψηφιακά συστήματα, δέχονται δεδομένα αλλά και τα αναπαράγουν με τη μορφή μιας σειράς δυαδικών ψηφίων (bits) 0 ή 1. Επομένως οποιαδήποτε πληροφορία θέλουμε να εισάγουμε στον υπολογιστή είμαστε υποχρεωμένοι να τη μετατρέπουμε (κωδικοποιούμε) σε δυαδική πληροφορία, δηλαδή, με τη μορφή μιας μοναδικής ακολουθίας δυαδικών ψηφίων 0 και 1.

Ένας κώδικας είναι ένα σύνολο κανόνων για τη μετατροπή ενός τμήματος πληροφορίας π.χ. έναν αριθμό, ένα γράμμα, μια λέξη, ή φράση σε μια συγκεκριμένη ομάδα συμβόλων.

Έτσι δημιουργήθηκε ο δυαδικός κώδικας επικοινωνίας του ανθρώπου με τη μηχανή.

Κατηγορίες δυαδικών κωδικών:

Κώδικες αριθμών (αναπαράσταση αριθμών στον υπολογιστή)

Κώδικες χαρακτήρων (αναπαράσταση χαρακτήρων στον υπολογιστή).

Κώδικας αριθμών BCD (Binary Coded Decimal)

Ο κώδικας BCD είναι ένας κώδικας παράστασης των 10 ψηφίων του δεκαδικού συστήματος με έναν δυαδικό κώδικα, όπου κάθε δεκαδικό ψηφίο αντιπροσωπεύεται από μια τετράδα δυαδικών ψηφίων. Όπως γνωρίζουμε, τα δεκαδικά ψηφία είναι τα 0, 1, 2, ..., 9.

Για την κωδικοποίηση των δέκα ψηφίων μέσω του δυαδικού συστήματος απαιτούνται τόσα δυαδικά ψηφία, έτσι ώστε ο συνδυασμός αυτών να μπορεί να δώσει 10 διαφορετικούς συνδυασμούς.

Με 3 bits μπορούμε να έχουμε $2^3 = 8$ μοναδικούς συνδυασμούς, άρα θα χρησιμοποιήσουμε υποχρεωτικά 4 bits αφού μπορούμε να έχουμε $2^4 = 16$ διαφορετικούς δυαδικούς συνδυασμούς, από τους οποίους όμως

χρειαζόμαστε μόνο τους δέκα. Οι υπόλοιποι 6 δεν χρησιμοποιούνται, αφού δεν υπάρχει γι' αυτούς αντίστοιχο ψηφίο.

Το κάθε bit της τετράδας έχει το δικό του βάρος στην αναπαράσταση με την έννοια ότι κάθε θέση περιλαμβάνει ένα βάρος με το οποίο πολλαπλασιάζεται το αντίστοιχο δυαδικό ψηφίο. Στην περίπτωση του κώδικα BCD, τα βάρη είναι 8421. Ο κώδικας BCD χαρακτηρίζεται ως δυαδικός κώδικας με βάρη. Π.χ. ο αριθμός 7_{10} αναπαρίσταται σε κώδικα BCD με την τετράδα 0111_2 ($0 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1$). Ο κώδικας BCD 8421 αποτελεί την πιο κοινή μορφή κωδικοποίησης.

Δεκαδικό ψηφίο	Δυαδική παράσταση
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Δεκαδικός: 5 8 3
 BCD: 0101 1000 0011
 Δυαδικό σύστημα: 1001000111

Αρχιτεκτονική Υπολογιστών, Δημήτρης Νικολός, Β. Γκιούρδας Εκδοτική, 2008

Όπως βλέπουμε στο παραπάνω σχήμα, η μετατροπή ενός δεκαδικού αριθμού στο δυαδικό σύστημα και η αναπαράστασή του σε κώδικα BCD είναι κάτι διαφορετικό. Π.χ., αν ο αριθμός 583_{10} μετατραπεί σε δυαδικό αριθμό θα γραφεί ως 1001000111_2 . Όμως, η παράσταση του 583_{10} σε κώδικα BCD είναι 0101 1000 0011. Η παράσταση αυτή προκύπτει αν λάβουμε χωριστά τις τιμές των δεκαδικών ψηφίων 5, 8 και 3 και έπειτα τις ενώσουμε.

Μετατροπή αριθμού από BCD σε δεκαδικό

Για τη μετατροπή ενός BCD αριθμού σε δεκαδικό αριθμό χωρίζουμε τον BCD αριθμό σε ομάδες τεσσάρων (4) bits και κάθε ομάδα μετατρέπεται στο ισοδύναμο δεκαδικό ψηφίο. Για παράδειγμα, ο BCD αριθμός 1001011000111000 αντιστοιχεί στο δεκαδικό αριθμό 9638 αφού:

1001 0110 0011 1000
 9 6 3 8

Μετατροπή αριθμού από δεκαδικό σε BCD

Για τη μετατροπή ενός δεκαδικού αριθμού σε BCD αριθμό, μετατρέπουμε κάθε ψηφίο του δεκαδικού αριθμού σε μία ομάδα τεσσάρων (4) bits που αποτελούν τον ισοδύναμο BCD αριθμό του κάθε δεκαδικού ψηφίου. Για παράδειγμα, ο δεκαδικός αριθμός 3628 αντιστοιχεί στον BCD αριθμό 0011011000101000 αφού:

3 6 2 8
 0011 0110 0010 1000

Ο κώδικας BCD χρησιμοποιείται συχνά σε αριθμομηχανές και ενσωματωμένα συστήματα, όταν χρειάζονται συνεχείς μετατροπές από το δεκαδικό στο δυαδικό σύστημα. Ωστόσο, παρουσιάζει μία δυσκολία στην εκτέλεση των αριθμητικών πράξεων.

Π.χ., έστω η πρόσθεση

25	οι αριθμοί σε κώδικα BCD	0010 0101
+46		+ 0100 0110
71		0110 1011

Η πρώτη τετράδα bits (0110) αντιστοιχεί στον αριθμό 6_{10} και η δεύτερη (1011) στον αριθμό 11_{10} , ο οποίος δεν έχει αντίστοιχο στον κώδικα BCD. Η λύση δίνεται προσθέτοντας το 6_{10} (0110) στο αποτέλεσμα της πρόσθεσης:

$$\begin{array}{r} 0110\ 1011 \\ +\ 0110 \\ \hline 0111\ 1001 \end{array}$$

Το τελικό αποτέλεσμα προκύπτει ενώνοντας τις δεκαδικές τιμές των δύο νέων τετράδων και είναι $(71)_{10}$. Ο κώδικας BCD δεν είναι ένα άλλο αριθμητικό σύστημα (όπως το δεκαδικό, το δυαδικό, το οκταδικό, το δεκαεξαδικό), αλλά είναι ένας τρόπος παράστασης των 10 ψηφίων του δεκαδικού συστήματος, το κάθε ένα από τα οποία αντιστοιχεί σε μία τετράδα bits.

Επομένως, είναι σημαντική η διαφορά ανάμεσα στη δυαδική κωδικοποίηση ενός δεκαδικού αριθμού και στη μετατροπή ενός δεκαδικού αριθμού στο δυαδικό σύστημα.

Κώδικας αριθμών Gray

Ο κώδικας Gray χαρακτηρίζεται ως δυαδικός κώδικας χωρίς βάρη. Στους δυαδικούς κώδικες χωρίς βάρη η θέση κάθε ψηφίου (bit) του κώδικα δεν έχει σταθερή «αξία», δηλαδή δεν αντιστοιχεί σε κάποιο βάρος, όπως γίνεται στους δυαδικούς κώδικες με βάρη. Οι κώδικες χωρίς βάρη προκύπτουν από κάποιους κανόνες διαφορετικούς για τον καθένα.

Βασική ιδιότητα του κώδικα Gray είναι ότι η παράσταση των αριθμών γίνεται με τέτοιο τρόπο ώστε κατά τη μετάβαση μεταξύ δύο διαδοχικών αριθμών του δεκαδικού συστήματος να αλλάζει μόνο ένα ψηφίο (1 bit).

Για παράδειγμα, η παράσταση της δεκαδικής τιμής "1" στο δυαδικό σύστημα είναι "001" και το "2" είναι "010". Στον κώδικα Gray, αυτές οι τιμές αντιπροσωπεύονται ως "001" και "011". Με αυτόν τον τρόπο, η αύξηση μιας τιμής από το 1 στο 2 απαιτεί μόνο ένα bit για αλλαγή, αντί για δύο.

Ένας τρόπος δημιουργίας του κώδικα Gray είναι με αντανάκλαση. Ειδικότερα, ο κώδικας Gray για τέσσερα δυαδικά ψηφία, παρουσιάζεται στον παρακάτω πίνακα:

Δεκαδικός αριθμός	Κωδικοποίηση Gray			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

Η πρώτη στήλη από δεξιά ξεκινάει πρώτα με ένα "0" και μετά με ένα "1". Τα επόμενα 2 κατακόρυφα bits είναι κατοπτρικά των 2 πρώτων bits (υπάρχει συμμετρία ως προς τη μέση τους). Έτσι, δημιουργούνται 4 bits. Τα επόμενα 4 κατακόρυφα bits είναι κατοπτρικά των 4 πρώτων bits. Έτσι, δημιουργούνται 8 bits. Τα επόμενα 8 bits είναι κατοπτρικά των 8 πρώτων bits. Η δεύτερη στήλη από δεξιά ξεκινάει πρώτα με δύο "0" και μετά με δύο "1". Τα επόμενα 4 bits είναι κατοπτρικά των 4 πρώτων bits. Έτσι, δημιουργούνται 8 bits. Τα επόμενα 8 bits είναι κατοπτρικά των 8 πρώτων bits. Η τρίτη στήλη από δεξιά ξεκινάει πρώτα με τέσσερα "0" και μετά με τέσσερα "1" και είναι κατοπτρική ως προς το μέσον της. Η τέταρτη στήλη από δεξιά ξεκινάει πρώτα με οκτώ "0" και μετά με οκτώ "1".

Παρατηρείστε ότι δύο διαδοχικοί αριθμοί διαφέρουν πάντα κατά ένα μόνο bit.

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

Η κωδικοποίηση ενός δεκαδικού αριθμού στον αντίστοιχο του κώδικα Gray ακολουθεί τα παρακάτω βήματα:

1. Βρίσκουμε τον αντίστοιχο δυαδικό του δεκαδικού αριθμού.
 2. Το πιο σημαντικό bit (MSB) του αριθμού σε κώδικα Gray είναι πάντα ίδιο με το πιο σημαντικό bit του δεδομένου δυαδικού αριθμού.
 3. Τα επόμενα bit του αριθμού σε κώδικα Gray προκύπτουν από τη σύγκριση των bit του δυαδικού αριθμού με το αμέσως προηγούμενο τους. Αν κατά τη σύγκριση έχουμε τα ίδια ψηφία, θέτουμε «0». Εάν έχουμε διαφορετικά θέτουμε «1» κ.ο.κ., έως ότου εξαντλήσουμε όλα τα δυαδικά ψηφία.
- Παρακάτω με τη βοήθεια μερικών παραδειγμάτων θα εξηγήσουμε τον τρόπο κωδικοποίησης του συγκεκριμένου κώδικα.

Παράδειγμα 1: Έστω ότι θέλουμε να μετατρέψουμε τον αριθμό 43_{10} .

Ο αντίστοιχος δυαδικός αριθμός του είναι: 101011_2

Το πιο σημαντικό bit (MSB) του αριθμού σε κώδικα Gray είναι πάντα ίσο με το πιο σημαντικό bit του δεδομένου δυαδικού αριθμού:

```

1 0 1 0 1 1   Δυαδικός
↓
1              Gray

```

Το επόμενο bit προκύπτει από τη σύγκριση του 2ου bit του δυαδικού αριθμού με το αμέσως προηγούμενο του. Έχουμε διαφορετικά ψηφία οπότε θέτουμε 1.

```

1 0 1 0 1 1   Δυαδικός
  ↓
1 1           Gray

```

Το επόμενο bit προκύπτει από τη σύγκριση του 3ου bit του δυαδικού αριθμού με το αμέσως προηγούμενο του. Έχουμε διαφορετικά ψηφία οπότε θέτουμε 1.

```

1 0 1 0 1 1   Δυαδικός
  ↓
1 1 1         Gray

```

Το επόμενο bit προκύπτει από τη σύγκριση του 4ου bit του δυαδικού αριθμού με το αμέσως προηγούμενο του. Έχουμε διαφορετικά ψηφία οπότε θέτουμε 1.

```

1 0 1 0 1 1   Δυαδικός
  ↓
1 1 1 1       Gray

```

Το επόμενο bit προκύπτει από τη σύγκριση του 5ου bit του δυαδικού αριθμού με το αμέσως προηγούμενο του. Έχουμε διαφορετικά ψηφία οπότε θέτουμε 1.

```

1 0 1 0 1 1   Δυαδικός
  ↓
1 1 1 1 1     Gray

```

Το επόμενο bit προκύπτει από τη σύγκριση του 6ου bit του δυαδικού αριθμού με το αμέσως προηγούμενο του. Έχουμε ίδια ψηφία οπότε θέτουμε 0.

```

1 0 1 0 1 1   Δυαδικός
  ↓
1 1 1 1 1 0   Gray

```

Τελειώνοντας, η κωδικοποίηση του δεκαδικού αριθμού 43_{10} σε κώδικα Gray είναι 111110.

Παράδειγμα 2: Έστω ότι θέλουμε να μετατρέψουμε τον αριθμό 564_{10} .

