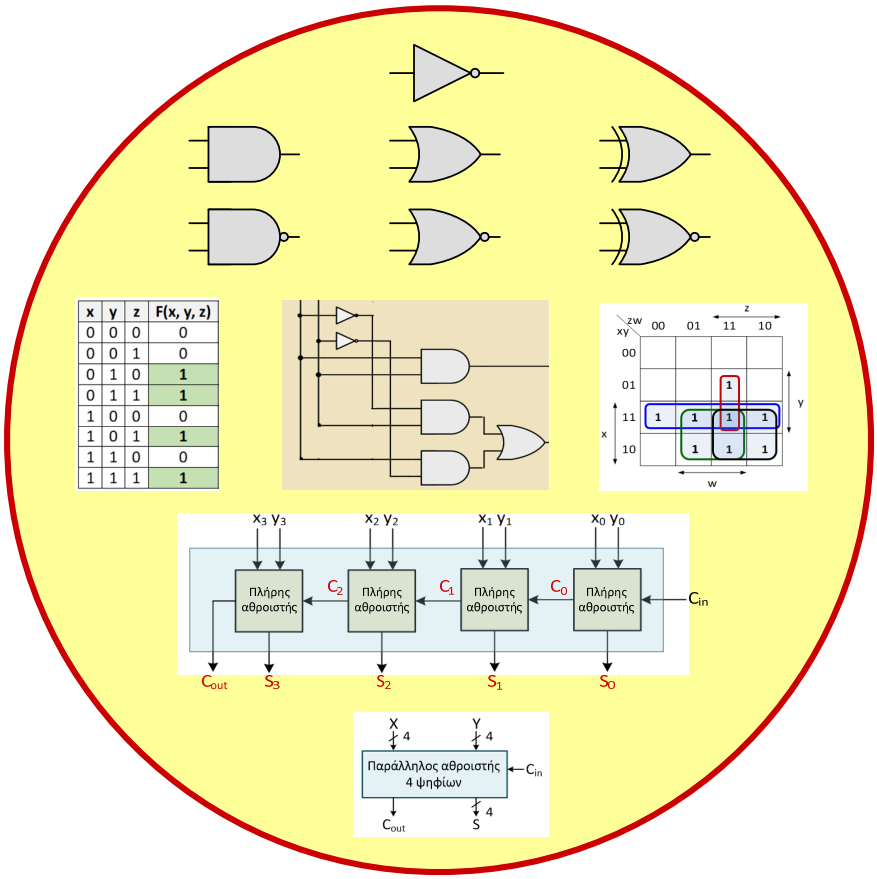
ΨΗΦΙΑΚΗ ΛΟΓΙΚΗ ΣΧΕΔΙΑΣΗ

– ΣΗΜΕΙΩΣΕΙΣ ΔΙΔΑΣΚΑΛΙΑΣ –



Λάμπρος Μπισδούνης

Καθηγητής

Πάτρα 2022

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Περιεχόμενα του μαθήματος

Ενότητα 1: Αναλογικά και ψηφιακά σήματα, μετατροπή σημάτων, ψηφιακά συστήματα.

Ενότητα 2: Αριθμητικά συστήματα (με έμφαση στο δυαδικό σύστημα), αριθμητικές πράξεις, αριθμοί κινητής υποδιαστολής, δυαδικοί κώδικες.

Ενότητα 3: Μαθηματική και δυαδική λογική, λογικές πύλες και άλγεβρα Boole, λογικές συναρτήσεις και λογικά κυκλώματα.

Ενότητα 4: Ελαχιστοποίηση λογικών συναρτήσεων με τη μέθοδο του χάρτη Karnaugh, υλοποίηση ελαχιστοποιημένων συναρτήσεων, μερικώς καθορισμένες συναρτήσεις.

Ενότητα 5: Σύνθεση και ανάλυση συνδυαστικών κυκλωμάτων και τυποποιημένα συνδυαστικά κυκλώματα (αθροιστές, αφαιρέτες, πολλαπλασιαστές, κωδικοποιητές, αποκωδικοποιητές, πολυπλέκτες, αποπολυπλέκτες).

1. Αναλογικά και ψηφιακά σήματα, μετατροπή σημάτων, ψηφιακά συστήματα

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Ψηφιακή και αναλογική παράσταση σημάτων

Στις επιστήμες, την τεχνολογία, την επιχειρηματικότητα, χειριζόμαστε πολύ συχνά ποσοτικά μεγέθη, που μετρούνται, συλλέγονται, επεξεργάζονται, καταγράφονται ή αποθηκεύονται και είναι σημαντικό να παριστάνονται με αποδοτικό και ακριβή τρόπο.

Οι βασικοί τρόποι παράστασης ποσοτικών μεγεθών είναι δύο: ο αναλογικός και ο ψηφιακός και η μορφή της πληροφορίας λέγεται αναλογικό ή ψηφιακό σήμα, αντίστοιχα.

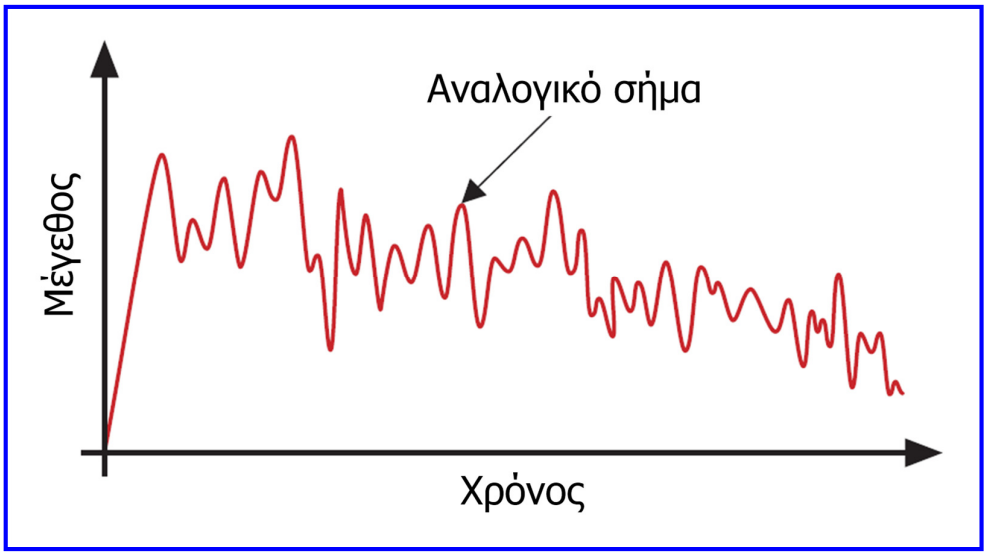
Παραδείγματα: επιλογές του διακόπτη που ελέγχει το «μάτι» ηλεκτρικής κουζίνας (ψηφιακό), ρεύμα που διαρρέει ηλεκτρικό κύκλωμα (αναλογικό), θερμοκρασία χώρου (αναλογικό), πάπιες σε λίμνη (ψηφιακό).

Ψηφιακή και αναλογική παράσταση σημάτων

Στην αναλογική παράσταση, ένα μέγεθος λαμβάνει συνεχείς τιμές, ενώ στην ψηφιακή παράσταση λαμβάνει διακριτές τιμές.

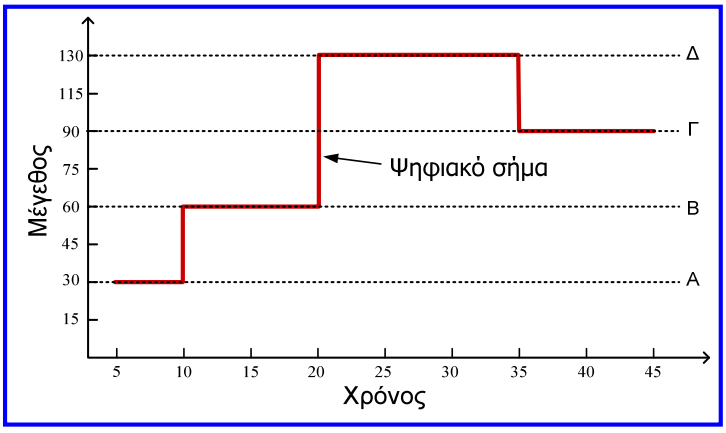
Τα περισσότερα μεγέθη που μετρούνται ποσοτικά, όπως θερμοκρασία, πίεση, ταχύτητα, ένταση ηλεκτρικού ρεύματος κ.ά. εμφανίζονται στη φύση με αναλογική μορφή και επειδή τα ψηφιακά συστήματα χειρίζονται ψηφιακά σήματα, απαιτούνται κατάλληλα συστήματα μετατροπής, τα οποία επιτελούν τη μετατροπή ενός μεγέθους που παριστάνεται με αναλογικό τρόπο σε ψηφιακή μορφή.

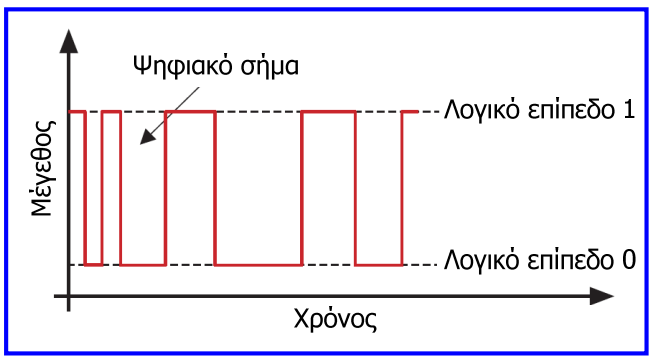
Αναλογικό σήμα



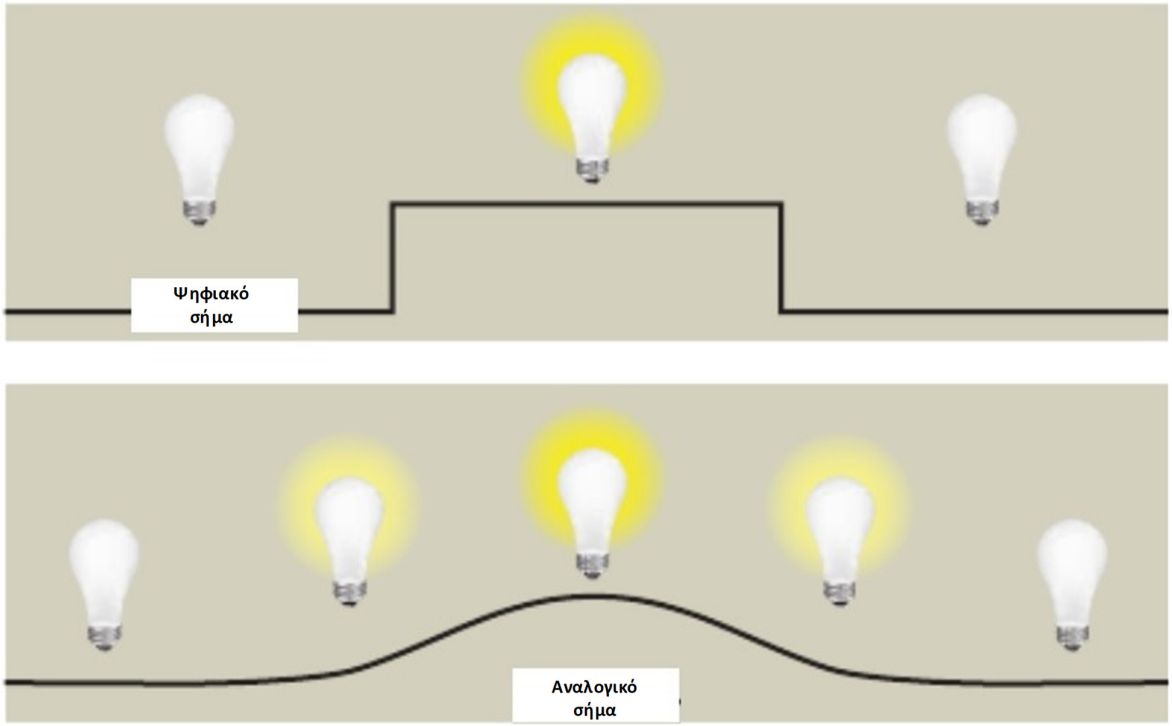
Αναλογική παράσταση μεγέθους (αναλογικό σήμα): γραφική παράστασή ως προς το χρόνο.

Ψηφιακό σήμα





Ψηφιακά και αναλογικά σήματα



Μετατροπή αναλογικών σημάτων σε ψηφιακά

Όπως αναφέρθηκε η αναλογική παράσταση μεγέθους (αναλογικό σήμα) αντιστοιχεί στη γραφική παράστασή του με το χρόνο.

Δειγματοληψία (sampling): λήψη δειγμάτων του μεγέθους σε διακριτές χρονικές στιγμές.

Η αναπαράσταση της τιμής κάθε δείγματος με έναν αριθμό πεπερασμένου πλήθους ψηφίων, αποτελεί την ψηφιοποίηση (digitization), αποτέλεσμα της οποίας είναι η ψηφιακή παράσταση του μεγέθους, δηλαδή μια απλή ακολουθία αριθμών που αναπαριστά το μέγεθος των διαδοχικών δειγμάτων (ψηφιακό σήμα).

Η ψηφιοποίηση προϋποθέτει αρχικά τη διαίρεση της κλίμακας του μεγέθους σε επίπεδα (στάθμες), ώστε κάθε δείγμα του μεγέθους να αντιστοιχιστεί στο πλησιέστερο επίπεδο, διαδικασία που αναφέρεται ως κβαντισμός (quantization).

Μετατροπή αναλογικών σημάτων σε ψηφιακά

Η διαδικασία της ψηφιοποίησης ολοκληρώνεται με την κωδικοποίηση (encoding, coding), κατά την οποία κάθε επίπεδο κβαντισμού και κατά συνέπεια κάθε δείγμα που αντιστοιχεί στο επίπεδο αυτό, κωδικοποιείται με μία απλή ακολουθία αριθμών.

Συνήθως, ακολουθείται η δυαδική κωδικοποίηση, στην οποία χρησιμοποιούνται δυαδικά ψηφία (binary digits ή bits) που μπορούν να λάβουν δύο τιμές 0 και 1.

Για το λόγο αυτό τα ψηφιακά σήματα που προκύπτουν αναφέρονται ως δυαδικά σήματα.

Μετατροπή αναλογικών σημάτων σε ψηφιακά

Για τη δυαδική κωδικοποίηση 2^N τιμών ενός μεγέθους, απαιτούνται N δυαδικά ψηφία.

Για παράδειγμα, όταν ένα μέγεθος λαμβάνει 2 τιμές (2^1), αυτές μπορούν να κωδικοποιηθούν με 1 δυαδικό ψηφίο που μπορεί να έχει 2 τιμές (0 και 1), δηλαδή μία τιμή για κάθε τιμή του μεγέθους.

$4 = 2^2$ τιμές ενός μεγέθους μπορούν να κωδικοποιηθούν με 2 δυαδικά ψηφία ως: 00, 01, 10, 11.

8 τιμές ενός μεγέθους κωδικοποιούνται με 3 ψηφία ως: 000, 001, 010, 011, 100, 101, 110, 111.

9 έως 16 τιμές ενός μεγέθους κωδικοποιούνται με 4 ψηφία, κ.ο.κ.

Μετατροπή αναλογικών σημάτων σε ψηφιακά

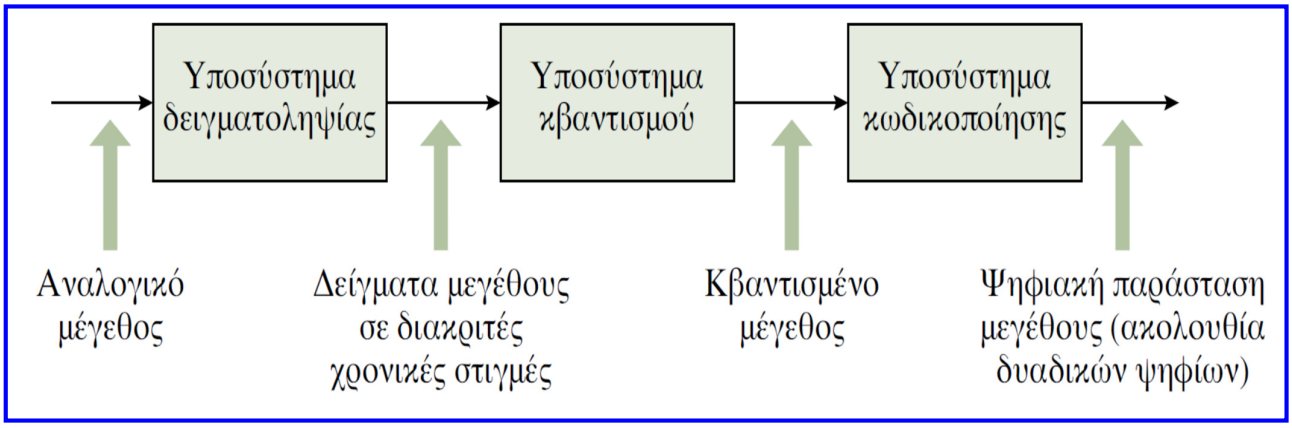
Η παράσταση που προκύπτει από τον κβαντισμό (κβαντισμένο σήμα) και κατά συνέπεια η ψηφιακή παράσταση (ψηφιακό σήμα), δεν αποδίδουν με ακρίβεια το αναλογικό μέγεθος, το οποίο λαμβάνει άπειρες μέσα στο πεδίο τιμών του.

Με τη δειγματοληψία, από το άπειρο πλήθος τιμών του αναλογικού σήματος, κρατάμε μόνο ένα σύνολο διακριτών τιμών ανά κάποιο σταθερό χρονικό διάστημα.

Στην ψηφιακή παράσταση δεν περιγράφονται όλες οι δυνατές τιμές του σήματος, αλλά μόνο κάποιο πεπερασμένο υποσύνολο αυτών, δηλαδή συγκεκριμένα επίπεδα κβαντισμού.

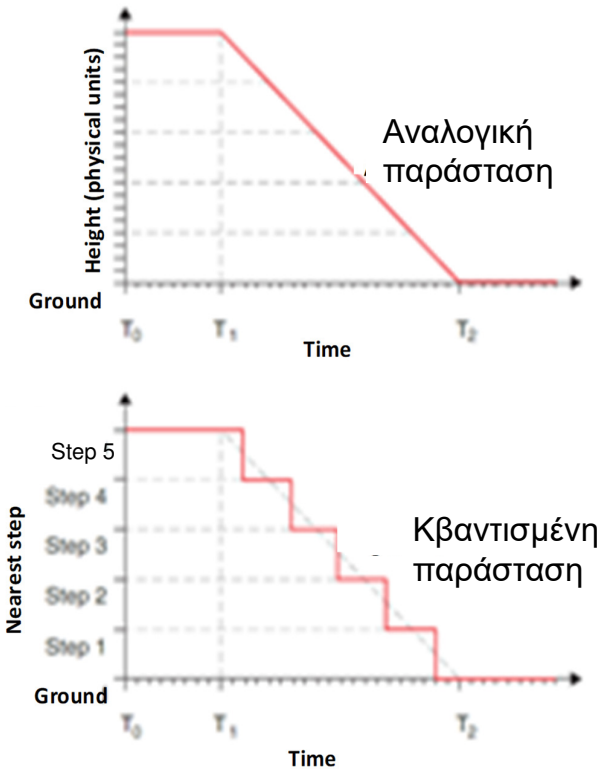
Ο κβαντισμός είναι μη αντιστρέψιμη διαδικασία, δηλαδή μετά την αντιστοίχιση των δειγμάτων σε επίπεδα κβαντισμού, δεν μπορούν να αποκατασταθούν τα δείγματα και το αντίστοιχο σφάλμα που εισάγεται αναφέρεται ως σφάλμα κβαντισμού.

Μετατροπή αναλογικών σημάτων σε ψηφιακά



Μετατροπή αναλογικών σημάτων σε ψηφιακά

Μετατροπή αναλογικών σημάτων σε ψηφιακά



Δειγματοληψία και κβαντισμός

Η δειγματοληψία ενός αναλογικού σήματος $x_\alpha(t)$ επιτυγχάνεται λαμβάνοντας δείγματα αυτού ανά T δευτερόλεπτα: $x = x_\alpha(nT)$.

x είναι το σήμα διακριτού χρόνου που προκύπτει από την δειγματοληψία, T το χρονικό διάστημα μεταξύ διαδοχικών δειγμάτων (περίοδος δειγματοληψίας) και n ο αριθμός των δειγμάτων. $F = 1 / T$: ρυθμός ή συχνότητα δειγματοληψίας.

Στον ομοιόμορφο κβαντισμό η απόσταση (διαφορά) Δ μεταξύ δύο επιπέδων κβαντισμού, δηλαδή των προκαθορισμένων τιμών στις οποίες στρογγυλοποιούνται οι τιμές του δειγματοληπτημένου σήματος, είναι σταθερή.

Δειγματοληψία και κβαντισμός

Σφάλμα κβαντισμού: διαφορά μεταξύ αρχικής (x) και κβαντισμένης τιμής (x_q). Η απόλυτη τιμή του περιορίζεται (μέγιστη τιμή) στο μισό της διαφοράς μεταξύ δύο επιπέδων κβαντισμού.

$$e_q = x - x_q$$
$$-\frac{\Delta}{2} \le e_q \le \frac{\Delta}{2}$$

$$\Delta = \frac{x_{max} - x_{min}}{L-1} = \frac{R}{L-1}$$

R: περιοχή κβαντισμού

Η αύξηση του πλήθους των επιπέδων κβαντισμού (L) οδηγεί στη μείωση του σφάλματος.

Παράδειγμα μετατροπής

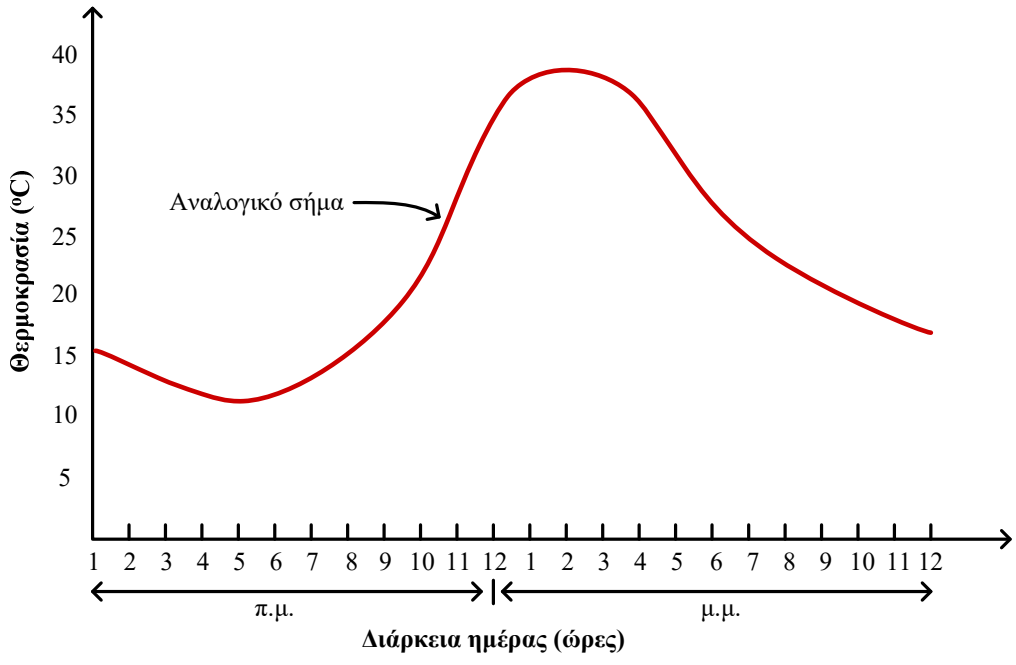
Ας δούμε ένα παράδειγμα ψηφιοποίησης ενός αναλογικού σήματος, μελετώντας τη διακύμανση της θερμοκρασίας κατά τη διάρκεια μιας καλοκαιρινής ημέρας.

Η θερμοκρασία λαμβάνει συνεχείς τιμές, αφού δεν μπορεί να μεταβληθεί στιγμιαία, για παράδειγμα από τους 20 στους 30 βαθμούς Κελσίου, χωρίς να λάβει όλες τις ενδιάμεσες τιμές.

Η γραφική παράσταση που προκύπτει είναι μία συνεχής καμπύλη γραμμή και αποτελεί την αναλογική παράσταση του ποσοτικού μεγέθους της θερμοκρασίας.

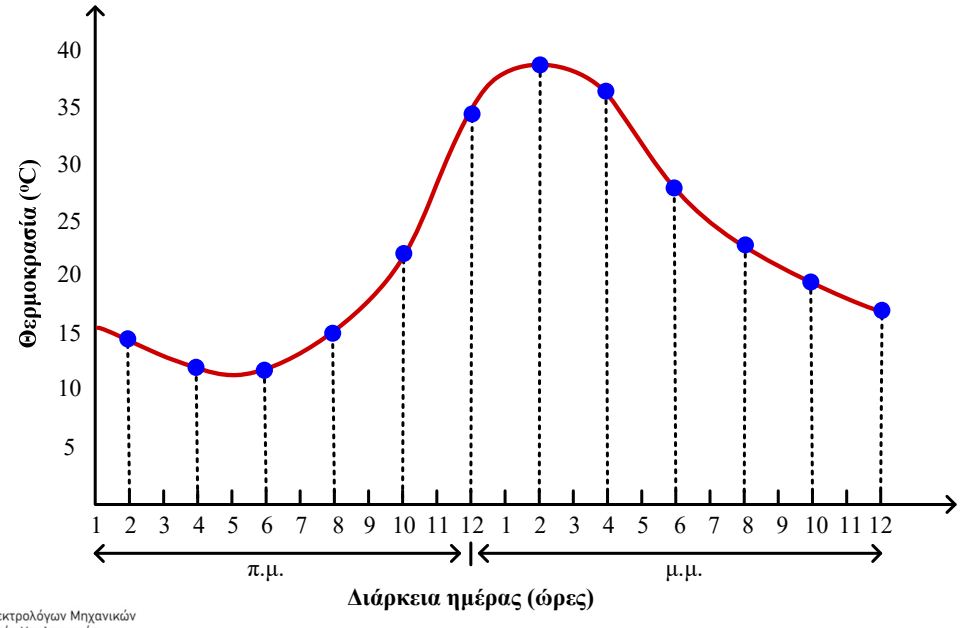
Παράδειγμα μετατροπής

Γραφική παράσταση της διακύμανσης της θερμοκρασίας



Παράδειγμα μετατροπής

Υποθέστε στη συνέχεια ότι μετράμε την τιμή της θερμοκρασίας ανά δίωρο. Αυτό έχει αποτέλεσμα τη λήψη δώδεκα διακριτών δειγμάτων που παριστάνουν την τιμή της θερμοκρασίας σε διακριτές χρονικές στιγμές.



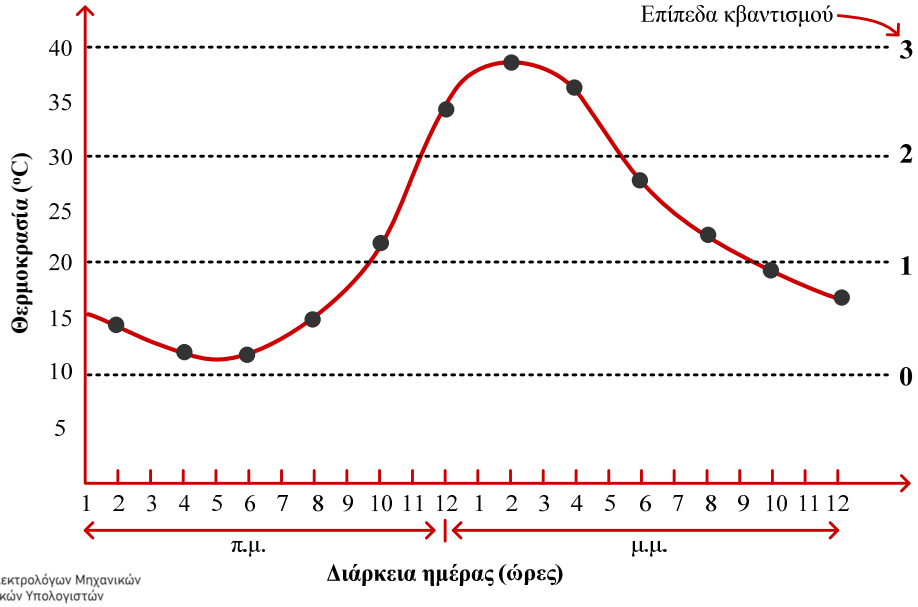
Παράδειγμα μετατροπής

Η διαδικασία με την οποία παράγουμε ένα διακριτό από ένα συνεχές σήμα ονομάζεται δειγματοληψία και προκύπτει από την καταγραφή των τιμών του συνεχούς σήματος σε μια σειρά από διακριτά και ισαπέχοντα σημεία στο χρόνο (ομοιόμορφη δειγματοληψία).

Η δειγματοληψία δημιουργεί διακριτοποίηση του σήματος στο χρόνο. Οι τιμές των δειγμάτων (πλάτος σήματος) όμως μπορούν να είναι οποιεσδήποτε.

Παράδειγμα μετατροπής

Στη συνέχεια καθορίζουμε 4 επίπεδα θερμοκρασίας που διαφέρουν κατά 10 ^{o}C. Τα επίπεδα αυτά καλύπτουν την περιοχή τιμών που λαμβάνει η θερμοκρασία. Με αυτόν τον τρόπο κάθε δείγμα (μέτρηση στιγμιαίας θερμοκρασίας) μπορεί να αντιστοιχιστεί σε κάποιο από τα 4 επίπεδα.

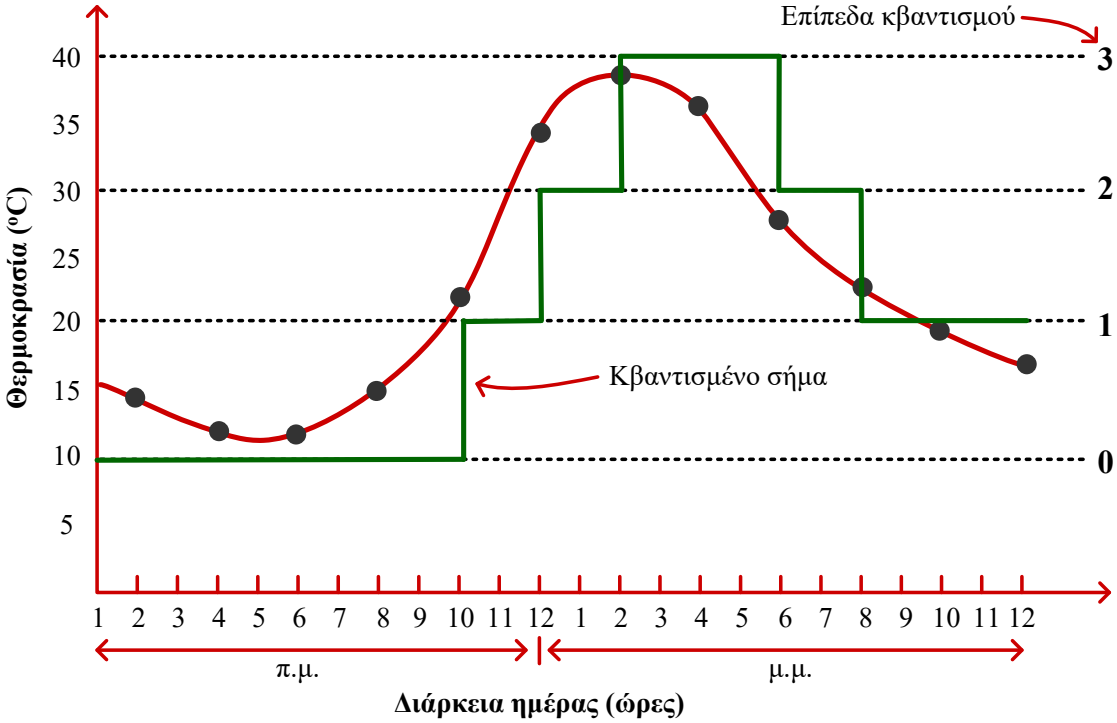


Παράδειγμα μετατροπής

Η διαδικασία αντιστοίχισης των δειγμάτων σε πεπερασμένο πλήθος επιπέδων, ονομάζεται κβαντισμός.

Κατά τη διαδικασία του κβαντισμού, επομένως, κάθε δείγμα πρέπει να αντιστοιχιστεί στο πλησιέστερο επίπεδο.

Παράδειγμα μετατροπής



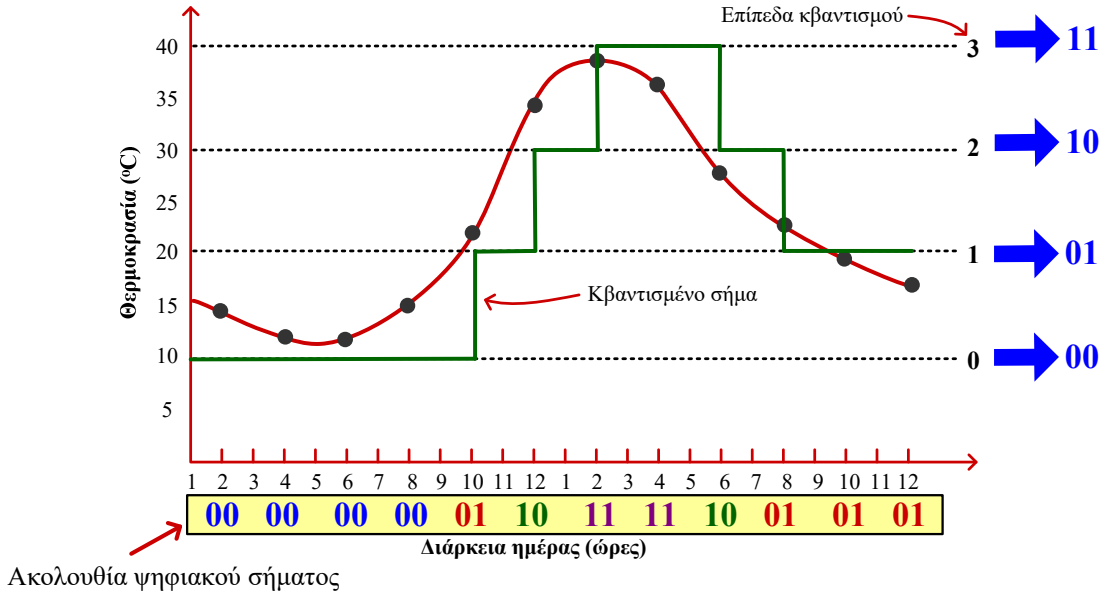
Παράδειγμα μετατροπής

Το πλήθος των επιπέδων κβαντισμού M συνδέεται με το πλήθος N των bits που αφιερώνεται για την κωδικοποίηση (περιγραφή) τους με τη σχέση: $M = 2^N$.

Στο παράδειγμά μας έχουμε 4 επίπεδα κβαντισμού της θερμοκρασίας ($M = 4 = 2^2$), επομένως χρειαζόμαστε 2 bits ($N = 2$), για να τα κωδικοποιήσουμε.

Παράδειγμα μετατροπής

Περιοχή κβαντισμού R = 40 – 10 = 30 °C
Πλήθος επιπέδων κβαντισμού L = 4
Απόσταση επιπέδων κβαντισμού Δ = R / L – 1 = 30 / (4 – 1) = 10 °C
Μέγιστο σφάλμα κβαντισμού = Δ / 2 = 10 / 2 = 5 °C



Παράδειγμα μετατροπής

Αφού αντιστοιχίσαμε κάθε δείγμα θερμοκρασίας με ένα πεπερασμένο πλήθος ψηφίων, επιτελέσαμε τη διαδικασία που αναφέρεται ως ψηφιοποίηση (digitization), αποτέλεσμα της οποίας είναι η ψηφιακή παράσταση της θερμοκρασίας, δηλαδή μια απλή ακολουθία δυαδικών ψηφίων που αναπαριστά το μέγεθος των διαδοχικών δειγμάτων.

Παράδειγμα μετατροπής

Το ψηφιακό σήμα που προέκυψε δεν αποτελεί ακριβή προσέγγιση του αναλογικού σήματος.

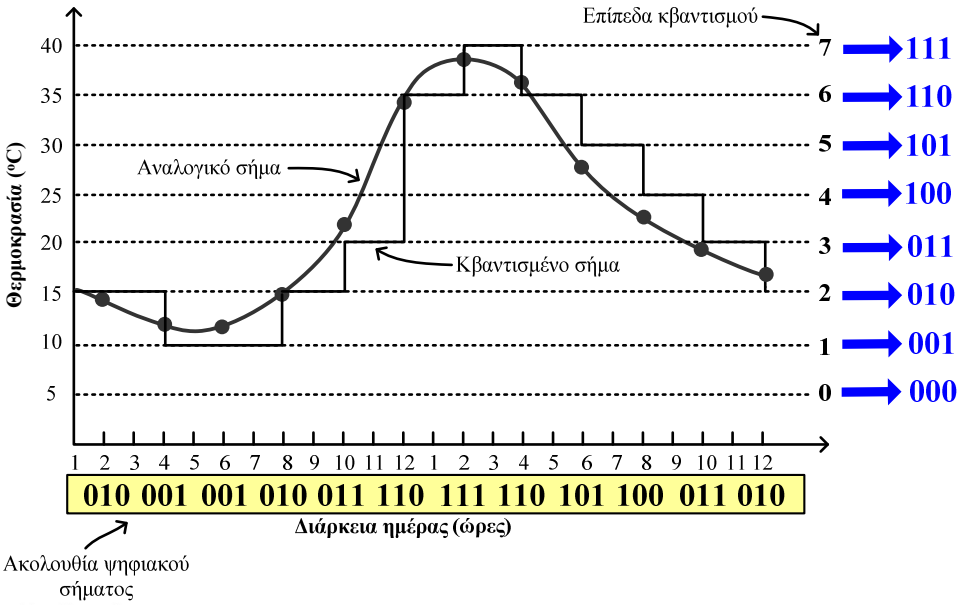
Όσο μεγαλύτερο είναι το πλήθος των επιπέδων κβαντισμού που χρησιμοποιούμε, τόσο ακριβέστερη θα είναι η προσέγγιση του ψηφιακού προς το αναλογικό σήμα.

Στο παράδειγμά μας, μπορούμε να βελτιώσουμε την προσέγγιση εάν ορίσουμε 8 επίπεδα κβαντισμού (ανά 5 ^{o}C).

Στην περίπτωση αυτή απαιτούνται N = 3 bits (8 = 2^3) για να τα κωδικοποιήσουμε.

Παράδειγμα μετατροπής

Περιοχή κβαντισμού R = 40 – 5 = 35 ^{o}C
Πλήθος επιπέδων κβαντισμού L = 8
Απόσταση επιπέδων κβαντισμού Δ = R / L – 1 = 35 / 8 – 1 = 5 ^{o}C
Μέγιστο σφάλμα κβαντισμού = Δ / 2 = 5 / 2 = 2,5 ^{o}C



Ψηφιακά συστήματα

Η πρόοδος στον τομέα των ηλεκτρονικών και της πληροφορικής, έχει οδηγήσει στην ανάπτυξη ψηφιακών συστημάτων και υπηρεσιών που χρησιμοποιούνται σε διάφορα είδη υπολογιστών και σε διάφορες εφαρμογές (κινητή τηλεφωνία, ευρυζωνικά δίκτυα, τηλεόραση, συστήματα αναπαραγωγής ήχου, φωτογραφικές μηχανές, κάμερες, πλατφόρμες ηλεκτρονικών παιχνιδιών, ιατρική κ.ά.)

Τα ψηφιακά συστήματα επεξεργάζονται μεγέθη ή δεδομένα που παριστάνονται με ψηφιακό τρόπο (δηλαδή λαμβάνουν διακριτές τιμές σε μία ορισμένη περιοχή τιμών).

Η ψηφιακή επεξεργασία αφορά μετασχηματισμούς των δεδομένων που το ψηφιακό σύστημα λαμβάνει στις εισόδους του, το αποτέλεσμα των οποίων παρέχεται στις εξόδους του συστήματος.

Ψηφιακά συστήματα

Στα ψηφιακά συστήματα τα διακριτά στοιχεία πληροφορίας παριστάνονται με ποσότητες (ψηφιακά σήματα) τα οποία έχουν τη μορφή διακριτών τιμών ηλεκτρικής τάσης.

Τα ψηφιακά σήματα είναι συνήθως δυαδικά σήματα ηλεκτρικής τάσης, χρησιμοποιούν, δηλαδή, δύο διακριτές τιμές ή καταστάσεις, οι οποίες αναφέρονται και ως στάθμες.

Η χαμηλή και η υψηλή στάθμη ενός δυαδικού σήματος αντιστοιχούν σε δύο διακριτές τιμές τάσης (π.χ. 0 και 3,3 Volts, αντίστοιχα), με τις ενδιάμεσες τιμές να είναι πρακτικά μη υπαρκτές. Οι δύο στάθμες εκφράζονται με δύο λογικές τιμές, τη λογική τιμή 0 και τη λογική τιμή 1, αντίστοιχα.

Ψηφιακά συστήματα

Στην πραγματικότητα, στα ψηφιακά συστήματα, στη λογική τιμή 1 δεν αντιστοιχεί μία τιμή (στάθμη) τάσης αλλά μία περιοχή τάσης και στη λογική τιμή 0 αντιστοιχεί μία δεύτερη περιοχή τάσης, μη επικαλυπτόμενη με την πρώτη.

Συνεπώς, μία τάση των 2.5 V γίνεται αποδεκτή από το ψηφιακό κύκλωμα ως λογική τιμή 1, ενώ μία τάση του 0.5 V ως λογική τιμή 0.

Ψηφιακά συστήματα

Η τιμή της τάσης στις εισόδους ή τις εξόδους ενός ψηφιακού κυκλώματος μπορεί να βρεθεί ανάμεσα στις δύο περιοχές μόνο κατά τη διάρκεια της μετάβασης μεταξύ των δύο περιοχών τάσης.