Ο αντίστοιχος δυαδικός αριθμός του είναι: 1000110100_2

1	0	0	0	1	1	0	1	0	0	Δυαδικός
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	
1	1	0	0	1	0	1	1	1	0	Gray

Η αντίστροφη διαδικασία της μετατροπής από τον κώδικα Gray στον αντίστοιχο δυαδικό αριθμό ακολουθεί τα παρακάτω βήματα:

1. Το πιο σημαντικό bit (MSB) του αριθμού σε κώδικα Gray είναι πάντα ίδιο με το πιο σημαντικό bit του ζητούμενου δυαδικού αριθμού.
2. Τα επόμενα bit του δυαδικού αριθμού προκύπτουν από τη σύγκριση του πιο σημαντικού bit του δυαδικού αριθμού με το δεύτερο bit του κώδικα Gray. Εάν κατά τη σύγκριση έχουμε τα ίδια ψηφία, θέτουμε «0». Εάν έχουμε διαφορετικά θέτουμε «1» κ.ο.κ., έως ότου εξαντλήσουμε όλα τα ψηφία του κώδικα Gray.

Παράδειγμα 3: Μετατροπή του κώδικα Gray 101101 σε δυαδικό αριθμό.

1	0	1	1	0	1	Gray					
↓											
1						Δυαδικός					
1	0	1	1	0	1	Gray					
↓	↓	↓	↓	↓	↓						
1	→	1	→	0	→	1	→	1	→	0	Δυαδικός

Τελειώνοντας, ο δυαδικός αριθμός του κώδικα Gray 101101 είναι ο 110110.

Ο κώδικας Gray χρησιμοποιείται σε εφαρμογές ψηφιακών συστημάτων που τα ψηφιακά δεδομένα αυξάνονται ή μειώνονται κατά ένα. Χρησιμοποιείται επίσης στις ηλεκτρομηχανικές εφαρμογές πολλών ψηφιακών συστημάτων και διατάξεων (εργαλειομηχανές, συστήματα φρένων αυτοκινήτου, φωτοαντιγραφικά κ.λπ.), όπου ένας αισθητήρας εισόδου δίνει μια ψηφιακή τιμή (σε κώδικα Gray), η οποία αναπαριστά μια μηχανική θέση. Ο κώδικας Gray παίζει σημαντικό ρόλο στις σύγχρονες ψηφιακές επικοινωνίες στη διόρθωση σφαλμάτων.

Αν χρησιμοποιούνταν δυαδικοί αριθμοί για τη μετάβαση από έναν αριθμό στον επόμενο, τότε υπάρχει η πιθανότητα σφάλματος: π.χ. η μετάβαση από το 0111 (7) στο 1000 (8) απαιτεί αλλαγή κατάστασης και των τεσσάρων bit πράγμα που μπορεί να οδηγήσει (για μικρό χρονικό διάστημα) στο 0110 (4) αν το λιγότερο σημαντικό bit (LSB) δεν συγχρονιστεί και αλλάξει κατάσταση γρηγορότερα από τα άλλα bits, με αποτέλεσμα να γίνει λάθος στη μετατροπή. Αν χρησιμοποιείται ο κώδικας Gray για τη μετάβαση από έναν αριθμό στον επόμενο, τότε η πιθανότητα σφάλματος εξαλείφεται: η μετάβαση από το 0100 (7) στο 1100 (8) επιτυγχάνεται με την αλλαγή ενός (1) μόνο bit.

Κώδικες χαρακτήρων

Στα υπολογιστικά συστήματα αποθηκεύονται εκτός από τα αριθμητικά δεδομένα και δεδομένα που αποτελούνται από γράμματα της αλφαβήτου ή σύμβολα όπως τα σημεία στίξης (όπως το '!' ή το ':') τα ειδικά σύμβολα (όπως το '€' ή το '@'), κάποιους μη εκτυπώσιμους χαρακτήρες (όπως το κενό, η αλλαγή γραμμής, οι στηλοθέτες) κτλ. Αυτά τα γράμματα της αλφαβήτου ή τα σύμβολα ονομάζονται χαρακτήρες (characters) που, όπως οι αριθμοί, παριστάνονται στους υπολογιστές σαν μία σειρά από δυαδικά ψηφία. Σε κάθε χαρακτήρα αντιστοιχεί ένας συγκεκριμένος συνδυασμός από bits (δηλ. ένας δυαδικός αριθμός). Υπάρχουν πολλοί κώδικες για αναπαράστασεις χαρακτήρων. Οι πιο γνωστοί είναι: ASCII, ISO & UNICODE.

Κώδικας χαρακτήρων ASCII

Ο κώδικας ASCII (American Standard Code for Information Interchange, Αμερικανικός Πρότυπος Κώδικας για Ανταλλαγή Πληροφοριών) είναι ένα κωδικοποιημένο σύνολο χαρακτήρων του λατινικού αλφάβητου. Αναπτύχθηκε στη δεκαετία του 1960 από τον Αμερικάνικο Οργανισμό Τυποποίησης, ώστε να επιτρέπει

σε ψηφιακές συσκευές να επικοινωνούν μεταξύ τους και να επεξεργάζονται, να αποθηκεύουν και να μεταδίδουν πληροφορίες σχετικές με χαρακτήρες. Σχεδιάστηκε αρχικά ως κώδικας των 8 bits όπου τα 7 χρησιμοποιούνταν για την κωδικοποίηση των χαρακτήρων ενώ το 8ο bit για τον έλεγχο της ορθότητας των υπολοίπων 7. Τα 7 διαθέσιμα bit δίνουν τη δυνατότητα 128 (2^7) διαφορετικών συνδυασμών και επομένως την αναπαράσταση 128 διαφορετικών χαρακτήρων, 94 εκτυπώσιμους (αλφαριθμητικά και σύμβολα), 33 μη εκτυπώσιμους (χαρακτήρες ελέγχου) και το κενό διάστημα. Ο παρακάτω πίνακας περιέχει το πλήρες σύνολο των 128 χαρακτήρων του κώδικα και την αντίστοιχη κωδικοποίηση τους στο δεκαδικό, δυαδικό, οκταδικό και δεκαεξαδικό σύστημα.

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	\$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

Πίνακας κώδικα ASCII

Παράδειγμα κωδικοποίησης ASCII:

```
01100010 01101001 01110100 01110011
b       i       t       s
```

Κωδικοσελίδες (character sets) ISO 8859

Σημαντικό μειονέκτημα του κώδικα ASCII είναι ότι περιλαμβάνει χαρακτήρες μόνο από το λατινικό αλφάβητο. Με την εξάπλωση των υπολογιστών σε χώρες με διαφορετικά αλφάβητα δημιουργήθηκε η ανάγκη αναπαράστασης χαρακτήρων και άλλων αλφαβητών. Γι' αυτό το λόγο χρησιμοποιήθηκε και το 8ο bit για την κωδικοποίηση χαρακτήρων και δημιουργήθηκαν επεκτάσεις του κώδικα.

Το πρότυπο ISO 8859 χρησιμοποιεί 8 bit για να κωδικοποιήσει κάθε χαρακτήρα δηλαδή, συνολικά μπορούμε να αναπαραστήσουμε μόνο 256 ($2^8=256$) διαφορετικούς χαρακτήρες. Οι πρώτες 128 θέσεις δεσμεύονται για τους 128 χαρακτήρες του κώδικα ASCII (Λατινικό αλφάβητο και σύμβολα), ενώ οι υπόλοιπες 128 θέσεις χρησιμοποιούνται από κάθε χώρα για την αναπαράσταση γραμμάτων του τοπικού αλφαβήτου, καθώς και άλλων χαρακτήρων (π.χ. μαθηματικά σύμβολα). Υπάρχουν πολλές παραλλαγές του προτύπου ISO 8859 για διαφορετικές οικογένειες γλωσσών:

- 8859-1 Γλώσσες Δυτικής Ευρώπης
- 8859-2 Γλώσσες Κεντρικής Ευρώπης

8859-3 Γλώσσες Νότιας Ευρώπης και Εσπεράντο
 8859-4 Γλώσσες Βόρειας Ευρώπης
 ...
 8859-7 Ελληνικά
 κτλ

Bits	b8b7b6b5	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
b4b3b2b1	HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			SP	0	@	P	`	p			NBSP	°	ı	Π	Û	π
0001	1			!	1	A	Q	a	q			`	±	A	P	α	ρ
0010	2			"	2	B	R	b	r			`	²	B		β	ς
0011	3			#	3	C	S	c	s			£	³	Γ	Σ	γ	σ
0100	4			\$	4	D	T	d	t				`	Δ	Τ	δ	τ
0101	5			%	5	E	U	e	u				ˆ	E	Υ	ε	υ
0110	6			&	6	F	V	f	v				A	Z	Φ	ζ	φ
0111	7			'	7	G	W	g	w			§	·	H	X	η	χ
1000	8			(8	H	X	h	x			ˆ	Ε	Θ	Ψ	θ	ψ
1001	9)	9	I	Y	i	y			©	Η	Ι	Ω	ι	ω
1010	A			*	:	J	Z	j	z				ı	K	İ	κ	ı
1011	B			+	;	K	[k	{			«	»	Λ	ÿ	λ	Û
1100	C			,	<	L	\	l				¬	Ο	M	ά	μ	ό
1101	D			-	=	M]	m	}			SHY	½	N	έ	ν	ύ
1110	E			.	>	N	^	n	-			®	Υ	Ξ	ή	ξ	ώ
1111	F			/	?	O	-	o	DEL			ˉ	Ω	O	ί	ο	

Εικόνα 1.3.5. Κωδικοποίηση ISO 8859-7

Κώδικας UNICODE

Ο κώδικας Unicode δημιουργήθηκε για να ξεπεράσει τους περιορισμούς των παλαιότερων προτύπων κωδικοποίησης χαρακτήρων όπως για παράδειγμα του προτύπου ISO 8859, το οποίο χρησιμοποιήθηκε από πολλές χώρες στον κόσμο, αλλά παρουσίαζε προβλήματα ασυμβατότητας μεταξύ των διαφορετικών παραλλαγών του αφού επέτρεπε την κωδικοποίηση μόνο δύο αλφαβήτων σε ένα συγκεκριμένο υπολογιστή, συνήθως του Λατινικού και ενός τοπικού, δηλαδή δεν υποστήριζε πολλά αλφάβητα στον ίδιο υπολογιστή. Για παράδειγμα, ένας υπολογιστής στην Ελλάδα που χρησιμοποιεί το πρότυπο ISO 8859-7, υποστηρίζει μόνο το Λατινικό και το Ελληνικό αλφάβητο. Έτσι, η εμφάνιση κειμένου που περιέχει κάποιο άλλο αλφάβητο (όπως για παράδειγμα Ρωσικό, Εβραϊκό κλπ) θα είναι προβληματική σε Ελληνικό υπολογιστή, αφού οι (Ρωσικοί, Εβραϊκοί κλπ) χαρακτήρες δεν θα εμφανίζονται σωστά.

Ο κώδικας Unicode αποτελεί διεθνές πρότυπο του ISO και παρέχει τη δυνατότητα κωδικοποίησης όλων των χαρακτήρων που χρησιμοποιούνται από ένα μεγάλο αριθμό γλωσσών του κόσμου. Για να κωδικοποιήσει τις χιλιάδες διαφορετικών χαρακτήρων που χρησιμοποιούνται στα αλφάβητα των διαφόρων γλωσσών, το Unicode χρησιμοποιεί ένα κώδικα των 8,16 ή και 32 bits, εξασφαλίζοντας έτσι τη δυνατότητα κωδικοποίησης 2^8 , 2^{16} ή και 2^{32} διαφορετικών χαρακτήρων. Το πρότυπο, το οποίο διατηρείται από την Unicode Consortium, ορίζει στην τρέχουσα έκδοση του (15.0) 149.186 χαρακτήρες.

Κάθε χαρακτήρας έχει τη δική του μοναδική αναπαράσταση ως μια σειρά από bit και αναφέρεται ως κωδικό σημείο (code point). Οι τιμές σύμφωνα με το Unicode είναι γραμμένες ως δεκαεξαδικοί αριθμοί και έχουν ένα πρόθεμα του U+.

Για παράδειγμα, το πεζό λατινικό γράμμα b εκφράζεται ως U+0062 ενώ το κεφαλαίο ελληνικό γράμμα Ω εκφράζεται ως U+03A9.

Έχει τρεις τύπους κωδικοποίησης χαρακτήρων:

UTF-8: Το UTF-8 χρησιμοποιείται ευρέως σε συστήματα ηλεκτρονικού ταχυδρομείου και στο διαδίκτυο. Είναι μια κωδικοποίηση μεταβλητού μήκους, δηλαδή έχει τη δυνατότητα να χρησιμοποιεί έως και 4 byte για την αναπαράσταση των χαρακτήρων σύμφωνα με τον παρακάτω πίνακα:

Περιοχή κώδικα σε δεκαεξαδική μορφή	UTF-8 κωδικοποίηση (bytes)	Σύνολο χαρακτήρων
-------------------------------------	----------------------------	-------------------

0...7F	0XXXXXXXX	128 χαρακτήρες, το λατινικό αλφάβητο και σύμβολα (συμβατό με το πρότυπο ASCII)
80...7FF	110XXXXX 10XXXXXX	1920 χαρακτήρες, πρόσθετα λατινικά και μεσανατολικά γράμματα και σύμβολα.
800...FFFF	1110XXXX 10XXXXXX 10XXXXXX	Ασιατικά γράμματα και σύμβολα.
10000... 10FFFF	11110XXX 10XXXXXX 10XXXXXX 10XXXXXX	Άλλοι πρόσθετοι χαρακτήρες

Όταν υπάρχουν πολλαπλά bytes στην κωδικοποίηση (δεύτερη στήλη πίνακα), το πρώτο byte ακολουθείται από έναν αριθμό συνοδευτικών bytes, τα οποία ξεκινούν με τον συνδυασμό bits 10....