Είναι σημαντικό οι περιοχές τάσεων που αντιστοιχούν στις δύο λογικές τιμές να είναι όσο το δυνατόν ευρύτερες, έτσι ώστε να εξασφαλίζεται η στιβαρότητα των ψηφιακών κυκλωμάτων, όσον αφορά την αλλοίωση των τιμών των σημάτων τους, λόγω της παρουσίας θορύβου.

Ψηφιακά συστήματα

Τα ψηφιακά συστήματα συνίστανται σε ψηφιακά ηλεκτρονικά κυκλώματα, στα οποία οι βασικές πράξεις μεταξύ δυαδικών σημάτων επιτελούνται από απλά δομικά στοιχεία που αναφέρονται ως λογικές πύλες (logic gates).

Οι λογικές πύλες ανταποκρίνονται στις δύο προαναφερθείσες στάθμες τάσης και υλοποιούνται με διασυνδεδεμένα ηλεκτρονικά στοιχεία που ονομάζονται τρανζίστορ (transistor).

Το σημαντικότερο πλεονέκτημα των στοιχείων αυτών, σε σχέση με το χειρισμό δυαδικών σημάτων, είναι ότι έχουν τη δυνατότητα να λειτουργούν ως διακόπτες, παρουσιάζοντας δύο επιτρεπτές καταστάσεις λειτουργίας αντίστοιχες με τις λογικές τιμές 0 και 1.

Ψηφιακά συστήματα

Υπάρχουν αρκετοί σημαντικοί λόγοι για τους οποίους τα ψηφιακά συστήματα χρησιμοποιούνται στην πλειονότητα των σύγχρονων εφαρμογών, έναντι των αναλογικών συστημάτων.

Τα σύγχρονα ψηφιακά συστήματα είναι προγραμματιζόμενα και παρέχουν ευελιξία, αφού η ίδια πλατφόρμα υλικού μπορεί να χρησιμοποιηθεί για διάφορες εφαρμογές.

Η αλματώδης πρόοδος της τεχνολογίας των ψηφιακών ολοκληρωμένων κυκλωμάτων έχει επιφέρει δραματική μείωση του κόστους των ψηφιακών συστημάτων.

Το πλήθος των τρανζίστορ που μπορούν να ολοκληρωθούν σε μια ψηφίδα πυριτίου αυξάνεται διαρκώς, με αποτέλεσμα, εκτός από τη μείωση του κόστους των ψηφιακών συστημάτων, τη δυνατότητα των ψηφιακών συστημάτων για εκτέλεση όλο και πιο περίπλοκων λειτουργιών.

Ψηφιακά συστήματα

Τα ψηφιακά συστήματα που δομούνται από ψηφιακά ολοκληρωμένα κυκλώματα, μπορούν να λειτουργήσουν σε υψηλές ταχύτητες (εκτέλεση εκατοντάδων εκατομμυρίων πράξεων ανά δευτερόλεπτο).

Παρέχουν επίσης, υψηλή αξιοπιστία και ακρίβεια κατά την επεξεργασία, αποθήκευση και μεταφορά δεδομένων, μέσω των δυνατοτήτων ανίχνευσης και διόρθωσης σφαλμάτων που διαθέτουν.

Το βασικό μειονέκτημα των ψηφιακών συστημάτων είναι η ανάγκη για μετατροπή της κατά φύση αναλογικής μορφής των μεγεθών σε ψηφιακή, που εισάγει σφάλματα και οδηγεί σε πρόσθετη καθυστέρηση, πολυπλοκότητα και κόστος.

2. Αριθμητικά συστήματα, πράξεις και δυαδικοί κώδικες

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Αριθμητικά συστήματα

Το δεκαδικό σύστημα αποτελεί το πιο οικείο αριθμητικό σύστημα, αφού χρησιμοποιείται σε πολλές δραστηριότητες (μέτρηση διάφορων μεγεθών, οι εμπορικές συναλλαγές κ.ά.).

Ωστόσο, έχουν αναπτυχθεί και άλλα αριθμητικά συστήματα, η γνώση των οποίων είναι αναγκαία για την κατανόηση και την ανάλυση της λειτουργίας των ψηφιακών συστημάτων.

Η αναγκαιότητα αυτή προκύπτει από το γεγονός ότι τα ψηφιακά συστήματα σχεδιάζονται ώστε να λειτουργούν με δυαδικά σήματα, συνεπώς τα αριθμητικά συστήματα που χρησιμοποιούνται στο εσωτερικό τους θα πρέπει να είναι συμβατά με τον τρόπο λειτουργίας τους και την τεχνολογία με την οποία αναπτύσσονται.

Αριθμητικά συστήματα

Για να είναι δυνατή η υιοθέτηση του οικείου δεκαδικού αριθμητικού συστήματος στα ψηφιακά κυκλώματα και συστήματα, θα έπρεπε να χρησιμοποιηθούν ηλεκτρονικά στοιχεία με δέκα καταστάσεις λειτουργίας, ώστε να μπορούν να παρασταθούν όλα τα αναγκαία ψηφία.

Ωστόσο, λόγω του ότι δεν είναι διαθέσιμα τέτοιου είδους στοιχεία, αλλά έχουν αναπτυχθεί ηλεκτρονικά στοιχεία όπως τα τρανζίστορ με δύο καταστάσεις λειτουργίας, το δυαδικό αριθμητικό σύστημα έχει πρωταρχική σημασία και χρησιμότητα για τους σχεδιαστές ψηφιακών κυκλωμάτων και συστημάτων.

Αριθμητικά συστήματα, όπως το οκταδικό και το δεκαεξαδικό, χρησιμοποιούνται, επίσης, κατά το σχεδιασμό και την ανάλυση ψηφιακών συστημάτων.

Αριθμητικά συστήματα

Βασικό διακριτικό χαρακτηριστικό των αριθμητικών συστημάτων είναι ένας φυσικός αριθμός μεγαλύτερος του 1 που αναφέρεται ως βάση (base ή radix) του συστήματος.

Τα αριθμητικά συστήματα λαμβάνουν το όνομά τους από τον αριθμό που αποτελεί τη βάση τους. Έτσι, η βάση του δεκαδικού, του δυαδικού, του οκταδικού και του δεκαεξαδικού συστήματος είναι ο αριθμός 10, 2, 8 και 16, αντίστοιχα.

Η βάση ενός συστήματος καθορίζεται από το πλήθος των ψηφίων ή συμβόλων που μπορούν να χρησιμοποιηθούν σε αυτό.

Αριθμητικά συστήματα

Τα ψηφία αυτά είναι φυσικοί αριθμοί μικρότεροι από τη βάση ή σύμβολα που αντιστοιχούν σε φυσικούς αριθμούς μικρότερους από τη βάση του συστήματος:

- στο δυαδικό σύστημα χρησιμοποιούνται τα ψηφία 0 και 1,
- στο οκταδικό τα ψηφία από 0 έως 7,
- στο δεκαδικό σύστημα τα ψηφία 0 έως 9
- στο δεκαεξαδικό σύστημα τα ψηφία 0 έως 9 και τα πρώτα 6 γράμματα του λατινικού αλφαβήτου (A, B, C, D, E, F) που αντιστοιχούν στους δεκαδικούς αριθμούς 10 έως 15.

Κάθε ψηφίο, ανάλογα με τη θέση στην οποία βρίσκεται, κατέχει διαφορετική βαρύτητα. Για το λόγο αυτόν, τα εν λόγω αριθμητικά συστήματα αναφέρονται ως θεσιακά (positional) αριθμητικά συστήματα.

Αριθμητικά συστήματα

Το πρώτο από αριστερά ψηφίο ενός αριθμού αναφέρεται ως περισσότερο σημαντικό ψηφίο (most significant digit, MSD) ή περισσότερο σημαντικό δυαδικό ψηφίο (most significant bit, MSB), όταν πρόκειται για δυαδικό αριθμό.

Το τελευταίο αναφέρεται ως λιγότερο σημαντικό ψηφίο (least significant digit, LSD) ή λιγότερο σημαντικό δυαδικό ψηφίο (least significant bit, LSB), όταν πρόκειται για δυαδικό αριθμό.

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Ένας ακέραιος μη-προσημασμένος δεκαδικός αριθμός, για παράδειγμα ο 5792, παριστάνει μια ποσότητα ίση με 5 χιλιάδες συν 7 εκατοντάδες συν 9 δεκάδες συν 2 μονάδες:

$$5 \times 1000 + 7 \times 100 + 9 \times 10 + 2 \times 1$$

Οι χιλιάδες, εκατοντάδες, δεκάδες και μονάδες, είναι δυνάμεις του 10 που προσδιορίζονται από τη θέση των ψηφίων του αριθμού:

$$5 \times 10^3 + 7 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$$

Επομένως, ο 5792 είναι μια παράσταση (συντομογραφία) της παραπάνω ποσότητας.

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Η σύμβαση που ακολουθείται είναι να γράφονται μόνο οι συντελεστές (δηλαδή, τα ψηφία με τα οποία πολλαπλασιάζονται οι αντίστοιχες δυνάμεις του 10) και από τη θέση των συντελεστών να προσδιορίζονται οι αναγκαίες δυνάμεις του 10.

Γενικά, ένας ακέραιος μη-προσημασμένος δεκαδικός αριθμός A παριστάνεται από μια ακολουθία συντελεστών:

$$A = \alpha_{n-1}\alpha_{n-2} \ldots \alpha_1\alpha_0$$

Οι συντελεστές α_i μπορεί να είναι οποιοδήποτε από τα δεκαδικά ψηφία (0, 1, 2, ..., 9), ενώ η τιμή του δείκτη i είναι η τάξη (θέση) κάθε συντελεστή, δηλαδή τη δύναμη του 10 με την οποία πρέπει να πολλαπλασιαστεί ο συγκεκριμένος συντελεστής:

$$A = \alpha_{n-1}\alpha_{n-2} \ldots \alpha_1\alpha_0 = \alpha_{n-1} \times 10^{n-1} + \alpha_{n-2} \times 10^{n-2} + \ldots + \alpha_1 \times 10^1 + \alpha_0 \times 10^0$$

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Η σύμβαση αυτή ακολουθείται σε όλα τα αριθμητικά συστήματα (δεκαδικό, δυαδικό, τετραδικό, οκταδικό, δεκαεξαδικό κλπ.).

Σε οποιοδήποτε αριθμητικό σύστημα, με βάση τον αριθμό B, ένας ακέραιος μη προσημασμένος αριθμός A με πλήθος ψηφίων n, εκφράζεται ως ακολούθως:

$$(A)_B = \alpha_{n-1}\alpha_{n-2} \ldots \alpha_1\ \alpha_0 = \alpha_{n-1} \times B^{n-1} + \alpha_{n-2} \times B^{n-2} + \ldots + \alpha_1 \times B^1 + \alpha_0 \times B^0$$

όπου B η βάση του αριθμητικού συστήματος και οι συντελεστές α_i μπορεί να είναι οποιοδήποτε από τα ψηφία του αριθμητικού συστήματος, τα οποία παίρνουν ακέραιες τιμές από 0 μέχρι και B – 1. Οι τιμές του δείκτη i είναι η τάξη (θέση) κάθε συντελεστή και κατά συνέπεια η δύναμη της βάσης B με την οποία πρέπει να πολλαπλασιαστεί ο συντελεστής αυτός.

Στη βιβλιογραφία η βάση B (base) αναφέρεται και ως r (radix).

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Στο δεκαδικό σύστημα, για να παρασταθούν αριθμοί μεγαλύτεροι του μεγαλύτερου επιτρεπτού ψηφίου (του 9), απαιτούνται περισσότερα από ένα ψηφία (2 ψηφία για αριθμούς μεταξύ του 10 και του 99, 3 ψηφία για αριθμούς μεταξύ του 100 και του 999, κ.ο.κ).

Γενικεύοντας, προκύπτει ότι για την παράσταση έως 10^n δεκαδικών αριθμών απαιτούνται n δεκαδικά ψηφία, για την παράσταση έως 2^n δυαδικών αριθμών απαιτούνται n δυαδικά ψηφία και για την παράσταση έως 8^n οκταδικών αριθμών και έως 16^n δεκαεξαδικών αριθμών απαιτούνται n οκταδικά και δεκαεξαδικά ψηφία, αντίστοιχα.

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Ο μεγαλύτερος αριθμός που μπορεί να παρασταθεί με n δυαδικά ψηφία είναι εκείνος που αποτελείται από n μονάδες και η αντίστοιχη δεκαδική τιμή είναι:

$$1 \times 2^{n-1} + 1 \times 2^{n-2} + \ldots + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 2^n - 1.$$

Παρομοίως, προκύπτει ότι ο μεγαλύτερος αριθμός που μπορεί να παρασταθεί με n ψηφία σε ένα αριθμητικό σύστημα με βάση B, είναι ο $B^n - 1$.

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Παραδείγματα:

Δεκαδικό αριθμητικό σύστημα (Βάση = 10)

Τα ψηφία του δεκαδικού αριθμητικού συστήματος είναι οι ακέραιοι αριθμοί από 0 μέχρι και 9.

$(795)_{10} = 7 \times 100 + 9 \times 10 + 5 \times 1 = 7 \times 10^2 + 9 \times 10^1 + 5 \times 10^0$

Τετραδικό αριθμητικό σύστημα (Βάση = 4)

Τα ψηφία του τετραδικού αριθμητικού συστήματος είναι οι ακέραιοι αριθμοί από 0 μέχρι και 3.

$(132)_4 = 1 \times 4^2 + 3 \times 4^1 + 2 \times 4^0 = 1 \times 16 + 3 \times 4 + 2 \times 1 = (30)_{10}$

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Οκταδικό αριθμητικό σύστημα (Βάση = 8)

Τα ψηφία του οκταδικού αριθμητικού συστήματος είναι οι ακέραιοι αριθμοί από 0 μέχρι και 7.

$(370)_8 = 3 \times 8^2 + 7 \times 8^1 + 0 \times 8^0 = 3 \times 64 + 7 \times 8 + 0 \times 1 = (248)_{10}$

Δεκαεξαδικό αριθμητικό σύστημα (Βάση = 16)

Τα ψηφία του δεκαεξαδικού αριθμητικού συστήματος είναι οι ακέραιοι αριθμοί 0 και 15, όμως οι αξίες 10, 11, 12, 13, 14 και 15 εκφράζονται με τους λατινικούς χαρακτήρες A, B, C, D, E και F αντίστοιχα.

$(1AF)_{16} = 1 \times 16^2 + A \times 16^1 + F \times 16^0 =$

$= 1 \times 256 + 10 \times 16 + 15 \times 1 = (431)_{10}$

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Δυαδικό αριθμητικό σύστημα (Βάση = 2)

Τα ψηφία του δυαδικού αριθμητικού συστήματος είναι οι ακέραιοι αριθμοί 0 και 1.

$(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 8 + 1 \times 1 = (9)_{10}$

Επομένως, στο δεκαδικό σύστημα έχουμε μονάδες (10^0), δεκάδες (10^1), εκατοντάδες (10^2), χιλιάδες (10^3), κλπ., ενώ στο δυαδικό σύστημα θα έχουμε μονάδες (2^0), δυάδες (2^1), τετράδες (2^2), οκτάδες (2^3), κλπ., αντίστοιχα.

Στον πίνακα που ακολουθεί παρουσιάζεται η παράσταση των δεκαδικών αριθμών 0 έως και 15 στο δυαδικό, τεραδικό, οκταδικό και δεκαεξαδικό αριθμητικό σύστημα αντίστοιχα.

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Δεκαδικό	Δυαδικό	Τεταδικό	Οκταδικό	Δεκαεξαδικό
0	0 0 0 0	0 0	0 0	0
1	0 0 0 1	0 1	0 1	1
2	0 0 1 0	0 2	0 2	2
3	0 0 1 1	0 3	0 3	3
4	0 1 0 0	1 0	0 4	4
5	0 1 0 1	1 1	0 5	5
6	0 1 1 0	1 2	0 6	6
7	0 1 1 1	1 3	0 7	7
8	1 0 0 0	2 0	1 0	8
9	1 0 0 1	2 1	1 1	9
10	1 0 1 0	2 2	1 2	A
11	1 0 1 1	2 3	1 3	B
12	1 1 0 0	3 0	1 4	C
13	1 1 0 1	3 1	1 5	D
14	1 1 1 0	3 2	1 6	E
15	1 1 1 1	3 3	1 7	F

Παράσταση αριθμών σε θεσιακά αριθμητικά συστήματα

Στον παρακάτω πίνακα παρουσιάζονται οι δυνάμεις του 2 (από 0 έως 20)

n	2^n	n	2^n	n	2^n
0	1	7	128	14	16384
1	2	8	256	15	32768
2	4	9	512	16	65536
3	8	10	1024	17	131072
4	16	11	2048	18	262144
5	32	12	4096	19	524288
6	64	13	8192	20	1048576

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Η μετατροπή ενός αριθμού από το σύστημα με βάση Β, στο δεκαδικό αριθμητικό σύστημα, γίνεται αν αναπτύξουμε τον αριθμό σε μία ακολουθία δυνάμεων της βάσης Β και προσθέσουμε όλους τους όρους, όπως δείξαμε προηγουμένως:

$(370)_8 = 3 \times 8^2 + 7 \times 8^1 + 0 \times 8^0 = 3 \times 64 + 7 \times 8 + 0 \times 1 = (248)_{10}$

$(1AF)_{16} = 1 \times 16^2 + A \times 16^1 + F \times 16^0 = 1 \times 256 + 10 \times 16 + 15 \times 1 = (431)_{10}$

$(1001)_2 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = (9)_{10}$

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Η μετατροπή ενός αριθμού από το δεκαδικό αριθμητικό σύστημα σε οποιοδήποτε άλλο αριθμητικό σύστημα με βάση B γίνεται με επαναλαμβανόμενη διαίρεση του δεκαδικού αριθμού με τη βάση B του επιθυμητού συστήματος, μέχρι το πηλίκο της διαίρεσης να γίνει 0.

Οι συντελεστές του αριθμού στο επιθυμητό σύστημα με βάση B λαμβάνονται από τα υπόλοιπα των διαιρέσεων.

Το υπόλοιπο της πρώτης διαίρεσης αντιστοιχεί στο λιγότερο σημαντικό ψηφίο (Least Significant Digit – LSD).

Το υπόλοιπο της τελευταίας διαίρεσης αντιστοιχεί στο περισσότερο σημαντικό ψηφίο (Most Significant Digit – MSD).

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Παράδειγμα : Να μετατραπεί ο αριθμός $(141)_{10}$ στο δυαδικό, τετραδικό, οκταδικό και δεκαεξαδικό σύστημα

Μετατροπή αριθμού στο δυαδικό σύστημα:

	141	2						
LSD	**1**	70	2					
		0	35	2				
			1	17	2			
				1	8	2		
					0	4	2	
						0	2	2
							0	1	2
							MSD	**1**	0

Επομένως, $\mathbf{(141)_{10} = (10001101)_2}$

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Μετατροπή αριθμού στο τετραδικό σύστημα:

	141	4			
LSD	**1**	35	4		
		3	8	4	
			0	2	4
			MSD	**2**	0

Επομένως, $(141)_{10} = (2031)_4$

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Μετατροπή αριθμού στο οκταδικό σύστημα:

	141	8		
LSD	**5**	17	8	
		1	2	8
		MSD	**2**	0

Επομένως, $(141)_{10} = (215)_8$

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Μετατροπή αριθμού στο δεκαεξαδικό σύστημα:

	141	16	
LSD	**13**	8	16
	MSD **8**		0

Επομένως, $(141)_{10} = (8D)_{16}$

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Η μετατροπή ενός αριθμού μεταξύ αριθμητικών συστημάτων που έχουν βάση Β, η οποία μπορεί να εκφραστεί ως δύναμη του 2, δηλαδή:

δυαδικό:	$B = 2 = 2^1$,
τετραδικό:	$B = 4 = 2^2$,
οκταδικό:	$B = 8 = 2^3$,
δεκαεξαδικό:	$B = 16 = 2^4$,

γίνεται μέσω του δυαδικού αριθμητικού συστήματος.

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Παράδειγμα: Να μετατραπεί ο αριθμός 10001101_2 στο τετραδικό, οκταδικό και δεκαεξαδικό σύστημα και να επαληθευτεί η μετατροπή (αντίστροφη μετατροπή).

Μετατροπή του δυαδικού αριθμού στο τετραδικό σύστημα
Χωρίζουμε δυάδες ψηφίων από δεξιά προς τα αριστερά:

10	**00**	**11**	**01**	δυαδικό
2	**0**	**3**	**1**	τετραδικό

Επαλήθευση: εκφράζουμε κάθε ψηφίο του τετραδικού αριθμού σε δυαδική μορφή:

2	**0**	**3**	**1**	τετραδικό
10	**00**	**11**	**01**	δυαδικό

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Μετατροπή του δυαδικού αριθμού στο οκταδικό σύστημα

Χωρίζουμε τριάδες ψηφίων από δεξιά προς τα αριστερά:

(0)**10**	**001**	**101**	δυαδικό
2	**1**	**5**	οκταδικό

Επαλήθευση:

Εκφράζουμε κάθε ψηφίο του οκταδικού αριθμού σε δυαδική μορφή

2	**1**	**5**	οκταδικό
(0)**10**	**001**	**101**	δυαδικό

Μετατροπές αριθμών σε συστήματα με άλλη βάση

Μετατροπή του δυαδικού αριθμού στο δεκαεξαδικό σύστημα

Χωρίζουμε τριάδες ψηφίων από δεξιά προς τα αριστερά:

1000	**1101**	δυαδικό
8	**D**	δεκαεξαδικό

Επαλήθευση:

Εκφράζουμε κάθε ψηφίο του δεκαεξαδικού αριθμού σε δυαδική μορφή:

8	**D**	δεκαεξαδικό
1000	**1101**	δυαδικό

Παράσταση αριθμών με κλασματικό μέρος

Όπως είδαμε, η σύμβαση που ακολουθείται στην παράσταση των αριθμών σε όλα τα αριθμητικά συστήματα, είναι να γράφονται μόνο οι συντελεστές (δηλαδή, τα ψηφία με τα οποία πολλαπλασιάζονται οι αντίστοιχες δυνάμεις της βάσης του συστήματος) και από τη θέση τους να προσδιορίζονται οι αναγκαίες δυνάμεις της βάσης:

$$A = \alpha_{n-1}\alpha_{n-2}\ldots\alpha_1\alpha_0 = \alpha_{n-1} \times B^{n-1} + {}_{n-2} \times B^{n-2} + \ldots + \alpha_1 \times B^1 + \alpha_0 \times B^0$$

όπου B η βάση του αριθμητικού συστήματος και οι συντελεστές α_i μπορεί να είναι οποιοδήποτε από τα ψηφία του αριθμητικού συστήματος, τα οποία παίρνουν ακέραιες τιμές από 0 μέχρι και B – 1. Οι τιμές του δείκτη i είναι η τάξη (θέση) κάθε συντελεστή και κατά συνέπεια τη δύναμη της βάσης B με την οποία πρέπει να πολλαπλασιαστεί ο συντελεστής αυτός.

Παράσταση αριθμών με κλασματικό μέρος

Όταν έχουμε μη προσημασμένους αριθμούς, οι οποίοι έχουν ακέραιο και κλασματικό μέρος, η παράσταση ακολουθεί την ίδια λογική:

$$A = \alpha_{n-1} \ldots \alpha_1 \alpha_0 \, . \, \alpha_{-1} \alpha_{-2} \ldots \alpha_{-m} =$$

$$\alpha_{n-1} B^{n-1} + \ldots + \alpha_1 B^1 + \alpha_0 B^0 . \, \alpha_{-1} B^{-1} + \alpha_{-2} B^{-2} + \ldots + \alpha_{-m} B^{-m}$$

Για παράδειγμα, ο δεκαδικός αριθμός $(739.24)_{10}$ αντιστοιχεί στη μαθηματική έκφραση:

$$7 \times 10^2 + 3 \times 10^1 + 9 \times 10^0 + 2 \times 10^{-1} + 4 \times 10^{-2}$$

Ομοίως, η δεκαδική παράσταση του δυαδικού αριθμού $(11010.11)_2$ υπολογίζεται από τη μαθηματική έκφραση:

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = (26.75)_{10}$$

και του οκταδικού αριθμού $(127.4)_8$ υπολογίζεται ως εξής:

$$1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = 64 + 16 + 7 + 0.5 = (87.5)_{10}$$

Μετατροπή αριθμών με κλασματικό μέρος

Η μετατροπή δεκαδικών αριθμών που έχουν ακέραιο και κλασματικό μέρος γίνεται χωριστά για το ακέραιο και το κλασματικό μέρος και, ακολούθως, συνδυάζονται τα δύο αποτελέσματα.

Για τη μετατροπή του κλασματικού μέρους ενός δεκαδικού αριθμού σε ένα σύστημα με βάση B ακολουθείται η εξής διαδικασία:

- Πολλαπλασιάζουμε το κλασματικό μέρος του δεκαδικού αριθμού με τη βάση B και προκύπτει ένα ακέραιο και ένα κλασματικό μέρος.
- Στη συνέχεια το νέο κλασματικό μέρος πολλαπλασιάζεται με τη βάση B και προκύπτει νέο ακέραιο και νέο κλασματικό μέρος.
- Η διαδικασία συνεχίζεται μέχρι το νέο κλασματικό μέρος που προκύπτει να είναι 0 ή μέχρι ο αριθμός των ψηφίων να γίνει τέτοιος, ώστε να έχουμε την επιθυμητή ακρίβεια. Οι συντελεστές του κλασματικού μέρους του αριθμού στο σύστημα με βάση B λαμβάνονται από τα ακέραια μέρη των γινομένων.

Μετατροπή αριθμών με κλασματικό μέρος

Παράδειγμα : Να εκφραστεί ο δεκαδικός αριθμός $(0.6875)_{10}$ στο δυαδικό σύστημα.

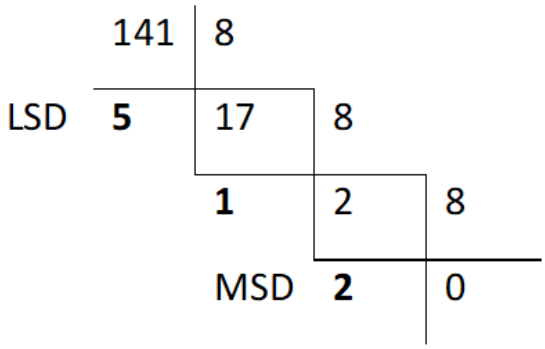
		Ακέραιο μέρος		Κλασματικό μέρος	Συντελεστής
0.6875 x 2	=	1	+	0.3750	$\alpha_{-1} = 1$
0.3750 x 2	=	0	+	0.7500	$\alpha_{-2} = 0$
0.7500 x 2	=	1	+	0.5000	$\alpha_{-3} = 1$
0.5000 x 2	=	1	+	0.0000	$\alpha_{-4} = 1$

Επομένως: $(0.6875)_{10} = (0.1011)_2$

Μετατροπή αριθμών με κλασματικό μέρος

Παράδειγμα: Να εκφραστεί ο δεκαδικός αριθμός $(141.513)_{10}$ στο οκταδικό σύστημα.

Μετατροπή του ακέραιου μέρους του αριθμού στο οκταδικό σύστημα:



Επομένως, το ακέραιο μέρος του αριθμού θα είναι: $(141)_{10} = (215)_8$

Μετατροπή αριθμών με κλασματικό μέρος

Μετατροπή του κλασματικού μέρους του αριθμού στο οκταδικό σύστημα:

$$0.513 \times 8 = 4.104$$
$$0.104 \times 8 = 0.832$$
$$0.832 \times 8 = 6.656$$
$$0.656 \times 8 = 5.244$$

Επομένως, με ακρίβεια τεσσάρων σημαντικών ψηφίων, το κλασματικό μέρος του αριθμού θα είναι: $(0.513)_{10} = (0.4065)_8$.

Άρα, θα έχουμε: $(141.513)_{10} = (215.4065)_8$.

Μετατροπή αριθμών με κλασματικό μέρος

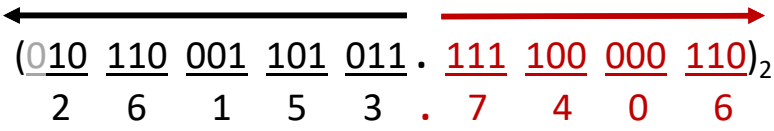
Η μετατροπή ενός αριθμού μεταξύ αριθμητικών συστημάτων με βάση Β, η οποία μπορεί να εκφραστεί ως δύναμη του 2 (δυαδικό: $B = 2 = 2^1$, τετραδικό: $B = 4 = 2^2$, οκταδικό: $B = 8 = 2^3$, δεκαεξαδικό: $B = 16 = 2^4$), γίνεται μέσω του δυαδικού αριθμητικού συστήματος.

Υπενθυμίζεται ότι, για να μετατρέψουμε έναν αριθμό από το δυαδικό σύστημα σε ένα σύστημα με βάση B^n (τετραδικό: n = 2, οκταδικό: n = 3, δεκαεξαδικό: n = 4), χωρίζουμε τον δυαδικό αριθμό σε ομάδες των n bits, ξεκινώντας από την υποδιαστολή και προχωρώντας προς τα αριστερά για το ακέραιο μέρος και προς τα δεξιά για το κλασματικό.

Στη συνέχεια, αντικαθιστούμε την κάθε ομάδα bits με τον αντίστοιχο συντελεστή του συστήματος με βάση B^n.

Μετατροπή αριθμών με κλασματικό μέρος

Για παράδειγμα, ο δυαδικός αριθμός $(10110001101011.111100000110)_2$, στο οκταδικό σύστημα (B = 8 = 2^3, n = 3) θα είναι:



$(010\ 110\ 001\ 101\ 011\ .\ 111\ 100\ 000\ 110)_2 = (26153.7406)_8$

2 6 1 5 3 . 7 4 0 6

Η μετατροπή από το δυαδικό στο δεκαεξαδικό σύστημα είναι παρόμοια, με τη διαφορά ότι χωρίζουμε σε ομάδες 4 δυαδικών ψηφίων, ενώ στο τετραδικό σε ομάδες των 2 δυαδικών ψηφίων.

Αριθμητικές πράξεις στο δυαδικό σύστημα: Πρόσθεση

Ο αλγόριθμος της πρόσθεσης σε όλα τα αριθμητικά συστήματα είναι ο ίδιος.

Έστω ότι θέλουμε να προσθέσουμε τους αριθμούς $X = X_3X_2X_1X_0$ και $Y = Y_3Y_2Y_1Y_0$.

Αθροίζουμε τους συντελεστές κάθε τάξης, ξεκινώντας από το ελάχιστο σημαντικό ψηφίο (μονάδες).

Κάθε επιμέρους άθροιση των X_i και Y_i δίνει ένα άθροισα S_i και ένα κρατούμενο C_i.

Το κρατούμενο C_i που προκύπτει προστίθεται στους συντελεστές της επόμενης τάξης X_{i+1} και Y_{i+1}.

Αριθμητικές πράξεις στο δυαδικό σύστημα: Πρόσθεση

$X =$	(0)	X_3	X_2	X_1	X_0
$+ Y =$	(0)	Y_3	Y_2	Y_1	Y_0
	C_3	C_2	C_1	C_0	
Άθροισμα **S** =	S_4	S_3	S_2	S_1	S_0
Επιμέρους Κρατούμενα C_i		C_3	C_2	C_1	C_0

Επομένως το άθροισμα που προκύπτει είναι $S = S_4S_3S_2S_1S_0$, όπου $S_4 = C_3$

Αριθμητικές πράξεις στο δυαδικό σύστημα: Πρόσθεση

Παράδειγμα: Να προστεθούν οι δυαδικοί αριθμοί:
$X = 1101$ και $Y = 111$.

Στην δυαδική πρόσθεση έχουμε τις εξής δυνατές περιπτώσεις:

	Κρατούμενο	Άθροισμα	
	C	**S**	
0 + 0	0	0	$= (0)_{10}$
0 + 1 ή 1 + 0	0	1	$= (1)_{10}$
1 + 1	1	0	$= (2)_{10}$
1 + 1 +1	1	1	$= (3)_{10}$

Αριθμητικές πράξεις στο δυαδικό σύστημα: Πρόσθεση

Επομένως, για το παράδειγμά μας θα έχουμε:

A =	(0)	1	1	0	1	$= (13)_{10}$
+ B =	(0)	(0)	1	1	1	$= (7)_{10}$
	1	1	1	1		
Άθροισμα **S** =	**1**	**0**	**1**	**0**	**0**	$\mathbf{= (20)_{10}}$
Κρατούμενα		1	1	1	1	

Άρα, το άθροισμα που προκύπτει είναι: $S = S_4S_3S_2S_1S_0 = 10100$

Αριθμητικές πράξεις στο δυαδικό σύστημα: Αφαίρεση

Η αφαίρεση στο δυαδικό σύστημα μπορεί να γίνει με τον αλγόριθμο της αφαίρεσης που εφαρμόζουμε στο δεκαδικό αριθμητικό σύστημα.

Κατά την αφαίρεση, προσθέτουμε το δανειζόμενο ψηφίο για την αφαίρεση της προηγούμενης θέσης στον αφαιρετέο και το άθροισμά τους αφαιρείται από τον μειωτέο:

$$(\text{Διαφορά})_i = (\text{Μειωτέος})_i - [(\text{Αφαιρετέος})_i + (\text{Δανειζόμενο})_{i-1}]$$

Η διαδικασία όμως αυτή δεν είναι εύκολα υλοποιήσιμη στα ψηφιακά συστήματα και για το λόγο αυτό ακολουθείται η μέθοδος της χρήσης συμπληρωμάτων.

Τα συμπληρώματα χρησιμοποιούνται στους ψηφιακούς υπολογιστές για την απλοποίηση της πράξης της αφαίρεσης. Η απλοποίηση των αριθμητικών πράξεων οδηγεί σε απλούστερα και πιο οικονομικά κυκλώματα υλοποίησής τους.

Συμπληρώματα αριθμών

Υπάρχουν 2 τύποι συμπληρωμάτων σε κάθε αριθμητικό σύστημα: το συμπλήρωμα ως προς βάση και το συμπλήρωμα ως προς ελαττωμένη βάση ή συμπλήρωμα ως προς βάση μείον ένα.

Σε ένα αριθμητικό σύστημα με βάση Β, το πρώτο αναφέρεται ως συμπλήρωμα ως προς Β, Σ_B, και το δεύτερο αναφέρεται ως συμπλήρωμα ως προς Β-1, Σ_{B-1}.

Επομένως, στο δεκαδικό σύστημα θα αναφερόμαστε για τους δύο τύπους των συμπληρωμάτων σε συμπλήρωμα ως προς 10, Σ_{10} και συμπλήρωμα ως προς 9, Σ_9. Στο δυαδικό σύστημα θα αναφερόμαστε σε συμπλήρωμα ως προς 2, Σ_2, και συμπλήρωμα ως προς 1, Σ_1, αντίστοιχα.

Συμπληρώματα αριθμών

Έστω ένας αριθμός $X = X_{n-1}X_{n-2}\ldots X_1X_0$, με n ψηφία, σε ένα αριθμητικό σύστημα με βάση Β.

Το συμπλήρωμα του Χ ως προς ελαττωμένη βάση ορίζεται ως:

$$\Sigma_{B-1}(X) = (B^n - 1) - X$$

Στο δεκαδικό αριθμητικό σύστημα, Β = 10 και Β – 1 = 9, και επομένως, το συμπλήρωμα ως προς ελαττωμένη βάση, δηλαδή το συμπλήρωμα ως προς 9, ενός δεκαδικού αριθμού Χ είναι:

$$\Sigma_9(X) = (10^n - 1) - X$$

Συμπληρώματα αριθμών

Το 10^n παριστάνει έναν αριθμό που αποτελείται από μία μονάδα που ακολουθείται από n μηδενικά.

Το $10^n - 1$ είναι αριθμός που παριστάνεται από n φορές το ψηφίο 9.

Για παράδειγμα, εάν n = 3, έχουμε $10^3 = 1000$ και $10^3 - 1 = 999$.
Άρα το συμπλήρωμα ως προς 9 ενός δεκαδικού αριθμού προκύπτει αν αφαιρέσουμε κάθε ψηφίο από το 9.

Παραδείγματα:

$\Sigma_9(56399) = 43600$ (αφού: 9 – 5 = 4, 9 – 6 = 3, 9 – 3 = 6, 9 – 9 = 0)

$\Sigma_9(024619) = 975380$

Συμπληρώματα αριθμών

Στο δυαδικό αριθμητικό σύστημα, B = 2 και B – 1 = 1 και το συμπλήρωμα ως προς 1 ενός αριθμού δυαδικού αριθμού X είναι:

$$\Sigma_1(X) = (2^n - 1) - X$$

Και σε αυτή την περίπτωση το 2^n παριστάνεται από ένα δυαδικό αριθμό που αποτελείται από μία μονάδα που ακολουθείται από n μηδενικά και το $2^n - 1$ είναι δυαδικός αριθμός που παριστάνεται από n μονάδες.

Για παράδειγμα, εάν n = 3, $2^3 = (1000)_2$ και $2^3 - 1 = (111)_2$.

Συμπληρώματα αριθμών

Άρα το συμπλήρωμα ως προς 1 ενός δυαδικού αριθμού προκύπτει αν αφαιρέσουμε κάθε ψηφίο από το 1.

Ωστόσο, όταν αφαιρούμε δυαδικά ψηφία από το 1, μπορούμε να έχουμε μόνο 1 – 0 = 1 ή 1 – 1 = 0, που σημαίνει ότι η αφαίρεση των δυαδικών ψηφίων από το 1 ισοδυναμεί με εναλλαγή των μονάδων και των μηδενικών (δηλαδή, αντικαθιστούμε τα 0 με 1 και τα 1 με 0).

Επομένως, το συμπλήρωμα ενός δυαδικού αριθμού ως προς 1 προκύπτει με εναλλαγή των μονάδων και των μηδενικών.

Παραδείγματα:

$\Sigma_1(101011101) = 010100010$

$\Sigma_1(01101011) = 10010100$

Συμπληρώματα αριθμών

Έστω ένας αριθμός $X = X_{n-1}X_{n-2} \ldots X_1X_0$, με n το πλήθος ψηφία, σε ένα αριθμητικό σύστημα με βάση B.

Το συμπλήρωμα του X ως προς τη βάση ορίζεται ως: $\Sigma_B(X) = B^n - X$.

$$\Sigma_B(X) = B^n - X = B^n - X + (1 - 1) = (B^n - 1) - X + 1 = \Sigma_{B-1}(X) + 1$$

Επομένως, το συμπλήρωμα ενός αριθμού X ως προς βάση ισούται με το συμπλήρωμα του αριθμού ως προς ελαττωμένη βάση συν 1:

$$\Sigma_B(X) = \Sigma_{B-1}(X) + 1$$

Συμπληρώματα αριθμών

Επομένως, στο δεκαδικό σύστημα έχουμε: $\Sigma_{10}(X) = \Sigma_9(X) + 1$.

Για παράδειγμα, στο δεκαδικό αριθμητικό σύστημα, το συμπλήρωμα του αριθμού 735 ως προς βάση μείον 1 (Σ_9) είναι ο αριθμός 264, αφού 735 + 264 = 999 και το συμπλήρωμα ως προς βάση (Σ_{10}) είναι 264 + 1 = 265.

Στο δυαδικό σύστημα θα έχουμε: $\Sigma_2(X) = \Sigma_1(X) + 1$.

Το συμπλήρωμα ως προς βάση μείον 1 (Σ_1) προκύπτει με εναλλαγή των ψηφίων του δυαδικού αριθμού (αντικατάσταση των 0 με 1 και των 1 με 0).

Για παράδειγμα, το συμπλήρωμα του δυαδικού αριθμού 1001 ως προς 1 είναι ο αριθμός 0110 και το συμπλήρωμα ως προς 2 αυτού του δυαδικού αριθμού είναι 0110 + 1 = 0111.

Συμπληρώματα αριθμών

Ο υπολογισμός του συμπληρώματος ως προς 2 ενός δυαδικού αριθμού με πρόσθεση μιας μονάδας στο συμπλήρωμα ως προς 1, είναι ισοδύναμος με το ακόλουθο υπολογισμό:

Στον δυαδικό αριθμό του οποίου επιθυμούμε να υπολογίσουμε το συμπλήρωμα ως προς 2, ξεκινώντας από τη λιγότερο σημαντική θέση (δεξιά) αφήνουμε αναλλοίωτα τα συνεχόμενα μηδενικά που συναντάμε και την πρώτη μονάδα και στη συνέχεια αντικαθιστούμε τα μηδενικά με μονάδες και τις μονάδες με μηδενικά.

Για παράδειγμα, το συμπλήρωμα του δυαδικού αριθμού 100100 ακολουθώντας τον προαναφερόμενο τρόπο υπολογισμού είναι 011100 που είναι ίδιο με το συμπλήρωμα ως προς 1 του αριθμού συν 1 (011011 + 1 = 011100).

Αφαίρεση με χρήση συμπληρωμάτων

Σε όλα τα αριθμητικά συστήματα, αντί του κλασσικού αλγόριθμου της αφαίρεσης, μπορούμε να εφαρμόσουμε τη μέθοδο των συμπληρωμάτων, ειδικότερα μάλιστα στο δυαδικό αριθμητικό σύστημα, λόγω της ευκολίας εφαρμογής του στο συγκεκριμένο αριθμητικό σύστημα.

Για την αφαίρεση Χ (μειωτέος) – Υ (αφαιρετέος) δύο μη προσημασμένων ακέραιων αριθμών, $X = X_{n-1}X_{n-2} \ldots X_1 X_0$ και $Y = Y_{n-1}Y_{n-2} \ldots Y_1Y_0$, με το ίδιο πλήθος n ψηφίων ο καθένας, στο σύστημα με βάση Β ακολουθούμε τα εξής βήματα:

Αφαίρεση με χρήση συμπληρωμάτων

Εκφράζουμε τους αριθμούς με το ίδιο πλήθος ψηφίων (προσθήκη, αν χρειάζεται, μηδενικών στον αριθμό με το μικρότερο πλήθος ψηφίων, αριστερά από το μέγιστο σημαντικό ψηφίο του).

Προσδιορίζουμε το συμπλήρωμα ως προς βάση (Σ_B) του αφαιρετέου.

Προσθέτουμε στον μειωτέο Χ το συμπλήρωμα ως προς βάση (Σ_B) του αφαιρετέου Υ, δηλαδή υπολογίζουμε το:

$$X + \Sigma_B(Y) = X + (B^n - Y) = X - Y + B^n$$

Αν $X \geq Y$, τότε προκύπτει τελικό κρατούμενο 1 (ένα επιπλέον ψηφίο, που είναι ο συντελεστής της τάξης B^v), το οποίο παραλείπεται.
Ο αριθμός που απομένει είναι η διαφορά Χ – Υ.

Αφαίρεση με χρήση συμπληρωμάτων

Αν X < Y, τότε στο άθροισμα που υπολογίζεται δεν προκύπτει κρατούμενο και το άθροισμα ισούται με $B^n - (Y - X)$ που είναι το συμπλήρωμα ως προς βάση του Y – Y.

Το αποτέλεσμα της αφαίρεσης είναι προφανώς ένας αρνητικός αριθμός, αφού ο μειωτέος είναι μικρότερος από τον αφαιρετέο.

Για να λάβουμε το αποτέλεσμα στην κανονική μορφή, υπολογίζουμε το συμπλήρωμα ως προς βάση του αθροίσματος που προκύπτει (δηλαδή, παίρνουμε το συμπλήρωμα του συμπληρώματος) και τοποθετούμε στην αρχή ένα αρνητικό πρόσημο.

Αφαίρεση με χρήση συμπληρωμάτων

Παράδειγμα: Με χρήση συμπληρωμάτων να γίνει η αφαίρεση $(619573)_{10} - (4237)_{10}$.

Ο αφαιρετέος έχει 2 ψηφία λιγότερα από τον μειωτέο, άρα συμπληρώνουμε 2 μηδενικά αριστερά από το περισσότερο σημαντικό ψηφίο: 4237 = 004237.

$\Sigma_9(004237) = 995762$

$\Sigma_{10}(004237) = 995762 + 1 = 995763$

Μειωτέος	619573
+ Συμπλήρωμα ως προς 10 του αφαιρετέου	995763
Άθροισμα	1615336
Διαφορά	**615336**

Αφού 619573 > 4237, παραλείπεται το τελικό κρατούμενο που προκύπτει

Αφαίρεση με χρήση συμπληρωμάτων

Παράδειγμα: Με χρήση συμπληρωμάτων να γίνει η αφαίρεση $(9573)_{10} - (14237)_{10}$.

Στην περίπτωση αυτή ο μειωτέος είναι μικρότερος από τον αφαιρετέο (X < Y), οπότε το αποτέλεσμα της αφαίρεσης θα είναι προφανώς ένας αρνητικός αριθμός.

$\Sigma_9(14237) = 85762$

$\Sigma_{10}(14237) = \Sigma_9(14237) + 1 = 85762 + 1 = 85763$

Μειωτέος	09573
+ Συμπλήρωμα ως προς 10 του αφαιρετέου	85763
Άθροισμα	95336

Αφαίρεση με χρήση συμπληρωμάτων

Το άθροισμα που προέκυψε είναι το συμπλήρωμα ως προς βάση της διαφοράς Y – X.

Για να λάβουμε το αποτέλεσμα σε κανονική μορφή, υπολογίζουμε το συμπλήρωμα ως προς βάση του αθροίσματος που προέκυψε και τοποθετούμε αρνητικό πρόσημο:

$\Sigma_9(95336) =$	04663
$\Sigma_{10}(95336) = \Sigma_9(95336) + 1 = 4663 + 1 =$	4664
Διαφορά	**– 4664**

Αφαίρεση με χρήση συμπληρωμάτων

Παράδειγμα: Με χρήση συμπληρωμάτων να γίνει η αφαίρεση $(10101)_2 - (1101)_2$.

Ο αφαιρετέος έχει ένα ψηφίο λιγότερο από τον μειωτέο, άρα συμπληρώνουμε ένα μηδενικό αριστερά από το περισσότερο σημαντικό ψηφίο: $(1101)_2 = (01101)_2$.

$\Sigma_1(01101) = 10010$
$\Sigma_2(01101) = \Sigma_1(01101) + 1 = 10010 + 1 = 10011$

Μειωτέος	10101
+ Συμπλήρωμα ως προς 2 του αφαιρετέου	10011
Άθροισμα	~~1~~01000
Διαφορά	**01000**

Αφαίρεση με χρήση συμπληρωμάτων

Παράδειγμα: Με χρήση συμπληρωμάτων να γίνει η αφαίρεση $(1111)_2 - (10101)_2$

Στην περίπτωση αυτή ο μειωτέος είναι μικρότερος από τον αφαιρετέο ($X < Y$), οπότε το αποτέλεσμα της αφαίρεσης θα είναι προφανώς αρνητικός αριθμός.

Επιπλέον να σημειωθεί ότι, αφού ο αφαιρετέος $(10101)_2$ αποτελείται από 5 ψηφία, όλη η διαδικασία θα πρέπει να πραγματοποιηθεί με 5 ψηφία.

$\Sigma_1(10101) = 01010$
$\Sigma_2(10101) = \Sigma_1(10101) + 1 = 01010 + 1 = 01011$

Αφαίρεση με χρήση συμπληρωμάτων

Μειωτέος	01111
+ Συμπλήρωμα ως προς 2 του αφαιρετέου	01011
Άθροισμα	11010

Το άθροισμα που προέκυψε είναι το συμπλήρωμα ως προς βάση της διαφοράς Υ – Χ.

Για να πάρουμε το αποτέλεσμα σε κανονική μορφή, υπολογίζουμε το συμπλήρωμα ως προς βάση του αθροίσματος που προέκυψε και τοποθετούμε στην αρχή ένα αρνητικό πρόσημο:

$\Sigma_1(11010) =$	00101
$\Sigma_2(11010) = \Sigma_1(11010) + 1 =$ 00101 + 1 =	00110
Διαφορά	**– 110**

Αριθμητικές πράξεις στο δυαδικό σύστημα: Πολ/σμός

Για τον πολλαπλασιασμό των δυαδικών αριθμών ισχύουν οι εξής περιπτώσεις:

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

Ο πολλαπλασιασμός δύο δυαδικών αριθμών μπορεί να γίνει με την ίδια μέθοδο με το δεκαδικό σύστημα και συνίσταται σε διαδοχικές προσθέσεις και μετατοπίσεις.

Αριθμητικές πράξεις στο δυαδικό σύστημα: Πολ/σμός

Παράδειγμα: Πολλαπλασιασμός των μη προσημασμένων αριθμών $(1110)_2$ και $(1011)_2$:

Πολλαπλασιαστέος:					1	1	1	0
Πολλαπλασιαστής:				x	1	0	1	1
					1	1	1	0
				1	1	1	0	
			0	0	0	0		
		1	1	1	0			
	κρατούμενα:	1	1	1	1	1	0	0
Γινόμενο:		**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**

Αριθμητικές πράξεις στο δυαδικό σύστημα: Πολ/σμός

Κατά την επιμέρους πρόσθεση των ψηφίων κάθε στήλης, δημιουργούνται επιμέρους κρατούμενα, τα οποία μεταφέρονται στην επόμενη στήλη και αθροίζονται με τα ψηφία αυτής.

Το πλήθος των ψηφίων του γινόμενου ισούται με το άθροισμα του πλήθους των ψηφίων του πολλαπλασιαστέου και του πολλαπλασιαστή.

Κάθε επιμέρους γινόμενο που προκύπτει από τον πολλαπλασιασμό του εκάστοτε ψηφίου του πολλαπλασιαστή με τα ψηφία του πολλαπλασιαστέου είναι είτε ίσο με τον πολλαπλασιαστέο, όταν το ψηφίο του πολλαπλασιαστή είναι 1, είτε μηδέν (με πλήθος ψηφίων ίσο με το πλήθος των ψηφίων του πολλαπλασιαστέου), όταν το ψηφίο του πολλαπλασιαστή είναι 0.

Αριθμητικές πράξεις στο δυαδικό σύστημα: Πολ/σμός

Στα ψηφιακά συστήματα, είναι πιο αποδοτικό ως προς την υλοποίηση, αντί να προσθέτουμε στο τέλος του πολλαπλασιασμού όλα τα ψηφία των επιμέρους γινομένων κατά στήλη, να διενεργούμε διαδοχικές προσθέσεις ώστε σε κάθε βήμα να παράγεται ένα μερικό γινόμενο, μέχρι και την εξαγωγή του τελικού γινομένου:

Πολλαπλασιαστέος:					1	1	1	0
Πολλαπλασιαστής:				x	1	0	1	1
					1	1	1	0
			+	1	1	1	0	
			1	0	1	0	1	
		+	0	0	0	0		
			1	0	1	0		
	+	1	1	1	0			
Γινόμενο:	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**

Αριθμητικές πράξεις στο δυαδικό σύστημα: Πολ/σμός

Ένας δυαδικός αριθμός Α μπορεί να πολλαπλασιαστεί επί 2, εάν τοποθετηθεί ένα μηδενικό δεξιά του λιγότερου σημαντικού ψηφίου του.

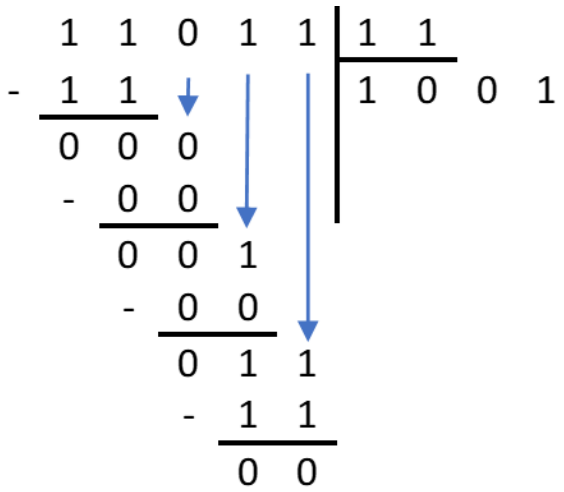
Με αυτόν τον τρόπο, όλα τα ψηφία του αριθμού Α μετατοπίζονται μια θέση προς τα αριστερά, δηλαδή ο αριθμός ολισθαίνει προς τα αριστερά (shift left) κατά μία θέση.

Επομένως, εάν $A = \alpha_{n-1}\alpha_{n-2}...\alpha_1\alpha_0$, τότε θα είναι:

$$2 \times A = \alpha_{n-1}\alpha_{n-2}...\alpha_1\alpha_0\mathbf{0}.$$

Αριθμητικές πράξεις στο δυαδικό σύστημα: Διαίρεση

Η διαίρεση δύο δυαδικών αριθμών μπορεί να γίνει με την ίδια μέθοδο με το δεκαδικό σύστημα και συνίσταται σε διαδοχικές συγκρίσεις και αφαιρέσεις. Για παράδειγμα, να διαιρέσουμε τον αριθμό $(11011)_2 = (27)_{10}$ δια του αριθμού $(11)_2 = (3)_{10}$:



Στην περίπτωση που το υπόλοιπο της διαίρεσης είναι διαφορετικό από το μηδέν, προκύπτει επιπλέον του ακέραιου μέρους (που ισούται με το πηλίκο της διαίρεσης) και κλασματικό μέρος.

Προσημασμένοι δυαδικοί αριθμοί

Όπως έχουμε ήδη αναφέρει, στα ψηφιακά συστήματα οποιαδήποτε πληροφορία αναπαριστάνεται αποκλειστικά με τη χρήση δυαδικών ψηφίων.

Αυτό ισχύει και για την παράσταση προσημασμένων δυαδικών αριθμών.

Στο δυαδικό αριθμητικό σύστημα υπάρχουν τρεις τρόποι παράστασης των προσημασμένων αριθμών:

Παράσταση προσημασμένου μεγέθους (signed-magnitude)

Σε αυτόν τον τρόπο παράστασης που αναφέρεται και ως παράσταση πρόσημο-μέτρο, χρησιμοποιείται ένα επιπλέον ψηφίο για το πρόσημο, που καταλαμβάνει την περισσότερο σημαντική θέση του αριθμού, δηλαδή, το περισσότερο σημαντικό ψηφίο παριστάνει το πρόσημο του αριθμού + (συν) ή – (πλην).

Η σύμβαση που ακολουθείται είναι:
εάν το ψηφίο προσήμου είναι 0, ο αριθμός είναι θετικός, ενώ εάν το ψηφίο προσήμου είναι 1, ο αριθμός είναι αρνητικός.

Για παράδειγμα, στην παράσταση αυτή ο προσημασμένος δεκαδικός αριθμός **+** 9_{10} εκφράζεται με την ακολουθία δυαδικών ψηφίων **0**1001, ενώ ο αριθμός **–** 9_{10}, εκφράζεται ως **1**1001.

Παράσταση προσημασμένου μεγέθους (signed-magnitude)

Στην παράσταση αυτή δεσμεύουμε ένα ψηφίο για την παράσταση του προσήμου και προκύπτουν δύο παραστάσεις για το μηδέν, το **+** 0 (0000) και το **–** 0 (1000).

Κατά την πρόσθεση σε παράσταση προσημασμένου μεγέθους, αρχικά διενεργείται σύγκριση των προσήμων. Εάν οι αριθμοί έχουν ίδιο πρόσημο, προσθέτουμε τα μεγέθη τους και διατηρούμε το πρόσημο, ενώ εάν έχουν διαφορετικό πρόσημο, τότε διενεργούμε σύγκριση του μεγέθους, αφαιρούμε το μικρότερο από το μεγαλύτερο μέγεθος και διατηρούμε το πρόσημο του αριθμού με το μεγαλύτερο μέγεθος.

Η απαιτούμενη σύγκριση προσήμων και μεγεθών των αριθμών αποτελεί μειονέκτημα της χρήσης της παράστασης προσημασμένου μεγέθους για την υλοποίηση αριθμητικών πράξεων με προσημασμένους δυαδικούς αριθμούς στα ψηφιακά συστήματα.

Παράσταση προσημασμένου συμπληρώματος (signed-complement)

Η παράσταση των θετικών αριθμών είναι ίδια και στους 3 τρόπους παράστασης και προκύπτει από τους αντίστοιχους μη προσημασμένους αριθμούς, με την προσθήκη ενός επιπλέον ψηφίου προσήμου με τιμή 0, στη θέση του περισσότερο σημαντικού ψηφίου.

Στην παράσταση προσημασμένου συμπληρώματος, οι αρνητικοί αριθμοί παριστάνονται με το συμπλήρωμα ως προς 1, είτε το συμπλήρωμα ως προς 2 των αντίστοιχων θετικών. Στον υπολογισμό των συμπληρωμάτων των θετικών αριθμών περιλαμβάνεται και το ψηφίο πρόσημο.

Η χρήση της παράστασης προσημασμένου συμπληρώματος ως προς 2 είναι η πιο συνηθισμένη.

Παράσταση προσημασμένου συμπληρώματος (signed-complement)

Για παράδειγμα, θεωρούμε τον μη προσημασμένο δεκαδικό αριθμό 9, ο οποίος παριστάνεται στο δυαδικό σύστημα με 4 ψηφία (1001).

Στην παράσταση προσημασμένου συμπληρώματος ως προς 1, ο προσημασμένος δεκαδικός αριθμός **+** 9_{10} εκφράζεται με την ακολουθία δυαδικών ψηφίων **0**1001, ενώ ο αριθμός **–** 9_{10}, εκφράζεται ως **1**0110.

Στην παράσταση προσημασμένου συμπληρώματος ως προς 2, ο προσημασμένος δεκαδικός αριθμός **+** 9_{10} εκφράζεται με την ακολουθία δυαδικών ψηφίων **0**1001, ενώ ο αριθμός **–** 9_{10}, εκφράζεται ως **1**0111.

Στην περίπτωση αριθμών με ακέραιο και κλασματικό μέρος, η υποδιαστολή δεν λαμβάνεται υπόψη κατά τον υπολογισμό των συμπληρωμάτων και διατηρεί την αρχική της θέση και στα συμπληρώματα.

Προσημασμένοι δυαδικοί αριθμοί

Δεκαδικός	Παράσταση προσημασμένου μεγέθους	Παράσταση προσημασμένου συμπληρώματος ως προς 1	Παράσταση προσημασμένου συμπληρώματος ως προς 2
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
0	0000 ή 1000	0000 ή 1111	0000
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	-	-	1000

Προσημασμένοι δυαδικοί αριθμοί

Στις παραστάσεις προσημασμένου μεγέθους και προσημασμένου συμπληρώματος ως προς 1, το 0 εκφράζεται με 2 τρόπους, + 0 και – 0.

Και στους 3 τρόπους παράστασης με 4 δυαδικά ψηφία, ο μεγαλύτερος αριθμός που μπορεί να παρασταθεί είναι ο δεκαδικός αριθμός +7.

Ο μικρότερος αριθμός που μπορεί να παρασταθεί είναι ο δεκαδικός αριθμός – 7, με εξαίρεση την παράσταση προσημασμένου συμπληρώματος ως προς 2, στην οποία μπορεί να παρασταθεί και ο δεκαδικός αριθμός – 8.

Γενικεύοντας για n δυαδικά ψηφία, ο μεγαλύτερος αριθμός που μπορεί να παρασταθεί με τους 3 τρόπους είναι ο $2^{n-1} - 1$ και ο μικρότερος είναι ο αριθμός $-2^{n-1} + 1$, με εξαίρεση την παράσταση προσημασμένου συμπληρώματος ως προς 2, στην οποία μπορεί να παρασταθεί και ο αριθμός -2^{n-1} (μια μονάδα ακολουθούμενη από n – 1 μηδενικά).

Προσημασμένοι δυαδικοί αριθμοί

Παράδειγμα: Να εκφράσετε με 8 δυαδικά ψηφία στις παραστάσεις προσημασμένου μεγέθους και προσημασμένου συμπληρώματος ως προς 2 τους αριθμούς $+74_{10}$ και -74_{10}.

Το μέτρο του δεκαδικού αριθμού 74 με 7 δυαδικά ψηφία είναι 1001010. Το όγδοο ψηφίο χρησιμοποιείται για το πρόσημο.

Στην παράσταση προσημασμένου μεγέθους έχουμε:

$+74_{10} = \mathbf{0}1001010 \quad -74_{10} = \mathbf{1}1001010$

Στην παράσταση προσημασμένου συμπληρώματος ως προς 2 για $+74_{10}$ έχουμε την ίδια παράσταση, ενώ για το -74_{10} θα πρέπει να υπολογίσουμε το συμπλήρωμα ως προς 2 του θετικού αριθμού 01001010 (συμπεριλαμβανομένου και του ψηφίου προσήμου):

$-74_{10} = \Sigma_2(01001010) = \Sigma_1(01001010) + 1 = 10110101 + 1 = 10110110$

Πράξεις προσημασμένων δυαδικών αριθμών

Όπως προαναφέρθηκε, κατά την πρόσθεση 2 δυαδικών αριθμών εκφρασμένων σε παράσταση προσημασμένου μεγέθους, απαιτείται αρχικά η σύγκριση των προσήμων και του μέτρου των αριθμών και ακολούθως εκτελείται η πράξη της πρόσθεσης ή της αφαίρεσης.

Διακρίνουμε δύο περιπτώσεις:

Οι δύο αριθμοί είναι ομόσημοι (θετικοί ή αρνητικοί):

Στην περίπτωση αυτή προσθέτουμε τα μέτρα των δύο αριθμών και διατηρούμε το πρόσημο. Για παράδειγμα:

$(+74)_{10} + (+12)_{10} = 01001010 + 00001100 = 01010110 = (+86)_{10}$

$(-74)_{10} + (-12)_{10} = 11001010 + 10001100 = 11010110 = (-86)_{10}$

Παρατηρούμε ότι το μέτρο των αριθμών παραμένει το ίδιο, ενώ αλλάζει το ψηφίο πρόσημο.

Πράξεις προσημασμένων δυαδικών αριθμών

Οι δύο αριθμοί είναι ετερόσημοι:

Στην περίπτωση αυτή συγκρίνουμε τα μέτρα των αριθμών, αφαιρούμε το μικρότερο από το μεγαλύτερο και διατηρούμε το πρόσημο του μεγαλύτερου αριθμού.

Για παράδειγμα, για να προσθέσουμε τους αριθμούς $(+74)_{10}$ και $(-12)_{10}$, αρχικά αφαιρούμε τα μέτρα των δύο αριθμών:

$$(74)_{10} - (12)_{10} = 1001010 - 0001100 = 0111110 = (62)_{10}$$

Στη συνέχεια, θέτουμε στον αριθμό που προέκυψε το ψηφίο πρόσημο, το οποίο είναι 0, αφού το αποτέλεσμα της αφαίρεσης 74 – 12 είναι θετικός αριθμός.

Επομένως, το αποτέλεσμα της πρόσθεσης είναι ο θετικός αριθμός:

$$(+74)_{10} + (-12)_{10} = 00111110 = (+62)_{10}$$

Πράξεις προσημασμένων δυαδικών αριθμών

Αντίστοιχα, κατά την πρόσθεση των αριθμών $(-74)_{10}$ και $(+12)_{10}$, θα έχουμε ως αποτέλεσμα έναν αρνητικό αριθμό, με το ίδιο μέτρο όπως προηγουμένως, αλλά με ψηφίο προσήμου 1:

$$(-74)_{10} + (+12)_{10} = 10111110 = (-62)_{10}$$

Πράξεις προσημασμένων δυαδικών αριθμών

Η αφαίρεση δύο προσημασμένων δυαδικών αριθμών σε παράσταση προσημασμένου μεγέθους ανάγεται σε πρόσθεση του μειωτέου με τον αντίθετο αριθμό του αφαιρετέου, ο οποίος εκφράζεται ως το συμπλήρωμα ως προς 2 του αφαιρετέου και η πράξη εκτελείται σύμφωνα με τα προαναφερόμενα.

Στην πρόσθεση δύο προσημασμένων δυαδικών αριθμών σε παράσταση προσημασμένου συμπληρώματος ως προς 2, συμπεριλαμβάνεται και το ψηφίο που κατέχει θέση προσήμου και η πράξη γίνεται χωρίς προηγούμενη επεξεργασία.

Εάν στο αποτέλεσμα της πρόσθεσης, το οποίο, επίσης, παριστάνεται με μορφή προσημασμένου συμπληρώματος ως προς 2, προκύψει τελικό κρατούμενο (δηλαδή κρατούμενο στην πιο σημαντική θέση), αυτό παραλείπεται.

Πράξεις προσημασμένων δυαδικών αριθμών

Για την αφαίρεση δύο προσημασμένων δυαδικών αριθμών σε παράσταση προσημασμένου συμπληρώματος ως προς 2, υπολογίζουμε το συμπλήρωμα ως προς 2 του αφαιρετέου (συμπεριλαμβανομένου του ψηφίου προσήμου) και το προσθέτουμε στον μειωτέο. Εάν προκύψει κρατούμενο στη θέση του ψηφίου προσήμου, αυτό παραλείπεται.

Η διαδικασία αυτή έχει καθιερωθεί επειδή η πράξη της αφαίρεσης μπορεί να μετατραπεί σε πράξη πρόσθεσης, εάν απλά αλλάξουμε το πρόσημο του αφαιρετέου και εκτελέσουμε πρόσθεση:

$$(\pm A) - (+ B) = (\pm A) + (- B), \quad (\pm A) - (- B) = (\pm A) + (+ B)$$

Η αλλαγή ενός θετικού αριθμού σε αρνητικό προκύπτει απλά, υπολογίζοντας το συμπλήρωμα ως προς 2 του θετικού αριθμού. Ισχύει επίσης το αντίστροφο, αφού το συμπλήρωμα ως προς 2 ενός αρνητικού αριθμού, παράγει τον αντίστοιχο θετικό αριθμό.

Πράξεις προσημασμένων δυαδικών αριθμών

Παράδειγμα: Δίνονται οι δεκαδικοί αριθμοί A = +4 και B = –11. Να εκφραστούν οι αριθμοί αυτοί σε παράσταση προσημασμένου συμπληρώματος ως προς 2 και να πραγματοποιηθούν οι πράξεις: A + B, A – B, – A + B και – A – B.

Αρχικά προσδιορίζουμε το πλήθος των απαιτούμενων ψηφίων για την έκφραση των αριθμών σε παράσταση προσημασμένου συμπληρώματος ως προς 2. Αυτό ισούται με το πλήθος δυαδικών ψηφίων που απαιτούνται για την έκφραση του μέτρου των αριθμών σε δυαδική μορφή, πλέον ενός ψηφίου για το πρόσημο. Στην περίπτωσή μας έχουμε: $4_{10} = 100_2$ και $11_{10} = 1011_2$

Παρατηρούμε ότι για την έκφραση του μεγέθους σε δυαδική μορφή, οι δύο αριθμοί απαιτούν διαφορετικό πλήθος ψηφίων.

Πράξεις προσημασμένων δυαδικών αριθμών

Για την εκτέλεση πράξεων μεταξύ αριθμών σε παράσταση προσημασμένου μεγέθους ή προσημασμένου συμπληρώματος ως προς 2 και οι δύο αριθμοί θα πρέπει να εκφραστούν με το ίδιο πλήθος ψηφίων, άρα ο μεγαλύτερος κατά μέτρο αριθμός καθορίζει και το πλήθος των ψηφίων με το οποίο θα παρασταθούν.

Επομένως, σε παράσταση προσημασμένου μεγέθους, οι αριθμοί μας θα πρέπει να εκφραστούν με 5 δυαδικά ψηφία, 4 ψηφία για το μέτρο και ένα ψηφίο πρόσημο.

Ο αριθμός A είναι θετικός, άρα η έκφρασή σε παράσταση προσημασμένου συμπληρώματος ως προς 2 είναι ίδια με αυτήν σε παράσταση προσημασμένου μεγέθους και, με 5 δυαδικά ψηφία συμπεριλαμβανομένου του ψηφίου προσήμου, θα είναι:

A = +4 = **0**0100

Πράξεις προσημασμένων δυαδικών αριθμών

Η παράσταση προσημασμένου συμπληρώματος ως προς 2 του αρνητικού αριθμού B = –11 προκύπτει από τον υπολογισμό του συμπληρώματος ως προς 2 του αντίστοιχου θετικού αριθμού (01011).

$$B = \Sigma_2(01011) = \Sigma_1(01011) + 1 = 10100 + 1 = 10101$$

A + B

	0	0	1	0	0	$A = +4$
+	1	0	1	0	1	$B = -11$
	1	1	0	0	1	$(-7)_{10}$

A – B

	0	0	1	0	0	$A = +4$
+	0	1	0	1	1	$-B = +11$
	0	1	1	1	1	$(+15)_{10}$

$\Sigma_2(11001) = 00111 = +7$

– A + B

	1	1	1	0	0	$-A = -4$
+	1	0	1	0	1	$B = -11$
	1	0	0	0	1	$(-15)_{10}$

– A – B

	1	1	1	0	0	$-A = -4$
+	0	1	0	1	1	$-B = +11$
	0	0	1	1	1	$(+7)_{10}$

$\Sigma_2(10001) = 01111 = +15$

Πράξεις προσημασμένων δυαδικών αριθμών

- Κατά την πρόσθεση ή αφαίρεση δύο προσημασμένων αριθμών n δυαδικών ψηφίων, υπάρχει πιθανότητα το αποτέλεσμα να απαιτεί για την ορθή παράστασή του n + 1 ψηφία.

  Στην περίπτωση αυτή συμβαίνει μετατόπιση του ψηφίου-προσήμου από την αναμενόμενη θέση και η κατάσταση αυτή αναφέρεται ως υπερχείλιση (overflow).

  Η υπερχείλιση αποτελεί πρόβλημα στα ψηφιακά συστήματα, στα οποία το πλήθος των θέσεων μνήμης όπου αποθηκεύονται οι αριθμοί είναι συγκεκριμένο, με συνέπεια να μην μπορεί να αποθηκευτεί το αποτέλεσμα μιας πράξης που το πλήθος των ψηφίων του υπερβαίνει το πλήθος των διαθέσιμων θέσεων μνήμης.

  Στους ψηφιακούς επεξεργαστές, το πρόβλημα της υπερχείλισης αντιμετωπίζεται με χρήση ειδικής θέσης μνήμης, που το περιεχόμενό της υποδεικνύει την υπερχείλιση στην πράξη που εκτελέστηκε.

Πράξεις προσημασμένων δυαδικών αριθμών

Υπερχείλιση κατά την πρόσθεση 2 προσημασμένων αριθμών συμβαίνει όταν οι αριθμοί είναι ομόσημοι και το αποτέλεσμα που προκύπτει έχει διαφορετικό πρόσημο από αυτούς.

Κατά την πρόσθεση των θετικών δυαδικών αριθμών 0111 = $(+7)_{10}$ και 0100 = $(+4)_{10}$ προκύπτει λανθασμένο αρνητικό αποτέλεσμα, δηλαδή 1011 = $(-5)_{10}$, λόγω της υπερχείλισης.

Αυτό συμβαίνει διότι το ορθό αποτέλεσμα 01011 = $(+11)_{10}$ είναι μεγαλύτερο από το μεγαλύτερο δυνατό θετικό αριθμό που μπορεί να παρασταθεί με 4 δυαδικά ψηφία, δηλαδή τον αριθμό $(+7)_{10}$.

Επίσης, κατά την πρόσθεση των αρνητικών δυαδικών αριθμών 1100 = $(-4)_{10}$ και 1010 = $(-6)_{10}$ προκύπτει λανθασμένο θετικό αποτέλεσμα, δηλαδή 0110 = $(+6)_{10}$, λόγω της υπερχείλισης.

Αυτό συμβαίνει διότι το ορθό αποτέλεσμα 10110 = $(-10)_{10}$ είναι μικρότερο από το μικρότερο αρνητικό αριθμό με 4 ψηφία $(-8)_{10}$.

Αριθμοί κινητής υποδιαστολής

Στην παράσταση αριθμών με ακέραιο και κλασματικό μέρος που μελετήσαμε, η θέση της υποδιαστολής είναι προκαθορισμένη και σταθερή. Οι αριθμοί που ακολουθούν αυτή την παράσταση χαρακτηρίζονται ως αριθμοί σταθερής υποδιαστολής.

Το εύρος τιμών που μπορεί να εκφραστεί με αυτόν τον τρόπο παράστασης είναι σχετικά μικρό. Για παράδειγμα, ο μεγαλύτερος ακέραιος μη προσημασμένος αριθμός που μπορεί να παρασταθεί με 32 δυαδικά ψηφία (bits) είναι:

$$2^{32} - 1 \approx 4.3 \times 10^{9}.$$

Όμως στους επιστημονικούς υπολογισμούς χρησιμοποιούμαι αριθμούς πολύ μεγαλύτερης τάξης (π.χ. αριθμός Avogadro = 6.022×10^{23} $mole^{-1}$), καθώς και αριθμούς με πολύ μικρό κλασματικό μέρος (π.χ. σταθερά Boltzmann = 1.38×10^{-23} Joule/^{o}K), για την παράσταση των οποίων θα απαιτούνταν πολύ μεγάλο πλήθος δυαδικών ψηφίων.

Αριθμοί κινητής υποδιαστολής

Για να καλυφθεί αυτή η ανάγκη, στα ψηφιακά συστήματα έχει υιοθετηθεί η παράσταση αριθμών κινητής υποδιαστολής, στην οποία η θέση της υποδιαστολής των αριθμών δεν είναι προκαθορισμένη αλλά μεταβλητή, για να προσαρμόζεται στις υπολογιστικές ανάγκες.

Για την παράσταση αριθμών κινητής υποδιαστολής ακολουθείται το πρότυπο του IEEE 754 (Institute of Electrical & Electronic Engineering).

Σύμφωνα με αυτό, οι αριθμοί κινητής υποδιαστολής παριστάνονται με 32 δυαδικά ψηφία (παράσταση απλής ακρίβειας) ή με 64 δυαδικά ψηφία (παράσταση διπλής ακρίβειας) και περιλαμβάνουν 3 τμήματα:

- το ψηφίο πρόσημο **S** (που έχει τιμή 0 για θετικούς και τιμή 1 για αρνητικούς αριθμούς),
- τον εκθέτη **E**, και
- το κλασματικό μέρος **M** (mantissa).

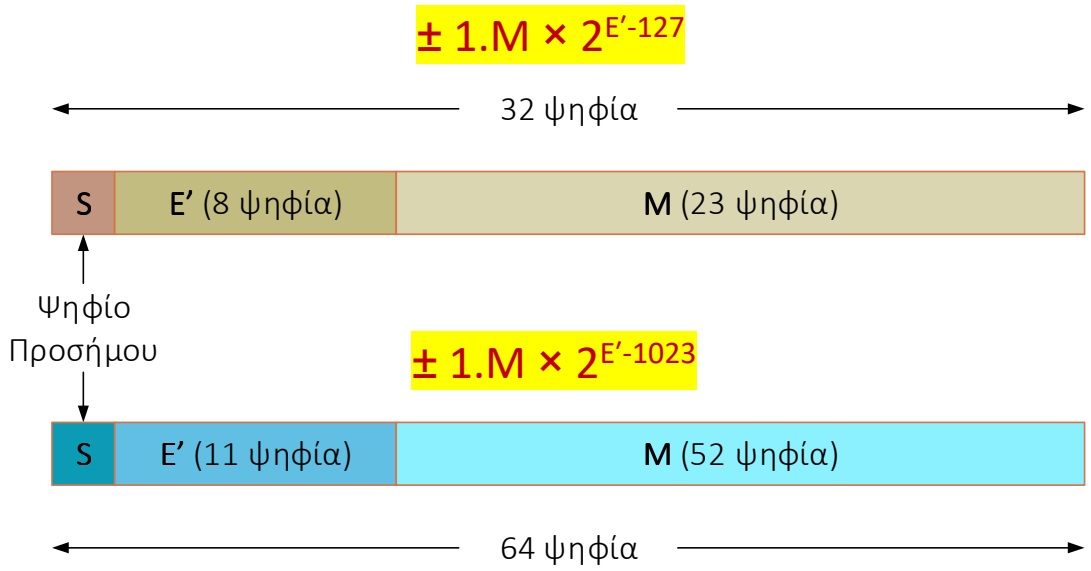
Αριθμοί κινητής υποδιαστολής

Στην παράσταση απλής ακρίβειας, ο εκθέτης εκφράζεται με 8 δυαδικά ψηφία που ακολουθούν μετά το πρόσημο και το κλασματικό μέρος εκφράζεται με τα 23 λιγότερο σημαντικά ψηφία.

Στην πραγματικότητα, στην παράσταση αυτή, αντί για τον προσημασμένο εκθέτη E, χρησιμοποιείται ένας μη προσημασμένος αριθμός E', τέτοιος ώστε: E' = E + 127 και λαμβάνει τιμές από 0 έως 255.

Στην παράσταση διπλής ακρίβειας (64 bits), ο εκθέτης εκφράζεται με 11 δυαδικά ψηφία που ακολουθούν μετά το πρόσημο και το κλασματικό μέρος εκφράζεται με 52 δυαδικά ψηφία. Ο αριθμός E' είναι τέτοιος ώστε: E' = E + 1023 και λαμβάνει τιμές από 0 έως 2047.

Αριθμοί κινητής υποδιαστολής



Δομή των αριθμών κινητής υποδιαστολής απλής και διπλής ακρίβειας, σύμφωνα με το πρότυπο IEEE 754

Αριθμοί κινητής υποδιαστολής

Παράδειγμα: Παράσταση κινητής υποδιαστολής απλής ακρίβειας του δεκαδικού αριθμού 25.75.

Ο αριθμός είναι θετικός, επομένως S = 0.

Το ακέραιο μέρος του αριθμού σε δυαδική μορφή είναι $25_{10} = 11001_2$

Το κλασματικό μέρος του αριθμού είναι $0.75_{10} = 0.11_2$

Άρα, $25.75 \times 2^0{}_{10} = 11001.11 \times 2^0{}_2$

Μετατοπίζουμε την υποδιαστολή προς τα αριστερά, αφήνοντας μόνο μια μονάδα αριστερά της, δηλαδή, στην περίπτωσή μας, μετατοπίζουμε την υποδιαστολή 4 θέσεις (ψηφία) προς τα αριστερά, αυξάνοντας αντίστοιχα τον εκθέτη. Επομένως ο δυαδικός αριθμός γίνεται 1.100111×2^4.

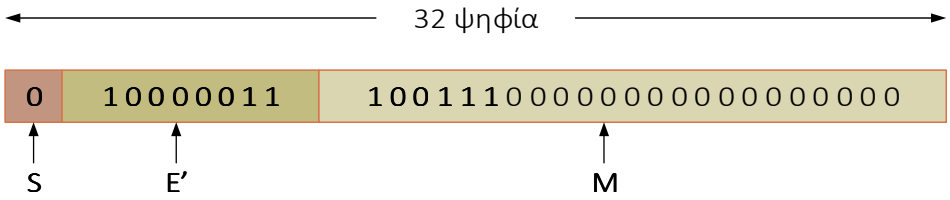
Ο εκθέτης του αριθμού είναι E = 4, οπότε E' = E + 127 = 131, επομένως E' = 10000011.

Αριθμοί κινητής υποδιαστολής

Η πρώτη μονάδα (ακέραιο μέρος του αριθμού) δεν παριστάνεται και δεν αποθηκεύεται, αλλά πάντα υπονοείται ότι υπάρχει.

Τα υπόλοιπα ψηφία αποτελούν το κλασματικό μέρος του αριθμού M (mantissa), που συμπληρώνεται με μηδενικά προς τα δεξιά, ώστε να περιλαμβάνει συνολικά 23 bits.

Έτσι το κλασματικό μέρος είναι M = 10011100000000000000000 (συμπληρώνουμε 17 μηδενικά).



Παράσταση του δεκαδικού αριθμού 25.75 σύμφωνα με το πρότυπο IEEE 754 απλής ακρίβειας

Αριθμοί κινητής υποδιαστολής

Αφού η μονάδα που βρίσκεται αριστερά της υποδιαστολής (ακέραιο μέρος) δεν παριστάνεται, αλλά θεωρείται ότι υπάρχει στη θέση αυτή, οι αριθμοί που παριστάνονται σύμφωνα με το πρότυπο IEEE 754 θα πρέπει να διαμορφώνονται με τέτοιο τρόπο ώστε να υπάρχει μόνο μια μονάδα στη θέση αριστερά της υποδιαστολής (ακέραιο μέρος πάντα ίσο με 1).

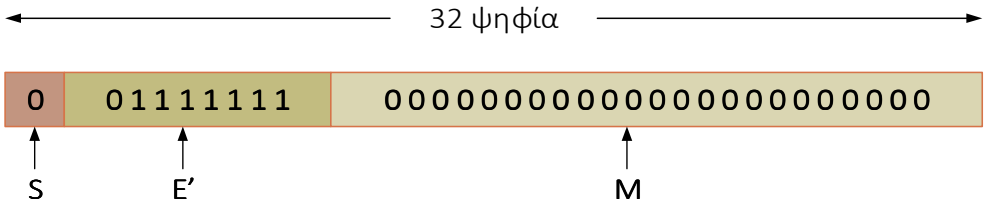
Η διαδικασία αυτή αναφέρεται ως κανονικοποίηση.

Σύμφωνα με τα παραπάνω βήματα, για να παραστήσουμε τον αριθμό $+1_{10}$ στο πρότυπο αυτό, θα πρέπει να γράψουμε τον αριθμό σε μορφή με ακέραιο και κλασματικό μέρος, π.χ. 1.00.

Το πεδίο προσήμου είναι S = 0.

Αριθμοί κινητής υποδιαστολής

Το κλασματικό μέρος M του αριθμού παριστάνεται με 23 μηδενικά ψηφία και η τιμή του εκθέτη είναι E = 0 (αφού $1 = 1 \times 2^0$), οπότε θα είναι E' = E + 127 = 0 + 127 = 127, με το πεδίο του εκθέτη θα έχει τιμή 01111111.

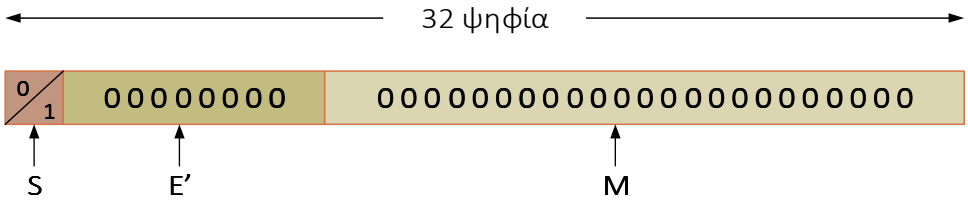


Παράσταση του δεκαδικού αριθμού 1 σύμφωνα με το πρότυπο IEEE 754 απλής ακρίβειας

Αριθμοί κινητής υποδιαστολής

Αν προσπαθήσουμε να παραστήσουμε τον αριθμό 0 σύμφωνα με την παραπάνω διαδικασία, προκύπτει πρόβλημα, αφού θα έχουμε την ίδια ακριβώς παράσταση όπως και για τον αριθμό 1.

Για το λόγο αυτό, έχει καθοριστεί στο πρότυπο IEEE 754, η τιμή του 0 να αναπαριστάνεται με τις τιμές E' = 0 (8 bits) και M = 0 (23 bits), ενώ το πρόσημο μπορεί να είναι 0 ή 1



Παράσταση του αριθμού 0 σύμφωνα με το πρότυπο IEEE 754 απλής ακρίβειας

Αριθμοί κινητής υποδιαστολής

Υπάρχουν ακόμα δύο σημαντικές συμβάσεις για τις παραστάσεις αριθμών κινητής υποδιαστολής στο πρότυπο IEEE 754.