UTF-16: Είναι επίσης μια κωδικοποίηση μεταβλητού μήκους. Χρησιμοποιεί είτε δύο bytes (16 bits) για να κωδικοποιήσει τους συνηθέστερα χρησιμοποιούμενους χαρακτήρες. Και 4 bytes (32 bits) για να κωδικοποιήσει τους επιπλέον χαρακτήρες. Το UTF-16 χρησιμοποιείται από συστήματα όπως το Microsoft Windows API, η γλώσσα προγραμματισμού Java και η JavaScript. Δεν είναι συμβατό με τον κώδικα χαρακτήρων ASCII

UTF-32: Το UTF-32 είναι μια κωδικοποίηση σταθερού μήκους, σε αντίθεση με τις άλλες δύο μορφές. Χρησιμοποιεί ακριβώς 32 bit (4 bytes) για να κωδικοποιήσει όλους τους χαρακτήρες.

Ο παρακάτω πίνακας μας δείχνει τον αριθμό των απαιτούμενων bytes για κάθε κωδικό σημείο στους τρεις διαφορετικούς τύπους κωδικοποίησης:

Περιοχή κώδικα σε δεκαεξαδική μορφή	UTF_8	UTF-16	UTF-32
0...7F	1 byte	2 byte	4 byte
80...7FF	2 byte		
800...FFFF	3 byte		
10000... 10FFFF	4 byte	4 byte	

Πηγή: https://en.wikipedia.org/wiki/Comparison_of_Unicode_encodings

Τα UTF-8 και UTF 16 είναι δύο από τα πιο καθιερωμένα πρότυπα κωδικοποίησης. Διαφέρουν μόνο σε πόσα bytes χρησιμοποιούν για την κωδικοποίηση κάθε χαρακτήρα. Δεδομένου ότι και οι δύο είναι κωδικοποίηση μεταβλητού εύρους, μπορούν να χρησιμοποιήσουν μέχρι και τέσσερα bytes για να κωδικοποιήσουν τα δεδομένα, αλλά όταν αυτό είναι το ελάχιστο, το UTF-8 χρησιμοποιεί μόνο 1 byte (8 bits) και το UTF-16 χρησιμοποιεί 2 bytes (16 bits). Αυτό έχει τεράστιο αντίκτυπο στο μέγεθος που προκύπτει από τα κωδικοποιημένα αρχεία. Όταν χρησιμοποιείτε μόνο χαρακτήρες ASCII, ένα αρχείο με κωδικοποίηση UTF-16 θα είναι περίπου διπλάσιο από το ίδιο αρχείο που κωδικοποιείται με το UTF-8.

Παράδειγμα κωδικοποίησης χαρακτήρων στους τρεις τύπους του Unicode:

Παράδειγμα κωδικοποίησης χαρακτήρων στους τρεις τύπους του Unicode:

Χαρακτήρας	Κωδικοποίηση	Bytes κώδικα
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001



Δραστηριότητες - Ασκήσεις

1. Να μετατραπεί ο δεκαδικός αριθμός 248,526 στον ισοδύναμο δυαδικό αριθμό.
2. Προσθέστε τους παρακάτω δυαδικούς αριθμούς σημειώνοντας και τα τυχόν κρατούμενα που θα προκύψουν:

$$\begin{array}{r}
 110010 \\
 + 1101 \\
 \hline
 \end{array}
 \qquad
 \begin{array}{r}
 11011101 \\
 + 110011 \\
 \hline
 \end{array}
 \qquad
 \begin{array}{r}
 01100 \\
 + 10001 \\
 \hline
 \end{array}
 \qquad
 \begin{array}{r}
 01111 \\
 + 10111 \\
 \hline
 \end{array}$$

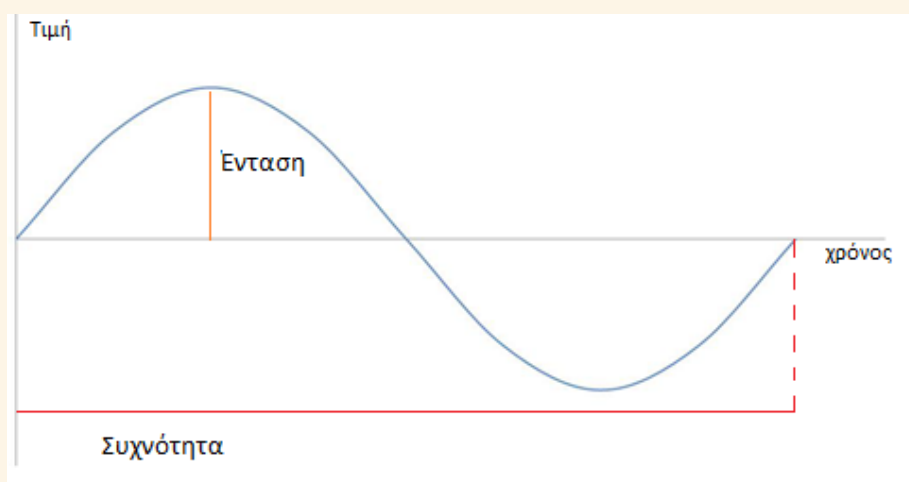
3. Να κάνετε την πράξη της αφαίρεσης για τους παραπάνω αριθμούς.
4. Να γραφούν σε κώδικα BCD οι δεκαδικοί αριθμοί 41,22,278 και 325
5. Να μετατραπεί ο δυαδικός αριθμός 011001001101 στον αντίστοιχο δυαδικό του κώδικα Gray.
6. Ανοίξτε τη σελίδα [Ascii Text to Binary Converter \(binaryhexconverter.com\)](http://binaryhexconverter.com). Πειραματιστείτε με τον online μετατροπέα ASCII σε δυαδικό και αντίστροφα. Γράψτε λέξεις για να δείτε πως μετατρέπονται σε ακολουθίες δυαδικών αριθμών ή αντίστροφα πληκτρολογήστε ακολουθίες δυαδικών αριθμών για να δείτε σε ποια γράμματα ή σύμβολα αντιστοιχούν.

1.3.6 Κωδικοποίηση ήχου

Ο ήχος, είναι η αίσθηση που αντιλαμβανόμαστε όταν τα ακουστικά νεύρα ενεργοποιούνται από μόρια που δονούνται στον αέρα. Οποιοσδήποτε ήχος απαιτεί μια **πηγή**, όπως μια ανθρώπινη φωνή ή ένα μουσικό όργανο, και ένα **μέσο** για να ταξιδέψει, τον αέρα, ένα υγρό όπως το νερό ή ένα στερεό.

Όταν μιλάμε, ο ήχος που παράγουμε δημιουργεί μια δόνηση. Αυτές οι δονήσεις αποτελούνται από την ταλάντωση των μορίων του αέρα που χτυπούν μεταξύ τους και δημιουργούν ένα **ηχητικό κύμα**: ένα αναλογικό σήμα που ταξιδεύει στον αέρα.

Ο ήχος διαφέρει από τους χαρακτήρες ή τους αριθμούς που περιγράψαμε παραπάνω. Ένα κείμενο αποτελείται από ένα πλήθος συγκεκριμένων χαρακτήρων που μπορεί να εύκολα να μετατραπεί σε ψηφιακή μορφή με την βοήθεια των κωδικών που είδαμε παραπάνω. Ο ήχος όμως δεν είναι μετρήσιμος, είναι μια οντότητα που μεταβάλλεται, σε σχέση με το χρόνο, σε ένταση (πλάτος) και συχνότητα (τόνος).



Εικόνα 1.3.6. Αναλογική παράσταση ήχου

Ο ήχος είναι ένα παράδειγμα αναλογικών δεδομένων. Όπως φαίνεται στην παραπάνω γραφική παράσταση, το πλήθος των τιμών της έντασης του ήχου σε μια χρονική περίοδο είναι άπειρο και δεν μπορούμε να τις αποθηκεύσουμε στη μνήμη του υπολογιστή.

Για να τον επεξεργαστούμε, αποθηκεύσουμε, αναπαράγουμε, μεταδώσουμε με ψηφιακά συστήματα (υπολογιστής, κινητό τηλέφωνο, ψηφιακή τηλεόραση και ραδιόφωνο, media player, Διαδίκτυο κ.τ.λ.) θα

πρέπει να μετατραπεί σε ψηφιακή μορφή, δηλαδή σε μια ακολουθία από διακριτές τιμές. Για τη μετατροπή του ήχου από αναλογικό σε ψηφιακό είναι απαραίτητη η διαδικασία της **ψηφιοποίησης** του. Η διαδικασία αυτή περιλαμβάνει τρία βασικά στάδια: την «δειγματοληψία» του αρχικού σήματος, την «κβάντιση» των τιμών του σήματος που προκύπτει από την δειγματοληψία και τέλος την «κωδικοποίηση».

Δειγματοληψία

Δειγματοληψία είναι η διαδικασία λήψης ενός πεπερασμένου πλήθους τιμών του πλάτους του αναλογικού σήματος, σε δεδομένες στιγμές του χρόνου, σε ίσες αποστάσεις μεταξύ τους. Παίρνουμε δηλαδή περιοδικά **δείγματα (samples)** σε μια συχνότητα που επιλέγουμε. Ο αριθμός λήψης δειγμάτων ανά δευτερόλεπτο ονομάζεται **ρυθμός ή συχνότητα δειγματοληψίας (sampling rate)**.

Πόσα δείγματα πρέπει να πάρουμε ώστε να πετύχουμε ένα ακριβές αντίγραφο του αρχικού αναλογικού ήχου;

Η απάντηση στο ερώτημα δίνεται από το θεώρημα του Nyquist, το οποίο αποδεικνύει ότι: Ένα συνεχές στο χρόνο σήμα που περιέχει συχνότητες όχι μεγαλύτερες από f_{max} μπορεί να ανακατασκευαστεί ακριβώς από τα δείγματα του, αν τα δείγματα αυτά έχουν ληφθεί με συχνότητα δειγματοληψίας που είναι μεγαλύτερη από $2f_{max}$. Η μέγιστη συχνότητα δειγματοληψίας πρέπει να είναι τουλάχιστον διπλάσια από τη μέγιστη συχνότητα του αναλογικού σήματος στο οποίο γίνεται δειγματοληψία.

Για παράδειγμα, αν η μέγιστη συχνότητα του αναλογικού συστήματος είναι 10 KHz, η συχνότητα δειγματοληψίας πρέπει να είναι τουλάχιστον 20 KHz.

Οι μουσικοί οπτικοί δίσκοι (Audio CD-ROM) χρησιμοποιούν συχνότητα δειγματοληψίας ίση με 44.1KHz (44100 δείγματα ανά δευτερόλεπτο) για την αποθήκευση του ηχητικού σήματος. Η συχνότητα αυτή είναι λίγο μεγαλύτερη από 2 φορές τη συχνότητα των 20 KHz που κατά γενική παραδοχή αποτελεί το άνω όριο για τις συχνότητες που γίνονται αντιληπτές από το ανθρώπινο αυτί. Με ένα τέτοιο ρυθμό δειγματοληψίας διασφαλίζεται ότι όλες οι συχνότητες που είναι μικρότερες ή ίσες με 20 kHz θα αποδοθούν σωστά στο το ψηφιακό σήμα.



Εικόνα 1.3.6.α. Δειγματοληψία διαφόρων συχνοτήτων. Πηγή: <https://www.headphonesty.com/2019/07/sample-rate-bit-depth-bit-rate>

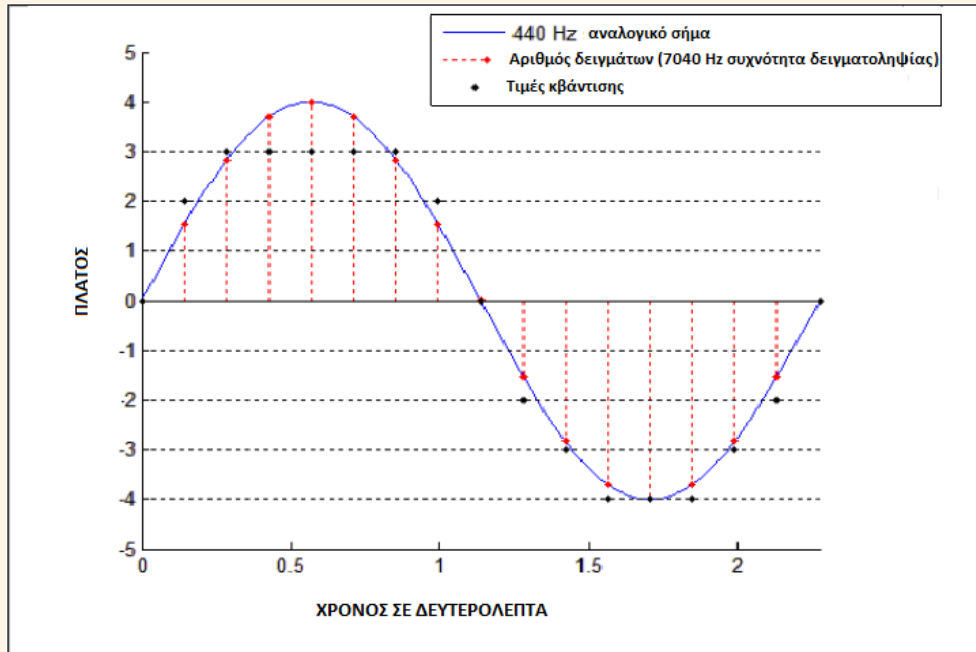
Κβάντιση

Το δεύτερο στάδιο είναι η κβάντιση, δηλαδή η μετατροπή της τιμής των δειγμάτων σε ακέραιους αριθμούς εκφρασμένους στο δυαδικό σύστημα. Πόσα bit θα χρειαστούν για να εκφραστούν οι τιμές των δειγμάτων στο δυαδικό σύστημα;