Η πρώτη αφορά την περίπτωση που είναι E’ = 255 (και τα 8 ψηφία έχουν τιμή 1) και M ≠ 0**.**

Τότε, η παράσταση θεωρείται ως μη-αριθμός (Not a Number – NaN) και αντιστοιχεί στο αποτέλεσμα μιας άκυρης πράξης (π.χ. η τετραγωνική ρίζα ενός αρνητικού αριθμού).

Η δεύτερη αφορά την περίπτωση που είναι E’ = 255 και M = 0. Η παράσταση αυτή αντιστοιχεί στη διαίρεση ενός αριθμού με το 0, το αποτέλεσμα της οποίας είναι η τιμή του απείρου (∞).

Αριθμοί κινητής υποδιαστολής

Παρατηρούμε λοιπόν ότι οι ακραίες τιμές του εκθέτη (0 ή 255) χρησιμοποιούνται κατά σύμβαση στο πρότυπο IEEE για την παράσταση ειδικών τιμών.

Επομένως, το πραγματικό εύρος του εκθέτη E’ για τις υπόλοιπες (εκτός των ακραίων) τιμών είναι από 1 έως και 254, δηλαδή ο πραγματικός εκθέτης E λαμβάνει τιμές από -126 έως και 127.

Όλα τα προαναφερθέντα για την παράσταση αριθμών σύμφωνα με το πρότυπο κινητής υποδιαστολής απλής ακρίβειας IEEE 754, ισχύουν παρομοίως και στην παράσταση διπλής ακρίβειας, με τις αντίστοιχες βεβαίως προσαρμογές ως πρός τις ακραίες τιμές του εκθέτη, που στην περίπτωση αυτή αποτελείται από 11 bits και λαμβάνει τιμές στο διάστημα 0 έως 2047.

Πράξεις αριθμών κινητής υποδιαστολής

Για να εκτελέσουμε την πρόσθεση ή την αφαίρεση δύο αριθμών σε παράσταση κινητής υποδιαστολής ακολουθούμε την εξής διαδικασία:

1. Θα πρέπει και οι δύο αριθμοί να έχουν τον ίδιο εκθέτη. Αν έχουν διαφορετικό εκθέτη, υπολογίζουμε τη διαφορά των εκθετών των 2 αριθμών και μετατοπίζουμε προς τα δεξιά τον αριθμό με τον μικρότερο εκθέτη (το κλασματικό μέρος του και το ψηφίο που βρίσκεται αριστερά της υποδιαστολής) τόσες θέσεις όσες είναι η διαφορά. Ως εκθέτης του αποτελέσματος λαμβάνεται ο μεγαλύτερος από τους εκθέτες των δύο αριθμών.
2. Εκτελούμε την πρόσθεση ή αφαίρεση των κλασματικών μερών των δύο αριθμών και καθορίζουμε το πρόσημο του αθροίσματος ή της διαφοράς.
3. Κανονικοποιούμε το αποτέλεσμα, εάν χρειάζεται.

Πράξεις αριθμών κινητής υποδιαστολής

Παράδειγμα: Δίνονται οι αριθμοί $A = +28.625_{10}$ και $B = 118.5_{10}$. Να εκφραστούν σε παράσταση κινητής υποδιαστολής απλής ακρίβειας και να εκτελεστούν οι πράξεις A + B και A – B.

$A = (28.625)_{10} = (11100.101)_2 \times 2^0$

Μετακινούμε την υποδιαστολή 4 θέσεις αριστερά, ώστε να έχουμε μόνο μια μονάδα αριστερά της (ακέραιο μέρος), οπότε θα έχουμε $A = 1.1100101 \times 2^4$.

Επομένως, σε παράσταση κινητής υποδιαστολής απλής ακρίβειας, θα έχουμε: S = 0, E = 4, άρα E’ = 4 + 127 = 131 = 10000011, και M = 11001010000000000000000.

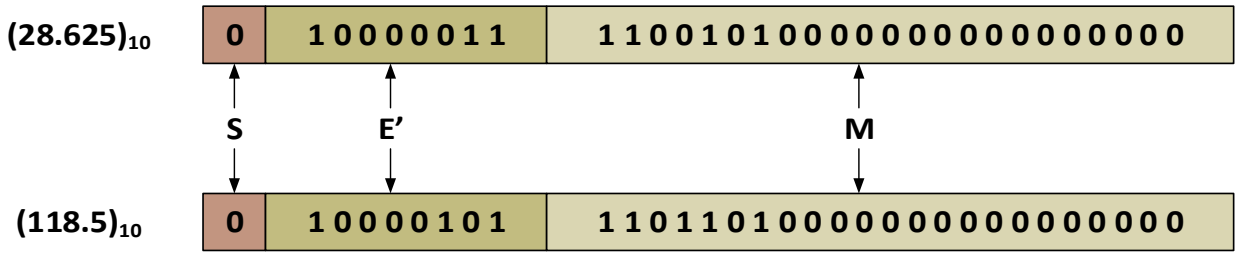
Το κλασματικό μέρος του αριθμού συμπληρώνεται με 16 μηδενικά, ώστε να περιλαμβάνει συνολικά 23 ψηφία.

Πράξεις αριθμών κινητής υποδιαστολής

Ακολουθούμε την ίδια διαδικασία για τον αριθμό Β.

$B = (118.5)_{10} = (1110110.1)_2 \times 2^0 = 1.1101101 \times 2^6$

Άρα, S = 0, E = 6, E' = 6 + 127 = 133 = 100000101,
M = 11011010000000000000000



Παράσταση των δεκαδικών αριθμών 28.625 και 118.5 σύμφωνα με το πρότυπο IEEE 754 απλής ακρίβειας

Πράξεις αριθμών κινητής υποδιαστολής

Για να εκτελεστούν οι πράξεις πρέπει οι δύο αριθμοί να έχουν τον ίδιο εκθέτη. Επειδή ο Α έχει εκθέτη 4 και ο Β έχει εκθέτη 6, θα πρέπει να εκφραστεί και ο Α με τον μεγαλύτερο εκθέτη, δηλαδή 6.

Η διαφορά των εκθετών είναι 6 – 4 = 2. Άρα, μετατοπίζουμε κατά 2 θέσεις προς τα δεξιά τα ψηφία του αριθμού Α, συνεπώς η μετατοπισμένη μορφή του είναι 0.011100101×2^6.

Ακολούθως, εκτελούμε τις ζητούμενες αριθμητικές πράξεις.

Πρόσθεση A + B

0 . 0 1 1 1 0 0 1 0 1
+ 1 . 1 1 0 1 1 0 1 0 0
10 . 0 1 0 0 1 1 0 0 1

Πράξεις αριθμών κινητής υποδιαστολής

Παρατηρούμε ότι προκύπτει τελικό κρατούμενο με αποτέλεσμα το άθροισμα να μην είναι σε κανονική μορφή με μόνο μια μονάδα αριστερά της υποδιαστολής.

Επομένως, απαιτείται κανονικοποίηση του.

Αυτό γίνεται μετατοπίζοντας την υποδιαστολή κατά μία θέση προς τα αριστερά, που ισοδυναμεί με την αύξηση της τιμής του εκθέτη κατά μία μονάδα, δηλαδή E= 6 + 1 =7, οπότε το τελικό αποτέλεσμα, που ήταν 10.010011001×2^6, θα γίνει

1.0010011001×2^7, και θα έχει

S = 0, E = 7, E' = 7 + 127 = 134 = 100000110

M = 00100110010000000000000

Πράξεις αριθμών κινητής υποδιαστολής

$(28.625)_{10}$	0	10000011	11001010000000000000000
+ $(118.5)_{10}$	0	10000101	11011010000000000000000
= $(147.125)_{10}$	0	10000110	00100110010000000000000

Παράσταση των δεκαδικών αριθμών 28.625, 118.5 και του αθροίσματός τους (147.125), σύμφωνα με το πρότυπο IEEE 754 απλής ακρίβειας

Πράξεις αριθμών κινητής υποδιαστολής

Αφαίρεση A – B: Αφαιρούμε τα κλασματικά μέρη των δύο αριθμών. Αυτό ανάγεται σε πρόσθεση του κλασματικού μέρους του A με το συμπλήρωμα ως προς 2 του κλασματικού μέρους του B.

Επειδή το κλασματικό μέρος του B είναι μεγαλύτερο από εκείνο του A, η διαφορά A – B θα είναι ένας αρνητικός αριθμός, συνεπώς S = 1.

Στο παράδειγμά μας, το κλασματικό μέρος του B είναι 1101101, επομένως το συμπλήρωμά του ως προς 2 θα είναι:

$$\Sigma_2(1101101) = \Sigma_1(1101101) + 1 = 0010010 + 1 = 0010011$$

$$\begin{array}{r} 0.0111001\,01 \\ +\,0.0010011\,00 \\ \hline 0.1001100\,01 \end{array}$$

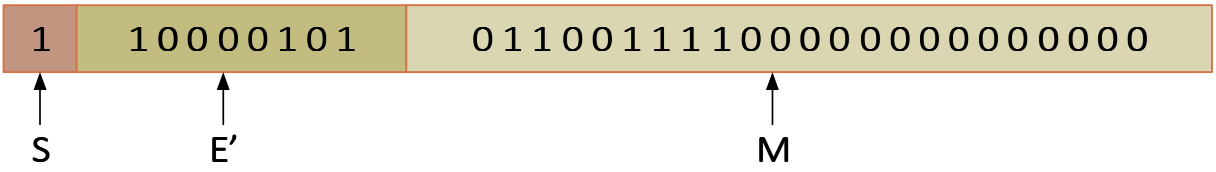
Πράξεις αριθμών κινητής υποδιαστολής

Επειδή το κλασματικό μέρος του B είναι μεγαλύτερο από εκείνο του A, το παραπάνω αποτέλεσμα παριστάνεται σε μορφή συμπληρώματος ως προς 2, άρα, για λάβουμε την κανονική μορφή του, θα πρέπει να υπολογίσουμε το συμπλήρωμά του ως προς 2:

$$\begin{aligned} \Sigma_2(0.100110001) &= \Sigma_1(0.100110001) + 1 \\ &= 1.011001110 + 1 \\ &= 1.011001111 \end{aligned}$$

Συνεπώς, η διαφορά που προκύπτει είναι 1.011001111×2^6 και παριστάνεται ως αριθμός κινητής υποδιαστολής απλής ακρίβειας.

Πράξεις αριθμών κινητής υποδιαστολής



Παράσταση του δεκαδικού αριθμού $A - B = (28.625)_{10} - (118.5)_{10} = (-89.875)_{10}$ σύμφωνα με το πρότυπο IEEE 754 απλής ακρίβειας

Πράξεις αριθμών κινητής υποδιαστολής

Για να εκτελέσουμε την πράξη του πολλαπλασιασμού δύο αριθμών εκφρασμένων σε παράσταση κινητής υποδιαστολής ακολουθούμε την εξής διαδικασία:

1. Προσθέτουμε τους εκθέτες των δύο αριθμών και από το άθροισμα που προκύπτει αφαιρούμε τον αριθμό 127, όταν πρόκειται για αριθμούς απλής ακρίβειας, ή τον αριθμό 1023, όταν πρόκειται για αριθμούς διπλής ακρίβειας.
2. Ακολούθως, εκτελούμε τον πολλαπλασιασμό των δύο αριθμών (το κλασματικό μέρος συν το ψηφίο που βρίσκεται αριστερά της υποδιαστολής).
3. Στη συνέχεια, καθορίζουμε το πρόσημο του γινομένου.
4. Τέλος, κανονικοποιούμε το αποτέλεσμα, εάν χρειάζεται.

Πράξεις αριθμών κινητής υποδιαστολής

Αντίστοιχα, για να εκτελέσουμε την πράξη της διαίρεσης δύο αριθμών εκφρασμένων σε παράσταση κινητής υποδιαστολής ακολουθούμε την εξής διαδικασία:

1. Αφαιρούμε τους εκθέτες των δύο αριθμών και στη διαφορά που προκύπτει προσθέτουμε τον αριθμό 127, όταν πρόκειται για αριθμούς απλής ακρίβειας, ή τον αριθμό 1023, όταν πρόκειται για αριθμούς διπλής ακρίβειας.
2. Ακολούθως, εκτελούμε τη διαίρεση των δύο αριθμών (το κλασματικό μέρος συν το ψηφίο που βρίσκεται αριστερά της υποδιαστολής).
3. Στη συνέχεια, καθορίζουμε το πρόσημο του πηλίκου.
4. Τέλος, κανονικοποιούμε το αποτέλεσμα, εάν χρειάζεται.

Δυαδικοί κώδικες

Στα ψηφιακά συστήματα, εκτός από τους δυαδικούς αριθμούς, παριστάνονται, επεξεργάζονται, αποθηκεύονται και μεταδίδονται διακριτά στοιχεία πληροφορίας, όπως γράμματα, αριθμοί, χαρακτήρες, σύμβολα κλπ.

Κάθε διακριτό στοιχείο πληροφορίας μπορεί να παρασταθεί με τη χρήση ενός δυαδικού κώδικα, δηλαδή, με τη μορφή ενός μοναδικού συνδυασμού δυαδικών ψηφίων 0 και 1.

Από έναν δυαδικό κώδικα με n bits προκύπτουν 2^n διαφορετικοί συνδυασμοί από n bits ο καθένας.

Κάθε συνδυασμός παριστάνει ένα και μοναδικό (διακριτό) στοιχείο της πληροφορίας που κωδικοποιείται.

Απαγορεύεται η χρήση του ίδιου συνδυασμού ψηφίων για περισσότερα από ένα στοιχεία, γιατί τότε θα είχαμε έναν ασαφώς ορισμένο κώδικα.

Δυαδικοί κώδικες

Αν και ο ελάχιστος αριθμός ψηφίων που απαιτείται για την κωδικοποίηση 2^n διακριτών στοιχείων είναι n, δεν τίθεται μέγιστο πλήθος ψηφίων που μπορούν να χρησιμοποιηθούν για τη δημιουργία ενός δυαδικού κώδικα.

Για παράδειγμα, τα 10 ψηφία του δεκαδικού συστήματος (0, 1, 2, ..., 9) μπορούν να κωδικοποιηθούν με 10 δυαδικά ψηφία το καθένα, δηλαδή κάθε δεκαδικό ψηφίο να παριστάνεται με ένα συνδυασμό από δέκα δυαδικά ψηφία 0 και 1.

Το ψηφίο 6 θα μπορούσε να αντιστοιχεί στο συνδυασμό 0001000000, το ψηφίο 3 στο συνδυασμό 0000001000 και το ψηφίο 0 στο συνδυασμό 0000000001.

Οι 3 κύριοι τύποι δυαδικών κωδίκων είναι οι αριθμητικοί, οι αλφαριθμητικοί και οι κώδικες ανίχνευσης / διόρθωσης σφαλμάτων.

Κώδικας BCD

Στον αριθμητικό κώδικα BCD, Binary Coded Decimal (δυαδικά κωδικοποιημένοι δεκαδικοί) κάθε ψηφίο ενός δεκαδικού αριθμού εκφράζεται ξεχωριστά σε δυαδική μορφή.

Η παράσταση ενός δεκαδικού αριθμού σε κώδικα BCD απαιτεί περισσότερα δυαδικά ψηφία από την αντίστοιχη δυαδική.

Το πλεονέκτημα, ωστόσο, της κωδικοποίησης αυτής είναι η εξοικείωση του ανθρώπου με τους δεκαδικούς αριθμούς, σε συνδυασμό με τη χρήση του δυαδικού συστήματος που χρησιμοποιείται στα ψηφιακά συστήματα.

Στα ψηφιακά υπολογιστικά συστήματα εισάγουμε (π.χ. πληκτρολογούμε) δεκαδικούς αριθμούς, αυτοί μετατρέπονται σε δυαδική μορφή, εκτελούνται οι αριθμητικοί υπολογισμοί με δυαδικό τρόπο και τα αποτελέσματα μετατρέπονται και λαμβάνονται σε δεκαδική μορφή.

Κώδικας BCD

Επειδή το μεγαλύτερο δεκαδικό ψηφίο (9), εκφράζεται με 4 δυαδικά ψηφία (1001), όλα τα δεκαδικά ψηφία θα εκφράζονται επίσης με 4 δυαδικά ψηφία, όπως παρουσιάζεται στον πίνακα που ακολουθεί.

Κώδικας BCD	
Δεκαδικό ψηφίο	Κωδικοποίηση BCD
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Κώδικας BCD

Από τους 16 δυνατούς συνδυασμούς 4 δυαδικών ψηφίων, στην κωδικοποίηση BCD χρησιμοποιούνται μόνο οι 10 πρώτοι (0000 έως 1001), ενώ οι υπόλοιποι 6 συνδυασμοί (1010 έως 1111, δεν χρησιμοποιούνται.

Ένας δεκαδικός αριθμός με n ψηφία, σε κωδικοποίηση BCD περιλαμβάνει n × 4 δυαδικά ψηφία (4 δυαδικά ψηφία για κάθε δεκαδικό ψηφίο)

Για παράδειγμα, ο αριθμός 295_{10} σε δυαδική παράσταση και σε κωδικοποίηση BCD κωδικοποίηση είναι:

$$(295)_{10} = (100100111)_2 = (\ 0010\ 1001\ 0101\ )_{BCD}.$$

Παρατηρούμε ότι, η κωδικοποίηση BCD απαιτεί 12 bits, ενώ ο ισοδύναμος δυαδικός αριθμός αποτελείται από 9 bits, δηλαδή η κωδικοποίηση BCD απαιτεί περισσότερα δυαδικά ψηφία.

Κώδικας BCD

Είναι σημαντικό να επισημάνουμε ότι οι αριθμοί BCD αναπαριστούν δεκαδικούς αριθμούς και δεν είναι δυαδικοί αριθμοί.

Ωστόσο, επειδή εκφράζονται σε δυαδική μορφή, μπορούμε να εφαρμόζουμε τους κανόνες που ισχύουν στο δυαδικό αριθμητικό σύστημα, λαμβάνοντας φυσικά υπόψη μας τους περιορισμούς που τίθενται από τον τρόπο που γίνεται η κωδικοποίηση.

Κατά την πρόσθεση δύο δεκαδικών ψηφίων κωδικοποιημένων κατά BCD, όταν το δυαδικό άθροισμα είναι ίσο ή μικρότερο του 1001 (δηλαδή του 9), τότε ταυτίζεται με το άθροισμα κατά BCD.

Κώδικας BCD

Στην περίπτωση όμως που το δυαδικό άθροισμα είναι μεγαλύτερο του 1001 (και προφανώς δεν περιλαμβάνεται στον κώδικα BCD), για να ληφθεί το σωστό άθροισμα κατά BCD, θα πρέπει να προσθέσουμε στο δυαδικό άθροισμα τον αριθμό 0110 (= 6_{10}), που είναι ίσος με το πλήθος των μη χρησιμοποιούμενων στον κώδικα BCD συνδυασμών των 4 δυαδικών ψηφίων.

Αυτό θα έχει αποτέλεσμα τη δημιουργία κρατούμενου, το οποίο θα πρέπει να ληφθεί υπόψη κατά την πρόσθεση των κωδικοποιημένων κατά BCD ψηφίων της αμέσως επόμενης πιο σημαντικής θέσης.

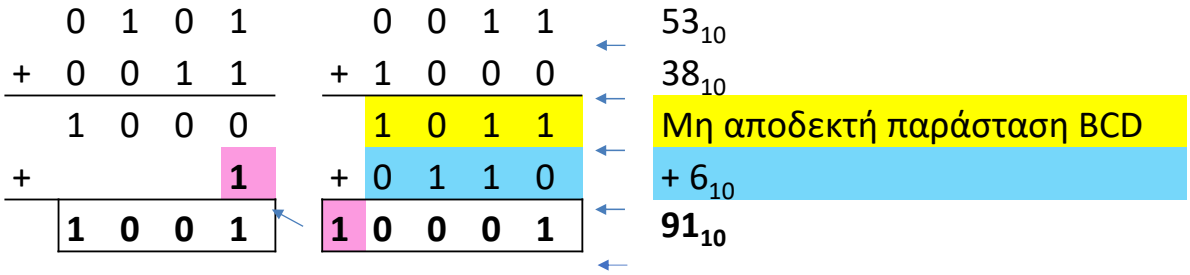
Η διαδικασία αυτή αναφέρεται ως διόρθωση κρατούμενου.

Κώδικας BCD

Παράδειγμα: Να εκφραστούν οι δεκαδικοί αριθμοί 53 και 38 σε κώδικα BCD και να εκτελεστεί η πράξη της πρόσθεσης.

Η BCD κωδικοποίηση των δύο αριθμών είναι: 53_{10} = 0101 0011 και 38_{10} = 0011 1000

Η πρόσθεση των δυαδικών ψηφίων γίνεται σύμφωνα με τους κανόνες του δυαδικού αριθμητικού συστήματος. Επομένως:



Κώδικας Gray

Ο κώδικας Gray κωδικοποιεί αριθμούς με δυαδικά ψηφία κατά τέτοιο τρόπο ώστε, κατά τη μετάβαση μεταξύ δύο διαδοχικών αριθμών να αλλάζει τιμή μόνο ένα δυαδικό ψηφίο.

Ο κώδικας Gray για τέσσερα δυαδικά ψηφία, παρουσιάζεται στον πίνακα που ακολουθεί.

Κώδικας Gray

Δεκαδικό ψηφίο	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

Κώδικας Gray

Ένας πρακτικός τρόπος συμπλήρωσης του παραπάνω πίνακα στηρίζεται στην ανακλαστική ιδιότητα.

Κατά τη δημιουργία της πρώτης από δεξιά στήλης ξεκινάμε με ένα μηδενικό και μία μονάδα.

Κατά τη δημιουργία της δεύτερης στήλης ξεκινάμε με δύο μηδενικά και δύο μονάδες.

Κατά τη δημιουργία της τρίτης στήλης με τέσσερα μηδενικά και τέσσερις μονάδες κ.ο.κ.

Δημιουργώντας, λοιπόν, τον κώδικα ανά στήλη, στην πρώτη στήλη τοποθετούμε τα υπόλοιπα ψηφία ανά 2 σε αντίστροφη σειρά, στη δεύτερη στήλη ανά 4 σε αντίστροφη σειρά, στην τρίτη στήλη ανά 8 σε αντίστροφη σειρά κ.ο.κ.

Κώδικας Gray

Ο κώδικας Gray χρησιμοποιείται σε εφαρμογές όπου η κανονική ακολουθία δυαδικών αριθμών μπορεί να οδηγήσει σε σφάλμα κατά τη μετάβαση μεταξύ δύο διαδοχικών αριθμών.

Ας θεωρήσουμε, για παράδειγμα, τη μετάβαση από τον αριθμό 0111 (7_{10}) στον επόμενο 1000 (8_{10}) μιας κανονικής δυαδικής ακολουθίας, κατά την οποία, εάν η αλλαγή τιμής του λιγότερο σημαντικού ψηφίου γίνει καθυστερημένα, θα προκύψει προσωρινά η λανθασμένη τιμή 1001 (9_{10}).

Με χρήση της ακολουθίας του κώδικα Gray αυτό αντιμετωπίζεται, αφού αλλάζει η τιμή μόνο ενός ψηφίου κατά τη μετάβαση μεταξύ διαδοχικών αριθμών.

Αλφαριθμητικός κώδικας ASCII

Με δεδομένο ότι, η επεξεργασία οποιαδήποτε πληροφορίας στα ψηφιακά συστήματα πραγματοποιείται αποκλειστικά με τη χρήση των δυαδικών ψηφίων 0 και 1, προέκυψε η ανάγκη κωδικοποίησης χαρακτήρων, όπως γράμματα και σύμβολα, αλλά και εντολών για την εκτέλεση συγκεκριμένων ενεργειών, όπως, για παράδειγμα, την αλλαγή γραμμής ή παραγράφου σε ένα κείμενο.

Η πλέον χρησιμοποιούμενη κωδικοποίηση είναι η ASCII (American Standard Code for Information Interchange), ένας δυαδικός κώδικας αλφαριθμητικών χαρακτήρων που χρησιμοποιεί 7 δυαδικά ψηφία για την κωδικοποίηση 128 χαρακτήρων, 94 εκτυπώσιμων (26 κεφαλαία και 26 μικρά λατινικά γράμματα, τα 10 δεκαδικά ψηφία και 32 ειδικά σύμβολα), καθώς και 34 μη εκτυπώσιμων (για πράξεις ελέγχου).

Αλφαριθμητικός κώδικας ASCII

Παραδείγματα κωδικοποιήσεων ASCII:

- η φράση yes!: 1111001 1100101 1110011 0100001
- το δεκαδικό ψηφίο 7 σαν χαρακτήρας (όχι ο αριθμός): 0110111
- η αλλαγή παραγράφου (Enter/CR – carriage return): 0001101
- η διαγραφή ενός χαρακτήρα (DEL – delete): 1111111

Στον πίνακα που ακολουθεί παρουσιάζεται η κωδικοποίηση ASCII για τους 94 εκτυπώσιμους χαρακτήρες και τους χαρακτήρες ελέγχου.

Αλφαριθμητικός κώδικας ASCII

$b_4 b_3 b_2 b_1$	$b_7 b_6 b_5$ 000	$b_7 b_6 b_5$ 001	$b_7 b_6 b_5$ 010	$b_7 b_6 b_5$ 011	$b_7 b_6 b_5$ 100	$b_7 b_6 b_5$ 101	$b_7 b_6 b_5$ 110	$b_7 b_6 b_5$ 111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	g	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	–	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Αλφαριθμητικός κώδικας ASCII

Χαρακτήρες ελέγχου

NUL	Null	DLE	Data link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronize
BEL	Bell	ETB	End transmitted block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
SP	Space	DEL	Delete or rubout

Αλφαριθμητικός κώδικας ASCII

Ο Ελληνικός Οργανισμός Τυποποίησης (ΕΛΟΤ) έχει αναπτύξει τον κώδικα ΕΛΟΤ-928 (επέκταση του ASCII) που χρησιμοποιεί 8 δυαδικά ψηφία και παρέχει ενιαία κωδικοποίηση λατινικών και ελληνικών χαρακτήρων.

Έως σήμερα έχουν γίνει πολλές προσπάθειες ανάπτυξης κωδίκων αλφαριθμητικών χαρακτήρων, με σημαντικότερη αυτήν της κοινοπραξίας Unicode, η οποία αναπτύσσει το ομώνυμο πρότυπο, για την κωδικοποίηση των χαρακτήρων όλων των υπαρχόντων συστημάτων γραφής, χρησιμοποιώντας έως και 32 δυαδικά ψηφία ανά χαρακτήρα.

Κώδικες ανίχνευσης σφαλμάτων

Κατά τη μετάδοση ή την επεξεργασία ψηφιακής πληροφορίας είναι πιθανό, λόγω ηλεκτρονικού θορύβου, δηλαδή ανεπιθύμητων διακυμάνσεων των ψηφιακών σημάτων, να συμβεί αλλοίωση της τιμής ενός ή περισσότερων δυαδικών ψηφίων.

Σφάλματα τέτοιου είδους είναι δυνατό να εντοπισθούν και να διορθωθούν χρησιμοποιώντας κώδικες ανίχνευσης ή κώδικες ανίχνευσης και διόρθωσης σφαλμάτων.

Ένας τρόπος ανίχνευσης των σφαλμάτων είναι η προσθήκη ενός επιπλέον ψηφίου (ψηφίο ισοτιμίας, parity bit) σε κάθε κωδικοποιημένο χαρακτήρα ή αριθμό, έτσι ώστε το πλήθος των ψηφίων 1 που περιέχονται σε αυτόν να είναι περιττό ή άρτιο, δηλαδή να δημιουργείται κώδικας με περιττή ή άρτια ισοτιμία, αντίστοιχα.

Κώδικες ανίχνευσης σφαλμάτων

Στα συστήματα που χρησιμοποιούν κωδικοποίηση ASCII (στην οποία, όπως είδαμε, χρησιμοποιούνται 7 δυαδικά ψηφία για την αναπαράσταση 128 χαρακτήρων), αυτό γίνεται με την προσθήκη ενός επιπλέον ψηφίου στην πιο σημαντική θέση κάθε κωδικοποιημένου χαρακτήρα.

Αυτό το όγδοο ψηφίο αναφέρεται ως ψηφίο ισοτιμίας (parity bit).

Ο έλεγχος ισοτιμίας (parity check) είναι ένας τρόπος ανίχνευσης σφαλμάτων.

Το ψηφίο ισοτιμίας μπορεί να πάρει τιμή 1 ή 0, και τοποθετείται στη δυαδική ακολουθία έτσι ώστε, το πλήθος των 1 στα 8 συνολικά ψηφία της ακολουθίας να γίνεται είτε άρτιο, οπότε γίνεται λόγος για άρτια ισοτιμία (even parity), είτε περιττό, οπότε προκύπτει περιττή ισοτιμία (odd parity).

Κώδικες ανίχνευσης σφαλμάτων

Παράδειγμα: Η λέξη ΤΟ (με λατινικά κεφαλαία γράμματα) συνίσταται από τους χαρακτήρες Τ και Ο των οποίων η κωδικοποίηση σε κώδικα ASCII είναι 1010100 και 1001111, αντίστοιχα.

Στην πιο σημαντική θέση προσθέτουμε ένα ψηφίο ισοτιμίας (1), για να υποδηλώσουμε άρτια ισοτιμία (το πλήθος των 1 είναι άρτιος αριθμός), οπότε θα έχουμε **1**1010100 και **1**1001111.

Στη συνέχεια, οι κωδικοποιημένοι χαρακτήρες 8 ψηφίων μεταδίδονται και η ισοτιμία τους ελέγχεται από το δέκτη. Εάν η ισοτιμία των χαρακτήρων που ελήφθησαν δεν είναι άρτια, αυτό σημαίνει ότι κατά τη διάρκεια της μετάδοσης έχει αλλάξει η τιμή (σφάλμα) τουλάχιστον ενός ψηφίου.

Ωστόσο, όταν έχουμε άρτιο πλήθος σφαλμάτων (π.χ. 2 σφάλματα), δεν εξασφαλίζεται η ανίχνευσή τους, οπότε απαιτούνται άλλου είδους κώδικες ανίχνευσης σφαλμάτων.

Κώδικες ανίχνευσης και διόρθωσης σφαλμάτων

Όλοι οι κώδικες διόρθωσης / ανίχνευσης σφαλμάτων προσθέτουν πλεονασματική πληροφορία στα δεδομένα που αποστέλλονται.

Πληρέστερος κώδικας ανίχνευσης σφαλμάτων είναι ο κώδικας Hamming, ο οποίος εκτός από τη δυνατότητα ανίχνευσης της ύπαρξης σφαλμάτων, έχει τη δυνατότητα προσδιορισμού της θέσης των σφαλμάτων σε έναν χαρακτήρα, έτσι ώστε να μπορούν να ανακτηθούν τα αρχικά δεδομένα.

Με τον κώδικα αυτό, δημιουργούνται κωδικοποιημένοι χαρακτήρες όπου τα ψηφία ισοτιμίας συνδυάζονται με επιλεγμένες ομάδες ψηφίων των αρχικών χαρακτήρων.

Κατά τη λήψη κάθε χαρακτήρα, ανιχνεύονται τυχόν σφάλματα μέσω ελέγχου ισοτιμίας και σχηματίζεται ένας δυαδικός αριθμός με ψηφία ελέγχου που δηλώνει τη θέση του ψηφίου του οποίου η τιμή έχει αλλάξει, ώστε αυτό να μπορεί να διορθωθεί.

3. Δυαδική λογική, λογικές πύλες και άλγεβρα Boole, λογικές συναρτήσεις και λογικά κυκλώματα

Τμήμα Ηλεκτρολόγων Μηχανικών
& Μηχανικών Υπολογιστών
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Μαθηματική λογική

Στη μαθηματική λογική χρησιμοποιούμε τον όρο λογική πρόταση, δηλαδή μια έκφραση με αυτοτελές νόημα, που επιδέχεται έναν μόνο χαρακτηρισμό, αληθής (true) ή ψευδής (false), αποκλείοντας κάθε άλλη περίπτωση.

Οι χαρακτηρισμοί αληθής (true) και ψευδής (false) αναφέρονται ως τιμές αλήθειας ή λογικές τιμές της λογικής πρότασης.

Για παράδειγμα, η πρόταση «ο αριθμός 3 είναι περιττός» λαμβάνει τιμή «αληθής», ενώ η πρόταση «ο αριθμός 6 είναι περιττός» λαμβάνει τιμή «ψευδής».

Επειδή οι απλές λογικές προτάσεις δεν αρκούν πάντα για να εκφράσουμε αυτό που θέλουμε, μπορούμε να συνδέσουμε μεταξύ τους δύο ή περισσότερες προτάσεις, χρησιμοποιώντας λογικούς συνδέσμους, δημιουργώντας έτσι σύνθετες προτάσεις.

Μαθηματική λογική

Βασικοί λογικοί σύνδεσμοι είναι οι εκφράσεις: και, είτε, ή, όχι (δεν).

Οι διάφοροι τρόποι (συνδυασμοί) με τους οποίους μπορούμε να συνδέσουμε απλές λογικές προτάσεις για να δημιουργήσουμε μια σύνθετη πρόταση, αποτελούν τις λογικές πράξεις μεταξύ των λογικών προτάσεων.

Η λογική τιμή μιας σύνθετης πρότασης καθορίζεται από τις τιμές των απλών προτάσεων που την αποτελούν και, φυσικά, και από τον τρόπο που συνδυάζονται αυτές για να σχηματίσουν τη σύνθετη πρόταση.

Στη μαθηματική λογική, οι λογικές τιμές των σύνθετων προτάσεων συνήθως προσδιορίζονται μέσω του πίνακα λογικών τιμών ή πίνακα αλήθειας, ο οποίος περιλαμβάνει όλους τους δυνατούς συνδυασμούς των λογικών τιμών των προτάσεων που συνιστούν τη σύνθετη πρόταση, καθώς και τις λογικές τιμές που λαμβάνει η σύνθετη πρόταση για κάθε συνδυασμό.

Μαθηματική λογική και λογικές πράξεις

Σύζευξη δύο λογικών προτάσεων, έστω p και q, αποκαλούμε τη λογική πρόταση «p και q» ή «p ∧ q», η οποία είναι αληθής μόνο στην περίπτωση που και οι δύο προτάσεις είναι αληθείς και ψευδής σε κάθε άλλη περίπτωση.

Αυτό σημαίνει ότι για να είναι ψευδής η σύζευξη δύο, ή και περισσοτέρων, προτάσεων αρκεί να είναι ψευδής μία μόνο από τις λογικές προτάσεις, χωρίς να μας ενδιαφέρει η λογική τιμή των υπόλοιπων προτάσεων.

Πίνακας αλήθειας της λογικής πράξης σύζευξης (και)		
p	**q**	**p Λ q**
False	False	False
False	True	False
True	False	False
True	True	True

Μαθηματική λογική και λογικές πράξεις

Διάζευξη δύο λογικών προτάσεων, έστω p και q, αποκαλούμε τη λογική πρόταση «p είτε q» ή «p V q», η οποία είναι ψευδής μόνο στην περίπτωση που και οι δύο προτάσεις είναι ψευδείς και αληθής σε κάθε άλλη περίπτωση.

Αυτό σημαίνει ότι για να είναι αληθής η διάζευξη δύο ή περισσότερων προτάσεων αρκεί να είναι αληθής μία μόνο από τις λογικές προτάσεις, χωρίς να μας ενδιαφέρει η λογική τιμή των υπόλοιπων προτάσεων

Πίνακας αλήθειας της λογικής πράξης διάζευξης (είτε)		
p	**q**	**p V q**
False	False	False
False	True	True
True	False	True
True	True	True

Μαθηματική λογική και λογικές πράξεις

Αποκλειστική διάζευξη δύο λογικών προτάσεων, έστω p και q, αποκαλούμε τη λογική πρόταση «p ή q» ή «p $\underline{V}$ q», η οποία είναι ψευδής στην περίπτωση που οι δυο προτάσεις έχουν την ίδια τιμή αληθείας και αληθής στην περίπτωση που οι δύο προτάσεις έχουν διαφορετικές τιμές αληθείας.

Αυτό σημαίνει ότι για να είναι αληθής η αποκλειστική διάζευξη δύο προτάσεων πρέπει να είναι αληθής μόνο μία από τις λογικές προτάσεις.

Πίνακας αλήθειας της λογικής πράξης αποκλειστικής διάζευξης (ή)		
p	**q**	**p $\underline{V}$ q**
False	False	False
False	True	True
True	False	True
True	True	False

Μαθηματική λογική και λογικές πράξεις

Άρνηση μιας λογικής πρότασης p αποκαλούμε την πρόταση «όχι p» ή «p’», η οποία είναι αληθής στην περίπτωση που η p είναι ψευδής και ψευδής στην περίπτωση που η p είναι αληθής.

Οι τιμές αληθείας των λογικών προτάσεων p και p’ είναι πάντα αντίθετες. Η άρνηση διαφέρει από τις άλλες λογικές πράξεις στο ότι είναι μια μονομελής πράξη.

Πίνακας αλήθειας της λογικής πράξης άρνησης (όχι)	
p	**p’**
False	True
True	False

Δυαδική λογική

Η δυαδική λογική διέπεται από τη μαθηματική λογική και αποτελεί μια περίπτωση εφαρμογής της.

Στη δυαδική λογική συμμετέχουν οι δυαδικές μεταβλητές, δηλαδή μεταβλητές που μπορούν να πάρουν δύο μόνο διακριτές τιμές, καθώς και οι λογικές πράξεις.

Στα ψηφιακά συστήματα οι δύο δυνατές τιμές των δυαδικών μεταβλητών είναι «λογικό 1» και «λογικό 0» ή απλά 1 και 0, οι οποίες είναι αντίστοιχες των λογικών τιμών «αληθής» και «ψευδής».

Οι λογικές πράξεις που χρησιμοποιούμε στη δυαδική λογική περιλαμβάνουν τη σύζευξη (και, AND), τη διάζευξη (είτε, OR), την αποκλειστική διάζευξη (ή, Exclusive OR ή XOR) και την άρνηση (όχι, NOT).

Δίτιμη άλγεβρα Boole

Η δίτιμη άλγεβρα Boole πραγματεύεται δυαδικές μεταβλητές και λογικές πράξεις, οι οποίες υλοποιούνται με ηλεκτρονικά κυκλώματα που ονομάζονται λογικές πύλες, τα κύρια στοιχεία των οποίων είναι τα τρανζίστορ.

Κάθε δυαδική μεταβλητή μπορεί να πάρει μία από δύο μόνο διαφορετικές τιμές, 0 ή 1.

Οι βασικές λογικές πράξεις της δίτιμης άλγεβρας Boole είναι οι πράξεις AND, OR και NOT, οι οποίες είναι αντίστοιχες με τις λογικές πράξεις που προαναφέρθηκαν.

Η δίτιμη άλγεβρα Boole αναφέρεται και ως άλγεβρα των διακοπτών, αφού υπάρχει αντιστοιχία των βασικών λογικών πράξεων με απλά ηλεκτρικά κυκλώματα διακοπτών.

Δίτιμη άλγεβρα Boole και κυκλώματα διακοπτών

Το απλούστερο δυαδικό στοιχείο είναι ένας διακόπτης s, που έχει 2 καταστάσεις, ανοιχτός (OFF) και κλειστός (ON).

Μπορούμε να θέσουμε s = 0 για την κατάσταση OFF και s = 1 για την κατάσταση ON.

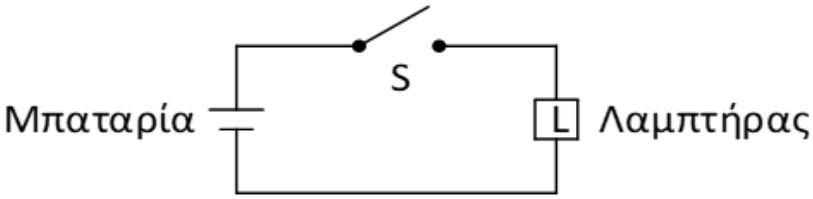
Επομένως, μια δυαδική μεταβλητή μπορεί να αντιστοιχιστεί με έναν διακόπτη δύο καταστάσεων.

Δίτιμη άλγεβρα Boole και κυκλώματα διακοπτών

Το παρακάτω κύκλωμα περιλαμβάνει μια μπαταρία, έναν διακόπτη s και ένα λαμπτήρα L.

Για να ανάψει ο λαμπτήρας (L = 1), πρέπει να διέλθει από αυτόν ηλεκτρικό ρεύμα και για να συμβεί αυτό πρέπει ο διακόπτης να είναι κλειστός (s = 1).

Εάν ο διακόπτης είναι ανοικτός (s = 0), ο λαμπτήρας δεν ανάβει (L = 0).



Δίτιμη άλγεβρα Boole και κυκλώματα διακοπτών

Η εξάρτηση της λειτουργίας του λαμπτήρα L από την κατάσταση του διακόπτη s περιγράφεται μέσω μιας λογικής έκφρασης: L(s) = s

Αυτή η λογική έκφραση αναφέρεται ως λογική συνάρτηση και περιγράφει την κατάσταση του λαμπτήρα L (έξοδος ή μεταβλητή εξόδου) ως συνάρτηση της κατάστασης του διακόπτη s (είσοδος ή μεταβλητή εισόδου).

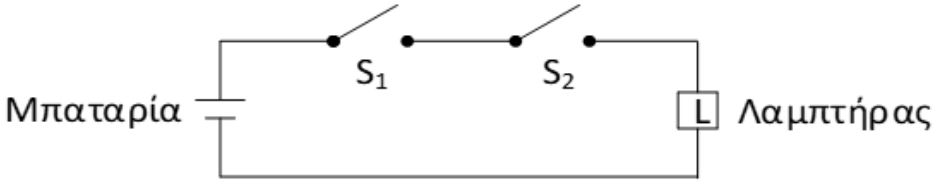
Η λογική συνάρτηση περιγράφεται από έναν πίνακα αλήθειας:

s	L(s)
0	0
1	1

Δίτιμη άλγεβρα Boole και κυκλώματα διακοπτών

Κατά τη σειριακή σύνδεση δύο διακοπτών, ο λαμπτήρας ανάβει ($L = 1$) μόνο όταν και οι δύο διακόπτες είναι κλειστοί, δηλαδή όταν: $s_1 = 1$ ΚΑΙ $s_2 = 1$.

Εάν τουλάχιστον ένας διακόπτης είναι ανοικτός ($s_i = 0$), ο λαμπτήρας δεν ανάβει ($L = 0$).



Η συμπεριφορά αυτή εκφράζεται με τη λογική συνάρτηση:

$$L(s_1, s_2) = s_1 \textbf{ AND } s_2 = s_1 \cdot s_2$$

Δίτιμη άλγεβρα Boole και κυκλώματα διακοπτών

Το σύμβολο « $\cdot$ » ονομάζεται τελεστής της λογικής πράξης AND (ή τελεστής λογικού γινόμενου) και το αντίστοιχο κύκλωμα διακοπτών υλοποιεί τη λογική πράξη AND.

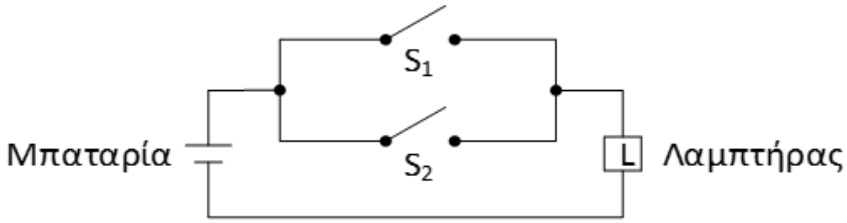
Ο πίνακας αλήθειας της λογικής πράξης AND είναι:

Πίνακας αλήθειας λογικής πράξης AND		
s_1	s_2	$s_1 \cdot s_2$
0	0	0
0	1	0
1	0	0
1	1	1

Δίτιμη άλγεβρα Boole και κυκλώματα διακοπτών

Κατά την παράλληλη σύνδεση των δύο διακοπτών, ο λαμπτήρας ανάβει (L = 1), εάν είναι κλειστός είτε ο διακόπτης s_1, είτε ο διακόπτης s_2, είτε και οι δύο διακόπτες ($s_i = 1$), δηλαδή όταν $s_1 = 1$ ΕΙΤΕ $s_2 = 1$.

Εάν και οι δύο διακόπτες είναι ανοικτοί ($s_i = 0$), ο λαμπτήρας δεν ανάβει (L = 0).



Η συμπεριφορά αυτή εκφράζεται με τη λογική συνάρτηση:

$$L(s_1, s_2) = s_1 \textbf{ OR } s_2 = s_1 + s_2$$

Δίτιμη άλγεβρα Boole και κυκλώματα διακοπτών

Το σύμβολο «+» ονομάζεται τελεστής της λογικής πράξης OR (ή τελεστής του λογικού αθροίσματος) και το αντίστοιχο κύκλωμα διακοπτών υλοποιεί τη λογική πράξη OR.

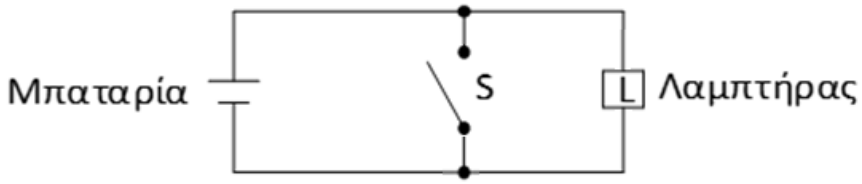
Ο πίνακας αλήθειας της λογικής πράξης OR είναι:

Πίνακας αλήθειας λογικής πράξης OR		
s_1	s_2	$s_1 + s_2$
0	0	0
0	1	1
1	0	1
1	1	1

Δίτιμη άλγεβρα Boole και κυκλώματα διακοπτών

Με σύνδεση ενός διακόπτη παράλληλα με τον λαμπτήρα, ο λαμπτήρας δεν ανάβει (L = 0) όταν ο διακόπτης είναι κλειστός (s = 1), αφού τότε το ρεύμα θα διέλθει εξ' ολοκλήρου από τον διακόπτη (που δεν έχει αντίσταση) και όχι από τον λαμπτήρα (που έχει αντίσταση και βραχυκυκλώνεται από τον κλειστό διακόπτη).

Αντιθέτως, όταν ο διακόπτης είναι ανοικτός (s = 0), το ρεύμα διέρχεται από τον λαμπτήρα και αυτός ανάβει (L = 1).



Η συμπεριφορά αυτή εκφράζεται με τη λογική συνάρτηση:

L(s) = **NOT(**s) = s'

Δίτιμη άλγεβρα Boole και κυκλώματα διακοπτών

Το σύμβολο « ' » ονομάζεται τελεστής της λογικής πράξης NOT (ή τελεστής της λογικής άρνησης ή τελεστής αντιστροφής ή συμπλήρωμα).

Ο πίνακας αλήθειας της λογικής πράξης NOT είναι:

Πίνακας αλήθειας λογικής πράξης NOT	
s	s'
0	1
1	0

Επέκταση βασικών λογικών πράξεων

Οι λογικές πράξεις AND και OR μπορούν να επεκταθούν για n μεταβλητές.

Ειδικότερα, η πράξη AND μεταξύ των μεταβλητών $x_1, x_2, \ldots, x_n$ έχει ως αποτέλεσμα τη λογική τιμή 1 μόνο όταν όλες οι μεταβλητές λαμβάνουν λογική τιμή 1.

Παρομοίως, η πράξη OR μεταξύ των μεταβλητών $x_1, x_2, \ldots, x_n$ έχει ως αποτέλεσμα τη λογική τιμή 1, όταν τουλάχιστον μία από τις μεταβλητές λαμβάνει τιμή 1.

x_1	x_2	x_3	$x_1 \cdot x_2 \cdot x_3$	$x_1 + x_2 + x_3$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Δίτιμη άλγεβρα Boole και λογικές συναρτήσεις

Οι λογικές συναρτήσεις περιγράφονται συνήθως από αλγεβρικές εκφράσεις που περιλαμβάνουν δυαδικές μεταβλητές, τις σταθερές τιμές 0 και 1, τους τελεστές των τριών βασικών λογικών πράξεων, παρενθέσεις και αγκύλες.

Οι αλγεβρικές εκφράσεις είναι σύνθεση λογικών πράξεων μεταξύ των δυαδικών μεταβλητών και των σταθερών τιμών.

Κάθε λογική συνάρτηση λαμβάνει τιμή 0 ή 1, ανάλογα με τις λογικές τιμές των δυαδικών μεταβλητών που συμμετέχουν σε αυτήν.

Οι πιθανές τιμές που μπορεί να λάβει μια λογική συνάρτηση προσδιορίζονται με τον υπολογισμό της τιμής της αλγεβρικής έκφρασης που την περιγράφει, για όλους τους δυνατούς συνδυασμούς τιμών των ανεξάρτητων μεταβλητών και αποτυπώνονται στον πίνακα αλήθειας της λογικής συνάρτησης.

Δίτιμη άλγεβρα Boole και λογικές συναρτήσεις

Εκτός, λοιπόν από την περιγραφή μιας λογικής συνάρτησης μέσω αλγεβρικής έκφρασης, μια λογική συνάρτηση περιγράφεται μέσω του πίνακα αλήθειας που περιλαμβάνει όλους τους δυνατούς συνδυασμούς τιμών των μεταβλητών που συμμετέχουν στη συνάρτηση, καθώς και την τιμή της συνάρτησης αυτής για κάθε συνδυασμό.

Το πλήθος των δυνατών συνδυασμών n μεταβλητών που συμμετέχουν σε μία συνάρτηση είναι 2^n και προκύπτουν εύκολα, εάν γράψουμε κατά σειρά τους δυαδικούς αριθμούς από 0 έως 2^{n-1} και αντιστοιχίσουμε κάθε δυαδικό ψηφίο σε μία από τις μεταβλητές.

Δίτιμη άλγεβρα Boole και λογικές συναρτήσεις

Παράδειγμα: Η λογική συνάρτηση F(x,y,z) που περιγράφεται από την αλγεβρική έκφραση F(x,y,z) = $x' \cdot y + x \cdot z$, περιλαμβάνει 3 δυαδικές μεταβλητές και λαμβάνει λογική τιμή 1, όταν η μεταβλητή x και η μεταβλητή y έχουν τιμή 0 και 1, αντίστοιχα, ή όταν οι μεταβλητές x και z έχουν τιμή 1, ενώ λαμβάνει λογική τιμή 0 για τους υπόλοιπους δυνατούς συνδυασμούς των μεταβλητών που συμμετέχουν σε αυτήν.

x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Ο πίνακας αλήθειας της F(x,y,z) περιλαμβάνει 8 (2^3) γραμμές και 4 στήλες (μία στήλη για κάθε μεταβλητή και μία στήλη για τη συνάρτηση).

Μειονέκτημα της περιγραφής μιας λογικής συνάρτησης μέσω πίνακα αλήθειας είναι ότι το μέγεθος του πίνακα αυξάνεται εκθετικά με το πλήθος των μεταβλητών (π.χ. 16 μεταβλητές, 2^{16} γραμμές).

Δίτιμη άλγεβρα Boole και λογικές συναρτήσεις

Η λογική πράξη NOT εκτελείται σε μία μεταβλητή, αλλά μπορεί να εφαρμοστεί και σε μία λογική συνάρτηση.

Για παράδειγμα, εάν εφαρμοστεί η λογική πράξη NOT στη λογική συνάρτηση $F(x,y,z) = x' \cdot y + x \cdot z$, δηλαδή $F'(x,y,z) = (x' \cdot y + x \cdot z)'$, η συνάρτηση $F'(x,y,z)$ που προκύπτει αναφέρεται ως συμπληρωματική συνάρτηση της $F(x,y,z)$.

Μια λογική συνάρτηση F' ονομάζεται συμπληρωματική συνάρτηση μιας συνάρτησης F, όταν λαμβάνει λογική τιμή 1 ή 0, για τους συνδυασμούς τιμών των μεταβλητών για τους οποίους η συνάρτηση F λαμβάνει τιμή 0 ή 1, αντίστοιχα.

Ο τιμές στον πίνακα αλήθειας της συμπληρωματικής συνάρτησης F', προκύπτουν εύκολα από εκείνες της συνάρτησης F για κάθε συνδυασμό τιμών των μεταβλητών.

Λογικές πύλες

Κάθε λογική πράξη υλοποιείται σε επίπεδο ηλεκτρονικού κυκλώματος με τρανζίστορ, οπότε προκύπτει ένα στοιχείο κυκλώματος που ονομάζεται λογική πύλη (logic gate).

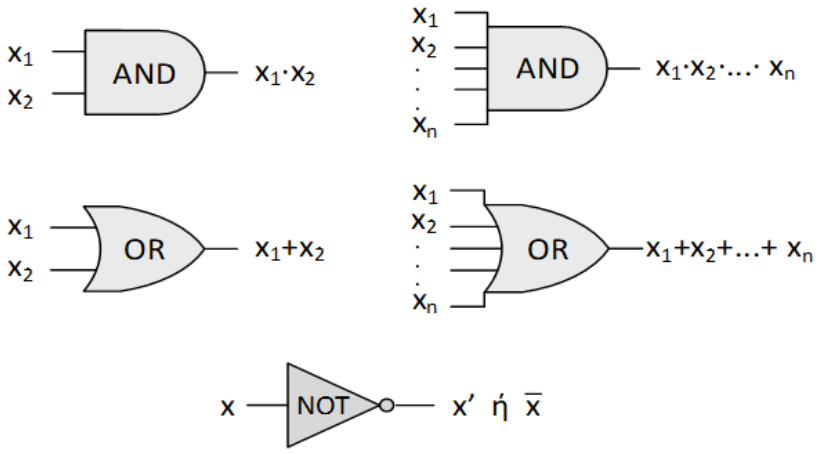
Μια λογική πύλη μπορεί να έχει μία ή περισσότερες εισόδους και μόνο μια έξοδο, που είναι συνάρτηση των εισόδων της.

Οι είσοδοι αντιστοιχούν στις μεταβλητές της λογικής πράξης, ενώ η τιμή της εξόδου προκύπτει από τη λογική πράξη που επιτελεί η πύλη μεταξύ των τρεχουσών τιμών των μεταβλητών.

Οι λογικές πύλες ανταποκρίνονται στη χαμηλή και την υψηλή στάθμη των δυαδικών σημάτων, οι οποίες εκφράζονται με τις λογικές τιμές 0 και 1, αντίστοιχα.

Λογικές πύλες

Τα σύμβολα των λογικών πυλών AND και OR με δύο και περισσότερες εισόδους, καθώς και το σύμβολο της λογικής πύλης NOT (που αναφέρεται και ως αντιστροφέας) είναι τα παρακάτω:



Υλοποίηση λογικών πυλών

Για την υλοποίηση λογικών πυλών χρησιμοποιούνται τρανζίστορ, τα οποία είναι ηλεκτρονικά στοιχεία που λειτουργούν ως διακόπτες, με δύο επιτρεπτές καταστάσεις λειτουργίας αντίστοιχες με τις λογικές τιμές 0, 1.

Η αποδοτικότερη τεχνολογία υλοποίησης βασίζεται στη χρήση τρανζίστορ επίδρασης πεδίου μετάλλου-οξειδίου-ημιαγωγού (metal-oxide-semiconductor field-effect transistors, MOSFETs), τα οποία διακρίνονται σε τρανζίστορ διαύλου αρνητικού φορτίου (nMOS) και τρανζίστορ διαύλου θετικού φορτίου (pMOS), ανάλογα με την πολικότητα των φορέων φορτίου που συμμετέχουν στη λειτουργία τους.

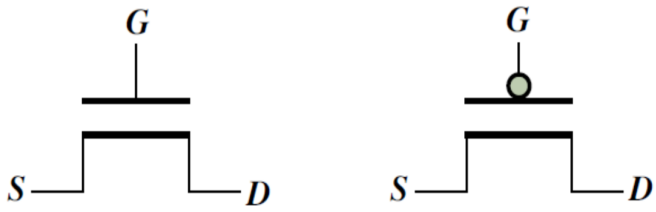
Η συμπληρωματική τεχνολογία μετάλλου-οξειδίου-ημιαγωγού (complementary metal-oxide-semiconductor, CMOS) βασίζεται στη συνδυασμένη χρήση των δύο τύπων MOSFETs.

Σε σχέση με άλλες τεχνολογίες υλοποίησης, η τεχνολογία CMOS παρουσιάζει μεγαλύτερη δυνατότητα ενσωμάτωσης μεγάλου αριθμού πυλών σε ένα κύκλωμα και χαμηλή κατανάλωση ενέργειας.

Υλοποίηση λογικών πυλών

Το τρανζίστορ nMOS συμπεριφέρεται ως κλειστός διακόπτης, όταν στον ακροδέκτη της πύλης του (G) εφαρμόζεται υψηλή στάθμη τάσης (λογική τιμή 1), ενώ όταν εφαρμόζεται χαμηλή στάθμη τάσης (λογική τιμή 0) συμπεριφέρεται ως ανοιχτός διακόπτης.

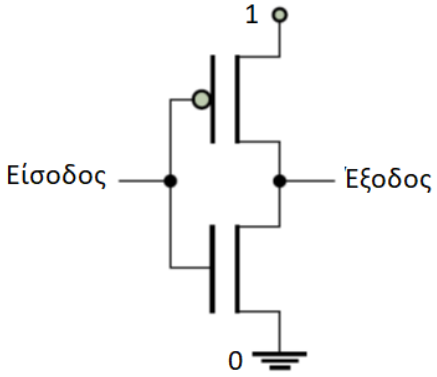
Το τρανζίστορ pMOS λειτουργεί αντίθετα.



nMOS pMOS

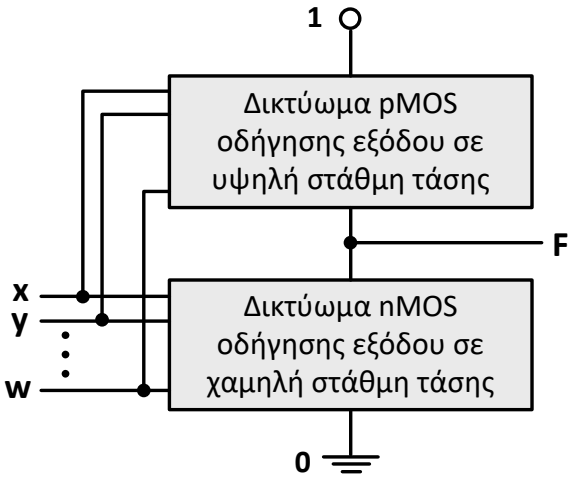
Υλοποίηση λογικών πυλών

Ο αντιστροφέας (πύλη NOT) υλοποιείται εύκολα με συνδυασμένη χρήση ενός τρανζίστορ nMOS και ενός τρανζίστορ pMOS.



Η βάση για την υλοποίηση άλλων λογικών πυλών είναι μια δομή που περιλαμβάνει δικτύωμα από τρανζίστορ nMOS για την οδήγηση της εξόδου του κυκλώματος σε χαμηλή στάθμη τάσης και δικτύωμα από τρανζίστορ pMOS για την οδήγηση της εξόδου σε υψηλή στάθμη τάσης.

Υλοποίηση λογικών πυλών

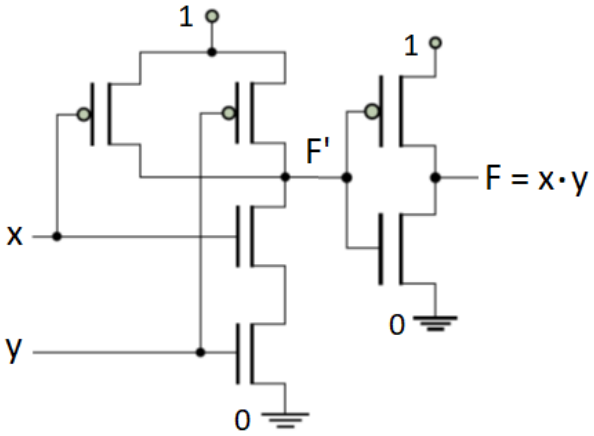


Τα δύο δικτυώματα δομούνται έτσι ώστε να δημιουργείται διαδρομή κλειστών διακοπτών στο δικτύωμα nMOS για τους συνδυασμούς τιμών των εισόδων, για τους οποίους η λογική συνάρτηση της πύλης λαμβάνει τιμή 0, ή στο δικτύωμα pMOS για τους συνδυασμούς τιμών των εισόδων, για τους οποίους η συνάρτηση λαμβάνει τιμή 1.

Υλοποίηση λογικών πυλών

Πύλη AND 2 εισόδων
τεχνολογίας CMOS

Πίνακας αλήθειας λογικής πράξης AND			
x	y	F'	F = x·y
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1



Παράγωγες λογικές πύλες

Όπως προαναφέρθηκε, η λογική πράξη NOT μπορεί να εφαρμοστεί σε μια μεταβλητή, αλλά και σε μία λογική συνάρτηση.

Επίσης, αναφέρθηκε ότι οι λογικές συναρτήσεις προκύπτουν με σύνθεση λογικών πράξεων.

Εκτός από τις 3 βασικές λογικές πύλες, υπάρχουν και άλλες πύλες, οι οποίες συντίθενται από τις βασικές και για το λόγο αυτό ονομάζονται παράγωγες λογικές πύλες και είναι οι λογικές πύλες XOR, NAND, NOR και XNOR.

Η λογική πύλη XOR (αποκλειστικού ή, Exclusive OR) υλοποιεί τη λογική πράξη της αποκλειστικής διάζευξης, σύμφωνα με την οποία, για να έχει η λογική πράξη XOR μεταξύ 2 μεταβλητών, ως αποτέλεσμα λογική τιμή 1, πρέπει μία μόνο μεταβλητή να έχει τιμή 1.

Η συμπεριφορά αυτή, εκφράζεται με τη λογική συνάρτηση:

$f(x_1, x_2) = x_1$ XOR $x_2 = x_1 \oplus x_2$, $\oplus$: τελεστής πράξης XOR

Παράγωγες λογικές πύλες

Πίνακας αλήθειας της λογικής πράξης XOR:

Πίνακας αλήθειας λογικής πράξης XOR δύο μεταβλητών		
x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

Για περισσότερες από δύο εισόδους, η έξοδος μιας πύλης XOR λαμβάνει λογική τιμή 1, όταν περιττός αριθμός εισόδων λαμβάνει τιμή 1, και λογική τιμή 0, όταν άρτιος αριθμός εισόδων λαμβάνει τιμή 1.

Παράγωγες λογικές πύλες

Η λογική πράξη XOR μπορεί να υλοποιηθεί με τη σύνθεση των λογικών πράξεων AND, OR και NOT, σύμφωνα με τη λογική έκφραση:

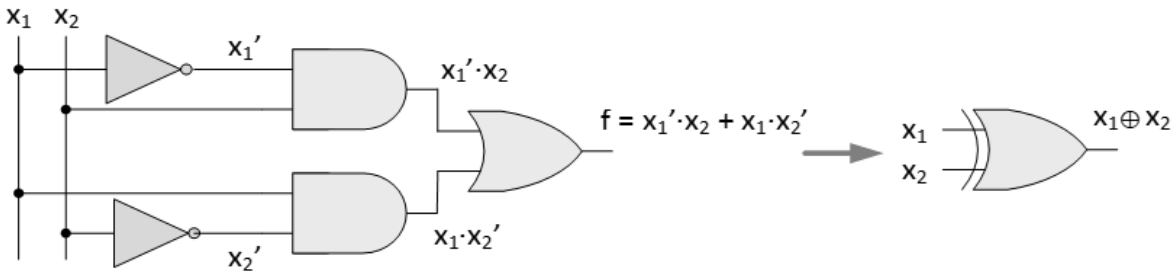
$$x_1 \oplus x_2 = x'_1 \cdot x_2 + x_1 \cdot x'_2$$

Η απόδειξη της ισχύος αυτής της λογικής σχέσης μπορεί να γίνει με τη χρήση διαδοχικών πινάκων αλήθειας.

x_1	x_2	x_1'	x_2'	$x_1' \cdot x_2$	$x_1 \cdot x_2'$	$x_1' \cdot x_2 + x_1 \cdot x_2'$	$x_1 \oplus x_2$
0	0	1	1	0	0	0	0
0	1	1	0	1	0	1	1
1	0	0	1	0	1	1	1
1	1	0	0	0	0	0	0

Παράγωγες λογικές πύλες

Λογικό κύκλωμα με βασικές λογικές πύλες, που υλοποιεί τη λογική πράξη XOR, και σύμβολο της λογικής πύλης XOR:

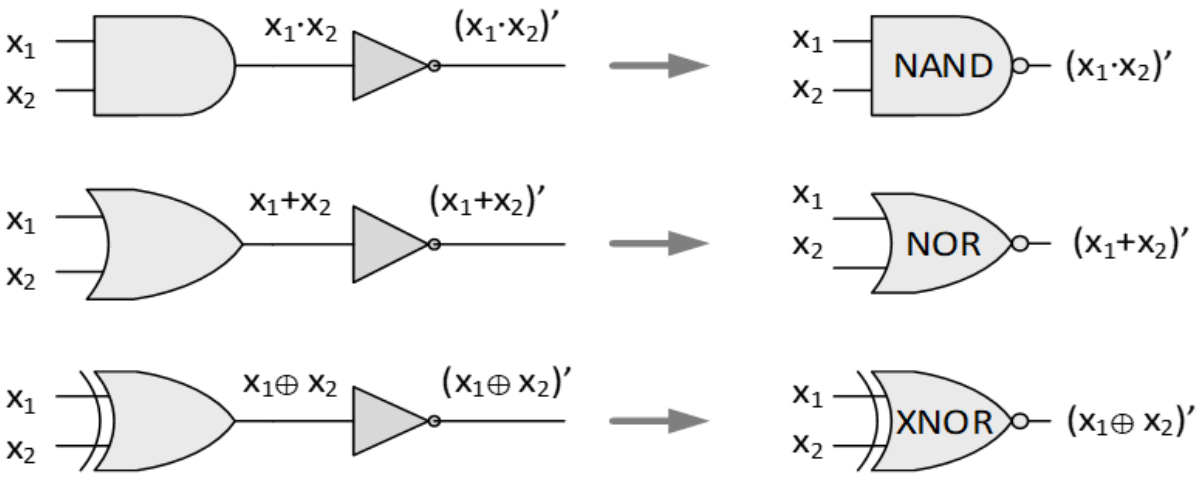


Παράγωγες λογικές πύλες

Από τον πίνακα αλήθειας της λογικής πύλης XOR 2 εισόδων, συμπεραίνουμε επίσης ότι η πύλη αυτή έχει τη δυνατότητα αναγνώρισης της διαφορετικότητας των τιμών των εισόδων, αφού η έξοδός λαμβάνει τιμή 1, μόνο όταν οι είσοδοί της λαμβάνουν διαφορετικές λογικές τιμές, ενώ λαμβάνει τιμή 0 όταν οι είσοδοί της λαμβάνουν την ίδια τιμή.

Οι άλλες τρεις παράγωγες λογικές πύλες, NAND, NOR και XNOR, προκύπτουν από τη σύνθεση των λογικών πυλών AND, OR και XOR με τη λογική πύλη NOT και, συγκεκριμένα, με τη σύνδεση μιας λογικής πύλης NOT στην έξοδο κάθε μιας λογικής πύλης AND, OR και XOR, αντίστοιχα.

Παράγωγες λογικές πύλες



Παράγωγες λογικές πύλες

Συγκεντρωτικός πίνακες αλήθειας για όλες τις λογικές πράξεις (βασικές και παράγωγες) μεταξύ δύο μεταβλητών:

Πίνακες αλήθειας δύο μεταβλητών							
x	y	x•y	(x•y)'	x+y	(x+y)'	x⊕y	(x⊕y)'
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Παράγωγες λογικές πύλες

Η πύλη NAND 2 εισόδων παράγει ως έξοδο το συμπλήρωμα της εξόδου της πύλης AND, δηλαδή, η έξοδός της λαμβάνει τιμή 1, όταν τουλάχιστον μία από τις 2 εισόδους της λαμβάνει τιμή 0, διαφορετικά η έξοδός της ισούται με 0.

Η πύλη NOR 2 εισόδων παράγει ως έξοδο το συμπλήρωμα της εξόδου της πύλης OR, δηλαδή, η έξοδος της πύλης NOR λαμβάνει τιμή 0, όταν τουλάχιστον μία από τις 2 εισόδους της λαμβάνει λογική τιμή 1, ενώ όταν και οι δύο είσοδοί της λαμβάνουν τιμή 0, η έξοδος της πύλης λαμβάνει τιμή 1.

Η πύλη XNOR 2 εισόδων παράγει ως έξοδο το συμπλήρωμα της εξόδου της πύλης XOR, δηλαδή, η έξοδος της πύλης XNOR ισούται με 1, όταν και οι δύο είσοδοί της λαμβάνουν την ίδια τιμή, ενώ όταν οι δύο είσοδοί της λαμβάνουν διαφορετική τιμή, η έξοδος της πύλης ισούται με 0.

Συμπεραίνουμε λοιπόν ότι η πύλη XNOR 2 εισόδων αναγνωρίζει την ομοιότητα των τιμών των εισόδων της και για το λόγο αυτό η λογική πράξη που επιτελεί αναφέρεται ως πράξη ισοδυναμίας.

Επέκταση λογικών πυλών

Συγκεντρωτικός πίνακες αλήθειας για όλες τις λογικές πράξεις (βασικές και παράγωγες) μεταξύ 3 μεταβλητών:

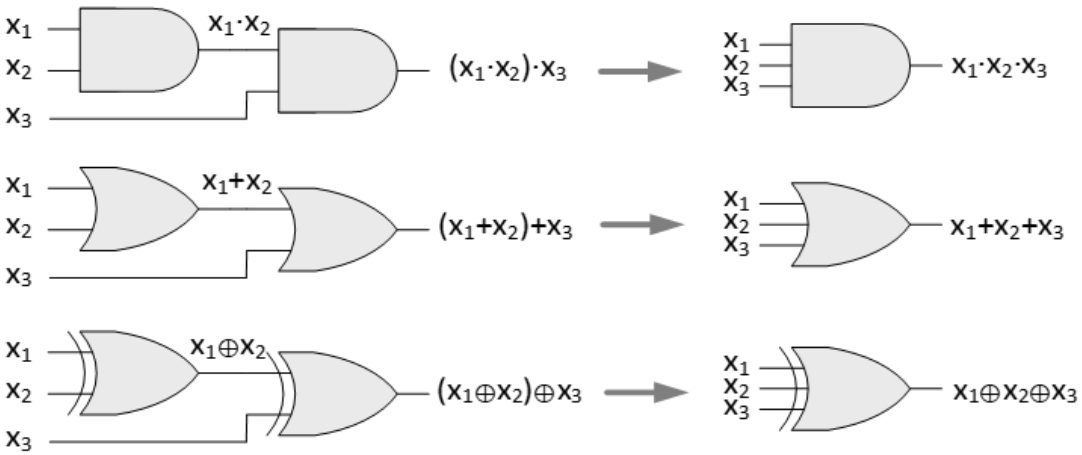
Πίνακες αλήθειας τριών μεταβλητών								
x	y	z	x·y·z	(x·y·z)'	x+y+z	(x+y+z)'	x⊕y⊕z	(x⊕y⊕z)'
0	0	0	0	1	0	1	0	1
0	0	1	0	1	1	0	1	0
0	1	0	0	1	1	0	1	0
0	1	1	0	1	1	0	0	1
1	0	0	0	1	1	0	1	0
1	0	1	0	1	1	0	0	1
1	1	0	0	1	1	0	0	1
1	1	1	1	0	1	0	1	0

XOR: Περιττή συνάρτηση

XNOR: Άρτια συνάρτηση

Επέκταση λογικών πυλών

Οι λογικές πύλες AND, OR και XOR τριών ή περισσότερων εισόδων μπορούν να υλοποιηθούν με διαδοχική σύνδεση πυλών AND, OR και XOR 2 εισόδων.



Επέκταση λογικών πυλών

Η επέκταση πυλών AND, OR και XOR 2 εισόδων σε πύλες 3 ή περισσότερων εισόδων με διαδοχική σύνδεση πυλών 2 εισόδων, είναι εφικτή διότι στις αντίστοιχες λογικές πράξεις ισχύουν η αντιμεταθετική και η προσεταιριστική ιδιότητα, όπως θα δούμε στις ιδιότητες της άλγεβρας Boole.

Αυτό όμως δεν είναι εφικτό για πύλες NAND και NOR, αφού για τις αντίστοιχες λογικές πράξεις δεν ισχύει η προσεταιριστική ιδιότητα Για παράδειγμα, οι εκφράσεις $(x \cdot y \cdot z)'$ και $[(x \cdot y)' \cdot z]'$ δεν είναι ισοδύναμες.

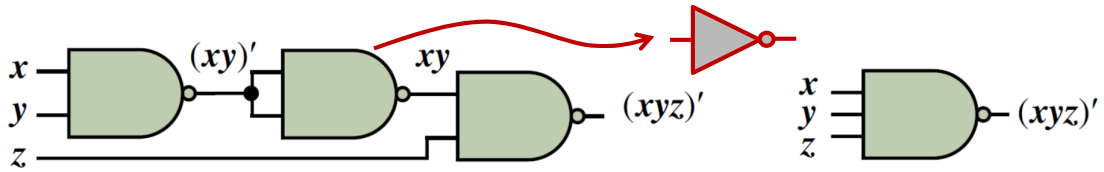
x	y	z	$(x \cdot y \cdot z)'$	$[(x \cdot y)' \cdot z]'$
0	0	0	1	1
0	0	1	1	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1

Επίσης, δεν είναι εφικτή η υλοποίηση πυλών XNOR 3 ή περισσότερων εισόδων με διαδοχική σύνδεση πυλών XNOR 2 εισόδων.

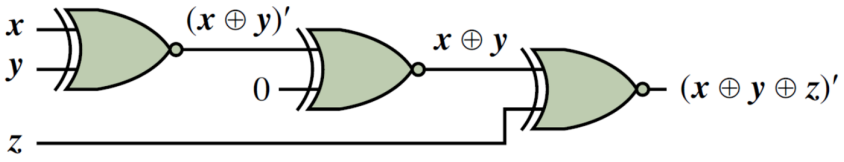
Επέκταση λογικών πυλών

Για την υλοποίηση μιας πύλης NAND 3 εισόδων, απαιτούνται 3 πύλες NAND 2 εισόδων, αφού $(x \cdot y \cdot z)' = \{ [(x \cdot y)']' \cdot z \}'$.

Όταν θέτουμε στις δύο εισόδους μιας πύλης NAND την ίδια μεταβλητή εισόδου, τότε η πύλη λειτουργεί ως αντιστροφέας, αφού $(A \cdot A)' = A'$.



Ομοίως υλοποιείται μια πύλη NOR 3 εισόδων με 3 πύλες NOR 2 εισόδων και με παρόμοιο τρόπο υλοποιούνται οι πύλες XNOR με περισσότερες από 2 εισόδους. Ισχύει ότι $(A \oplus 0)' = A'$.



Σύνθεση λογικών κυκλωμάτων με λογικές πύλες

Ένα λογικό κύκλωμα αποτελείται από γραμμές διασύνδεσης λογικών πυλών που αντιστοιχούν στις διαδρομές των δυαδικών σημάτων και από λογικές πύλες που επιτελούν τις λογικές πράξεις μεταξύ των σημάτων, οι οποίες δηλώνονται στη λογική συνάρτηση.

Για τη σύνθεση (σχεδίαση) ενός λογικού κυκλώματος, δίνεται μία λογική συνάρτηση και σχεδιάζεται το λογικό κύκλωμα που την υλοποιεί.

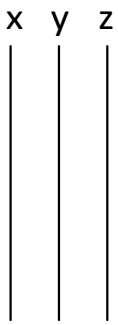
Ξεκινώντας από τις μεταβλητές της συνάρτησης που είναι οι είσοδοι του λογικού κυκλώματος, σχεδιάζουμε το λογικό κύκλωμα συνδέοντας κατάλληλα τις λογικές πύλες που απαιτούνται για την υλοποίηση των αλγεβρικών (λογικών) εκφράσεων που περιλαμβάνονται στη συνάρτηση.

Η σύνθεση γίνεται με συγκεκριμένη σειρά, ξεκινώντας από τις εκφράσεις σε παρένθεση και εφαρμόζοντας την προτεραιότητα των λογικών πράξεων, που είναι κατά σειρά NOT, AND, OR.

Σύνθεση λογικών κυκλωμάτων με λογικές πύλες

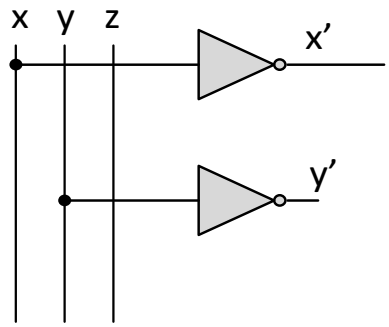
Παράδειγμα: σύνθεση λογικού κυκλώματος που υλοποιεί τη συνάρτηση:

$$f(x, y, z) = (x' + y' \cdot z) \cdot z$$



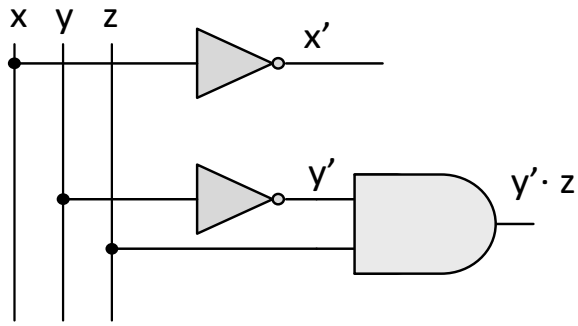
Σύνθεση λογικών κυκλωμάτων με λογικές πύλες

$$f(x, y, z) = (x' + y' \cdot z) \cdot z$$



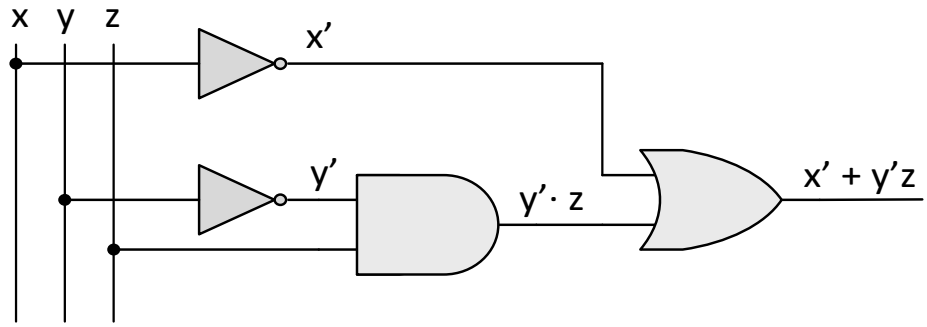
Σύνθεση λογικών κυκλωμάτων με λογικές πύλες

$$f(x, y, z) = (x' + y' \cdot z) \cdot z$$



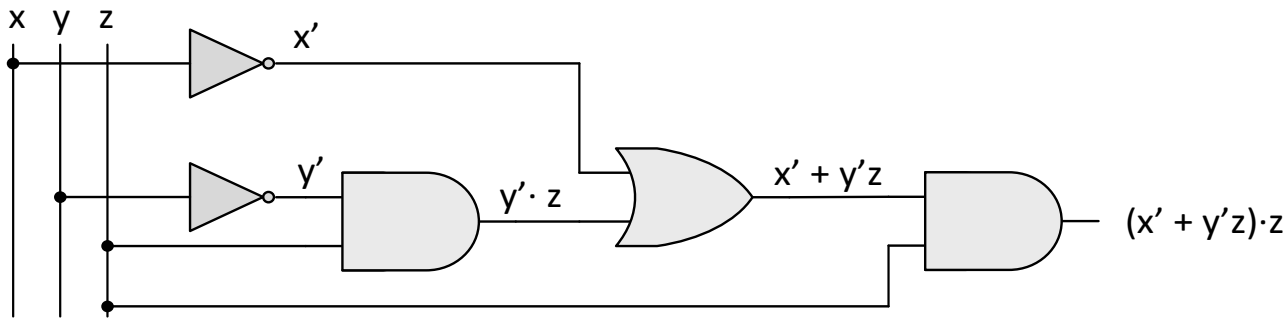
Σύνθεση λογικών κυκλωμάτων με λογικές πύλες

$f(x, y, z) = (x' + y' \cdot z) \cdot z$



Σύνθεση λογικών κυκλωμάτων με λογικές πύλες

$f(x, y, z) = (x' + y' \cdot z) \cdot z$



Άλγεβρα Boole

Για τη σχεδίαση και την ανάλυση των ψηφιακών συστημάτων είναι απαραίτητη η χρήση ενός αλγεβρικού συστήματος, οι μεταβλητές του οποίου λαμβάνουν μόνο τις λογικές τιμές 0 και 1.

Οι βάσεις για το σύστημα αυτό τέθηκαν το 1854 από τον George Boole, που ανέπτυξε ένα αλγεβρικό σύστημα για τη συστηματική αντιμετώπιση της λογικής.

Τα αξιώματα της άλγεβρας αυτής διατυπώθηκαν το 1904 από τον Edward Huntington και το 1938, ο Claude Elwood Shannon εισήγαγε τη δίτιμη άλγεβρα Boole, για να εκφράσει τις ιδιότητες ηλεκτρικών κυκλωμάτων διακοπτών με 2 καταστάσεις και μας επιτρέπει να εκφράσουμε και τη σχέση μεταξύ των εισόδων και των εξόδων ενός ψηφιακού κυκλώματος.

Η δίτιμη άλγεβρα Boole πραγματεύεται λογικές πράξεις (λογική αντιστροφή NOT, λογικό γινόμενο AND, λογικό άθροισμα OR) μεταξύ δυαδικών μεταβλητών.

Αξιώματα άλγεβρας Boole

A1. Κλειστότητα ως προς τους τρεις τελεστές, αφού σύμφωνα με τους ορισμούς των τριών λογικών πράξεων το αποτέλεσμά τους δεν μπορεί να είναι διαφορετικό από 0 ή 1, δηλαδή, εάν $x, y \in A = \{0, 1\}$, τότε $x \cdot y \in A$, $x + y \in A$ και $x', y' \in A$.

A2. Αντιμεταθετικότητα: η διάταξη των στοιχείων κατά την εκτέλεση των λογικών πράξεων AND και OR δεν επηρεάζει το αποτέλεσμα, δηλαδή, εάν $x, y \in A$, τότε $x \cdot y = y \cdot x$ και $x + y = y + x$.

A3. Επιμεριστικότητα: το λογικό γινόμενο επιμερίζεται στο λογικό άθροισμα και το λογικό άθροισμα επιμερίζεται στο λογικό γινόμενο, δηλαδή, εάν $x, y, z \in A$, τότε $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ και $x + y \cdot z = (x + y) \cdot (x + z)$.

A4. Ουδέτερα στοιχεία: για τις λογικές πράξεις AND και OR τα ουδέτερα στοιχεία είναι 1 και 0, αντίστοιχα, δηλαδή, εάν $x \in A$, τότε $x \cdot 1 = x$ και $x + 0 = x$.

Αξιώματα άλγεβρας Boole

Α5. Συμπλήρωμα: από τον ορισμό της λογικής αντιστροφής προκύπτει ότι, εάν $x \in A$, τότε $x \cdot x' = 0$ και $x + x' = 1$, όπου το στοιχείο x' αναφέρεται ως συμπλήρωμα του στοιχείου ή της μεταβλητής x. Το συμπλήρωμα συμβολίζεται και ως .

Σε κάθε ζεύγος αξιωμάτων το ένα τμήμα μπορεί να προκύψει από το άλλο, με αμοιβαία εναλλαγή των λογικών πράξεων AND, OR και των στοιχείων 0 και 1 (αρχή δυϊσμού).

Συνεπώς, κάθε αλγεβρική έκφραση η οποία μπορεί να παραχθεί από τα αξιώματα εξακολουθεί να ισχύει, εάν εναλλάξουμε τους τελεστές και τα ουδέτερα στοιχεία.