Έστω ότι έχουμε το αναλογικό ηχητικό σήμα της παρακάτω εικόνας (μπλε γραμμή). Ας υποθέσουμε επίσης ότι έχουμε στη διάθεση μας 3 bits για να εκφράσουμε τις διαφορετικές τιμές των δειγμάτων, δηλαδή οι διαφορετικές τιμές που μπορούμε να πάρουμε είναι $2^3=8$. Τα δείγματα αναπαρίστανται με τις κόκκινες τελείες, ενώ οι οριζόντιες διακεκομμένες γραμμές είναι οι 8 επιτρεπτές τιμές κβάντισης (0,1,2,3,-1,-2,-3,-4). Επειδή η κβάντιση εκφράζεται σε ακέραιους δυαδικούς αριθμούς, κάθε τιμή του δείγματος στρογγυλοποιείται στην πλησιέστερη ακέραιη τιμή κβάντισης. Π.χ. το τρίτο δείγμα στην εικόνα έχει τιμή περίπου 2,9 και στρογγυλοποιήθηκε στο 3. Έτσι αντί των κόκκινων τελειών, οι τιμές

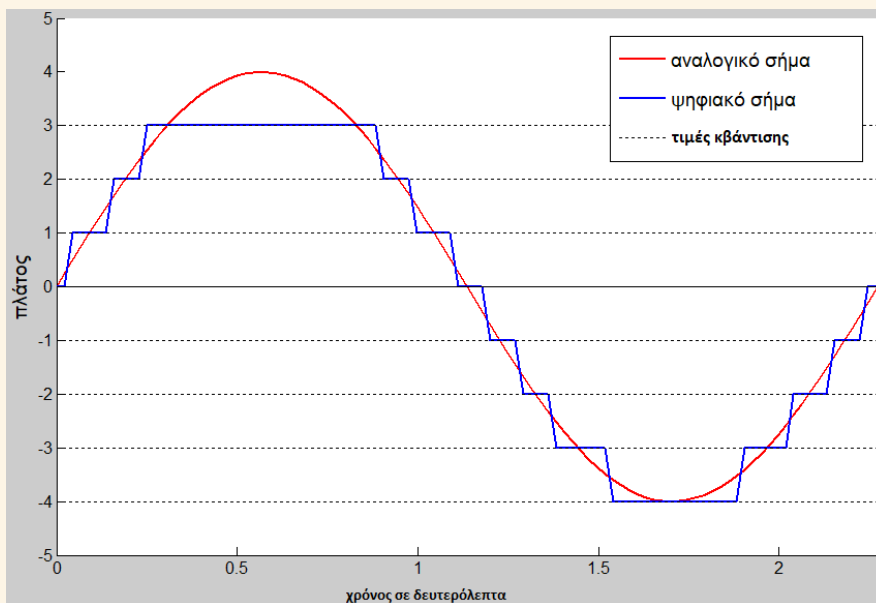
αντιστοιχούν πλέον στις μαύρες τελείες. Συνεπώς, κοντινές, αλλά διαφορετικές τιμές αναλογικής έντασης αντιστοιχούν στην ίδια ψηφιακή τιμή.

Αν τώρα ενώσουμε μεταξύ τους τις μαύρες τελείες, η γραμμή που θα σχηματιστεί είναι το κβαντισμένο, ψηφιακό σήμα.



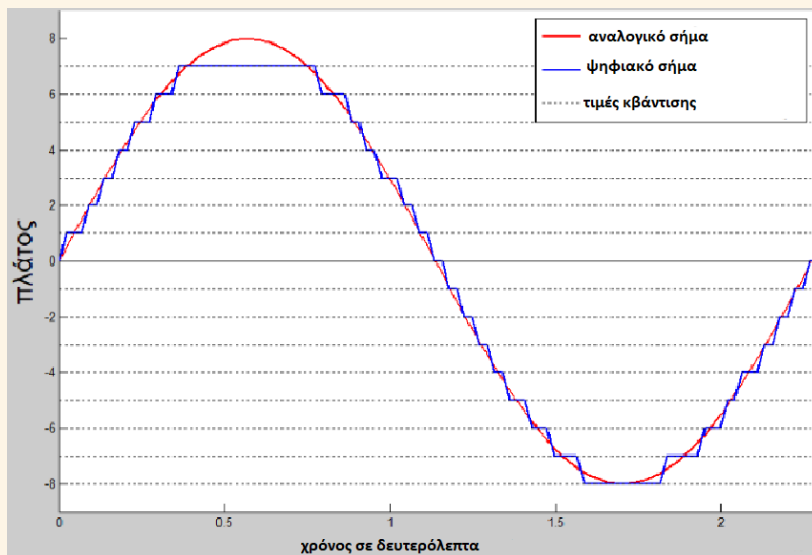
Εικόνα 1.3.6.β. Κβάντιση σήματος με 3 bit

Στην παρακάτω εικόνα (1.3.6.γ) βλέπουμε το αναλογικό σήμα (κόκκινη γραμμή) και το ψηφιακό (κυανή γραμμή) που προκύπτει έπειτα από κβάντιση των 3 bit με 8 επιτρεπτές τιμές.



Εικόνα 1.3.6.γ. Σύγκριση αναλογικού και ψηφιακού σήματος.

Ας δούμε την κβάντιση του ίδιου σήματος με 4 bit, δηλαδή με $2^4=16$ επιτρεπτές τιμές κβάντισης (Εικόνα 1.3.6.δ).



Εικόνα 1.3.6.δ. Κβάντιση με 4 bit

Συμπεραίνουμε ότι όσο περισσότερα bit διαθέτουμε για τις τιμές των δειγμάτων, τόσο το παραγόμενο ψηφιακό σήμα θα είναι καλύτερη προσέγγιση του αναλογικού.

Κωδικοποίηση

Το τελικό στάδιο για την ψηφιοποίηση ενός αναλογικού σήματος είναι η κωδικοποίηση, δηλαδή η μετατροπή των τιμών κβάντισης σε δυαδικούς αριθμούς, ώστε να είναι δυνατή η αποθήκευση τους στη μνήμη του υπολογιστή για να μπορούν στη συνέχεια να χρησιμοποιηθούν για την μεταφορά, την επεξεργασία και την ανασύνθεση του αρχικού αναλογικού ηχητικού σήματος.

Η πιο συχνά χρησιμοποιούμενη μέθοδος κωδικοποίησης, είναι η παλμοκωδική διαμόρφωση (Pulse Code Modulation, PCM). Ο αριθμός των δυαδικών ψηφίων (bit) που διατίθενται για την αποθήκευση κάθε τιμής καλείται **εύρος δείγματος** (sample width ή bit depth). Το εύρος δείγματος καθορίζει το πλήθος των διαφορετικών τιμών κβάντισης. Για παράδειγμα, ένα ηχητικό σήμα με εύρος δείγματος 16 bit (ποιότητα CD) μπορεί να περιέχει $65.536 (=2^{16})$ τιμές. [1]

Αν δεχθούμε ότι το εύρος δείγματος είναι A και ότι ρυθμός δειγματοληψίας ανά δευτερόλεπτο είναι B τότε για κάθε δευτερόλεπτο ήχου πρέπει ν' αποθηκεύσουμε $A \times B$ bits. Το γινόμενο αυτό αναφέρεται ως **ρυθμός μετάδοσης (bit rate)** και εκφράζει την ταχύτητα δημιουργίας και μετάδοσης ψηφιακών δεδομένων σε μια συγκεκριμένη μονάδα χρόνου και μετριέται σε bits/sec (bps).

Το μέγεθος ενός ασυμπίεστου ψηφιακού ηχητικού αρχείου εξαρτάται, από τέσσερις παραμέτρους: Τη χρονική του διάρκεια, την συχνότητα δειγματοληψίας, το εύρος δείγματος και το πλήθος των καναλιών του. Για να υπολογίσουμε το μέγεθος του αρχείου εφαρμόζουμε τον παρακάτω τύπο:

Μέγεθος (σε bytes) = συχνότητα δειγματοληψίας (Hz) x εύρος δείγματος (bits) x χρονική διάρκεια (sec) x αρ.καναλιών / 8 (bits) γιατί το ζητούμενο μέγεθος είναι σε bytes.

Για παράδειγμα ένα στερεοφωνικό αρχείο (με δύο κανάλια ήχου), με συχνότητα δειγματοληψίας 44.1 kHz (44.100 Hz), εύρος δείγματος 16 bit και με χρονική διάρκεια 5 λεπτά (300 sec), το μέγεθός του θα είναι περίπου 55,92 Mb.

Από τα παραπάνω καταλαβαίνουμε ότι όσο μεγαλύτερη είναι η συχνότητα δειγματοληψίας και το εύρος δείγματος, τόσο μεγαλύτερο το κομμάτι πληροφορίας που παράγεται και επομένως το ψηφιακό μας σήμα είναι καλύτερη προσέγγιση του αναλογικού από το οποίο προήλθε. Περισσότερη πληροφορία όμως σημαίνει και περισσότερη δυσκολία, καθώς η κατασκευή ενός ποιοτικού ψηφιακού σήματος απαιτεί μεγάλη υπολογιστική ισχύ και μνήμη κατά την αποθήκευση του, την επεξεργασία του και τη μετάδοση του. Για αυτούς τους λόγους είναι απαραίτητη η συμπίεση των αρχείων ήχου ώστε να μπορούν εύκολα να διαχειριστούν.

Συμπίεση

Στην συμπίεση των ηχητικών δεδομένων εφαρμόζονται αλγόριθμοι που συνδυάζουν απωλεστικές και μη απωλεστικές τεχνικές συμπίεσης. Έτσι, υπάρχουν δύο κατηγορίες αλγορίθμων συμπίεσης, οι απωλεστικοί (lossy) και οι μη απωλεστικοί (lossless).

Πολλά αρχεία περιέχουν μεγάλα τμήματα τα οποία επαναλαμβάνονται (όπως ένα αρχείο κειμένου μπορεί να περιέχει πολλές φορές τη λέξη Κωνσταντινούπολη οπότε αρκεί η λέξη να τοποθετηθεί σε έναν πίνακα με λέξεις και να αντικαθίσταται με έναν αριθμό που δείχνει τη θέση της στον πίνακα) ή περιέχουν δεδομένα που η απώλειά τους δεν προκαλεί σοβαρή μεταβολή του περιεχόμενου.

Η συμπίεση μπορεί να μειώσει τον όγκο των δεδομένων που στέλνονται ή αποθηκεύονται, με την ελαχιστοποίηση του πλεονασμού που παρουσιάζεται κατά τη δημιουργία των δεδομένων. Με τη διαδικασία της συμπίεσης η μεταφορά και η αποθήκευση γίνονται με πιο αποδοτικό τρόπο, ενώ παράλληλα διατηρείται η ακεραιότητα των δεδομένων.

Στην μη απωλεστική συμπίεση (lossless compression) διατηρείται η ακεραιότητα των δεδομένων. Τα αρχικά δεδομένα και τα δεδομένα μετά τη συμπίεση και την αποσυμπίεση είναι ακριβώς τα ίδια, και κατά τη διαδικασία δε χάνεται κανένα μέρος των δεδομένων. Τα πλεονάζοντα δεδομένα κωδικοποιούνται κατά τη συμπίεση και αποκωδικοποιούνται κατά την αποσυμπίεση. Αυτοί οι μέθοδοι χρησιμοποιούνται όταν δεν πρέπει να χαθεί ούτε ένα μπιτ δεδομένων όπως στην περίπτωση ενός αρχείου κειμένου ή ενός προγράμματος. [3]

Η απωλεστική συμπίεση στηρίζεται σε ψυχο-ακουστικά μοντέλα βάσει των οποίων εφαρμόζονται αλγόριθμοι που απορρίπτουν τους ήχους που δεν γίνονται αντιληπτοί από το ανθρώπινο αυτί, δηλαδή μεταξύ των συχνοτήτων 20 Hz και 20 KHz. Μπορούμε, λοιπόν, να απορρίψουμε τα δεδομένα που αφορούν σε συχνότητες εκτός των παραπάνω ορίων. Επίσης, όταν υπάρχει ένας ήχος συγκεκριμένης συχνότητας και έντασης, είναι δυνατόν άλλοι ήχοι σε κοντινές συχνότητες και μικρότερης έντασης να μη γίνονται αντιληπτοί από το ανθρώπινο αυτί (φαινόμενο ηχητικής σκίασης). Το ίδιο ισχύει και όταν κάποιος ήχος έχει πολύ χαμηλή ένταση. Στις περιπτώσεις αυτές η κωδικοποίηση απορρίπτει τις σχετικές πληροφορίες, επιτυγχάνοντας μείωση του όγκου του τελικού αρχείου. Οι πληροφορίες που χάνονται είναι τέτοιες που, έτσι και αλλιώς, δεν μπορεί να τις αντιληφθεί το ανθρώπινο αυτί.

Πρότυπα κωδικοποίησης ήχου

Η ποιότητα και η ομοιομορφία της δομής των κωδικοποιήσεων του ήχου απαιτούσε τη σύσταση μιας επιτροπής προτυποποίησης. Η επιτροπή ονομάστηκε Moving Pictures Expert Group (MPEG) και από το τέλος της δεκαετίας του 80 επιφορτίστηκε με την ανάπτυξη προτύπων για την κωδικοποιημένη αναπαράσταση κινούμενων εικόνων, ηχητικών σημάτων και του συνδυασμού τους.

Το πρώτο πρότυπο που αναπτύχθηκε από την επιτροπή ονομάστηκε MPEG-1 Audio και αποτελεί το πρώτο διεθνές πρότυπο που καθορίζει το ψηφιακό μορφότυπο για υψηλής ποιότητας ήχο. Η επιτυχία των προτύπων MPEG οδήγησε στην αποδοχή συμπιεσμένου ήχου υψηλής ποιότητας σε μια ευρεία γκάμα εφαρμογών, όπως η ψηφιακή μετάδοση και οι διαδικτυακές εφαρμογές. Σήμερα το επικρατέστερο πρότυπο MPEG είναι το MP3 (MPEG Layer III).