Θεωρήματα άλγεβρας Boole

Θ1. Οι λογικές πράξεις μιας μεταβλητής με τον εαυτό της έχουν αποτέλεσμα τη μεταβλητή αυτή, δηλαδή $x \cdot x = x$, $x + x = x$.

Θ2. Οι λογικές πράξεις μιας μεταβλητής με το αντίστοιχο ουδέτερο στοιχείο έχουν αποτέλεσμα το ουδέτερο στοιχείο, δηλαδή $x \cdot 0 = 0$, $x + 1 = 1$.

Θ3. Θεώρημα διπλής άρνησης: το συμπλήρωμα του συμπληρώματος μιας μεταβλητής ισούται με τη μεταβλητή αυτή, δηλαδή $(x')' = x$.

Θ4. Προσεταιριστική ιδιότητα λογικού αθροίσματος και λογικού γινομένου: $(x + y) + z = x + (y + z)$, $(x \cdot y) \cdot z = x \cdot (y \cdot z)$.

Θ5. Θεώρημα De Morgan: το συμπλήρωμα του αποτελέσματος μιας λογικής πράξης AND ή OR ανάμεσα σε δύο μεταβλητές λαμβάνεται, εάν συμπληρώσουμε κάθε μεταβλητή και εναλλάξουμε τους τελεστές, δηλαδή $(x + y)' = x' \cdot y'$, $(x \cdot y)' = x' + y'$.

Θεωρήματα άλγεβρας Boole

Θ6. Θεώρημα απορρόφησης: κατά το λογικό άθροισμα μιας μεταβλητής με το λογικό γινόμενο της μεταβλητής αυτής με άλλη μεταβλητή, το λογικό γινόμενο απορροφάται, δηλαδή $x + x \cdot y = x$.
Αντίστοιχα ισχύει και η απορρόφηση του λογικού αθροίσματος, δηλαδή $x \cdot (x + y) = x$.
Ειδική περίπτωση: $x + x' \cdot y = x + y$, $x \cdot (x' + y) = x \cdot y$.

Θ7. Θεώρημα ομοφωνίας: αποτελεί ειδική περίπτωση του θεωρήματος απορρόφησης, που εκφράζεται αλγεβρικά με τις παρακάτω σχέσεις

$$x \cdot y + x' \cdot z + y \cdot z = x \cdot y + x' \cdot z \quad \text{και}$$
$$(x + y) \cdot (x' + z) \cdot (y + z) = (x + y) \cdot (x' + z).$$

Τα προαναφερθέντα θεωρήματα που αναφέρθηκαν δεν είναι μοναδικά, ωστόσο είναι τα πιο βασικά και πολύ χρήσιμα για το χειρισμό αλγεβρικών εκφράσεων.

Αξιώματα και θεωρήματα άλγεβρας Boole

A1	$0 \cdot 0 = 0$	$1 + 1 = 1$
	$1 \cdot 1 = 1$	$0 + 0 = 0$
	$0 \cdot 1 = 0$	$1 + 0 = 1$
	Εάν $x = 0$, τότε $x' = 1$	Εάν $x = 1$, τότε $x' = 0$
A2	$x \cdot y = y \cdot x$	$x + y = y + x$
A3	$x \cdot (y + z) = x \cdot y + x \cdot z$	$x + (y \cdot z) = (x + y) \cdot (x + z)$
A4	$x \cdot 1 = x$	$x + 0 = x$
A5	$x \cdot x' = 0$	$x + x' = 1$
Θ1	$x \cdot x = x$	$x + x = x$
Θ2	$x \cdot 0 = 0$	$x + 1 = 1$
Θ3	$(x')' = x$	
Θ4	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$	$x + (y + z) = (x + y) + z$
Θ5	$(x \cdot y)' = x' + y'$	$(x + y)' = x' \cdot y'$
Θ6	$x + x \cdot y = x$	$x \cdot (x + y) = x$
	$x + x' \cdot y = x + y$	$x \cdot (x' + y) = x \cdot y$
Θ7	$x \cdot y + x' \cdot z + y \cdot z = x \cdot y + x' \cdot z$	$(x + y) \cdot (x' + z) \cdot (y + z) = (x + y) \cdot (x' + z)$

Αρχή του δυϊσμού: Κάθε αλγεβρική έκφραση της άλγεβρας Boole εξακολουθεί να ισχύει εάν εναλλάξουμε τους τελεστές (+ σε · και · σε +) και τα ουδέτερα στοιχεία (0 σε 1 και 1 σε 0).

Απόδειξη θεωρημάτων άλγεβρας Boole

Για την απόδειξη των θεωρημάτων χρησιμοποιούνται τα αξιώματα ή/και τα αποδεδειγμένα θεωρήματα και ξεκινώντας από το ένα μέλος της εξίσωσης καταλήγουμε στο άλλο, ή και από τα δύο μέλη καταλήγουμε στο ίδιο αποτέλεσμα.

Εναλλακτικά, ανάγουμε ένα θεώρημα σε αξίωμα.

Επίσης, μπορούμε να καταστρώσουμε τους πίνακες αλήθειας για τα δύο μέλη της εξίσωσης ενός θεωρήματος, αφού δύο αλγεβρικές εκφράσεις που έχουν τον ίδιο πίνακα αλήθειας είναι ισοδύναμες.

Απόδειξη θεωρημάτων άλγεβρας Boole

Παράδειγμα 1: απόδειξη της δεύτερης εξίσωσης του Θ2 με τη χρήση των αξιωμάτων της άλγεβρας Boole.

$x + 1 = 1 \cdot (x + 1)$	[αξίωμα Α4 για τα ουδέτερα στοιχεία]
$= (x + x') \cdot (x + 1)$	[αξίωμα Α5 για το συμπλήρωμα]
$= x + x' \cdot 1$	[αξίωμα Α3, επιμεριστικότητα]
$= x + x'$	[αξίωμα Α4 για τα ουδέτερα στοιχεία]
$= 1$	[αξίωμα Α5 για το συμπλήρωμα]

Παράδειγμα 2: απόδειξη της πρώτης εξίσωσης του Θ6.

$x + x \cdot y = x \cdot 1 + x \cdot y$	[αξίωμα Α4 για τα ουδέτερα στοιχεία]
$= x \cdot (1 + y)$	[αξίωμα Α3, επιμεριστικότητα]
$= x \cdot 1$	[θεώρημα Θ2]
$= x$	[αξίωμα Α4 για τα ουδέτερα στοιχεία]

Απόδειξη θεωρημάτων άλγεβρας Boole

Παράδειγμα 3: απόδειξη της δεύτερης εξίσωσης του Θ5 (θεώρημα De Morgan) με χρήση των πινάκων αλήθειας των δύο μελών τη εξίσωσης:

$$(x + y)' = x' \cdot y'$$

x	y	x'	y'	x + y	(x + y)'	x'· y'
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

Αλγεβρικοί μετασχηματισμοί

Οι αλγεβρικοί μετασχηματισμοί που βασίζονται στα αξιώματα και θεωρήματα της άλγεβρας Boole, εκτελούνται για την απλοποίηση αλγεβρικών εκφράσεων, έτσι ώστε αυτές να μπορούν να υλοποιηθούν με το μικρότερο δυνατό αριθμό λογικών πυλών.

Όπως σε κάθε αλγεβρικό σύστημα, έτσι και στην άλγεβρα Boole είναι καθορισμένη μια προτεραιότητα τελεστών, ώστε να επιτυγχάνεται ορθός χειρισμός των αλγεβρικών εκφράσεων.

Πρώτη προτεραιότητα, αποτελεί ο υπολογισμός των εκφράσεων που βρίσκονται εντός παρενθέσεων.

Ακολουθεί ο υπολογισμός των συμπληρωμάτων (τελεστής ’), στη συνέχεια ο υπολογισμός των λογικών γινομένων (τελεστής ·) και στο τέλος διενεργείται ο υπολογισμός των λογικών αθροισμάτων (τελεστής +).

Αλγεβρικοί μετασχηματισμοί

Παράδειγμα 1: Να απλοποιηθεί η αλγεβρική έκφραση $x \cdot y + x' \cdot z + y \cdot z$

$$
\begin{aligned}
x \cdot y + x' \cdot z + y \cdot z &= x \cdot y + x' \cdot z + 1 \cdot y \cdot z \\
&= x \cdot y + x' \cdot z + (x + x') \cdot y \cdot z \\
&= x \cdot y + x' \cdot z + x \cdot y \cdot z + x' \cdot y \cdot z \\
&= x \cdot y \cdot 1 + x \cdot y \cdot z + x' \cdot z \cdot 1 + x' \cdot z \cdot y \\
&= x \cdot y \cdot (1 + z) + x' \cdot z \cdot (1 + y) \\
&= x \cdot y \cdot 1 + x' \cdot z \cdot 1 \\
&= x \cdot y + x' \cdot z
\end{aligned}
$$

Αρχικά, αξιοποιώντας το αξίωμα για το συμπλήρωμα ($x + x' = 1$), έγινε προσπάθεια δημιουργίας λογικών γινομένων, τέτοιων ώστε στη συνέχεια να είναι δυνατή η εφαρμογή του αξιώματος της επιμεριστικότητας (παραγοντοποίηση) που θα οδηγήσει σε μείωση των λογικών γινομένων.

Αλγεβρικοί μετασχηματισμοί

Στη συνέχεια του μαθήματος, για λόγους απλότητας, όπως συμβαίνει και στη διατύπωση του πολλαπλασιασμού της κοινής άλγεβρας, ο τελεστής · θα παραλείπεται εντός των αλγεβρικών εκφράσεων, αλλά θα υπονοείται.

Παράδειγμα 2: Να απλοποιηθεί η αλγεβρική έκφραση $x'y + xz + yz + yzw'$

$$
\begin{aligned}
x'y + xz + yz + yzw' &= x'y + xz + yz1 + yzw' \\
&= x'y + xz + yz(1 + w') \\
&= x'y + xz + yz \\
&= x'y + xz + (x + x')yz \\
&= x'y + xz + xyz + x'yz \\
&= x'y \cdot 1 + xz \cdot 1 + xyz + x'yz \\
&= x'y(1 + z) + xz(1 + y) \\
&= x'y + xz
\end{aligned}
$$

Αλγεβρικοί μετασχηματισμοί

Παράδειγμα 3: Να απλοποιηθεί η αλγεβρική έκφραση xy + xy’ + x’y

$$xy + xy' + x'y = xy + xy' + x'y + xy$$
$$= x(y + y') + (x + x')y$$
$$= x1 + y1 = x + y$$

Αρχικά προσθέσαμε το λογικό γινόμενο xy (που περιλαμβάνεται στην αρχική έκφραση), αφού σύμφωνα με το θεώρημα Θ1, η ενέργεια αυτή δεν αλλοιώνει την αρχική έκφραση.

Στόχος ήταν η δημιουργία ενός συνδυασμού λογικών γινομένων, που μας δίνει δυνατότητα εφαρμογής του αξιώματος της επιμεριστικότητας, ώστε να καταλήξουμε σε απλοποιημένη μορφή της αρχικής έκφρασης.

Οι επιλογές που ακολουθήθηκαν στα παραδείγματα απλοποίησης λογικών εκφράσεων, αποτελούν χρήσιμες πρακτικές απλοποίησης, ωστόσο δεν συνιστούν μεθοδολογία απλοποίησης. Με συστηματική μεθοδολογία απλοποίησης θα ασχοληθούμε στην επόμενη ενότητα (χάρτης Karnaugh).

Λογικές συναρτήσεις

Αφού όπως διαπιστώσαμε μια λογική έκφραση μπορεί να απλοποιηθεί σε μία ισοδύναμη λογική έκφραση, γίνεται προφανές ότι μια λογική συνάρτηση μπορεί να περιγραφεί με διαφορετικές αλγεβρικές εκφράσεις.

Οι ισοδύναμες διαφορετικές αυτές λογικές εκφράσεις παράγουν τον ίδιο πίνακα αλήθειας, συνεπώς μια λογική συνάρτηση μπορεί να περιγραφεί με έναν και μοναδικό πίνακα αλήθειας.

Δύο ή περισσότερες αλγεβρικές εκφράσεις που παράγουν τον ίδιο πίνακα αλήθειας, είναι ισοδύναμες και αντιστοιχούν στην ίδια λογική συνάρτηση.

Όπως προαναφέρθηκε, μειονέκτημα της περιγραφής μιας λογικής συνάρτησης μέσω πίνακα αλήθειας είναι ότι το μέγεθος του πίνακα αυξάνεται εκθετικά με το πλήθος των δυαδικών μεταβλητών, αφού ο πίνακας αλήθειας μιας συνάρτησης n μεταβλητών έχει 2^n γραμμές.

Για παράδειγμα, στην περίπτωση συνάρτησης 16 μεταβλητών, απαιτείται πίνακας αλήθειας με περισσότερες από 65.5 χιλιάδες (2^{16}) γραμμές.

Λογικές συναρτήσεις

Παράδειγμα: Να διαπιστωθεί μέσω πινάκων αλήθειας, αλλά και μέσω αλγεβρικών μετασχηματισμών, ότι οι λογικές συναρτήσεις F(A, B, C) = AC’ + B + AB’C’ και G(A, B, C) = B + AC’, είναι ισοδύναμες.

Αφού οι δύο συναρτήσεις δίνονται σε μορφή αθροίσματος γινομένων, για ευκολία κατά τη συμπλήρωση των πινάκων αλήθειας, χρησιμοποιούμε το θεώρημα Θ2 (x + 1 = 1), σύμφωνα με το οποίο όταν οποιοδήποτε όρος (λογικό γινόμενο) ενός λογικού αθροίσματος λαμβάνει τιμή 1, η συνάρτηση λαμβάνει τιμή 1, ανεξάρτητα από τι τιμές που λαμβάνουν οι άλλοι όροι.

Επομένως, για να συμπληρώσουμε τον πίνακα αλήθειας μιας λογικής συνάρτησης, προσδιορίζουμε τους συνδυασμούς τιμών των μεταβλητών για τους οποίους κάθε όρος του λογικού αθροίσματος λαμβάνει την τιμή 1. Για τους συνδυασμούς αυτούς, η λογική συνάρτηση λαμβάνει τιμή 1, ενώ για τους υπόλοιπους λαμβάνει τιμή 0.

Λογικές συναρτήσεις

Μεταβλητές			F = AC’ + B + AB’C’				G = B + AC’		
A	B	C	AC’	B	AB’C’	F	B	AC’	G
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	1	0	0	1	0	1	1	0	1
0	1	1	0	1	0	1	1	0	1
1	0	0	1	0	1	1	0	1	1
1	0	1	0	0	0	0	0	0	0
1	1	0	1	1	0	1	1	1	1
1	1	1	0	1	0	1	1	0	1

Οι δύο συναρτήσεις (F, G) παράγουν τον ίδιο πίνακα αλήθειας, συνεπώς είναι ισοδύναμες.

Λογικές συναρτήσεις

Διενεργούμε αλγεβρικούς μετασχηματισμούς στη συνάρτηση F, εφαρμόζοντας αξιώματα και θεωρήματα της άλγεβρας Boole και καταλήγουμε στην G:

$$\begin{aligned} F(A, B, C) &= AC' + B + AB'C' \\ &= AC'(1 + B') + B \\ &= AC'1 + B \\ &= AC' + B = B + AC' = G(A, B, C) \end{aligned}$$

Επισημαίνεται ότι υπάρχουν και άλλες λογικές συναρτήσεις που έχουν τον ίδιο πίνακα αλήθειας με τις F και G, όπως οι:

$$K(A, B, C) = A'BC' + A'BC + AB'C' + ABC' + ABC$$

$$L(A, B, C) = B + AB'C'$$

Λογικές συναρτήσεις

Για να εξάγουμε την αλγεβρική έκφραση μιας συνάρτησης από τον πίνακα αλήθειας, εντοπίζουμε τους συνδυασμούς τιμών των μεταβλητών, για τους οποίους η συνάρτηση λαμβάνει τιμή 1.

Στη συνέχεια εκφράζουμε κάθε συνδυασμό ως λογικό γινόμενο των μεταβλητών με τιμή 1 και των συμπληρωμάτων των μεταβλητών με τιμή 0 και αθροίζουμε τα λογικά γινόμενα.

Ακολουθώντας τη διαδικασία αυτή για τον πίνακα αλήθειας της συνάρτησης F(x,y,z) = x'y + xz, προκύπτει η συνάρτηση F(x,y,z) = x'yz' + x'yz + xy'z + xyz, η οποία είναι ισοδύναμη με την αρχική συνάρτηση, δηλαδή την F(x,y,z) = x'y + xz.

x	y	z	F(x, y, z)
0	0	0	0
0	0	1	0
0	1	0	**1**
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	0
1	1	1	**1**

Συμπληρωματική λογική συνάρτηση

Μια λογική συνάρτηση F′ ονομάζεται συμπληρωματική συνάρτηση της συνάρτησης F, όταν λαμβάνει λογική τιμή 1 ή 0, για τους συνδυασμούς τιμών των μεταβλητών για τους οποίους η F λαμβάνει τιμή 0 ή 1, αντίστοιχα.

Ο τιμές στον πίνακα αλήθειας της συμπληρωματικής συνάρτησης της F(x,y,z) = x′y + xz, προκύπτουν από εκείνες της συνάρτησης F(x,y,z) για κάθε συνδυασμό τιμών των μεταβλητών.

Για να εξαγάγουμε την αλγεβρική έκφραση της F′ από τον πίνακα αλήθειας, ενεργούμε με τον τρόπο που ενεργήσαμε για την εξαγωγή της έκφρασης της F.

x	y	z	F(x, y, z)	F′(x, y, z)
0	0	0	0	**1**
0	0	1	0	**1**
0	1	0	**1**	0
0	1	1	**1**	0
1	0	0	0	**1**
1	0	1	**1**	0
1	1	0	0	**1**
1	1	1	**1**	0

F′(x,y,z) = x′y′z′ + x′y′z + xy′z′ + xyz′

Γενικευμένη μορφή θεωρήματος De Morgan

Η συμπληρωματική συνάρτηση F′ προκύπτει από τη συνάρτηση F, ως εξής:

$$F'(x_1, x_2, \ldots, x_n, +, \cdot, 0, 1) = F(x'_1, x'_2, \ldots, x'_n, \cdot, +, 1, 0)$$

$$(x_1 + x_2 + x_3 + \ldots + x_n)' = x'_1 \cdot x'_2 \cdot x'_3 \cdot \ldots \cdot x'_n$$

$$(x_1 \cdot x_2 \cdot x_3 \cdot \ldots \cdot x_n)' = x'_1 + x'_2 + x'_3 + \ldots + x'_n$$

Αυτό σημαίνει ότι κατά την εξαγωγή της F′, στη θέση κάθε μεταβλητής τίθεται το συμπλήρωμά της, οι τελεστές + και · εναλλάσσονται και οι σταθερές τιμές 0 και 1, επίσης εναλλάσσονται.

Αυτό ισοδυναμεί με την εφαρμογή αρχικά της αρχής του δυϊσμού και στη συνέχεια με την αντικατάσταση κάθε μεταβλητής με το συμπλήρωμά της.

Επισημαίνεται ότι, κατά την παραγωγή της συμπληρωματικής συνάρτησης, θα πρέπει να τηρείται η προτεραιότητα των πράξεων.

Γενικευμένη μορφή θεωρήματος De Morgan

Εάν κατά τον υπολογισμό του συμπληρώματος της συνάρτησης $F(x,y,z) = x + yz$ εφαρμόσουμε τη γενικευμένη μορφή του θεωρήματος De Morgan, συμπληρώνοντας τις μεταβλητές και εναλλάσσοντας τους τελεστές, χωρίς όμως, να τηρήσουμε την ορθή προτεραιότητα πράξεων, θα καταλήξουμε στη συνάρτηση $F'(x,y,z) = x'y' + z'$, που είναι εσφαλμένη.

Ακολουθώντας την ορθή προτεραιότητα πράξεων (παρενθέσεις, συμπληρώματα, λογικά γινόμενα, λογικά αθροίσματα), καταλήγουμε στη σωστή συμπληρωματική συνάρτηση:

$$F'(x,y,z) = (x + yz)' = x'(yz)' = x'(y' + z') = x'y' + x'z'$$

Γενικευμένη μορφή θεωρήματος De Morgan

Παράδειγμα 1: Να προσδιοριστούν οι συμπληρωματικές συναρτήσεις των λογικών συναρτήσεων: $F = x'yz' + x'y'z$ και $G = x(y'z' + yz)$.

$$\begin{aligned} F'(x,y,z) &= (x'yz' + x'y'z)' \\ &= (x'yz')'(x'y'z)' \\ &= (x + y' + z)(x + y + z') \end{aligned}$$

$$\begin{aligned} G'(x,y,z) &= [x(y'z' + yz)]' \\ &= x' + (y'z' + yz)' \\ &= x' + (y'z')'(yz)' \\ &= x' + (y + z)(y' + z') \\ &= x' + yy' + yz' + y'z + zz' \\ &= x' + yz' + y'z \end{aligned}$$

Γενικευμένη μορφή θεωρήματος De Morgan

Παράδειγμα 2: Να αποδείξετε ότι η συμπληρωματική συνάρτηση της συνάρτησης F(A,B,C,D) = (AC′ + A′B′)[(B′ + C′)D′ + D] περιγράφεται από την αλγεβρική έκφραση A′B + AC.

$$
\begin{aligned}
F'(A,B,C,D) &= \{(AC' + A'B')[(B' + C')D' + D]\}' \\
&= [(AC' + A'B')]' + [(B' + C')D' + D]' \\
&= (AC')'(A'B')' + [(B' + C')D']'D' \\
&= (A' + C)(A + B) + [(B' + C')' + D]D' \\
&= (A' + C)(A + B) + (BC + D)D' \\
&= A'A + A'B + AC + BC + BCD' + DD' \\
&= A'B + AC + BC + BCD' \\
&= A'B + AC + BC(1 + D') \\
&= A'B + AC + BC \\
&= A'B + AC
\end{aligned}
$$

Γενικευμένη μορφή θεωρήματος De Morgan

F(A,B,C,D) = (AC′ + A′B′)[(B′ + C′)D′ + D]

F′(A,B,C,D) = A′B + AC

A	B	C	D	F	F′
0	0	0	0	1	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	0	1

Ελαχιστόροι και μεγιστόροι λογικής συνάρτησης

Ένα λογικό γινόμενο, στο οποίο συμμετέχουν όλες οι μεταβλητές μιας συνάρτησης (στην κανονική μορφή τους ή στη μορφή συμπληρώματος) μία φορά, αναφέρεται ως ελαχιστόρος (minterm), ενώ ένα λογικό άθροισμα μεταβλητών με τα ίδια χαρακτηριστικά αναφέρεται ως μεγιστόρος (maxterm).

Για παράδειγμα, όταν πρόκειται για τρεις μεταβλητές x, y και z, το λογικό γινόμενο xy'z και το λογικό άθροισμα x' + y + z' είναι ένας ελαχιστόρος και ένας μεγιστόρος, αντίστοιχα.

Κατά τη δημιουργία του πίνακα αλήθειας μιας λογικής συνάρτησης με n μεταβλητές, γράφουμε κατά σειρά τους δυαδικούς αριθμούς από 0 έως 2^{n-1} και αντιστοιχίζουμε κάθε δυαδικό ψηφίο σε μία από τις μεταβλητές.

Ελαχιστόροι και μεγιστόροι λογικής συνάρτησης

Έτσι, για 3 μεταβλητές, όταν x = y = z = 0, σχηματίζεται ο μικρότερος δυαδικός αριθμός (0) και όταν x = y = z = 1 ο μεγαλύτερος δυνατός δυαδικός αριθμός (7).

Υπάρχει λοιπόν αντιστοιχία μεταξύ των 8 δυνατών συνδυασμών τιμών των 3 μεταβλητών και των 8 ελαχιστόρων.

Όταν η τιμή μιας μεταβλητής είναι 0, τότε στον αντίστοιχο ελαχιστόρο συμμετέχει το συμπλήρωμα της μεταβλητής αυτής, ενώ όταν η τιμή της μεταβλητής είναι 1, τότε στον αντίστοιχο ελαχιστόρο συμμετέχει η μεταβλητή αυτή με την κανονική μορφή της.

Ελαχιστόροι και μεγιστόροι λογικής συνάρτησης

Οι ελαχιστόροι ονομάζονται ως m_0 έως m_7 (ο δείκτης συμπίπτει με το δεκαδικό αριθμό που αντιστοιχεί σε κάθε ελαχιστόρο).

Οι μεγιστόροι (M_0 έως M_7) είναι τα συμπληρώματα των αντίστοιχων ελαχιστόρων και προκύπτουν από τους ελαχιστόρους με εφαρμογή του θεωρήματος De Morgan.

Για n μεταβλητές, σχηματίζονται 2^n ελαχιστόροι και 2^n μεγιστόροι.

x	y	z	Ελαχιστόροι		Μεγιστόροι	
0	0	0	x′y′z′	m_0	x + y + z	M_0
0	0	1	x′y′z	m_1	x + y + z′	M_1
0	1	0	x′y z′	m_2	x + y′ + z	M_2
0	1	1	x′y z	m_3	x + y′ + z′	M_3
1	0	0	x y′z′	m_4	x′ + y + z	M_4
1	0	1	x y′z	m_5	x′ + y + z′	M_5
1	1	0	x y z′	m_6	x′ + y ′+ z	M_6
1	1	1	x y z	m_7	x′ + y′ + z′	M_7

Κανονικές μορφές λογικής συνάρτησης

Για να εξάγουμε την αλγεβρική έκφραση μιας συνάρτησης από τον πίνακα αλήθειας, εντοπίζουμε τους συνδυασμούς τιμών των μεταβλητών για τους οποίους η συνάρτηση λαμβάνει τιμή 1, εκφράζουμε κάθε συνδυασμό ως λογικό γινόμενο των μεταβλητών με τιμή 1 και των συμπληρωμάτων των μεταβλητών με τιμή 0 και αθροίζουμε τα λογικά γινόμενα που προκύπτουν.

Στην ουσία, με τον τρόπο αυτό εκφράζουμε τη λογική συνάρτηση σε μορφή αθροίσματος ελαχιστόρων.

Για παράδειγμα, η συνάρτηση $F(x,y,z) = x'yz' + x'yz + xy'z + xyz$, μπορεί να γραφεί και ως άθροισμα ελαχιστόρων: $F(x,y,z) = m_2 + m_3 + m_5 + m_7$.

Κανονικές μορφές λογικής συνάρτησης

Η F′ μπορεί να εκφραστεί σε μορφή αθροίσματος ελαχιστόρων, εάν προσθέσουμε τους ελαχιστόρους που αντιστοιχούν στους συνδυασμούς τιμών των μεταβλητών, για τους οποίους η συνάρτηση λαμβάνει τιμή 0, που είναι οι ελαχιστόροι που λείπουν από την F:

$$F'(x,y,z) = x'y'z' + x'y'z + xy'z' + xyz' = m_0 + m_1 + m_4 + m_6$$

Η συμπληρωματική συνάρτηση της F′ είναι η F (σύμφωνα με το θεώρημα διπλής άρνησης) και προκύπτει εύκολα από την F′ με εφαρμογή του θεωρήματος De Morgan:

$$F(x,y,z) = (m_0 + m_1 + m_4 + m_6)' = m'_0 m'_1 m'_4 m'_6 = M_0 M_1 M_4 M_6$$
$$= (x + y + z)(x + y + z')(x' + y + z)(x' + y' + z)$$

Έτσι, εκφράσαμε τη συνάρτηση σε μορφή γινομένου μεγιστόρων. Σε αυτή συμμετέχουν οι μεγιστόροι που αντιστοιχούν στους συνδυασμούς τιμών των μεταβλητών, για τους οποίους η συνάρτηση λαμβάνει λογική τιμή 0.

Κανονικές μορφές λογικής συνάρτησης

Οι εκφράσεις μιας λογικής συνάρτησης σε μορφή αθροίσματος ελαχιστόρων και γινομένου μεγιστόρων είναι μοναδικές και αναφέρονται ως κανονικές μορφές (canonical forms).

Για να μετατρέψουμε την έκφραση μιας λογικής συνάρτησης από τη μια κανονική μορφή στην άλλη, εναλλάσσουμε τα σύμβολα m, M και τους τελεστές ·, +, και ως δείκτες θέτουμε τους δείκτες των ελαχιστόρων ή των μεγιστόρων που λείπουν από την αρχική κανονική μορφή.

Για να προκύψουν οι δείκτες αυτοί, λαμβάνουμε υπόψη ότι το πλήθος των ελαχιστόρων ή μεγιστόρων για συναρτήσεις n μεταβλητών είναι 2^n.

Το άθροισμα ελαχιστόρων και το γινόμενο μεγιστόρων μιας λογικής συνάρτησης αναφέρονται και ως Σ( ), Π( ), αντίστοιχα, θέτοντας εντός των παρενθέσεων τους δείκτες των ελαχιστόρων ή των μεγιστόρων, αντίστοιχα.

Πρότυπες μορφές λογικής συνάρτησης

Οι κανονικές μορφές μιας λογικής συνάρτησης προκύπτουν απευθείας από τον πίνακα αλήθειας και λόγω του ότι στους ελαχιστόρους και στους μεγιστόρους συμμετέχουν όλες οι μεταβλητές της συνάρτησης, οι μορφές αυτές δεν είναι συνήθως οι απλούστερες δυνατές που μπορούν να οδηγήσουν σε υλοποίηση με μικρό πλήθος λογικών πυλών.

Έτσι, οι λογικές συναρτήσεις που υλοποιούνται σε ψηφιακά κυκλώματα εκφράζονται συχνά σε μορφή αθροίσματος γινομένων ή σε μορφή γινομένου αθροισμάτων, χωρίς όμως στα γινόμενα και τα αθροίσματα αυτά να συμμετέχουν όλες οι μεταβλητές.

Οι μορφές αυτές αναφέρονται και ως πρότυπες μορφές (standard forms).

Πρότυπες μορφές λογικής συνάρτησης

Για να μετατρέψουμε μια πρότυπη μορφή αθροίσματος γινομένων σε κανονική, ελέγχουμε κάθε γινόμενο, ώστε να διαπιστώσουμε ποιες μεταβλητές δε συμμετέχουν σε αυτό.

Για κάθε μεταβλητή που δε συμμετέχει σε κάποιο από τα γινόμενα (έστω x), πολλαπλασιάζουμε το γινόμενο με τον όρο (x + x’).

Με βάση το αξίωμα της άλγεβρας Boole για το συμπλήρωμα μιας μεταβλητής, ο όρος αυτός ισούται με 1 και η σχετική πράξη δεν επηρεάζει τη λογική τιμή του γινομένου.

Στη συνέχεια εφαρμόζουμε το αξίωμα της επιμεριστικότητας σε κάθε ελλιπές γινόμενο και λαμβάνουμε την επιθυμητή κανονική μορφή αθροίσματος ελαχιστόρων.

Πρότυπες μορφές λογικής συνάρτησης

Στην περίπτωση πρότυπης μορφής γινομένου αθροισμάτων, για κάθε μεταβλητή που δε συμμετέχει σε κάποιο άθροισμα (έστω x), προσθέτουμε στο άθροισμα τον όρο xx′, αφού η πράξη αυτή δεν επηρεάζει τη λογική τιμή του αθροίσματος (λόγω του ότι xx′ = 0).

Κατόπιν, εφαρμόζουμε το αξίωμα της επιμεριστικότητας σε κάθε ελλιπές άθροισμα και λαμβάνουμε την επιθυμητή κανονική μορφή γινομένου μεγιστόρων.

Η μετατροπή μεταξύ πρότυπων μορφών γίνεται εύκολα με χρήση του αξιώματος της επιμεριστικότητας.

Πρότυπες μορφές λογικής συνάρτησης

Παράδειγμα 1: Μετατροπή της πρότυπης μορφής

$$F(x,y,z) = xy + x'z + yz$$

στις δύο κανονικές μορφές.

Η κανονική μορφή αθροίσματος ελαχιστόρων προκύπτει ως εξής:

$$\begin{aligned} F(x,y,z) &= xy + x'z + yz \\ &= xy(z + z') + x'z(y + y') + yz(x + x') \\ &= xyz + xyz' + x'yz + x'y'z + xyz + x'yz \\ &= m_7 + m_6 + m_3 + m_1 = \Sigma(1, 3, 6, 7) \end{aligned}$$

Οι όροι m_3 και m_7 προκύπτουν και δεύτερη φορά στο τελικό άθροισμα, αλλά απαλείφονται λόγω του θεωρήματος Θ1 της άλγεβρας Boole.

Η κανονική μορφή γινομένου μεγιστόρων προκύπτει με εφαρμογή του θεωρήματος De Morgan: $F(x,y,z) = \Sigma(1, 3, 6, 7) = \Pi(0, 2, 4, 5)$.

Πρότυπες μορφές λογικής συνάρτησης

Παράδειγμα 2: Μετατροπή της πρότυπης μορφής αθροίσματος γινομένων F(x,y,z) = xy + x′z + yz σε πρότυπη μορφή γινομένου αθροισμάτων.

$$
\begin{aligned}
F(x,y,z) &= xy + x'z + yz \\
&= (x + x')(x + z)(y + x')(y + z) + yz \\
&= (x + z)(x' + y)(y + z) + yz \\
&= (x + z + y)(x' + y + y)(y + z + y)(x + z + z)(x' + y + z)(y + z + z) \\
&= (x + y + z)(x' + y)(y + z)(x + z)(x' + y + z)(y + z) \\
&= (x + y + z)(x' + y)(y + z)(x + z)(x' + y + z)
\end{aligned}
$$

Η μετατροπή έγινε με επαναληπτική εφαρμογή του αξιώματος της επιμεριστικότητας και απαλοιφή των μεταβλητών που επαναλαμβάνονται στα αθροίσματα, καθώς και των αθροισμάτων που επαναλαμβάνονται στη νέα πρότυπη μορφή της συνάρτησης

Πρότυπες μορφές λογικής συνάρτησης

Παράδειγμα 3: Μετατροπή της πρότυπης μορφής

F(x,y,z) = (x + y + z)(x′ + y)(y + z)(x+z)(x′ + y + z)

σε κανονική μορφή γινομένου μεγιστόρων

$$
\begin{aligned}
F(x,y,z) &= (x + y + z)(x' + y)(y + z)(x + z)(x' + y + z) \\
&= (x + y + z)(x' + y + zz')(y + z + xx')(x + z + yy')(x' + y + z) \\
&= (x + y + z)(x' + y + z)(x' + y + z')(y + z + x)(y + z + x') \\
&\quad (x + z + y)(x + z + y') (x' + y + z) \\
&= M_0M_4M_5M_2 \\
&= \Pi(0, 2, 4, 5)
\end{aligned}
$$

Οι όροι M_0 και M_4 προκύπτουν περισσότερες από μία φορές στο τελικό γινόμενο, αλλά απαλείφονται λόγω του θεωρήματος Θ1 της άλγεβρας Boole.

Λογικά κυκλώματα με πύλες NOT, AND και OR

Όπως προαναφέρθηκε, ένα λογικό κύκλωμα αποτελείται από γραμμές διασύνδεσης λογικών πυλών που αντιστοιχούν στις διαδρομές των δυαδικών σημάτων και από λογικές πύλες που επιτελούν την επεξεργασία μεταξύ των σημάτων, η οποία δηλώνεται στη λογική συνάρτηση που υλοποιείται από κάθε πύλη.

Τα τρία λογικά γινόμενα που περιλαμβάνονται στη συνάρτηση F(x,y,z) = xy + x′z + yz, μπορούν να υλοποιηθούν με ισάριθμες πύλες AND.

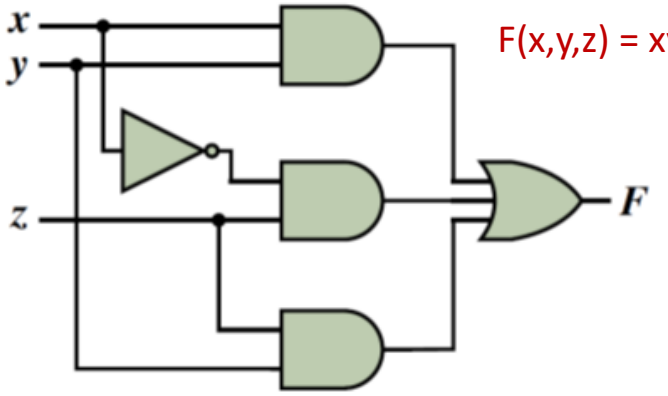
Λόγω του ότι στα γινόμενα συμμετέχουν δύο μεταβλητές, θα πρέπει να χρησιμοποιήσουμε πύλες AND με δύο εισόδους.

Η συμπληρωματική μορφή της μεταβλητής x που εμφανίζεται στο δεύτερο κατά σειρά γινόμενο θα πρέπει να παραχθεί με χρήση ενός αντιστροφέα, ο οποίος θα προηγείται της πύλης AND που υλοποιεί το δεύτερο κατά σειρά λογικό γινόμενο.

Λογικά κυκλώματα με πύλες NOT, AND και OR

Το λογικό άθροισμα των τριών γινομένων μπορεί να υλοποιηθεί με χρήση μιας πύλης OR τριών εισόδων, η οποία θα λαμβάνει ως εισόδους τις εξόδους των πυλών AND.

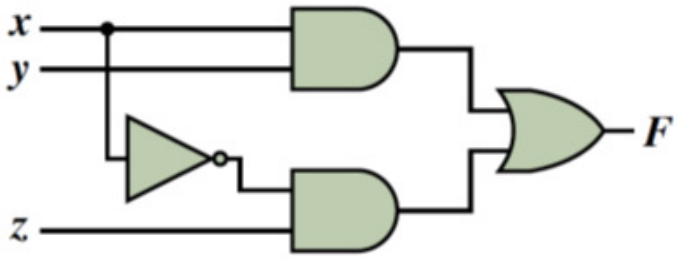
Το λογικό κύκλωμα που υλοποιεί την F περιλαμβάνει δύο επίπεδα πυλών, εάν δε συνυπολογίσουμε τον αντιστροφέα που απαιτείται για την παραγωγή της συμπληρωματικής μορφής της εισόδου x.



F(x,y,z) = xy + x′z + yz

Λογικά κυκλώματα με πύλες NOT, AND και OR

Εάν στη συνάρτηση εφαρμόσουμε το θεώρημα ομοφωνίας, προκύπτει η ισοδύναμη συνάρτηση F(x,y,z) = xy + x'z, η οποία μπορεί να υλοποιηθεί με μία πύλη AND λιγότερη, αφού τα λογικά γινόμενα που συμμετέχουν στη συνάρτηση μειώνονται κατά ένα.



Λογικά κυκλώματα με πύλες NOT, AND και OR

Εφαρμόζοντας το αξίωμα της επιμεριστικότητας στη προηγούμενη μορφή της συνάρτησης (F(x,y,z) = xy + x'z), καταλήγουμε σε ισοδύναμη συνάρτηση μορφής γινομένου αθροισμάτων:

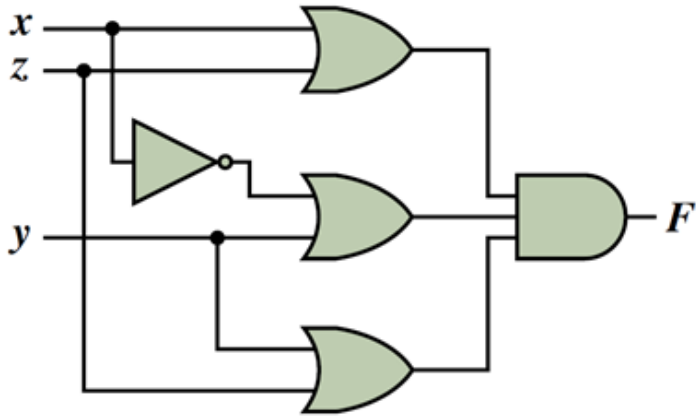
$$F(x,y,z) = xy + x'z = (xy + x')(xy + z) = (x + x')(y + x')(x + z)(y + z)$$
$$= (x' + y)(x + z)(y + z).$$

Τα τρία λογικά αθροίσματα που περιλαμβάνονται μπορούν να υλοποιηθούν με ισάριθμες πύλες OR δύο εισόδων και το λογικό γινόμενο των τριών αθροισμάτων μπορεί να υλοποιηθεί με χρήση μιας πύλης AND τριών εισόδων, η οποία θα λαμβάνει ως εισόδους τις εξόδους των πυλών OR.

Η συμπληρωματική μορφή της μεταβλητής x που εμφανίζεται στο πρώτο κατά σειρά άθροισμα, παράγεται με χρήση ενός αντιστροφέα

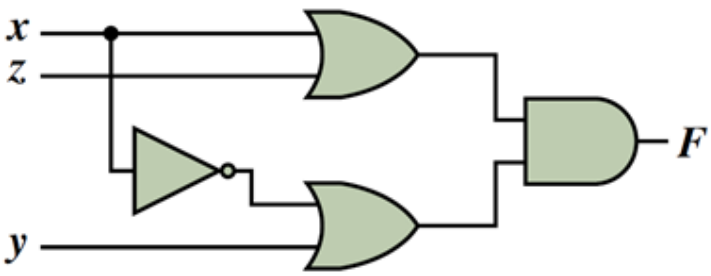
Λογικά κυκλώματα με πύλες NOT, AND και OR

$F(x,y,z) = (x' + y)(x + z)(y + z)$



Λογικά κυκλώματα με πύλες NOT, AND και OR

Εάν στη συνάρτηση που υλοποιήθηκε εφαρμόσουμε το θεώρημα ομοφωνίας, προκύπτει η επίσης ισοδύναμη συνάρτηση $F(x,y,z) = (x + z)(x' + y)$, η οποία υλοποιείται με μία πύλη OR λιγότερη, αφού τα λογικά αθροίσματα μειώνονται κατά ένα.



Λογικά κυκλώματα με πύλες NOT, AND και OR

Συμπεραίνουμε ότι μια δεδομένη λογική συνάρτηση μπορεί να υλοποιηθεί με λογικά κυκλώματα διαφορετικής δομής που αποτελούνται από:

- ✓ αντιστροφείς για την παραγωγή των συμπληρωματικών μορφών των μεταβλητών εισόδου (όπου αυτή είναι απαραίτητη) και
- ✓ δύο επίπεδα πυλών σε διάταξη AND-OR ή OR-AND, ανάλογα με το αν η συνάρτηση είναι σε μορφή αθροίσματος γινομένων ή γινομένου αθροισμάτων, αντίστοιχα.

Κάποια από τα λογικά κυκλώματα που υλοποιούν μια συνάρτηση είναι απλούστερα από άλλα και σημαντικός στόχος των σχεδιαστών είναι η μείωση του κόστους υλοποίησης, μια ένδειξη του οποίου είναι το άθροισμα του πλήθους των λογικών πυλών και του πλήθους των εισόδων των πυλών του κυκλώματος.

Λογικά κυκλώματα με πύλες NAND

Οι λογικές πύλες NAND και NOR υλοποιούνται, ως ηλεκτρονικά κυκλώματα, ευκολότερα και με μικρότερο κόστος συγκριτικά με τις πύλες AND και OR.

Επομένως, είναι χρήσιμο να μπορούμε να σχεδιάζουμε λογικά κυκλώματα αποκλειστικά μόνο με πύλες NAND ή NOR.

Εάν εφαρμόσουμε κατά σειρά τα θεωρήματα διπλής άρνησης και De Morgan στη λογική συνάρτηση μορφής αθροίσματος γινομένων $F(x,y,z) = xy + x'z + yz$, λαμβάνουμε την ισοδύναμη λογική συνάρτηση:

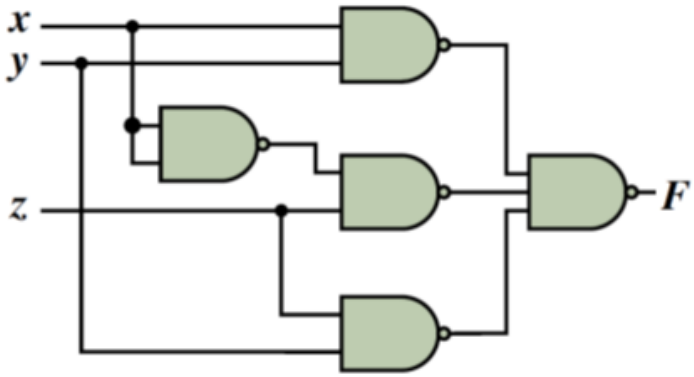
$$F(x,y,z) = [(xy + x'z + yz)']' = [(xy)'(x'z)'(yz)']'$$

Λογικά κυκλώματα με πύλες NAND

Τα συμπληρώματα των λογικών γινομένων μπορούν να υλοποιηθούν με 3 πύλες NAND 2 εισόδων, ενώ το συμπλήρωμα του συνολικού γινομένου και συνεπώς η συνολική συνάρτηση μπορεί να υλοποιηθεί, εάν θέσουμε τις εξόδους των πυλών αυτών ως εισόδους σε μία πύλη NAND 3 εισόδων.

Για την παραγωγή της συμπληρωματικής μορφής της μεταβλητής x που απαιτείται, μπορεί να χρησιμοποιηθεί μία πύλη NAND δύο εισόδων, στις εισόδους της οποίας θα πρέπει να τεθεί η μεταβλητή εισόδου x

$F(x,y,z)= [(xy)'(x'z)'(yz)']'$



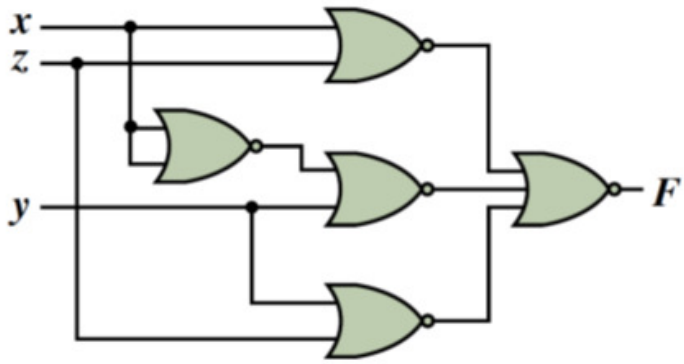
Λογικά κυκλώματα με πύλες NOR

Εάν εφαρμόσουμε όμοια διαδικασία στη συνάρτηση μορφής γινομένου αθροισμάτων $F(x,y,z) = (x + z)(x' + y)(y + z)$, λαμβάνουμε την ισοδύναμη λογική συνάρτηση:

$$F(x,y,z) = \{[(x + z)(x' + y)(y + z)]'\}' = [(x + z)' + (x' + y)' + (y + z)']'.$$

Τα συμπληρώματα των λογικών αθροισμάτων υλοποιούνται με 3 πύλες NOR 2 εισόδων, ενώ το συμπλήρωμα του συνολικού αθροίσματος υλοποιείται εάν θέσουμε τις εξόδους των πυλών αυτών ως εισόδους σε μία πύλη NOR 3 εισόδων.

Η αντιστροφή της μεταβλητής x υλοποιείται με μία επιπλέον πύλη NOR δύο εισόδων.



Λογικά κυκλώματα με πύλες NAND και NOR

Το συμπέρασμα που προκύπτει είναι ότι οποιοδήποτε λογικό κύκλωμα διάταξης AND-OR μπορεί να μετατραπεί σε λογικό κύκλωμα διάταξης NAND-NAND με την ίδια ακριβώς τοπολογία, ενώ οποιοδήποτε λογικό κύκλωμα διάταξης OR-AND μπορεί να μετατραπεί σε λογικό κύκλωμα διάταξης NOR-NOR ίδιας τοπολογίας.

Τα λογικά κυκλώματα που υλοποιούνται με τους 4 τρόπους διάταξης λογικών πυλών που προαναφέρθηκαν, περιλαμβάνουν 2 επίπεδα πυλών, πρόκειται δηλαδή για υλοποιήσεις λογικών συναρτήσεων σε δύο επίπεδα, εάν δεν συνυπολογίσουμε τους αντιστροφείς που απαιτούνται για την παραγωγή των συμπληρωματικών μορφών των μεταβλητών εισόδου.

Το πλήθος των επιπέδων πυλών ενός λογικού κυκλώματος είναι το μέγιστο πλήθος πυλών που πρέπει να διέλθουν τα δυαδικά σήματα εισόδου για να φτάσουν στην έξοδο.

Πλήρη (καθολικά) σύνολα λογικών πυλών

Με τις λογικές πράξεις AND, OR και NOT, είναι δυνατή η παράσταση κάθε λογικής συνάρτησης. Ένα τέτοιο σύνολο λογικών πράξεων αναφέρεται ως πλήρες.

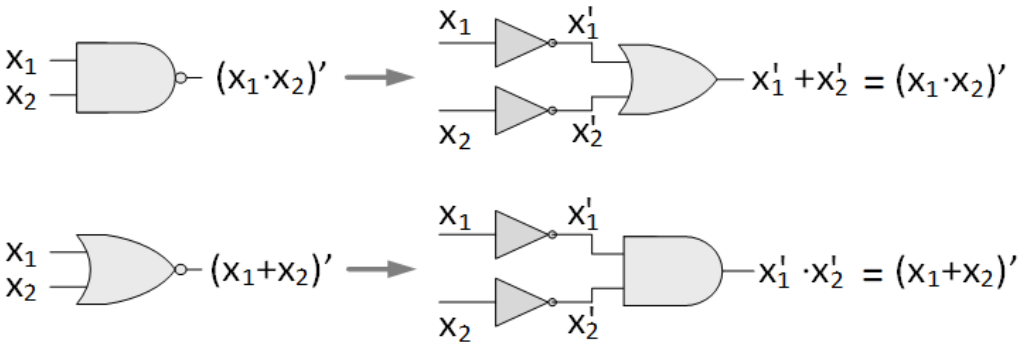
Επίσης, κάθε σύνολο λογικών πυλών με το οποίο μπορεί να υλοποιηθεί οποιαδήποτε λογική συνάρτηση αναφέρεται ως πλήρες (ή καθολικό) σύνολο πυλών.

Σύμφωνα με όσα έχουν αναφερθεί, το σύνολο που περιλαμβάνει αντιστροφείς και πύλες AND και OR 2 εισόδων αποτελεί πλήρες σύνολο πυλών.

Επίσης, οι πύλες NAND 2 εισόδων και οι πύλες NOR 2 εισόδων είναι πλήρη σύνολα πυλών.

Πλήρη (καθολικά) σύνολα λογικών πυλών

Πλήρη σύνολα πυλών είναι και το σύνολο που περιλαμβάνει αντιστροφείς και πύλες OR 2 εισόδων, καθώς και το σύνολο που περιλαμβάνει αντιστροφείς και πύλες AND 2 εισόδων, αφού είναι ισοδύναμα με τα πλήρη σύνολα που περιλαμβάνουν πύλες NAND 2 εισόδων και πύλες NOR 2 εισόδων, αντίστοιχα.



Λογικά κυκλώματα πολλών επιπέδων

Οι υλοποιήσεις λογικών συναρτήσεων με δύο επίπεδα πυλών είναι ικανοποιητικές στις περιπτώσεις λογικών συναρτήσεων με λίγες μεταβλητές.

Σε συναρτήσεις πολλών μεταβλητών, μπορεί να χρησιμοποιηθεί η μέθοδος της παραγοντοποίησης (δηλαδή το αξίωμα της επιμεριστικότητας), ώστε να επιτευχθεί υλοποίηση σε περισσότερα επίπεδα πυλών, αλλά με πύλες περιορισμένου πλήθους εισόδων.

Οι πύλες αυτές παρουσιάζουν καλύτερα χαρακτηριστικά σε σχέση με τις πύλες πολλών εισόδων, όπως μικρότερη καθυστέρηση απόκρισης και μεγαλύτερη ανεκτικότητα στην παρουσία θορύβου.

Λογικά κυκλώματα πολλών επιπέδων

Η συνάρτηση F(A, B, C, D, E, G, H) = AC'G + ADEG + BC'H + BDEH, απαιτεί για την υλοποίησή της έναν αντιστροφέα για την παραγωγή της συμπληρωματικής μορφής της μεταβλητής εισόδου C, 4 πύλες AND (2 με 3 εισόδους και 2 με 4 εισόδους) για την παραγωγή των λογικών γινομένων και μία πύλη OR 4 εισόδων για την παραγωγή του συνολικού λογικού αθροίσματος.

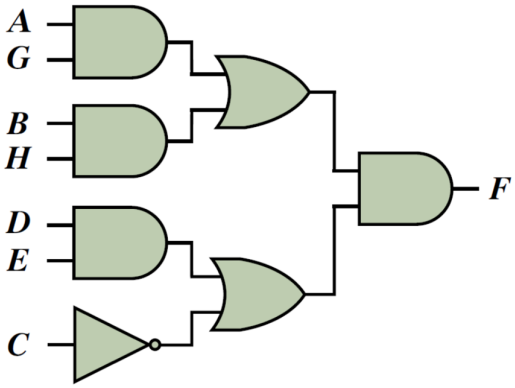
Χρησιμοποιώντας τη μέθοδο της παραγοντοποίησης, εφαρμόζοντας, δηλαδή, το αξίωμα επιμεριστικότητας, προκύπτει η ισοδύναμη συνάρτηση:

$$F = AC'G + ADEG + BC'H + BDEH$$
$$= AG(C' + DE) + BH(C' + DE)$$
$$= (AG + BH)(C' + DE)$$

Λογικά κυκλώματα πολλών επιπέδων

Η συνάρτηση F = (AG + BH)(C' + DE) που προέκυψε, σύμφωνα και με την προτεραιότητα πράξεων της άλγεβρας Boole, απαιτεί για την υλοποίησή της, τρία επίπεδα πυλών.

Το 1ο αφορά την παραγωγή των τριών λογικών γινομένων 2 μεταβλητών που περιλαμβάνονται στη συνάρτηση και την παραγωγή της συμπληρωματικής μορφής της μεταβλητής εισόδου C, το 2ο αφορά την παραγωγή των λογικών αθροισμάτων και το 3ο την παραγωγή του συνολικού λογικού γινομένου.



Λογικά κυκλώματα πολλών επιπέδων

Καταλήξαμε, λοιπόν, σε μια υλοποίηση η οποία περιλαμβάνει 7 λογικές πύλες, δηλαδή μία παραπάνω από το πλήθος των πυλών που απαιτεί η υλοποίηση δύο επιπέδων.

Ωστόσο, στο λογικό κύκλωμα, καμία λογική πύλη δε διαθέτει περισσότερες από δύο εισόδους, γεγονός που αποτελεί πλεονέκτημα.

Σε αρκετές περιπτώσεις, η χρήση λογικών κυκλωμάτων με περισσότερα από δύο επίπεδα πυλών μπορεί να οδηγήσει και σε μικρότερο πλήθος πυλών.

Για παράδειγμα, η παραγοντοποίηση μπορεί να μας προσφέρει τη δυνατότητα να χρησιμοποιήσουμε λογικές πύλες XOR ή XNOR, ώστε να μειώσουμε το κόστος υλοποίησης μίας λογικής συνάρτησης.

Επίσης, μία ή περισσότερες πύλες ή, ένα ή περισσότερα υποκυκλώματα, μπορούν να υλοποιούν λογικές εκφράσεις που χρησιμοποιούνται περισσότερες από μία φορές στο συνολικό λογικό κύκλωμα.

Λογικά κυκλώματα πολλών επιπέδων

Για την υλοποίηση σε 2 επίπεδα πυλών της συνάρτησης

$$F(x,y,z,w) = x'z'w' + yz'w + x'zw + yzw',$$

εάν εξαιρέσουμε τους αντιστροφείς παραγωγής των συμπληρωματικών μορφών των μεταβλητών εισόδου, απαιτούνται 4 πύλες AND 3 εισόδων για την παραγωγή των λογικών γινομένων και 1 πύλη OR 4 εισόδων για την παραγωγή του συνολικού λογικού αθροίσματος.

Για την υλοποίηση της ίδιας συνάρτησης μόνο με πύλες 2 εισόδων σε 4 επίπεδα πυλών, απαιτούνται (εκτός των αντιστροφέων) 8 πύλες AND και 4 πύλες OR, αφού κάθε πύλη 3 εισόδων υλοποιείται με 2 πύλες 2 εισόδων και η πύλη 4 εισόδων υλοποιείται με 3 πύλες 2 εισόδων.

Λογικά κυκλώματα πολλών επιπέδων

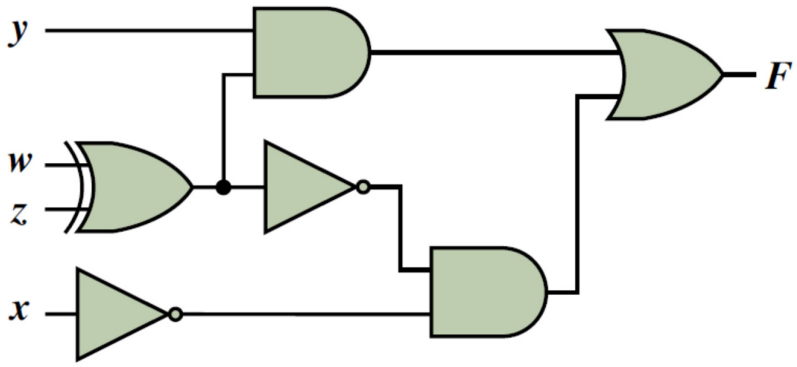
Ωστόσο, χρησιμοποιώντας τη μέθοδο της παραγοντοποίησης, μπορούμε να καταλήξουμε στην ακόλουθη ισοδύναμη λογική συνάρτηση:

$$F = x'z'w' + yz'w + x'zw + yzw'$$
$$= x'(z'w' + zw) + y(z'w + zw')$$
$$= x'(z \oplus w)' + y(z \oplus w)$$

Η υλοποίηση της F μπορεί τώρα να επιτευχθεί σε 4 επίπεδα πυλών, οι είσοδοι των συμμετεχουσών πυλών δεν είναι σε καμία περίπτωση περισσότερες από 2 και η έξοδος η πύλης XOR χρησιμοποιείται τόσο στην υλοποίηση της λογικής συνάρτησης αποκλειστικού OR, όσο και στην υλοποίηση της λογικής ισοδυναμίας των μεταβλητών z και w, με τη συνδρομή ενός αντιστροφέα.

Λογικά κυκλώματα πολλών επιπέδων

$$F = x'(z \oplus w)' + y(z \oplus w)$$



Το λογικό κύκλωμα περιλαμβάνει μικρότερο πλήθος πυλών από τις υλοποιήσεις διάταξης AND-OR.

Λογικά κυκλώματα πολλών επιπέδων

Μολονότι, η υλοποίηση λογικών συναρτήσεων με πολλά επίπεδα πυλών, μπορεί να περιορίζει το πλήθος των εισόδων των πυλών που χρησιμοποιούνται ή να προσφέρει τη δυνατότητα χρησιμοποίησης μικρότερου πλήθους λογικών πυλών, συχνά οδηγεί σε μεγαλύτερη καθυστέρηση απόκρισης των λογικών κυκλωμάτων, λόγω των πολλαπλών επιπέδων πυλών που παρεμβάλλονται μεταξύ των εισόδων και της εξόδου του λογικού κυκλώματος.

Καθυστέρηση απόκρισης μιας πύλης είναι η καθυστέρηση διάδοσης των δυαδικών σημάτων διαμέσου της πύλης.

Καθυστέρηση απόκρισης ενός λογικού κυκλώματος είναι το άθροισμα της καθυστέρησης απόκρισης του μέγιστου πλήθους πυλών που πρέπει να διέλθουν τα δυαδικά σήματα εισόδου για να φτάσουν στην έξοδο του κυκλώματος.

4. Ελαχιστοποίηση λογικών συναρτήσεων με τη μέθοδο του χάρτη Karnaugh και μερικώς καθορισμένες συναρτήσεις

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Ελαχιστοποίηση λογικών συναρτήσεων

Μια λογική συνάρτηση μπορεί να περιγραφεί με έναν και μοναδικό πίνακα αλήθειας, ενώ μπορεί να έχει πολλές διαφορετικές, αλλά ισοδύναμες, αλγεβρικές εκφράσεις.

Συνεπώς, μπορούμε να συνθέσουμε πολλά διαφορετικά λογικά κυκλώματα που υλοποιούν την ίδια λογική συνάρτηση, καθένα από τα οποία περιλαμβάνει διαφορετικούς τύπους πυλών και διαφορετικές διασυνδέσεις μεταξύ τους.

Η πολυπλοκότητα και το κόστος των λογικών κυκλωμάτων που υλοποιούν μια λογική συνάρτηση σχετίζεται άμεσα με την πολυπλοκότητα της αλγεβρικής έκφρασης της συνάρτησης αυτής.

Ελαχιστοποίηση λογικών συναρτήσεων

Κάποια λογικά κυκλώματα που υλοποιούν μια συνάρτηση είναι απλούστερα από άλλα και στόχος κατά τη σύνθεση ενός λογικού κυκλώματος είναι η μείωση του κόστους υλοποίησης.

Το κόστος υλοποίησης συνδέεται με το πλήθος των λογικών πυλών που χρησιμοποιούνται, καθώς και με το πλήθος των εισόδων των πυλών.

Επομένως, η οικονομικότερη υλοποίηση είναι η απλούστερη δυνατή, δηλαδή αυτή που συνίσταται από το μικρότερο δυνατό πλήθος πυλών με μικρότερο δυνατό πλήθος εισόδων ανά πύλη.

Κάθε υλοποίηση προκύπτει από μια αλγεβρική έκφραση της λογικής συνάρτησης και επομένως, η επιλογή της απλούστερης υλοποίησης αντιστοιχεί στην επιλογή της απλούστερης αλγεβρικής έκφρασης.

Γι’ αυτό είναι κρίσιμο, κατά τη διαδικασία της σύνθεσης, να επιδιώκεται ο προσδιορισμός της απλούστερης δυνατής αλγεβρικής έκφρασης της συνάρτησης.

Ελαχιστοποίηση λογικών συναρτήσεων

Η απλοποίηση αλγεβρικών εκφράσεων με τη χρήση αλγεβρικών μετασχηματισμών (δηλαδή των αξιωμάτων και των θεωρημάτων της άλγεβρας Boole) δεν οδηγεί πάντοτε με βεβαιότητα στην ελάχιστη δυνατή αλγεβρική έκφραση, λόγω του ότι οι αλγεβρικοί μετασχηματισμοί δεν μπορούν να συστήσουν συστηματική μεθοδολογία απλοποίησης.

Μια συστηματική γραφική μέθοδος για την ελαχιστοποίηση λογικών συναρτήσεων είναι η μέθοδος του χάρτη Karnaugh (Karnaugh map).

Η μέθοδος αυτή μας δίνει τη βέλτιστη απλοποίηση (ελαχιστοποίηση) μιας λογικής συνάρτησης σε επίπεδο βασικών λογικών πράξεων (AND, OR και NOT), που αντιστοιχεί σε υλοποίηση του κυκλώματος με τις αντίστοιχες βασικές πύλες.

Περαιτέρω ελαχιστοποίηση της συνάρτησης μπορεί να επιτευχθεί με αλγεβρικούς μετασχηματισμούς και χρήση παράγωγων λογικών πυλών (NAND, NOR, XOR, XNOR) και μπορεί να οδηγήσει σε απλούστερο κύκλωμα.

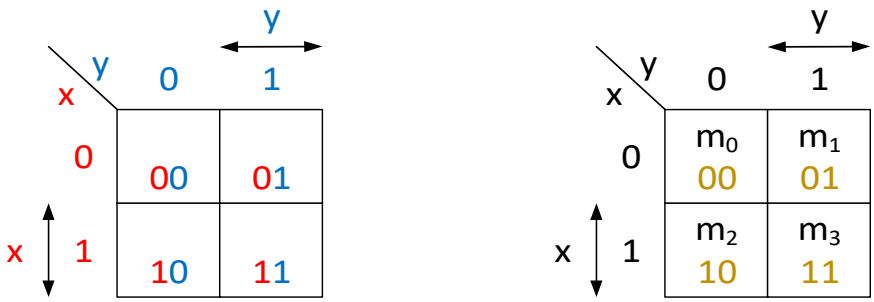
Χάρτης Karnaugh

Ο χάρτης Karnaugh μιας λογικής συνάρτησης αποτελεί απεικόνιση του πίνακα αλήθειας της συνάρτησης και είναι ισοδύναμος με αυτόν, δηλαδή από τον πίνακα αλήθειας μπορούμε να συμπληρώσουμε το χάρτη Karnaugh και αντιστρόφως.

Ο χάρτης Karnaugh αποτελείται από τετράγωνα, σε διάταξη γραμμών και στηλών, καθένα από τα οποία αντιστοιχεί σε ένα συνδυασμό τιμών των μεταβλητών του πίνακα αλήθειας και συνεπώς, σε ένα ελαχιστόρο της λογικής συνάρτησης.

Σε κάθε τετράγωνο του χάρτη Karnaugh μεταφέρουμε την τιμή που έχει η λογική συνάρτηση στον αντίστοιχο συνδυασμό τιμών των ανεξάρτητων μεταβλητών στον πίνακα αλήθειας.

Χάρτης Karnaugh συνάρτησης 2 μεταβλητών



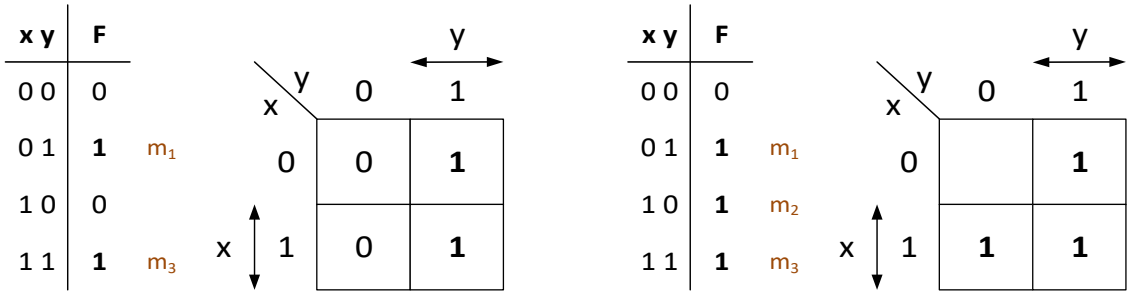
Η πρώτη γραμμή αντιστοιχεί στη συμπληρωματική τιμή της μεταβλητής **x** και η δεύτερη στην κανονική μορφή αυτής. Αντίστοιχα, η πρώτη στήλη του πίνακα αντιστοιχεί στη συμπληρωματική τιμή της μεταβλητής **y** και η δεύτερη στην κανονική μορφή αυτής.

Οι περιοχές του πίνακα στις οποίες κάθε μεταβλητή λαμβάνει λογική τιμή 1, υποδεικνύονται με βέλη συνδυαζόμενα με το όνομα της μεταβλητής.

Χάρτης Karnaugh συνάρτησης 2 μεταβλητών

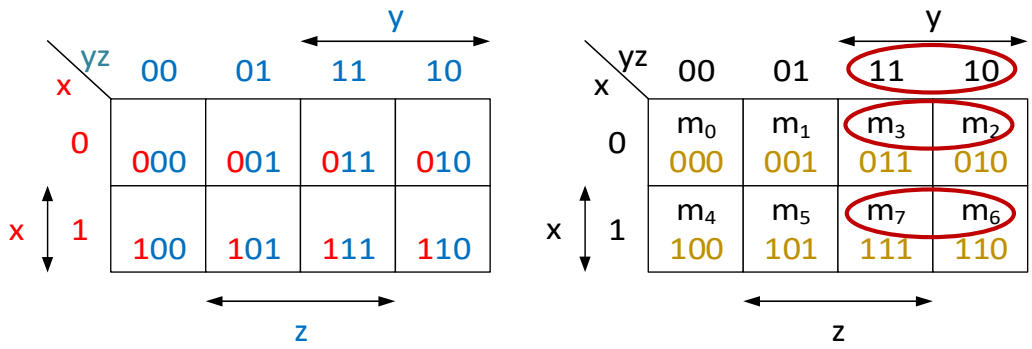
Οι τιμές (0 ή 1) που λαμβάνει μια λογική συνάρτηση για κάθε συνδυασμό τιμών των μεταβλητών του πίνακα αλήθειας, που αντιστοιχούν στους ελαχιστόρους που συμμετέχουν στη συνάρτηση, μεταφέρονται στα αντίστοιχα τετράγωνα του χάρτη Karnaugh.

Συνήθως, μεταφέρονται από τον πίνακα αλήθειας μόνο οι τιμές 1 της λογικής συνάρτησης και τα τετράγωνα του χάρτη που αντιστοιχούν στην τιμή 0 της συνάρτησης, αφήνονται κενά.



$F = \Sigma(1,3) = x'y + xy$ $F = \Sigma(1,2,3) = x'y + xy' + xy$

Χάρτης Karnaugh συνάρτησης 3 μεταβλητών



Ο χάρτης Karnaugh μιας συνάρτησης 3 μεταβλητών, αποτελείται από 8 τετράγωνα, αφού 3 μεταβλητές συνιστούν 8 ελαχιστόρους.

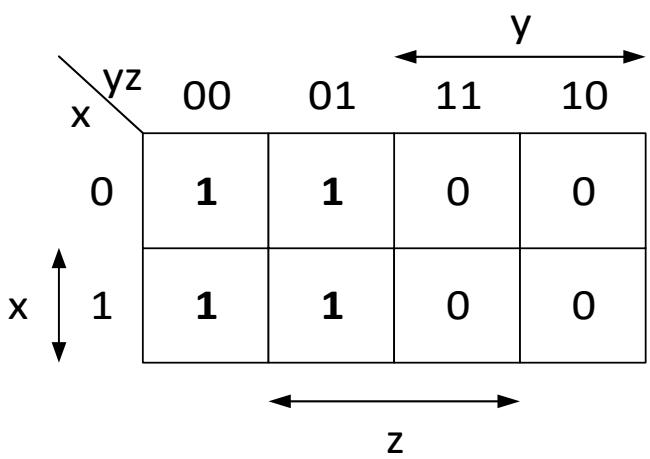
Οι λογικές τιμές των μεταβλητών **y** και **z**, στις οποίες αντιστοιχούν οι στήλες του χάρτη, διατάσσονται σύμφωνα με τον κώδικα Gray, στον οποίο 2 διαδοχικοί αριθμοί διαφέρουν μόνο σε 1 δυαδικό ψηφίο.

Έτσι, κατά τη μετάβαση από μία στήλη του χάρτη Karnaugh στη γειτονική της, παρουσιάζεται αλλαγή τιμής μόνο στη μία από τις δύο μεταβλητές.

Χάρτης Karnaugh συνάρτησης 3 μεταβλητών

Ο πίνακας αλήθειας και ο χάρτης Karnaugh της λογικής συνάρτησης F(x, y, z) = Σ(0, 1, 4, 5).

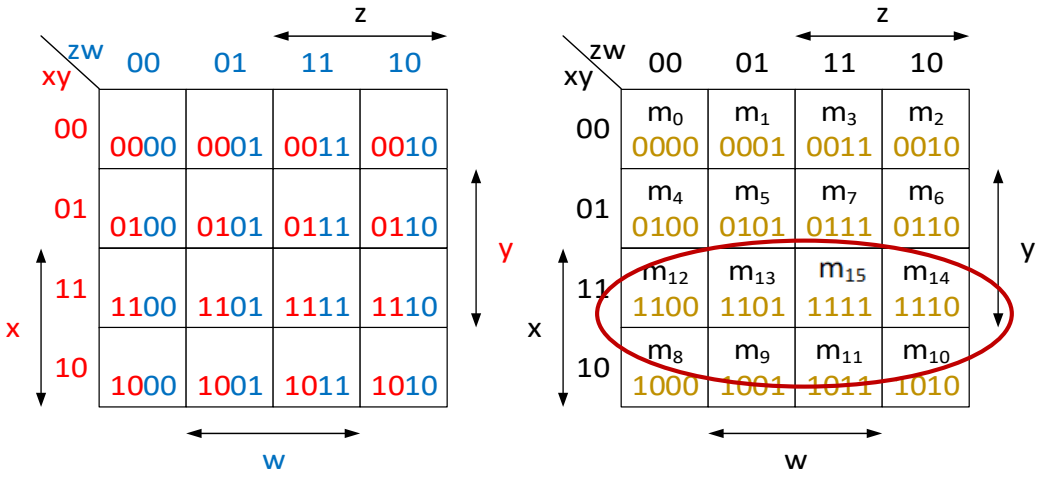
x y z	F	
0 0 0	**1**	m_0
0 0 1	**1**	m_1
0 1 0	0	
0 1 1	0	
1 0 0	**1**	m_4
1 0 1	**1**	m_5
1 1 0	0	
1 1 1	0	



Χάρτης Karnaugh συνάρτησης 4 μεταβλητών

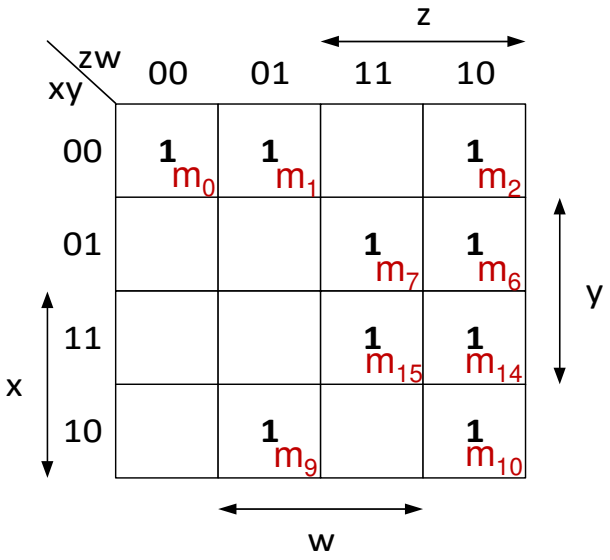
Ο χάρτης Karnaugh μιας συνάρτησης 4 μεταβλητών, αποτελείται από 16 τετράγωνα, αφού 4 μεταβλητές συνιστούν 16 ελαχιστόρους.

Οι λογικές τιμές των μεταβλητών **x** και **y**, στις οποίες αντιστοιχούν οι γραμμές του χάρτη, καθώς και οι λογικές τιμές των μεταβλητών **z** και **w**, στις οποίες αντιστοιχούν οι στήλες του χάρτη, διατάσσονται σύμφωνα με τον κώδικα Gray.



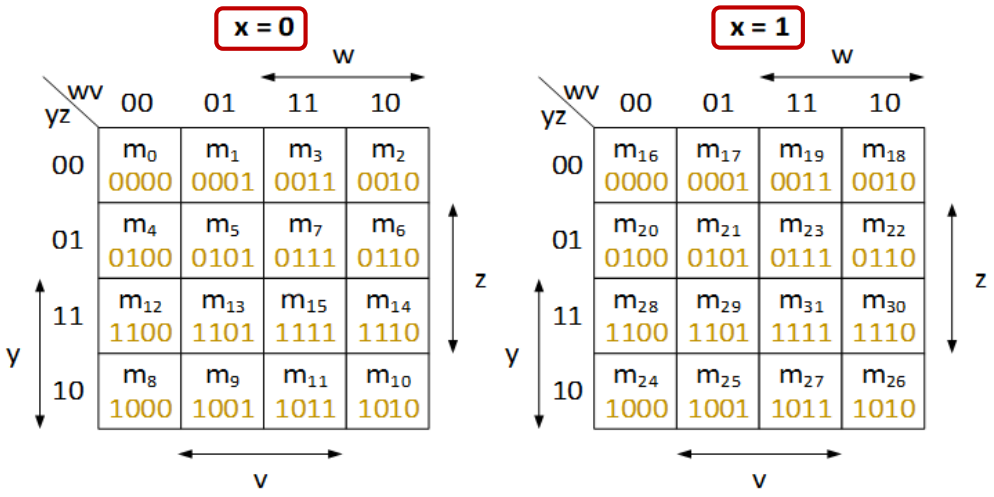
Χάρτης Karnaugh συνάρτησης 4 μεταβλητών

Χάρτης Karnaugh της λογικής συνάρτησης
F(x,y,z,w) = Σ(0, 1, 2, 6, 7, 9, 10, 14, 15)



Χάρτης Karnaugh συνάρτησης 5 μεταβλητών

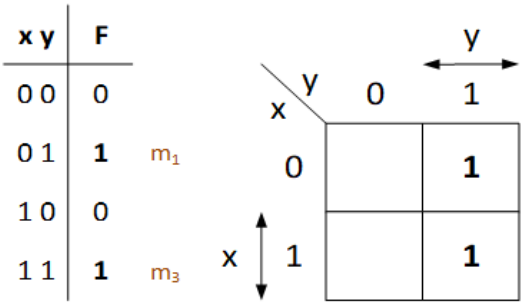
Για την περιγραφή μιας λογικής συνάρτησης 5 μεταβλητών, έστω F(x, y, z, w, v), χρησιμοποιούνται 2 χάρτες Karnaugh 4 μεταβλητών, στον πρώτο από τους οποίους σε μία από τις 5 μεταβλητές (π.χ. στη μεταβλητή x) τίθεται η λογική τιμή 0, ενώ στον δεύτερο χάρτη τίθεται στην ίδια μεταβλητή η τιμή 1, και αναπτύσσουμε 2 χάρτες με τις υπόλοιπες τέσσερις μεταβλητές.



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

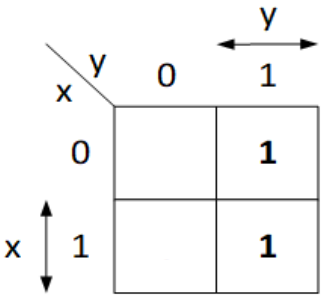
Δύο γειτονικά τετράγωνα του χάρτη (τετράγωνα με κοινή πλευρά), διαφέρουν κατά μία μεταβλητή, η οποία συμμετέχει με την κανονική της μορφή στον ελαχιστόρο που αντιστοιχεί στο ένα τετράγωνο και με τη συμπληρωματική της μορφή στον ελαχιστόρο που αντιστοιχεί στο άλλο τετράγωνο.

Για παράδειγμα, οι ελαχιστόροι m_1 = x'y και m_3 = xy, που αντιστοιχούν σε τετράγωνα με κοινή πλευρά, διαφέρουν στο ότι η μεταβλητή x συμμετέχει στον ελαχιστόρο m_1 με τη συμπληρωματική της μορφή και στον ελαχιστόρο m_3 με την κανονική της μορφή.



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

x y	F	
0 0	0	
0 1	**1**	m_1
1 0	0	
1 1	**1**	m_3



Εφαρμόζοντας το αξίωμα επιμεριστικότητας και το αξίωμα για το συμπλήρωμα, το λογικό άθροισμα των ελαχιστόρων $m_1 = x'y$, $m_3 = xy$ είναι:

$$m_1 + m_3 = x'y + xy = y(x' + x) = y.$$

Συμπεραίνουμε ότι κατά τη λογική άθροιση ελαχιστόρων που αντιστοιχούν σε γειτονικά τετράγωνα, δηλαδή σε ζεύγος τετραγώνων, απαλείφεται η μεταβλητή κατά την οποία αυτοί διαφέρουν.

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Δεν αποτελούν ζεύγη τα τετράγωνα που είναι διαγώνια μεταξύ τους, αφού στην περίπτωση αυτή αλλάζουν κατάσταση 2 μεταβλητές.

Στην περίπτωση που όλα τα τετράγωνα του χάρτη περιέχουν τιμή 1, τότε η λογική συνάρτηση ισούται με 1, ενώ, εάν όλα τα τετράγωνα περιέχουν τιμή 0, τότε η λογική συνάρτηση ισούται με 0.

Επομένως για την ελαχιστοποίηση μιας συνάρτησης 2 μεταβλητών, αρχικά μεταφέρουμε στον χάρτη Karnaugh τις τιμές της συνάρτησης (σημειώνουμε 1 στα τετράγωνα για τα οποία η συνάρτηση λαμβάνει τιμή 1 και αφήνουμε κενά τα υπόλοιπα για τα οποία η συνάρτηση έχει τιμή 0).

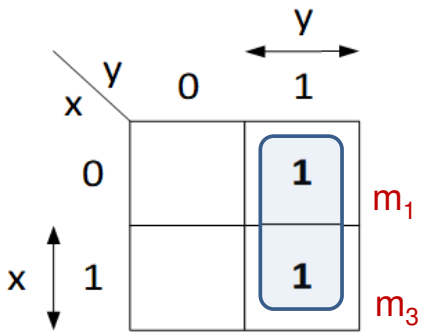
Επιλέγουμε στο χάρτη ζεύγη γειτονικών τετραγώνων με τιμή 1.

Εξάγουμε την ελαχιστοποιημένη συνάρτηση σε πρότυπη μορφή αθροίσματος γινομένων, λαμβάνοντας υπόψη ότι κάθε ζεύγος γειτονικών κυψελών οδηγεί στην απαλοιφή μιας μεταβλητής.

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 1:

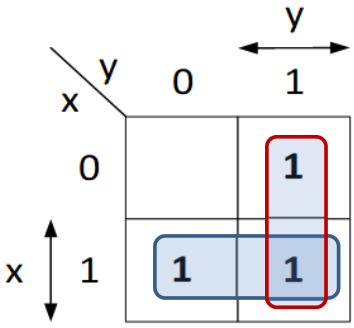
$F(x, y) = m_1 + m_3 = x'y + xy$



Επιλέγουμε στον χάρτη Karnaugh ζεύγη γειτονικών τετραγώνων που περιέχουν τιμή 1 και εξάγουμε την ελαχιστοποιημένη μορφή της συνάρτησης σε πρότυπη μορφή αθροίσματος γινομένων, λαμβάνοντας υπόψη ότι κάθε ζεύγος γειτονικών τετραγώνων οδηγεί στην απαλοιφή μιας μεταβλητής και, συγκεκριμένα, αυτής που στο ένα τετράγωνο είναι σε κανονική μορφή και στο γειτονικό τετράγωνο είναι σε συμπληρωματική μορφή. Επομένως: $F(x, y) = y$.

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 2: $F(x, y) = x'y + xy' + xy = \Sigma(1, 2, 3)$



Από το ζεύγος των τετραγώνων που αντιστοιχούν στους ελαχιστόρους m_2 και m_3 προκύπτει ότι το άθροισμα ελαχιστόρων $m_2 + m_3 = xy' + xy$ απλοποιείται στη μεταβλητή x.

Από το ζεύγος των τετραγώνων που αντιστοιχούν στους ελαχιστόρους m_1 και m_3 προκύπτει ότι το άθροισμα ελαχιστόρων $m_1 + m_3 = x'y + xy$ απλοποιείται στη μεταβλητή y. Επομένως: $F(x, y) = x + y$.

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Σε κάθε ομάδα (ζεύγος) μπορεί να ανήκουν τετράγωνα που περιλαμβάνουν μονάδα και ανήκουν και σε άλλη ομάδα, αφού σύμφωνα με το θεώρημα Θ1 της άλγεβρας Boole, ισχύει ότι A + A = A.

Η επιλογή αυτή ακολουθείται όταν η συμπερίληψη τετραγώνων που περιέχουν μονάδες σε περισσότερες από μία ομάδες, οδηγεί σε απαλοιφή περισσότερων μεταβλητών (δηλαδή, εκτενέστερη απλοποίηση).

Δεν ομαδοποιούνται τετράγωνα που περιέχουν μονάδες, τα οποία ανήκουν όλα σε άλλες ομάδες.