Το MP3 είναι ένας τύπος ψηφιακού συμπιεσμένου αρχείου ήχου που πετυχαίνει συμπίεση έως 12 φορές ως προς την αρχική μορφή χωρίς εμφανείς απώλειες στην ποιότητα. Παρέχει τη δυνατότητα αναπαράστασης ήχου κωδικοποιημένου με μορφή Pulse Code Modulation (PCM), δεσμεύοντας όμως πολύ λιγότερο χώρο (για δεδομένα). Αυτό γίνεται μέσω ενός αλγορίθμου σχεδιασμένου να μειώνει δραστικά το πλήθος των ψηφιακών δεδομένων που απαιτούνται για την αποθήκευση και ορθή αναπαραγωγή του ήχου, απορρίπτοντας τμήματα ή περιοχές του ηχητικού σήματος που δεν ακούει το ανθρώπινο αυτί και καταγράφοντας την υπόλοιπη πληροφορία με αποτελεσματικό τρόπο. [2]

Άλλοι τύποι αρχείων ήχου είναι:

.WAV (Waveform Audio File Format) ψηφιακός τύπος αρχείου βασισμένος στην κωδικοποίηση PCM Επιτρέπει την αποθήκευση ήχου με διάφορα sampling και bit rates. Συνήθως χρησιμοποιούνται για να

αποθηκεύουν αρχεία ποιότητας CD. Είναι παρόμοια με τα .AIF άλλα είναι πιο περίπλοκα και χρησιμοποιούνται περισσότερο από συστήματα Windows.

. AIFF (Audio Interchange File Format) είναι τύπος αρχείου όμοιος με το .WAV. Αναπτύχθηκε από την Apple. Ένα τυπικό αρχείο .AIF ψηφιακού δίσκου έχει συχνότητα δειγματοληψίας 44.1KHz, εύρος δείγματος 16 bit και έχει δυο κανάλια.

.WMA (Windows Media Audio File) αρχείο ήχου συμπιεσμένο με το Windows Media το οποίο κατασκευάστηκε από την Microsoft, έχει ομοιότητες με το MP3. Χρησιμοποιείται κυρίως για την αναπαραγωγή μουσικής στο Διαδίκτυο.

.MIDI (Musical Instrument Digital Interface, Ψηφιακή Διασύνδεση Μουσικών Οργάνων) είναι το παλαιότερο (1983) και το πλέον διαδεδομένο πρωτόκολλο μουσικής επικοινωνίας, το οποίο επιτρέπει σε ηλεκτρονικά μουσικά όργανα (π.χ. synthesizers), υπολογιστές και άλλο ηλεκτρονικό εξοπλισμό να επικοινωνούν και να συγχρονίζονται μεταξύ τους ελέγχοντας το ένα το άλλο.

Η πληροφορία που μεταφέρεται δεν αποτελεί σήμα ήχου αλλά μηνύματα (οδηγίες εκτέλεσης) που αφορούν το ποια νότα πρέπει να παιχθεί, για πόσο χρόνο και ένταση και από ποιο μουσικό όργανο. Επομένως η "γλώσσα" MIDI ορίζει τον τρόπο αναπαραγωγής της μουσικής με τρόπο παρόμοιο με αυτόν μιας παρτιτούρας.

Τα πλεονεκτήματα του MIDI είναι ότι υπάρχει μεγαλύτερη ευελιξία στην επεξεργασία της μουσικής MIDI ενώ απαιτείται και μικρότερος αποθηκευτικός χώρος (ένα αρχείο MIDI είναι από 200 έως 1.000 φορές μικρότερο από ένα αρχείο ψηφιοποιημένου ήχου ποιότητας CD). Για παράδειγμα, ένα λεπτό μουσικής αποθηκευμένο σε αρχείο WAV απαιτεί περίπου 10 MB, ενώ σε αρχείο MIDI απαιτεί περίπου 6 KB.



Δραστηριότητες - Ασκήσεις

1. Στη σελίδα <https://photodentro.edu.gr/lor/r/8521/556> θα βρείτε μια διαδραστική οπτικοποίηση με στόχο την αναπαράσταση της διαδικασίας της δειγματοληψίας και της μετατροπής αναλογικού ήχου σε ψηφιακό. Μπορείτε να μεταβάλετε τις βασικές παραμέτρους της δειγματοληψίας ενός συγκεκριμένου ηχητικού κομματιού και θα παρατηρήσετε την μεταβολή της κυματομορφής, αλλά και του μεγέθους του αντίστοιχου αρχείου ήχου.
2. Από τι εξαρτάται η ποιότητα του ψηφιακού ήχου;
3. Τι είναι η δειγματοληψία και ο ρυθμός δειγματοληψίας;
4. Ποιες διαφορές υπάρχουν ανάμεσα στο αρχείο ήχου που προκύπτει από ψηφιοποίηση και στο περιεχόμενο ενός αρχείου MIDI;

1.3.7 Ψηφιακή παράσταση εικόνας

Όπως αναφέραμε στην προηγούμενη ενότητα, η μετάβαση από τον αναλογικό κόσμο, στον ψηφιακό κόσμο γίνεται με τη διαδικασία της ψηφιοποίησης. Είδαμε δηλαδή ότι αν σε ένα συνεχή ήχο απομονώσουμε μια σειρά στιγμών που βρίσκονται πολύ κοντά η μία στην άλλη, τότε προκύπτει μια ακολουθία αριθμών που ο καθένας από αυτούς αντιπροσωπεύει μια ένταση του ήχου. Ομοίως, αν σε μια εικόνα απομονώσουμε μια σειρά σημείων πολύ κοντά το ένα στο άλλο, τότε προκύπτει μια σειρά αριθμών που ο καθένας αντιπροσωπεύει τον χρωματικό κώδικα κάθε σημείου. Και στις δύο περιπτώσεις η πληροφορία που παίρνουμε είναι μια σειρά αριθμών που μπορεί να παρασταθεί στο δυαδικό αριθμητικό σύστημα, δηλαδή στο σύστημα κατανόησης και επικοινωνίας των Η/Υ.

Αν η λήψη της εικόνας γίνεται με φωτογραφική μηχανή και η εικόνα αποτυπώνεται σε φιλμ ή σε χαρτί, τότε χρειάζεται η μετατροπή της αναλογικής πληροφορίας με τη χρήση ενός σαρωτή ο οποίος θα τη μετατρέψει σε ψηφιακή μορφή.

Αν η λήψη της εικόνας γίνεται με ψηφιακή φωτογραφική μηχανή, τότε η συσκευή λήψης εκτελεί απευθείας την ψηφιοποίηση της.

Ανάλογα με τον τρόπο δημιουργίας μιας εικόνας, μπορούμε να έχουμε τις παρακάτω κατηγορίες:

Ψηφιογραφικές (bitmap)

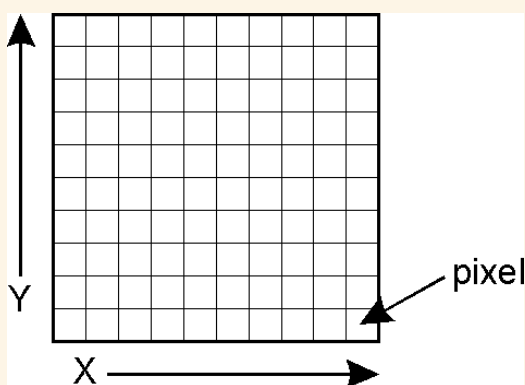
Διανυσματικές (vector)

Ψηφιογραφικές εικόνες (bitmap)

Στην κατηγορία αυτή ανήκουν οι εικόνες που προέρχονται από ψηφιακή φωτογραφική μηχανή ή εικόνες που ψηφιοποιήθηκαν μέσω σαρωτή, ή εικόνες που δημιουργήθηκαν από προγράμματα ζωγραφικής όπως το Microsoft Paint, το Photoshop, το GIMP κ.α.

Η διαδικασία ψηφιοποίησης εικόνων περιλαμβάνει δύο στάδια, την δειγματοληψία και τον κβαντισμό. Η δειγματοληψία στην περίπτωση της ψηφιογραφικής εικόνας ονομάζεται σάρωση και τα δείγματα που παίρνουμε ονομάζονται pixels (από το picture element, στοιχείο εικόνας ή εικονοστοιχείο).

Το εικονοστοιχείο ή pixel είναι ένα «σημείο» μιας εικόνας που εμφανίζεται στην οθόνη ενός υπολογιστικού συστήματος, δηλαδή, για το υπολογιστικό σύστημα, ένα δείγμα πληροφορίας. Στον υπολογιστή η εικόνα αναπαρίσταται υπό τη μορφή «ψηφιδωτού». Το εικονοστοιχείο είναι, απλά, μια ψηφίδα του ψηφιδωτού αυτού και, ως εκ τούτου, θεωρείται ως το μικρότερο πλήρες δείγμα μιας εικόνας. Στην οθόνη ενός υπολογιστή οι εικόνες αναπαρίστανται με «υποδιαίρεση» της οθόνης σε ένα δισδιάστατο πίνακα με στήλες και γραμμές. Κάθε «κελί» σε ένα τέτοιο πίνακα είναι ένα εικονοστοιχείο. Ο αριθμός των υποδιαίρεσεων είναι επαρκώς μεγάλος, τόσο ώστε το ανθρώπινο μάτι να μη μπορεί να διακρίνει το ένα εικονοστοιχείο από το άλλο και να βλέπει την εικόνα ενιαία. [4]



Εικόνα 1.3.7. Εικονοστοιχείο (pixel). Η θέση κάθε pixel ορίζεται από τις τιμές x & y.

Μια ψηφιογραφική εικόνα μπορεί να αποτελείται από εκατομμύρια pixels. Το πλήθος των pixels για συγκεκριμένες διαστάσεις της εικόνας, καθορίζουν την ποιότητα της και αναφέρονται στην **ανάλυση** (resolution) της, η οποία εκφράζει τον αριθμό των pixels στο μήκος της μιας ίντσας (pixels per inch, ppi). Η δειγματοληψία εδώ είναι ένα φαινόμενο στον χώρο και όχι στον χρόνο όπως ήταν στην περίπτωση του ήχου.

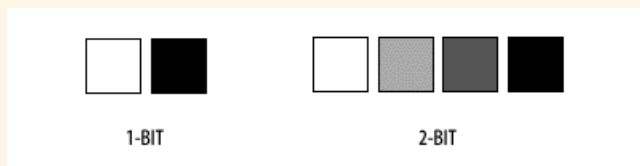
Οι τιμές ανάλυσης σε εικονοστοιχεία μπορούν να εκφραστούν:

- είτε ως ενιαίος αριθμός, όπως σε μια ψηφιακή φωτογραφική μηχανή, π.χ. «3 megapixel». Αυτό σημαίνει ότι η μηχανή έχει τρία εκατομμύρια εικονοστοιχεία,
- ή ως ζεύγος αριθμών, όπως, π.χ. εικόνα "640 επί 480". Η έκφραση αυτή σημαίνει ότι η εικόνα έχει 640 εικονοστοιχεία κατά μήκος και 480 καθ' ύψος και επομένως συνολικό αριθμό $640 \times 480 = 307.200$ εικονοστοιχεία (ή 0,3 megapixels). [4]

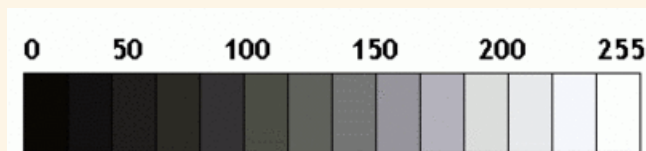
Στη συνέχεια τα δείγματα θα πρέπει να κβαντιστούν, δηλαδή να μετατραπούν οι τιμές τους σε πεπερασμένο αριθμό εκφρασμένο στο δυαδικό σύστημα. Πόσα bit θα χρειαστούν για να εκφραστούν οι τιμές των δειγμάτων στο δυαδικό σύστημα;

Ένα άλλο χαρακτηριστικό που καθορίζει την ποιότητα της εικόνας είναι το **βάθος χρώματος (bit depth)**, δηλαδή το πλήθος των bit που χρησιμοποιούνται για την καταγραφή της αξίας κάθε pixel. Μια μονόχρωμη εικόνα με βάθος 2 bit έχει $2^2 = 4$ διαφορετικές τιμές (διαβαθμίσεις) για τη χρωματική περιγραφή κάθε pixel, ενώ μια εικόνα των 3 bit θα έχει $2^3 = 8$ διαβαθμίσεις. Συνήθως χρησιμοποιούνται $2^8 = 256$ διαβαθμίσεις, όπου η τιμή 0 αντιστοιχεί στο μαύρο χρώμα, η τιμή 256 στο άσπρο και οι 254 διαβαθμίσεις ανάμεσα τους αντιστοιχούν στους διαφορετικούς τόνους του γκρι.

Οι παρακάτω εικόνες μας δείχνουν τις διαβαθμίσεις ανάλογα με τον αριθμό των bits που χρησιμοποιούνται.

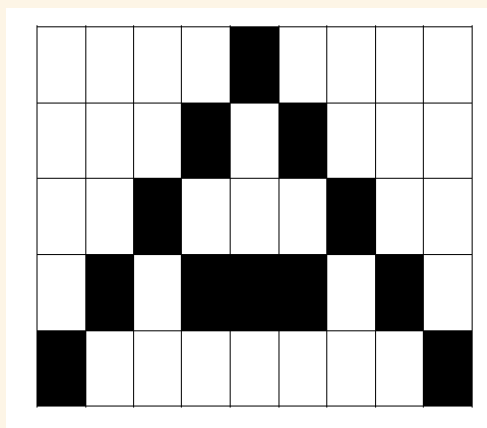


Εικόνα 1.3.7.α. Διαβαθμίσεις με 1 και 2 bit. Πηγή: http://users.softlab.ntua.gr/~klaskar/IEK/PsifiakiEikona_040603.pdf



Εικόνα 1.3.7.β Διαβαθμίσεις με 8 Bits. Πηγή: https://www.esa.int/SPECIALS/Eduspace_GR/SEM3JR4PVFG_0.html

Παράδειγμα: Έστω η παρακάτω ασπρόμαυρη εικόνα που αποτελείται από 45 pixels τα οποία βρίσκονται σε μια από τις δύο καταστάσεις, άσπρο ή μαύρο. Με δεδομένο ότι έχουμε μόνο δυο διαβαθμίσεις, θα χρησιμοποιήσουμε 1 bit για τη χρωματική περιγραφή κάθε pixel, όπου το 0 αντιστοιχεί στο άσπρο χρώμα και το 1 στο μαύρο.



	1	2	3	4	5	6	7	8	9
1	0	0	0	0	1	0	0	0	0
2	0	0	0	1	0	1	0	0	0
3	0	0	1	0	0	0	1	0	0
4	0	1	0	1	1	1	0	1	0
5	1	0	0	0	0	0	0	0	1

Στην ψηφιοποίηση της έγχρωμης εικόνας εφαρμόζουμε τις παραπάνω διαδικασίες με κάπως αυξημένη πολυπλοκότητα. Για την καλύτερη χρωματική απόδοση των έγχρωμων εικόνων, απαιτούνται 24 bit για το βάθος χρώματος, που αντιστοιχούν σε 16,8 εκατομμύρια χρωμάτων.

Για να καταλάβουμε καλύτερα πως λειτουργεί το ψηφιακό χρώμα θα πρέπει να ρίξουμε μια πολύ απλοποιημένη ματιά στο πώς αντιλαμβάνεται τις εικόνες το ανθρώπινο μάτι. Στον αμφιβληστροειδή χιτώνα του ματιού υπάρχουν τρεις διαφορετικές ομάδες αισθητήρων, που ονομάζονται κώνοι, με διαφορετική ευαισθησία η κάθε μία στο φάσμα του φωτός. Η πρώτη ομάδα είναι ευαίσθητη στο μπλε φως, η δεύτερη στο πράσινο και η τρίτη στο κόκκινο χρώμα. Αυτά τα τρία χρώματα ονομάζονται πρωτεύοντα χρώματα για τον άνθρωπο. Όλα τα υπόλοιπα χρώματα που αντιλαμβάνεται το ανθρώπινο μάτι (περίπου 7 εκατομμύρια χρώματα) προκύπτουν από το συσχετισμό του ποιες από τις τρεις ομάδες κώνων αντιδρούν στο ορισμένο ερέθισμα και σε τι ποσοστό η κάθε μία. Με την ταυτόχρονη και ισοδύναμη διέγερση και των τριών ομάδων, προκύπτει το λευκό χρώμα.

Όπως είναι εμφανές από τα παραπάνω, αναπτύχθηκαν διάφορα χρωματικά μοντέλα για την τυποποίηση των χρωμάτων, δηλαδή την ακριβή περιγραφή κάθε χρώματος από μια αυστηρά καθορισμένη μαθηματική μορφή, κατάλληλη για την επεξεργασία τους από ψηφιακά μέσα.

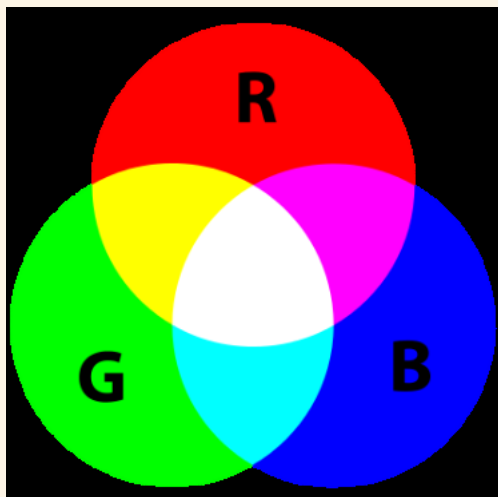
Ανάλογα με τη χρήση κάθε εικόνας, υιοθετείται και το ανάλογο χρωματικό μοντέλο.

Θα εξετάσουμε τα πιο σημαντικά χρωματικά μοντέλα.

Χρωματικό μοντέλο RGB

Η ονομασία είναι ακρωνύμιο των λέξεων Red Green Blue (Κόκκινο Πράσινο Μπλε). Το χρωματικό μοντέλο RGB βασίζεται στα τρία πρωτεύοντα **χρώματα**: Κόκκινο(R), Πράσινο (G) και Μπλε (B) και στο γεγονός πως προσθέτοντας στις κατάλληλες αναλογίες τα τρία αυτά χρώματα μπορούμε να πάρουμε οποιοδήποτε άλλο. Το RGB ονομάζεται **Προσθετικό (additive)** επειδή όλα τα χρώματα που μπορούν να εμφανιστούν σε μία οθόνη (συνήθως υπολογιστή) δημιουργούνται από τη πρόσθεση των σχετικών εντάσεων των τριών πρωτευόντων χρωμάτων.

Η παρακάτω εικόνα μας δείχνει τα πρωτεύοντα χρώματα του μοντέλου καθώς και τα συμπληρωματικά χρώματα που δημιουργούνται από την ανάμειξη μεταξύ δύο πρωτευόντων (κίτρινο, βιολετί, γαλάζιο). Τέλος, μας δείχνει πως η ανάμειξη και των τριών πρωτευόντων χρωμάτων μας δίνει το λευκό χρώμα.



Εικόνα 1.3.7.γ. Μοντέλο RGB. Πηγή: Wikimedia, <https://commons.wikimedia.org/wiki/File:AdditiveColor.svg>

Για να απεικονίσουμε μια έγχρωμη εικόνα, είναι απαραίτητο να διαχωρίσουμε την εικόνα στα τρία πρωτεύοντα χρώματα, Κόκκινο, Πράσινο και Μπλε (RGB). Κάθε ένα από τα χρώματα αυτά απεικονίζεται με βάθος 8 bit, επομένως θα χρειαστούν $3 \times 8 = 24$ bit για την απεικόνιση κάθε χρώματος. Οι δυνατότητες απεικόνισης διαφορετικών χρωματικών τόνων στα 24 Bit είναι $256 \times 256 \times 256$ επομένως απεικονίζονται 16.777.216 διαφορετικοί τόνοι χρώματος, περισσότερα από τα διπλάσια απ' όσα μπορεί να αντιληφθεί το μάτι.

Μερικά παραδείγματα γνωστών χρωμάτων στο μοντέλο RGB:

Χρώμα	Δεκαδική αναπαράσταση	Δεκαεξαδική αναπαράσταση
Μαύρο:	0,0,0	00,00,00
Λευκό	255,255,255	FF,FF,FF
Κόκκινο	255,0,0	FF,00,00
Πράσινο	0,255,0	00,FF,00
Μπλε	0,0,255	00,00,FF
Κίτρινο	255,255,0	FF,FF,00
Γαλάζιο	0,255,255	00,FF,FF
Ματζέντα (Magenta)	255,0,255	FF,00,FF
Πορτοκαλί	255,102,0	FF,66,00

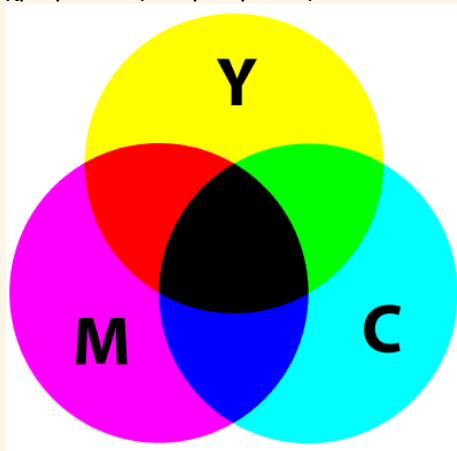
Το χρωματικό μοντέλο RGB ονομάζεται προσθετικό γιατί ένα χρώμα δημιουργείται από την πρόσθεση των σχετικών εντάσεων των τριών πρωτευόντων χρωμάτων στο μαύρο που απεικονίζει την πλήρη έλλειψη φωτός.

Το μοντέλο RGB εφαρμόζεται στα συστήματα που προβάλλουν εικόνες όπως οι οθόνες υπολογιστών, τηλεοράσεων, στις φωτογραφικές μηχανές κ.α.

Χρωματικό μοντέλο CMY ή CMYK

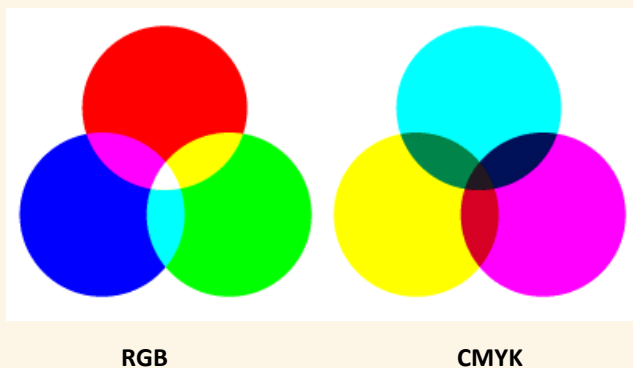
Η αρχή του μοντέλου CMY είναι παρόμοια με του RGB, με τη διαφορά ότι βασίζεται στα συμπληρωματικά χρώματα: Γαλάζιο (Cyan), Βιολετί (Magenta) και Κίτρινο (Yellow). Όπως είδαμε στην

Εικόνα 1.3.7.γ αν συνδυαστούν ανά δύο τα πρωτεύοντα χρώματα του RGB προκύπτουν τα τρία βασικά χρώματα του CMY. Γι' αυτό το λόγο το μοντέλο CMY ονομάζεται αφαιρετικό, επειδή τα χρώματα που προκύπτουν είναι αποτέλεσμα αφαίρεσης ποσοστών κόκκινου, πράσινου και μπλε φωτός. Το μοντέλο αυτό δημιουργήθηκε για να εξυπηρετεί τις τεχνολογίες εκτύπωσης (έντυπα, βιβλία αφίσες κλπ) καθώς αυτές γίνονται πάνω στη λευκή επιφάνεια του χαρτιού. Με το συνδυασμό μελανιών γαλάζιο, βιολετί και κίτρινο μπορούν να εκτυπωθούν στο χαρτί όλες οι χρωματικές διαβαθμίσεις.



Εικόνα 1.3.7.δ. Μοντέλο CMY. Πηγή: <https://www.pcsteps.gr>

Θεωρητικά, αναμειγνύοντας και τα τρία χρώματα του CMY θα έπρεπε να παράγεται το μαύρο. Στην πραγματικότητα όμως κάτι τέτοιο δεν είναι εφικτό λόγω της ατέλειας των μελανιών (δεν μπορούν εύκολα να δώσουν απόλυτα καθαρά μελάνια CYAN, MAGENTA και YELLOW) και αντί για μαύρο παράγεται ένα γκριζο-καφετί χρώμα. Για να αντιμετωπιστεί το πρόβλημα, στις συσκευές εκτύπωσης προστέθηκε ξεχωριστά και το μαύρο μελάνι. Αυτό το νέο μοντέλο ονομάστηκε CMYK. Το CMYK λειτουργεί ακριβώς όπως και το CMY, αλλά στις περιπτώσεις που χρειάζεται να τυπώσει μαύρο χρώμα χρησιμοποιεί ξεχωριστή μαύρη μελάνη. Το "K" που προστέθηκε τα αρχικά σημαίνει Key δηλαδή κλειδί.



RGB

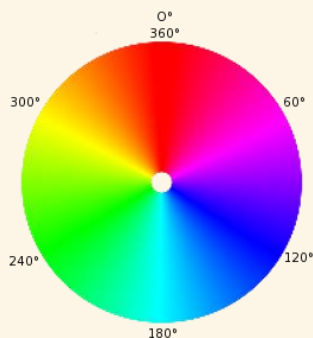
CMYK

Εικόνα 1.3.7.ε. Σύγκριση των δύο μοντέλων.

Χρωματικό μοντέλο HSV

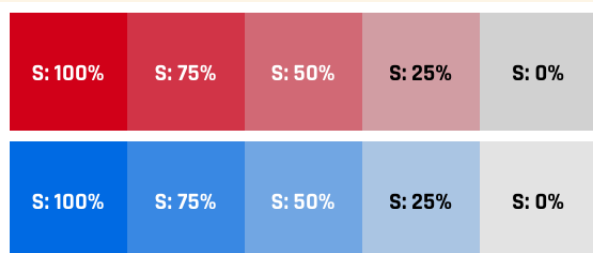
Το μοντέλο αυτό περιγράφει τα χρώματα χρησιμοποιώντας τρεις παραμέτρους: τη χροιά (hue), τον κορεσμό (saturation) και την αξία (value ή lightness).

Η χροιά διαιρεί το χρωματικό κύκλο σε μοίρες γωνιών όπου κάθε χρώμα αντιστοιχίζεται σε μία συγκεκριμένη γωνία από 0° έως 360°.



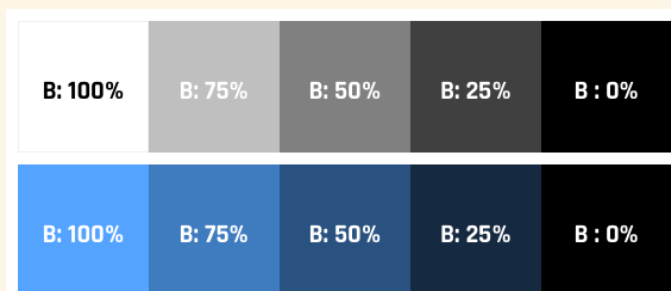
Εικόνα 1.3.7.στ. Ο χρωματικός κύκλος του μοντέλου HSV. Πηγή: <https://docs.gimp.org/2.10/el/glossary.html>

Ο κορεσμός αναφέρεται στην καθαρότητα ενός χρώματος και δίνει το επί τοις εκατό ποσοστό της απόστασης από το κέντρο του χρωματικού κύκλου. Με κορεσμό 100% το χρώμα είναι καθαρό (πλήρως κορεσμένο), ενώ με 0% άσπρο, μαύρο ή γκρι ανάλογα με τη βασική απόχρωση.



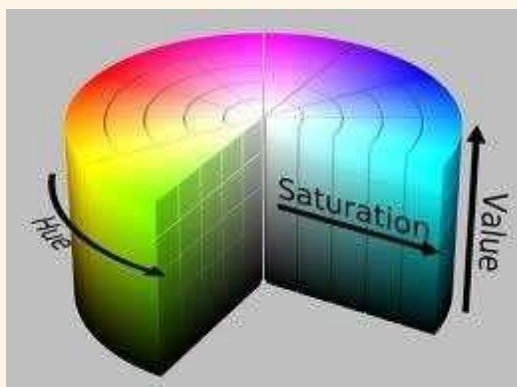
Εικόνα 1.3.7.ζ. Κορεσμός του κόκκινου και του μπλε χρώματος. Πηγή: <https://www.learnui.design/blog/the-hsb-color-system-practicioners-primer.html>

Η αξία ή φωτεινότητα δηλώνει την ένταση του χρώματος, δηλαδή το ποσοστό (0-100%) του φωτός που εκπέμπεται από ένα χρώμα.



Εικόνα 1.3.7.η. Φωτεινότητα χρωμάτων. Πηγή : <https://www.learnui.design/blog/the-hsb-color-system-practicioners-primer.html>

Στην Εικόνα 1.3.7.η παρατηρούμε ότι: α) όλα τα χρώματα με φωτεινότητα 0% δίνουν το μαύρο χρώμα, β) Με φωτεινότητα 100% μας δίνει ένα πολύ φωτεινό χρώμα, αλλά αν ο κορεσμός είναι 0% μας δίνει το λευκό χρώμα.



Εικόνα 1.3.7.θ. Χροιά, κορεσμός, αξία του μοντέλου HSV. Πηγή: http://repfiles.kallipos.gr/html_books/50/Chapter_5/index.html

Αυτό το μοντέλο παρουσιάζει το πλεονέκτημα ότι συνδέει τις παραμέτρους του με βασικές έννοιες του χρώματος, κάτι που το καθιστά εύκολο στη χρήση του σε λογισμικά επεξεργασίας εικόνας.

Διανυσματικές εικόνες

Σε αντίθεση με την ψηφιογραφική εικόνα που χρησιμοποιεί ένα πλέγμα εικονοστοιχείων για την αναπαράσταση της εικόνας, η διανυσματική εικόνα αναλύεται σε ένα συνδυασμό από γεωμετρικά σχήματα (σημεία, γραμμές, καμπύλες, ορθογώνια, ελλείψεις, πολύγωνα κ.α.) που αναπαρίστανται από ένα μαθηματικό τρόπο.

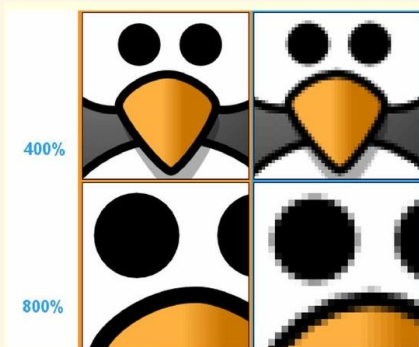
Για παράδειγμα, ένας κύκλος περιγράφεται από τις συντεταγμένες του κέντρου του, το μήκος της ακτίνας του και την περιγραφή των χαρακτηριστικών του όπως το πάχος γραμμής και το χρώμα γεμίματος.

Ο σχεδιασμός τέτοιων εικόνων γίνεται από σχεδιαστικά πακέτα όπως το Adobe Illustrator, Corell, AutoCad, Fireworks, Photoshop κ.α.

Αυτός ο τύπος εικόνας χρησιμοποιείται συχνά για λογότυπα, εικονογραφήσεις και γενικά για σχεδιασμό γραφικών από αρχιτέκτονες και γραφίστες.

Τα βασικά χαρακτηριστικά αυτής της κατηγορίας είναι ότι:

1. Μια εικόνα μπορεί να κλιμακωθεί σε οποιοδήποτε μέγεθος χωρίς καμία απώλεια ποιότητας. Αντίθετα, στις ψηφιογραφικές εικόνες τα εικονοστοιχεία μπορεί να γίνουν θολά ή να παραμορφωθούν όταν μεγεθύνονται.



Εικόνα 1.3.7.1. Σύγκριση διανυσματικής (αριστερά) και ψηφιογραφικής (δεξιά) εικόνας μετά από μεγέθυνση.

2. Αποθηκεύονται σε πολύ μικρότερα αρχεία από ότι οι ψηφιογραφικές εικόνες επειδή αποθηκεύονται μόνο οι πληροφορίες για το σχεδιασμό τους.

Τύποι αρχείων εικόνας - Συμπύεση

Σήμερα έχουμε στη διάθεση μας πολλούς και διαφορετικούς τύπους αρχείων εικόνας. Κάθε τύπος αρχείου δημιουργήθηκε με σκοπό να χρησιμοποιηθεί σε συγκεκριμένες εφαρμογές (εκτύπωση, δημοσίευση σε ιστοσελίδα κ.λ.π) και με συγκεκριμένο τρόπο.

Όταν αποθηκεύουμε μια εικόνα με συγκεκριμένο τύπο αρχείου, στην ουσία λέμε στην εφαρμογή που χρησιμοποιούμε πώς να γράψει τις πληροφορίες της εικόνας στο μέσο αποθήκευσης.

Όταν πρόκειται για ψηφιογραφική εικόνα, οι πληροφορίες της εικόνας αποθηκεύονται ως διάταξη από ρικελς. Όταν πρόκειται για διανυσματική εικόνα, οι πληροφορίες αποθηκεύονται ως μαθηματικά δεδομένα.

Ας ρίξουμε μια ματιά στα πλεονεκτήματα και στα μειονεκτήματα των ψηφιογραφικών και διανυσματικών αρχείων γραφικών :

Ψηφιογραφικά (bitmap) αρχεία:

- + Υψηλός βαθμός φωτορεαλισμού και πιστότητας της εικόνας
- Πολύ μεγάλο μέγεθος αρχείων
- Αλλαγές στην κλίμακα ή /και στο μέγεθος δεν εφαρμόζονται εύκολα
- Δυσκολίες παρουσιάζονται κατά την επεξεργασία τους

Διανυσματικά (vector) αρχεία:

- + Αρκετά μικρό μέγεθος αρχείων
- + Εύκολη αλλαγή κλίμακας και μεγέθους, χωρίς αισθητές απώλειες στην ανάλυση
- + Εύκολη επεξεργασία
- Συχνά απαιτούν αρκετό από τον χρόνο μας για τον σχεδιασμό τους (αναλόγως του μηχανήματος)
- Δεν προτείνεται αυτός ο τύπος για φωτορεαλιστικές εικόνες

Το μεγάλο μέγεθος των ψηφιογραφικών εικόνων οδήγησε στην επινόηση τρόπων συμπίεσης χωρίς να γίνονται ιδιαίτερα αντιληπτές οι απώλειες που μπορεί να υπάρχουν στην εικόνα από το ανθρώπινο μάτι. Έτσι επινοήθηκαν δυο τρόποι συμπίεσης της εικόνας:

α) Αλγόριθμοι συμπίεσης χωρίς απώλεια δεδομένων (lossless)

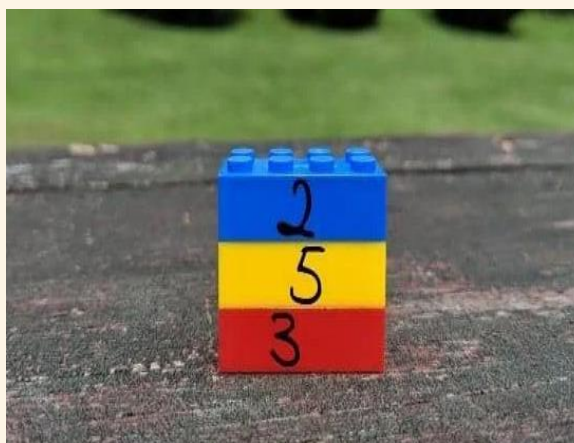
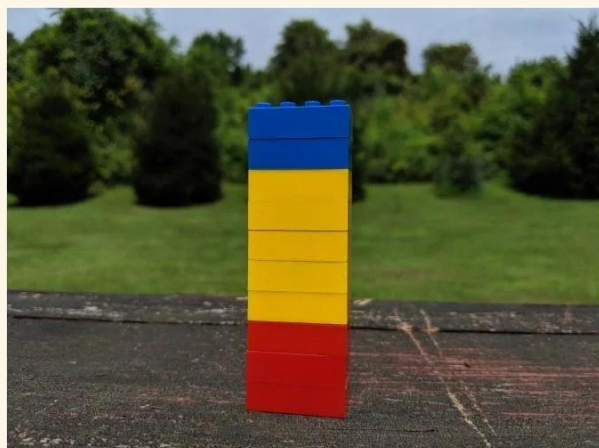
β) Αλγόριθμοι συμπίεσης με απώλεια δεδομένων (lossy)

Αλγόριθμοι συμπίεσης χωρίς απώλεια δεδομένων (lossless)

Οι αλγόριθμοι αυτοί βασίζονται στην αποφυγή αποθήκευσης των επαναλαμβανόμενων πληροφοριών που εμφανίζονται σε μια ομάδα από ριξελς. Το μέγεθος της συμπίεσης εξαρτάται λοιπόν από το εάν έχει η εικόνα πολλά ριξελς που είναι όμοια μεταξύ τους. Τον αλγόριθμο αυτό χρησιμοποιούν οι εικόνες TIFF (*.TIF), PNG (*.PNG) και οι εικόνες PHOTOSHOP (*.PSD).

Πώς μπορούμε να κάνουμε ένα τμήμα πληροφορίας μικρότερο από το πραγματικό του μέγεθος χωρίς να χάσουμε κανένα τμήμα της πληροφορίας;

Ας δούμε το παράδειγμα της Εικόνας 1.3.6.3 για να καταλάβουμε πως λειτουργεί ο αλγόριθμος. Αντί να δείξουμε και τα 10 τούβλα (αριστερή εικόνα), μπορούμε να δείξουμε ένα από κάθε χρώμα αφαιρώντας όλα τα υπόλοιπα (δεξιά εικόνα). Στη συνέχεια, αν χρησιμοποιήσουμε αριθμούς για να δείξουμε πόσα τούβλα από κάθε χρώμα υπήρχαν, έχουμε αντιπροσωπεύσει ακριβώς το ίδιο κομμάτι πληροφοριών χρησιμοποιώντας πολύ λιγότερα τούβλα. Αντί για 10 τούβλα, χρειαζόμαστε τώρα μόνο τρία.



Εικόνα 1.3.7.κ. Συμπίεση χωρίς απώλεια δεδομένων. Πηγή: <https://iguru.gr/compression-work/>

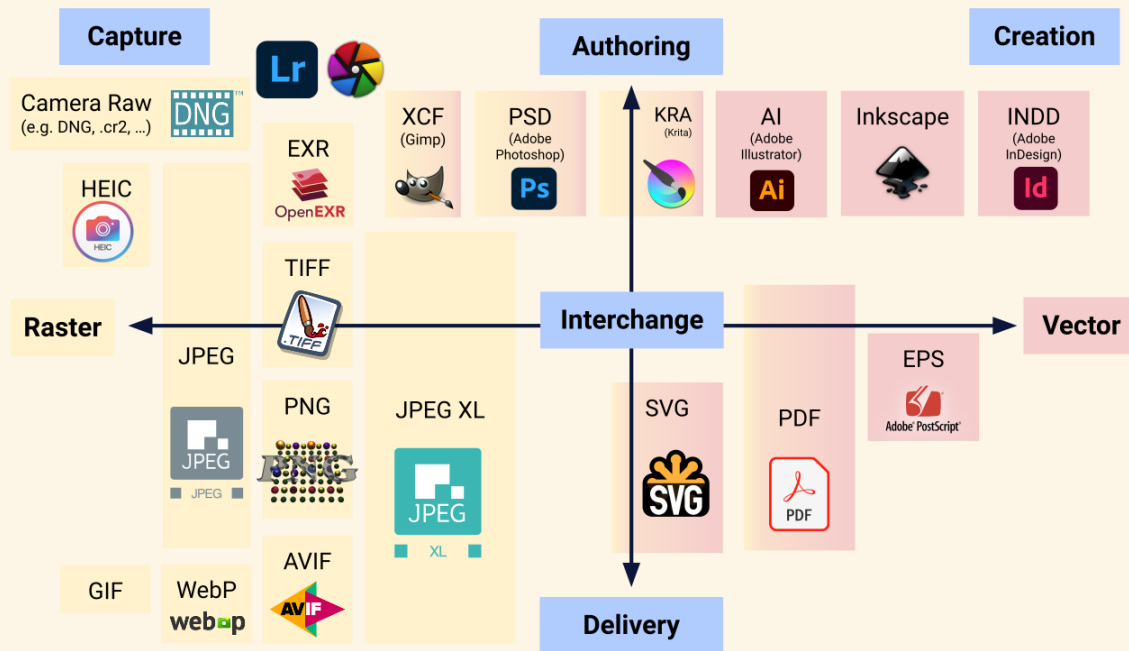
Η συμπίεση χωρίς απώλειες γενικά, αλλά όχι πάντα, έχει ως αποτέλεσμα μεγαλύτερα αρχεία από τη συμπίεση με απώλειες.

Αλγόριθμοι συμπίεσης με απώλεια δεδομένων (lossy)

Η συμπίεση με απώλεια δεδομένων μειώνει το μέγεθος του αρχείου, αφαιρώντας τα περιττά τμήματα πληροφοριών. Συνήθως χρησιμοποιείται σε αρχεία εικόνας, βίντεο και ήχου, όπου δεν είναι απαραίτητη η τέλεια αναπαράσταση της αρχικής πηγής. Η μέθοδος αυτή βασίζεται στο γεγονός ότι το ανθρώπινο μάτι δεν έχει την δυνατότητα να αντιληφθεί όλα τα μήκη κύματος των χρωμάτων που μπορεί να παράγει η οθόνη. Κόβοντας λοιπόν συγκεκριμένες αποχρώσεις, πρακτικά έχουμε την ίδια ποιότητα με μικρότερο όγκο.

Σήμερα το πιο διαδεδομένο πρότυπο συμπίεσης με απώλειες δεδομένων είναι το JPEG (Joint Photographic Experts Group). Χρησιμοποιείται για εικόνες και φωτογραφίες που επιβάλλεται να είναι μικρές σε μέγεθος αρχείου για να μπορούν να προβληθούν στο διαδίκτυο ή να σταλούν με mail. Το JPEG υποστηρίζει λόγο συμπίεσης της τάξης 20:1 με ικανοποιητική ποιότητα στην εικόνα, ενώ μπορεί να υποστηρίξει συμπίεση μέχρι 75:1 με απώλειες δεδομένων. Πριν συμπιέσουμε το αρχείο μας μπορούμε να επιλέξουμε τον σωστό συνδυασμό συμπίεσης – ποιότητας της JPEG συμπίεσης.

Στην παρακάτω εικόνα εμφανίζονται οι τύποι αρχείων ανά κατηγορία εικόνων και το σκοπό που εξυπηρετεί ο κάθε τύπος.



Εικόνα 1.3.7.λ. Πηγή: https://upload.wikimedia.org/wikipedia/commons/thumb/c/ce/Image_formats_by_scope.svg/990px-Image_formats_by_scope.svg.png

Βασικές μορφοποιήσεις ψηφιογραφικών αρχείων εικόνας

Όνομα Μορφοποίησης	Επέκταση Αρχείου	Πλεονεκτήματα	Μειονεκτήματα	Συνηθισμένη Χρήση
TIFF (Tagged Image File)	.tif	Συμπίεσμένη ή Ασυμπίεστη πληροφορία χωρίς απώλειες. Υποστηρίζεται από όλους τους εκτυπωτές και σαρωτές	Μεγάλο μέγεθος αρχείου. Δύσκολο στην αποθήκευση	Ψηφιακές εικόνες με όλη την αρχική πληροφορία διαθέσιμη
GIF (CompuServe Graphics Interface Format)	.gif	Μικρό μέγεθος, εύκολη αποθήκευση και μεταφορά, υποστηρίζει διαφάνεια στο φόντο του σχεδίου, παρέχει τη δυνατότητα animation	Περιορισμός χρωμάτων στα 256	Εικόνες grayscale, εικόνες με λίγα χρώματα
JPEG (Joint Photo graphics Expert Group)	.jpg	Προσφέρει «πραγματικό χρώμα» (16,8 εκ. χρώματα) και δυνατότητα καθορισμού του βαθμού συμπίεσης	Συμπίεση με απώλειες, δεν υποστηρίζει διαφάνεια	Παρουσίαση ψηφιογραφικών εικόνων (π.χ. φωτογραφίες στο Web)

PCX (PC Paintbrush Format)	.pcx	Ασυμπίεστη πληροφορία, καλή ποιότητα εικόνας, Βάθος χρώματος 24 bit	Μεγάλο μέγεθος αρχείου	Στο λογισμικό Paintbrush - Ζωγραφική
BMP (Standard Windows Bitmap)	.bmp	Αποδίδει καλά σε εφαρμογές των Windows και του OS/2	Δεν υποστηρίζει διαφάνεια, μεγάλο μέγεθος αρχείου	Σε εφαρμογές των Windows
PSD (PhotoShop images)	.psd	Διατηρεί πλήθος πληροφοριών (π.χ. Channels, layers, paths) για την ολοκληρωμένη επεξεργασία της εικόνας	Συμβατό μόνο με συγκεκριμένες εφαρμογές. Δεν χρησιμοποιείται στο Web	Ενδιάμεση φόρμα αποθήκευσης για εξειδικευμένη επεξεργασία εικόνας
PNG (Portable Network Graphics)	.png	Υποστηρίζει συμπίεση χωρίς απώλεια δεδομένων και διαφάνεια στο φόντο	Σχεδιάστηκε με απότερο στόχο την αντικατάσταση του προτύπου GIF και αποκλειστικά για την προβολή εικόνων μέσω internet	

Βασικές μορφοποιήσεις διανυσματικών αρχείων εικόνας

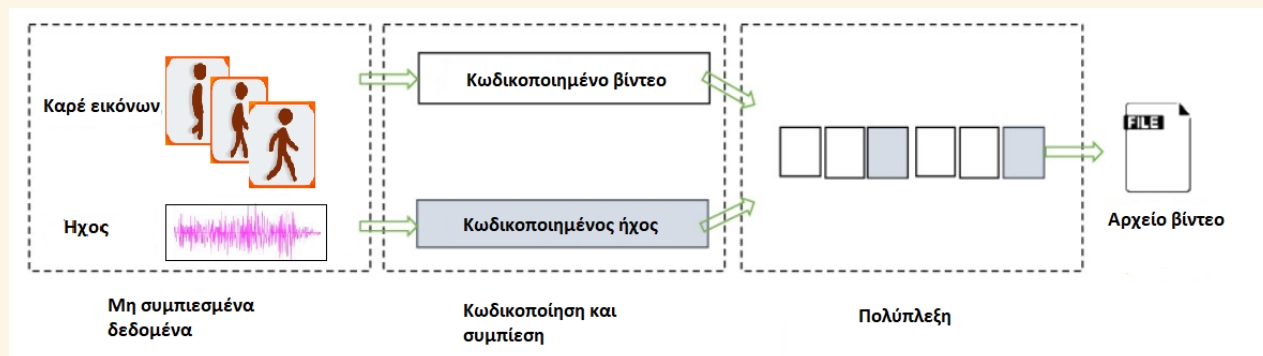
Όνομα Μορφοποίησης	Επέκταση Αρχείου	Πλεονεκτήματα	Μειονεκτήματα	Συνηθισμένη Χρήση
WMF (MS Word MetaFile)	.wmf	Μικρό μέγεθος αρχείου, εύκολο στην αποθήκευση	Μόνο γραφικά	Διανυσματικά γραφικά για εφαρμογές Πολυμέσων, MS Office, περιβάλλον Windows
EMF (MS Word Enhanced File)	.emf	Εύκολο στην αποθήκευση	Μέγεθος αρχείου μεγαλύτερο από τις wmf	Διανυσματικές εικόνες σε περιβάλλον Windows, για εφαρμογές Πολυμέσων, MS Office
AUTOCAD	.DWG ή DXF		Συμβατό μόνο με συγκεκριμένες εφαρμογές	Διανυσματικά σχέδια για εφαρμογές CAD/CAM όπως το AutoCad
Corel Draw images	.cdr	Δυνατότητες εφέ και ολοκληρωμένη επεξεργασία εικόνας	Συμβατό μόνο με συγκεκριμένες εφαρμογές, μεγάλο μέγεθος αρχείου	Ενδιάμεση φόρμα αποθήκευσης για επεξεργασία εικόνας στο Corel Draw
Adobe Illustrator	.AI	Δυνατότητες εφέ και ολοκληρωμένη επεξεργασία εικόνας	Συμβατό μόνο με συγκεκριμένες εφαρμογές, μεγάλο μέγεθος αρχείου	Ενδιάμεση φόρμα αποθήκευσης για επεξεργασία εικόνας στο Illustrator
SVG (Scalable Vector Graphics)	.svg	Μπορούν να επεξεργαστούν με οποιοδήποτε επεξεργαστή κειμένου, να εκτυπωθούν με υψηλή ποιότητα σε οποιαδήποτε ανάλυση, να μεγεθυνθούν (zoom), ή να γίνει αλλαγή του μεγέθους τους (resize) χωρίς υποβάθμιση της ποιότητας		Ανοιχτό πρότυπο, αναπτύχθηκε από την W3C και ορίζει τα γραφικά σε μορφή XML. Υποστηρίζει αλληλεπιδραστικά γραφικά Web υψηλής ποιότητας

1.3.8 Ψηφιακή παράσταση βίντεο

Με τον όρο βίντεο περιγράφουμε την συνεχή εναλλαγή εικόνων (καρέ) σε καθορισμένο χρόνο. Έτσι δίνεται η εντύπωση της κινούμενης εικόνας. Ο αριθμός των καρέ ανά δευτερόλεπτο ονομάζεται ρυθμός εναλλαγής (frame rate per second / fps). Ο ρυθμός εναλλαγής καθορίζει την ομαλότητα της εικόνας, αλλά και το μέγεθος του αρχείου. Θεωρητικά όσο πιο πολλά καρέ έχουμε το δευτερόλεπτο τόσο πιο ομαλά καταγράφεται η κίνηση. Για να δίνεται η αίσθηση της ομαλής κίνησης χωρίς απότομες αλλαγές στις εικόνες, πρέπει ο ρυθμός εναλλαγής να είναι μεταξύ 25 fps και 30 fps. Βέβαια, έτσι μεγαλώνει και το μέγεθος του αρχείου, γιατί κάθε απόσπασμα ασυμπίεστου βίντεο έχει μέγεθος όσο το συνολικό μέγεθος των πλαισίων που το αποτελούν. Ωστόσο, με τεχνικές συμπίεσης το μέγεθος του βίντεο μειώνεται.

Παράλληλα με τη ροή των εικόνων υπάρχει και ηχητική πληροφορία που συμπληρώνει το βίντεο.

Ένα ψηφιακό αρχείο βίντεο αποθηκεύεται ως μια ακολουθία εικόνων, η οποία συνοδεύεται και από ένα ηχητικό σήμα που έχουν κωδικοποιηθεί και συμπιεστεί. Επίσης, στο αρχείο περιλαμβάνονται τυχόν υπότιτλοι και μεταδεδομένα (τίτλος, δημιουργός κ.α).



Εικόνα 1.3.8. Ψηφιακό αρχείο βίντεο.



Δραστηριότητες - Ασκήσεις

1. Έστω ότι έχουμε μία εικόνα μεγέθους 300px*300px, έγχρωμη, στο πρότυπο RGB. Πόσο χώρο θα καταλάμβανε η εικόνα αυτή στο δίσκο; Το μέγεθος της εικόνας είναι $300 * 300 = 90.000$ pixels συνολικά.
2. Αν η παραπάνω εικόνα ήταν ασπρόμαυρη, πόσο χώρο θα καταλάμβανε στο δίσκο;
3. Επιλέξτε τη σωστή απάντηση:
Όταν διπλασιάζονται οι διαστάσεις της εικόνας, αλλά το βάθος χρώματος παραμένει σταθερό, τότε το μέγεθος της εικόνας
 - a. Υποδιπλασιάζεται
 - b. Τετραπλασιάζεται
 - c. Διπλασιάζεται
4. Ποιο είναι το πλεονέκτημα και ποιο το μειονέκτημα των διανυσματικών εικόνων έναντι των ψηφιογραφικών;
5. Επισκεφθείτε τη σελίδα <https://photodentro.edu.gr/lor/r/8521/975?locale=el> και χρησιμοποιήστε τη διαδραστική εφαρμογή που εμφανίζει την ίδια εικόνα παράλληλα σε ψηφιογραφική και διανυσματική μορφή και επιτρέπει τη σύγκρισή τους μέσα από τη δυναμική μεταβολή της κλίμακας απεικόνισης της κάθε μορφής. Πειραματιστείτε μεταβάλλοντας την κλίμακα των δύο μορφών εικόνων και συγκρίνοντάς το αποτέλεσμα ως προς την ποιότητα αναπαράστασης.
6. Επισκεφθείτε τη σελίδα [Describing Colours | Representation | Computing \(advanced-ict.info\)](#) και συνθέστε χρωματικές αποχρώσεις στο πρότυπο RGB.
7. Επισκεφθείτε τη σελίδα [Photodentro: Αναπαράσταση ασπρόμαυρης ψηφιακής εικόνας στον υπολογιστή](#) και χρησιμοποιήστε τη διαδραστική εφαρμογή για να συνθέσετε μια ασπρόμαυρη εικόνα είτε μέσω χρωματισμού, με άσπρο ή μαύρο, των εικονοστοιχείων στην οθόνη του υπολογιστή είτε αποδίδοντας την αντίστοιχη τιμή, 0 ή 1, για το χρώμα (άσπρο ή μαύρο) στο συγκεκριμένο εικονοστοιχείο. Επίσης, μπορείτε να διαλέξετε από τη διαθέσιμη βιβλιοθήκη έτοιμων εικόνων

ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ

1. Βασικά Θέματα Πληροφορικής – Β ΕΠΑΛ
2. <https://el.wikipedia.org/wiki/MP3>
3. https://el.wikipedia.org/wiki/Συμπίεση_δεδομένων
4. <https://el.wikipedia.org/wiki/Εικονοστοιχείο>
5. Μάρκος Δενδρινός, Δημήτρης Κούης . Βασικές Αρχές και Τεχνολογίες στην Επιστήμη της Πληροφόρησης. www.kallipos.gr
6. Σιδερίδης, Α., Γιαλούρης. Κ., Παπαδόπουλος. Α., Σταθόπουλος. Κ. (2001). Βασικές Αρχές Ψηφιακής Τεχνολογίας, Παιδαγωγικό Ινστιτούτο, Αθήνα.
7. What Every Programmer Absolutely, Positively Needs to Know About Encodings and Character Sets to Work With Text by David C. Zentgraf
8. Wikipedia- ISO/IEC 8859
9. Wikipedia (2014). Unicode
10. Behrouz Forouzan. Εισαγωγή στην επιστήμη των υπολογιστών - 3η έκδοση Κλειδάριθμος
11. Ceruzzi P. (2003). A History of Modern Computing (2nd edition) MIT Publishing.
12. Information in the Context of Philosophy and Cognitive Sciences