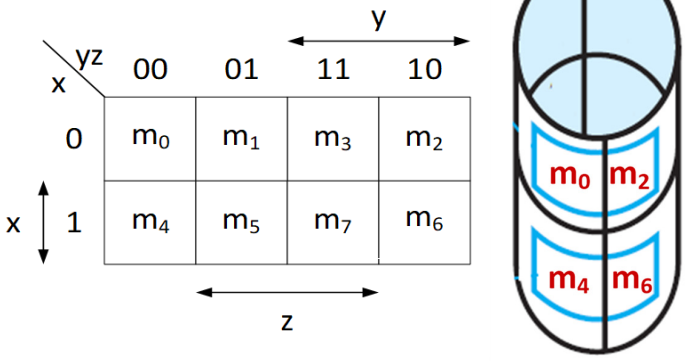
Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Στον χάρτη Karnaugh 3 μεταβλητών, για τους ελαχιστόρους που αντιστοιχούν σε 2 γειτονικά τετράγωνα (τετράγωνα με κοινή πλευρά), ισχύει ότι και στον χάρτη 2 μεταβλητών.

Εκτός των γειτονικών τετραγώνων, συναντώνται και άλλα ζεύγη τετραγώνων, των οποίων οι αντίστοιχοι ελαχιστόροι διαφέρουν κατά μία μόνο μεταβλητή.

Τέτοια ζεύγη τετραγώνων είναι εκείνα που αντιστοιχούν στους m_0, m_2 και στους m_4, m_6.

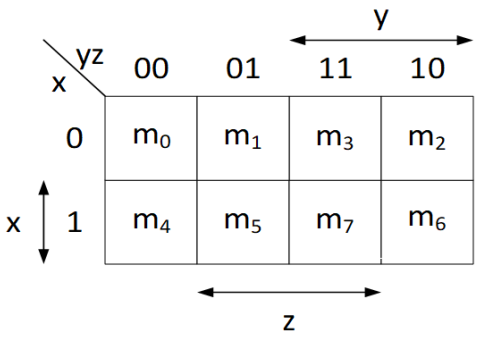
Τα τετράγωνα καθενός από τα ζεύγη αυτά λαμβάνονται ως γειτονικά, εάν θεωρήσουμε ότι η αριστερή και η δεξιά πλευρά του χάρτη εφάπτονται, σχηματίζοντας κύλινδρο.



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Τα λογικά γινόμενα που αντιστοιχούν σε γειτονικά ζεύγη τετραγώνων διαφέρουν κατά μία μόνο μεταβλητή, συνεπώς, το άθροισμα των ελαχιστόρων που αντιστοιχούν σε μία τέτοια τετράδα τετραγώνων έχει αποτέλεσμα έναν όρο μιας μόνο μεταβλητής.

Στην περίπτωση που περιέχονται μονάδες και στα 8 τετράγωνα του χάρτη, η λογική συνάρτηση ισούται με 1, αφού συμμετέχουν σε αυτήν όλοι οι ελαχιστόροι.



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Συμπερασματικά, για να ελαχιστοποιήσουμε μια λογική συνάρτηση 3 μεταβλητών μορφής αθροίσματος γινομένων:

α) σχηματίζουμε το χάρτη Karnaugh,

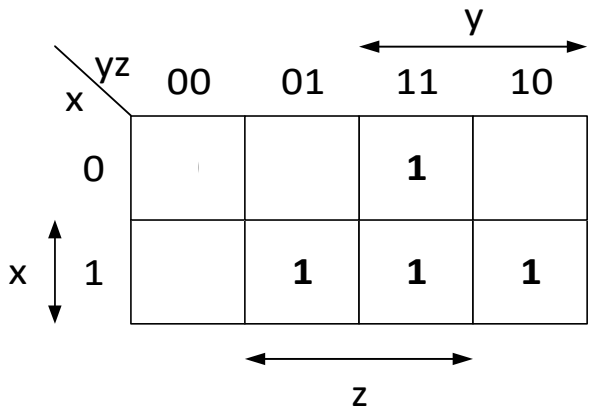
β) επιλέγουμε στο χάρτη ζεύγη γειτονικών τετραγώνων που περιέχουν μονάδες και, εάν υπάρχουν γειτονικά ζεύγη που περιέχουν μονάδες, επιλέγουμε την τετράδα τετραγώνων που αυτά σχηματίζουν,

γ) εξάγουμε την ελαχιστοποιημένη συνάρτηση σε μορφή αθροίσματος γινομένων, λαμβάνοντας υπόψη ότι κάθε ζεύγος γειτονικών τετραγώνων οδηγεί στην απαλοιφή μιας μεταβλητής και ότι κάθε τετράδα που αποτελείται από γειτονικά ζεύγη οδηγεί σε όρο μιας μεταβλητής.

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 1: $F(x, y, z) = \Sigma(3, 5, 6, 7) = x'yz + xy'z + xyz' + xyz$

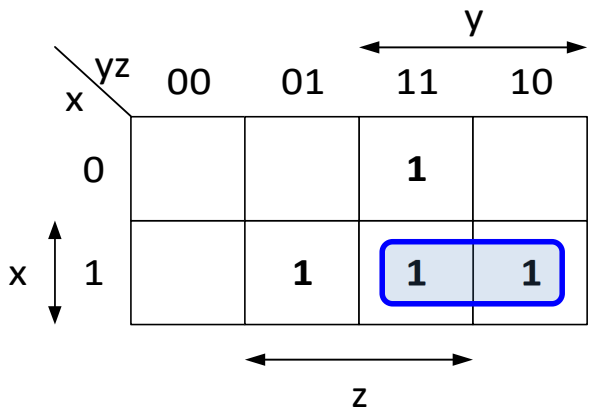
x	y	z	F(x, y, z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 1: $F(x, y, z) = \Sigma(3, 5, 6, 7) = x'yz + xy'z + xyz' + xyz$

x	y	z	F(x, y, z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

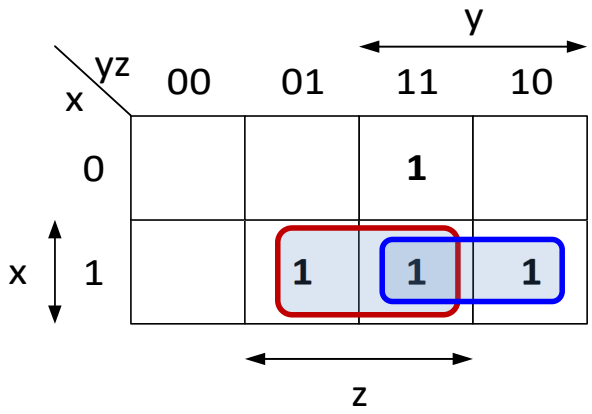


F = xy +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 1: F(x, y, z) = Σ(3, 5, 6, 7) = x'yz + xy'z + xyz' + xyz

x	y	z	F(x, y, z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

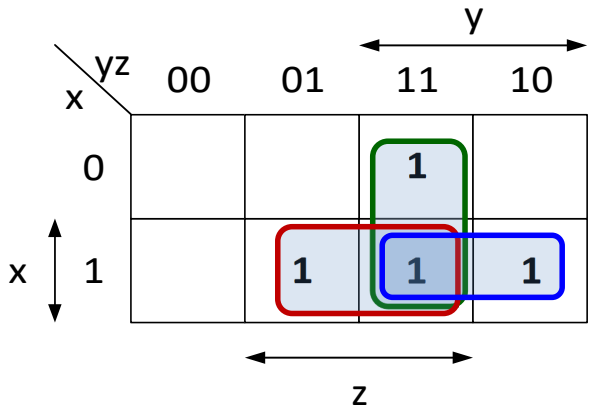


F = xy + xz +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 1: F(x, y, z) = Σ(3, 5, 6, 7) = x'yz + xy'z + xyz' + xyz

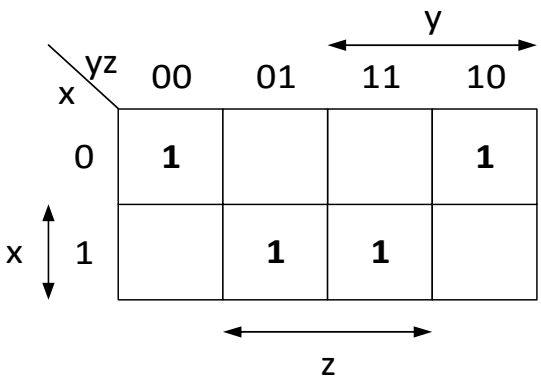
x	y	z	F(x, y, z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**



F = xy + xz + yz

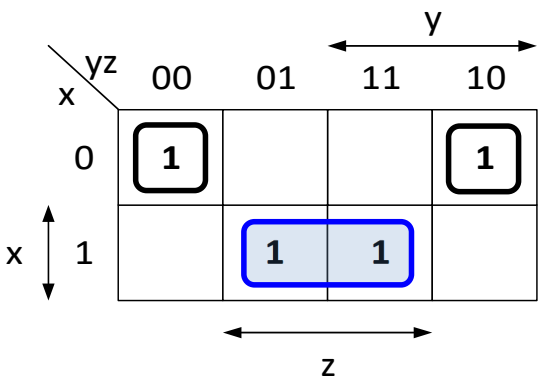
Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 2: F(x, y, z) = Σ(0, 2, 5, 7)



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

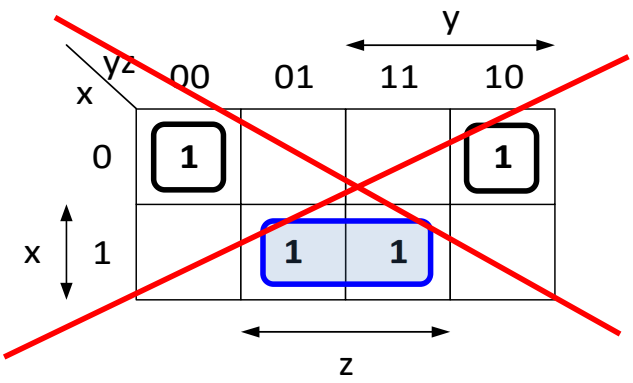
Παράδειγμα 2: F(x, y, z) = Σ(0, 2, 5, 7)



F = xz + x′y′z′ + x′yz′

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 2: F(x, y, z) = Σ(0, 2, 5, 7)

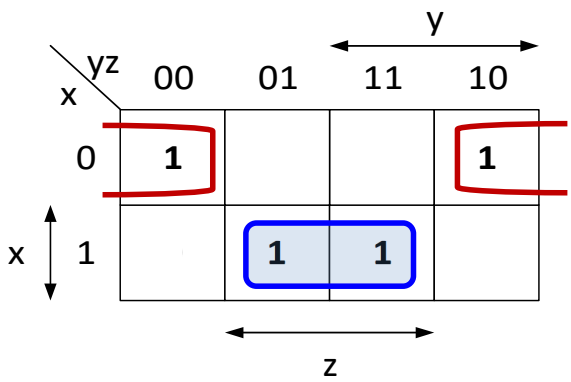


F = xz + x′y′z′ + x′yz′

Οι ελαχιστόροι που αντιστοιχούν στα τετράγωνα της πρώτης σειράς που περιέχουν μονάδες διαφέρουν κατά μία μόνο μεταβλητή.

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 2: F(x, y, z) = Σ(0, 2, 5, 7)



F = xz + x′z′

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Το πλήθος των μονάδων που ομαδοποιούνται αποτελεί δύναμη του 2:

$2^0 = 1$, δεν έχουμε ομαδοποίηση τετραγώνων και απαλοιφή μεταβλητής και ο αντίστοιχος ελαχιστόρος παραμένει ως είχε.

$2^1 = 2$, ένα ζεύγος τετραγώνων που περιέχουν μονάδες και οδηγεί στην απαλοιφή μιας μεταβλητής.

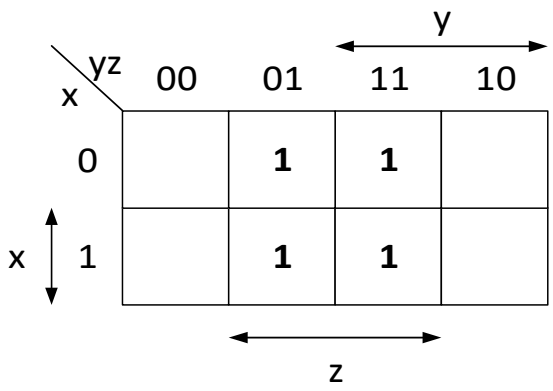
$2^2 = 4$, μια τετράδα τετραγώνων που περιέχουν μονάδες και οδηγεί στην απαλοιφή 2 μεταβλητών.

$2^3 = 8$, μια οκτάδα τετραγώνων που περιέχουν μονάδες και οδηγεί στην απαλοιφή 3 μεταβλητών, κ.ο.κ.

Επομένως, ο εκθέτης του 2, υποδεικνύει το πλήθος των μεταβλητών που απαλείφονται.

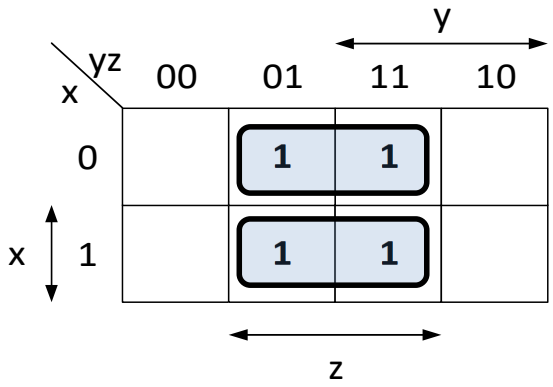
Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 3: F(x, y, z) = Σ(1, 3, 5, 7)



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

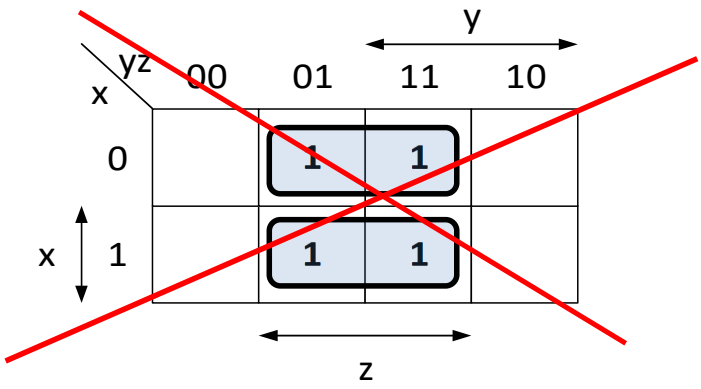
Παράδειγμα 3: F(x, y, z) = Σ(1, 3, 5, 7)



F = x'z + xz

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 3: F(x, y, z) = Σ(1, 3, 5, 7)

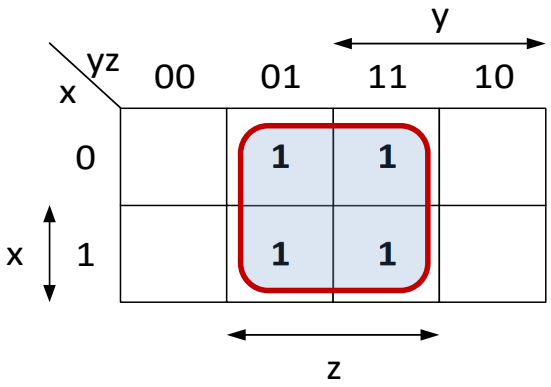


F = x'z + xz

Τα λογικά γινόμενα που αντιστοιχούν στα δύο ζεύγη τετραγώνων που περιέχουν μονάδες διαφέρουν σε μία μόνο μεταβλητή και σχηματίζουν τετράδα που οδηγεί συνολικά στην απαλοιφή 2 μεταβλητών

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

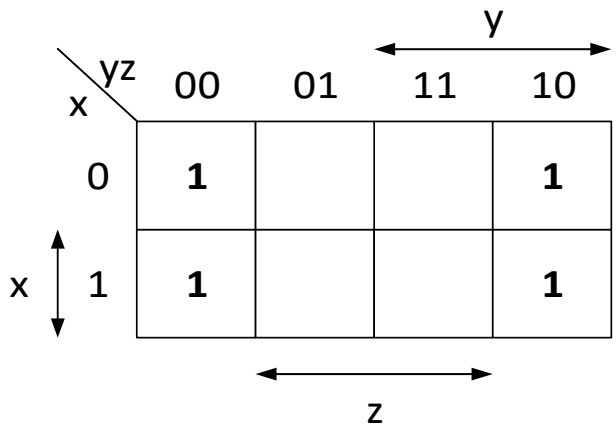
Παράδειγμα 3: F(x, y, z) = Σ(1, 3, 5, 7)



F = z

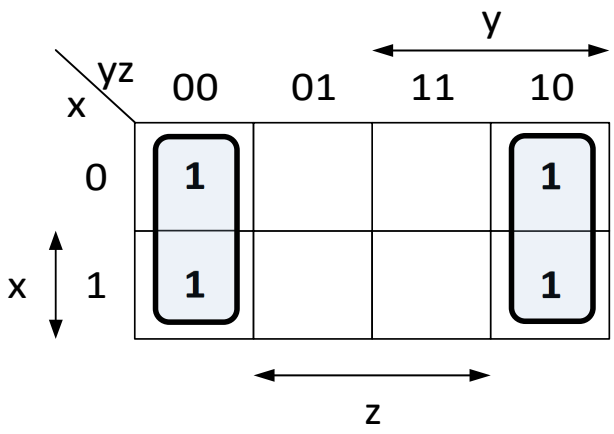
Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 4: F(x, y, z) = Σ(0, 2, 4, 6)



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

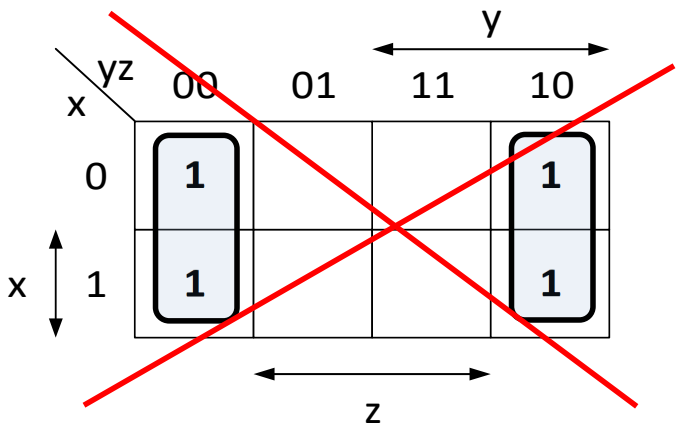
Παράδειγμα 4: F(x, y, z) = Σ(0, 2, 4, 6)



F = y'z' + yz'

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 4: F(x, y, z) = Σ(0, 2, 4, 6)

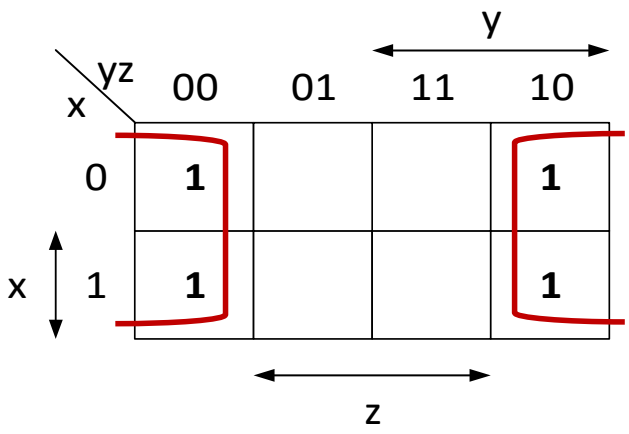


F = y'z' + yz'

Τα λογικά γινόμενα που αντιστοιχούν στα δύο ζεύγη τετραγώνων που περιέχουν μονάδες διαφέρουν σε μία μόνο μεταβλητή και σχηματίζουν τετράδα που οδηγεί συνολικά στην απαλοιφή 2 μεταβλητών

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

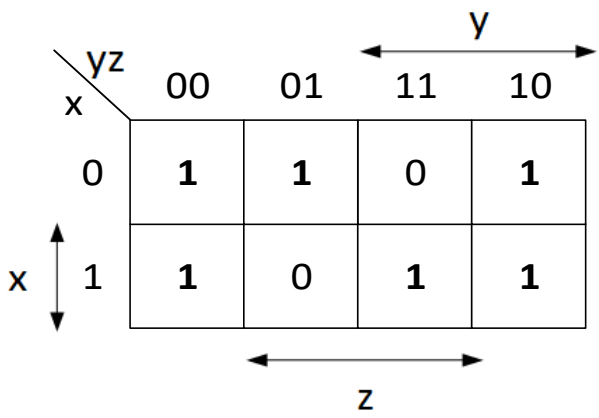
Παράδειγμα 4: F(x, y, z) = Σ(0, 2, 4, 6)



F = z′

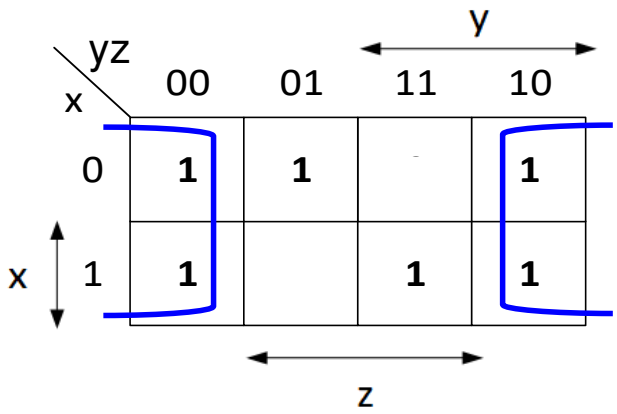
Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 5: F(x, y, z) = Σ(0, 1, 2, 4, 6, 7)



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

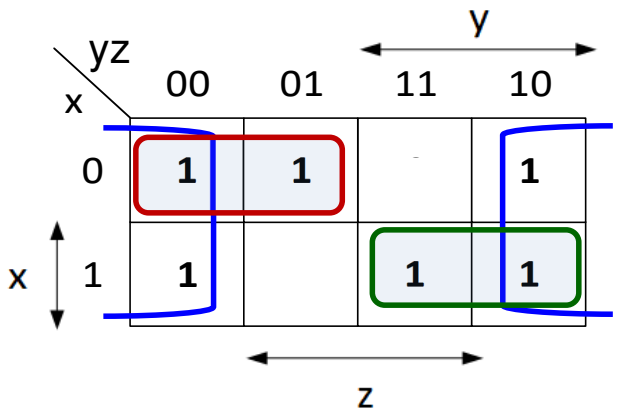
Παράδειγμα 5: F(x, y, z) = Σ(0, 1, 2, 4, 6, 7)



F = z’ +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 5: F(x, y, z) = Σ(0, 1, 2, 4, 6, 7)



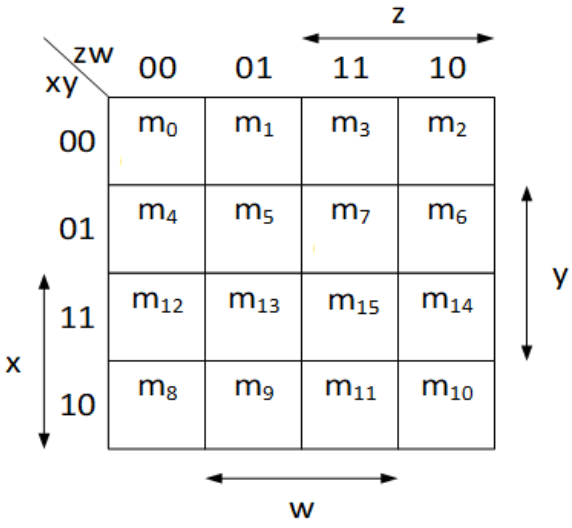
F = z’ + x’y’ + xy

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Στον χάρτη Karnaugh 4 μεταβλητών, το άθροισμα των ελαχιστόρων που αντιστοιχούν σε δύο γειτονικά τετράγωνα έχει ως αποτέλεσμα λογικό γινόμενο 3 μεταβλητών.

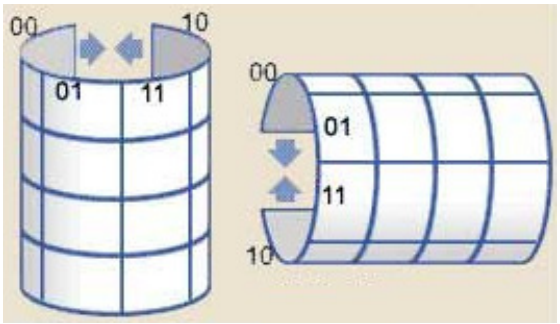
Τα λογικά γινόμενα που αντιστοιχούν σε γειτονικά ζεύγη τετραγώνων (τετράδα τετραγώνων) διαφέρουν κατά 2 μόνο μεταβλητές, με αποτέλεσμα το λογικό άθροισμα των γινομένων αυτών να απλοποιείται σε ένα γινόμενο 2 μεταβλητών.

Το άθροισμα των ελαχιστόρων που αντιστοιχούν σε μία οκτάδα τετραγώνων που σχηματίζεται από 2 τετράδες με τα παραπάνω χαρακτηριστικά, απλοποιείται σε έναν όρο μιας μεταβλητής.



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

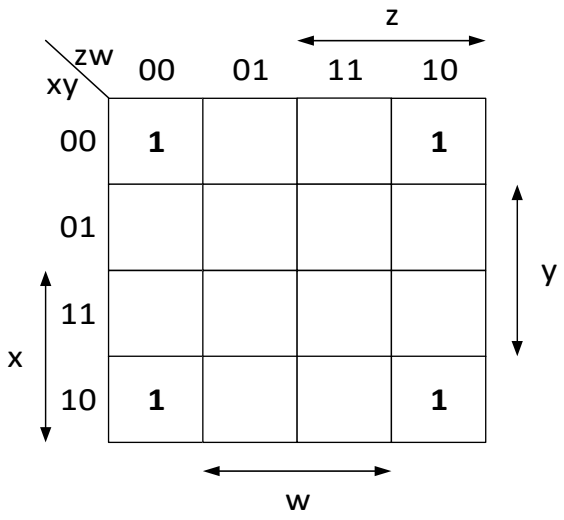
Οι ακραίες στήλες περιλαμβάνουν γειτονικά μεταξύ τους τετράγωνα, αφού οι ελαχιστόροι που αντιστοιχούν σε αυτά διαφέρουν κατά μία μόνο μεταβλητή. Το ίδιο συμβαίνει και με τις ακραίες γραμμές του χάρτη.



Τα τετράγωνα που βρίσκονται στις 4 γωνίες του χάρτη σχηματίζουν μία τετράδα τετραγώνων που αντιστοιχεί σε λογικό γινόμενο δύο μεταβλητών.

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 1: F(x, y, z, w) = Σ(0, 2, 8, 10)

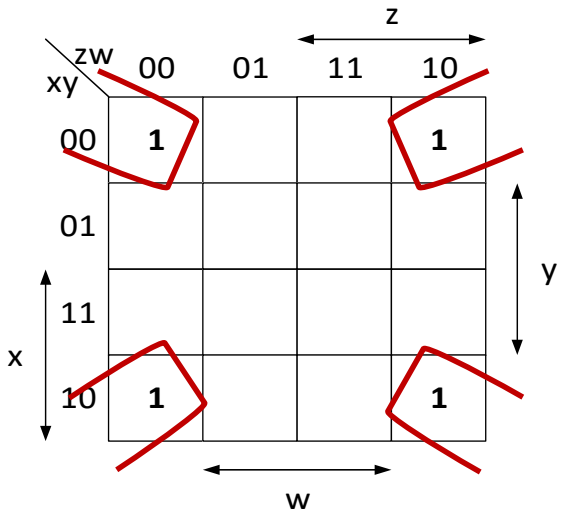


Τα τετράγωνα που βρίσκονται στις 4 γωνίες του χάρτη σχηματίζουν μία τετράδα τετραγώνων που αντιστοιχεί σε λογικό γινόμενο δύο μεταβλητών.

F(x, y, z, w) = x'y'z'w' + x'y'zw' + xy'z'w' + xy'zw' = x'y'w'(z' + z) + xy'w'(z' + z)
= x'y'w' + xy'w' = (x' + x)y'w' = y'w'

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

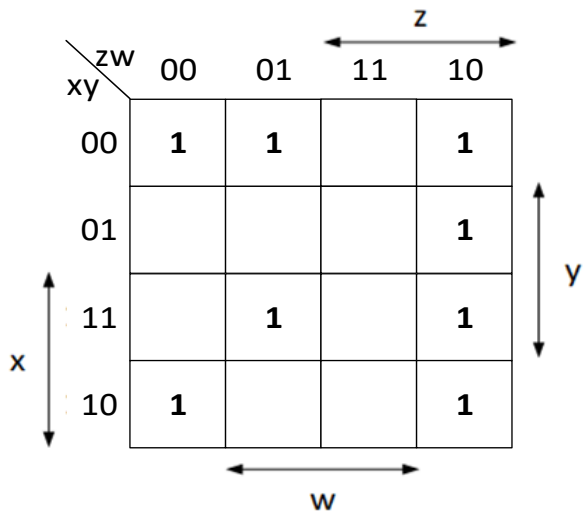
Παράδειγμα 1: F(x, y, z, w) = Σ(0, 2, 8, 10)



F(x, y, z, w) = y'w'

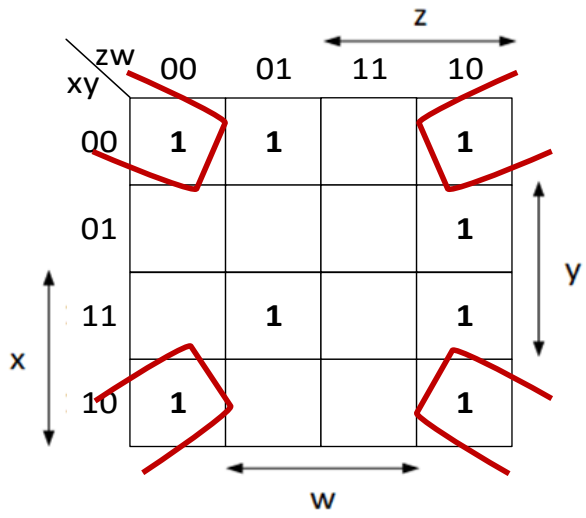
Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 2: F(x,y,z,w) = Σ(0, 1, 2, 6, 8, 10, 13, 14)



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

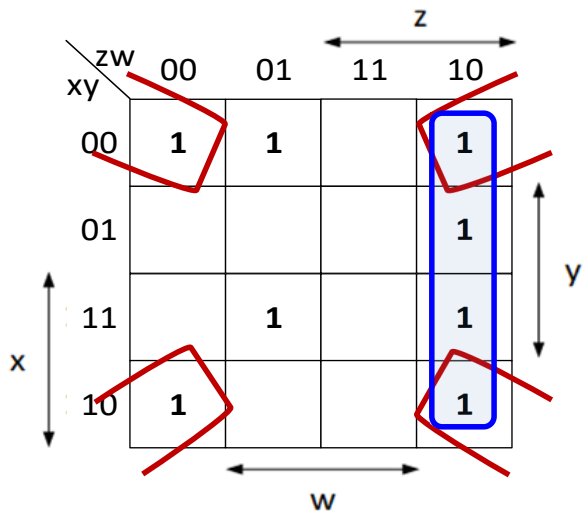
Παράδειγμα 2: F(x,y,z,w) = Σ(0, 1, 2, 6, 8, 10, 13, 14)



F = y'w' +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

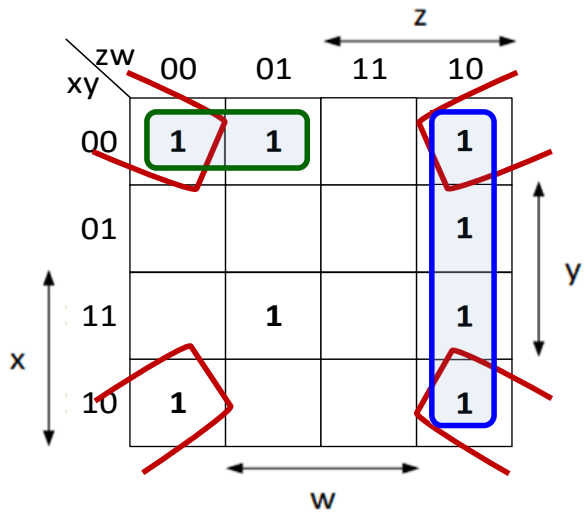
Παράδειγμα 2: F(x,y,z,w) = Σ(0, 1, 2, 6, 8, 10, 13, 14)



F = y'w' + zw' +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

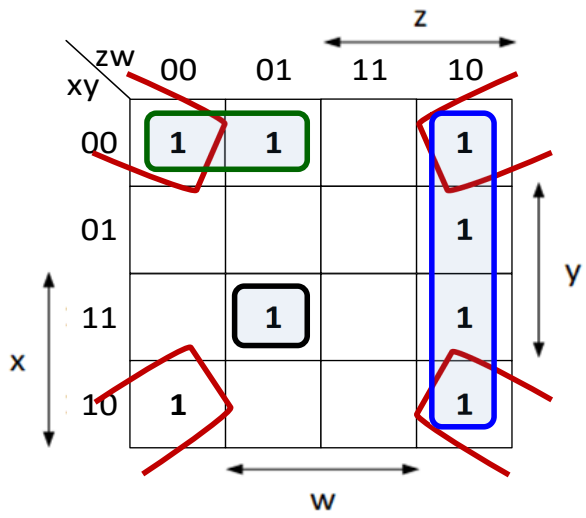
Παράδειγμα 2: F(x,y,z,w) = Σ(0, 1, 2, 6, 8, 10, 13, 14)



F = y'w' + zw' + x'y'z' +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

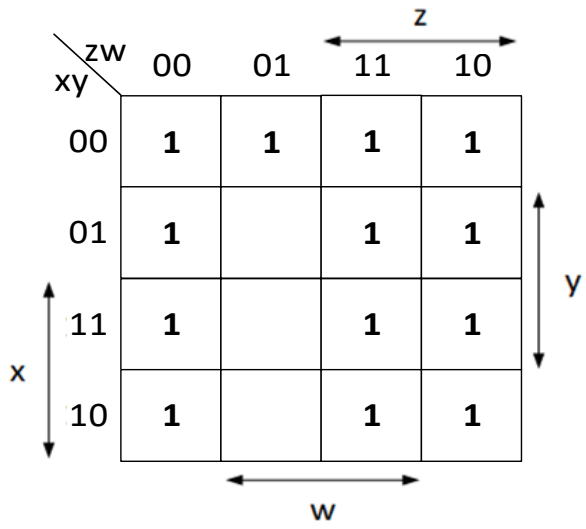
Παράδειγμα 2: F(x,y,z,w) = Σ(0, 1, 2, 6, 8, 10, 13, 14)



F = y'w' + zw' + x'y'z' + xyz'w

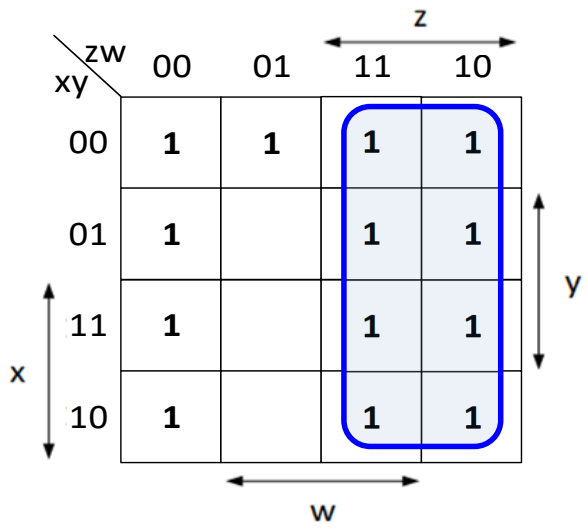
Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 3: F(x,y,z,w) = Σ(0, 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15)



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

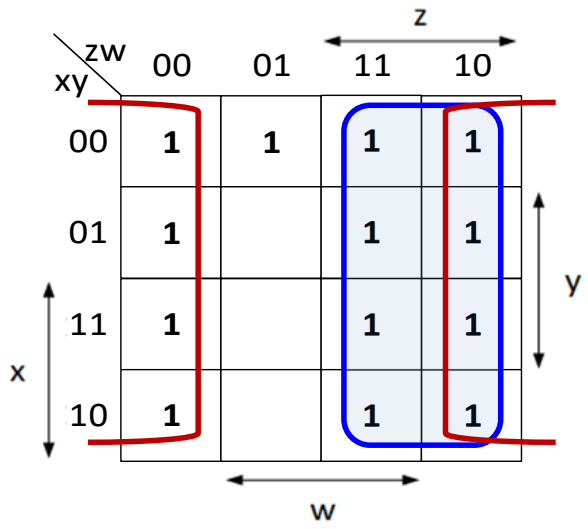
Παράδειγμα 3: F(x,y,z,w) = Σ(0, 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15)



F = z +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

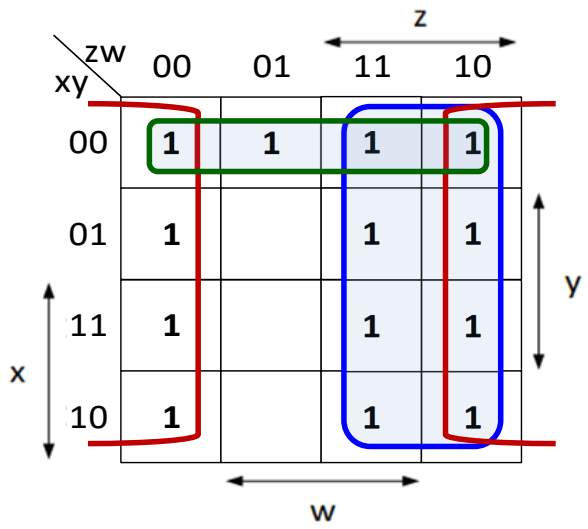
Παράδειγμα 3: F(x,y,z,w) = Σ(0, 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15)



F = z + w’ +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

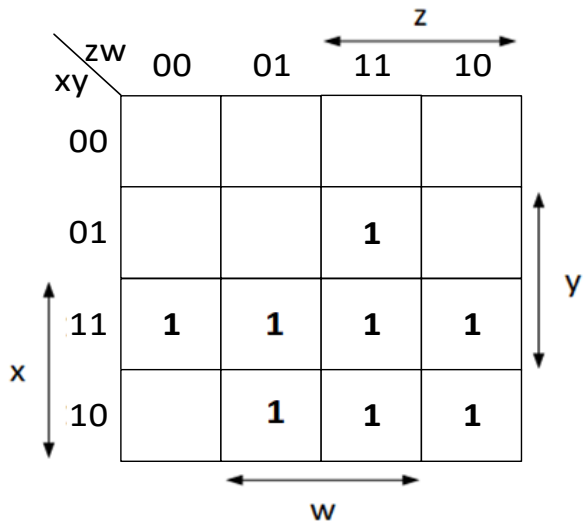
Παράδειγμα 3: F(x,y,z,w) = Σ(0, 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 14, 15)



F = z + w’ + x’y’

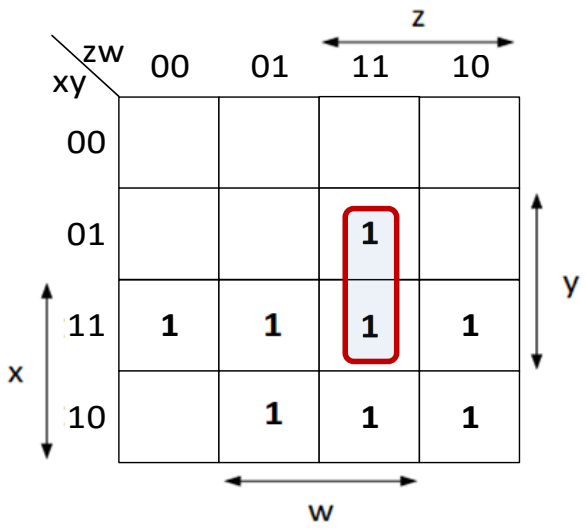
Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 4: F(x,y,z,w) = Σ(7, 9, 10, 11, 12, 13, 14, 15)



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

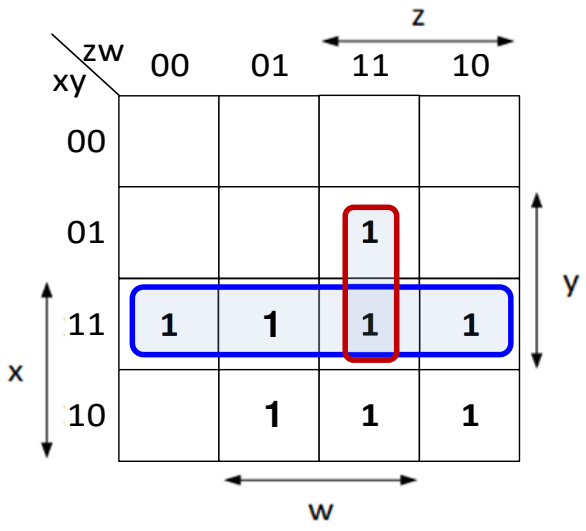
Παράδειγμα 4: F(x,y,z,w) = Σ(7, 9, 10, 11, 12, 13, 14, 15)



F = yzw +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

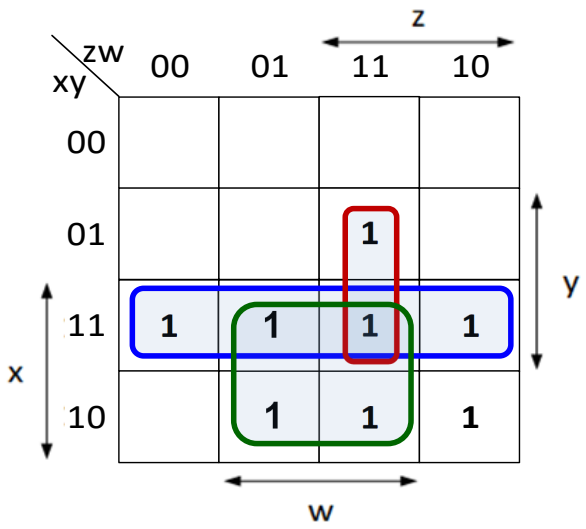
Παράδειγμα 4: F(x,y,z,w) = Σ(7, 9, 10, 11, 12, 13, 14, 15)



F = yzw + xy +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

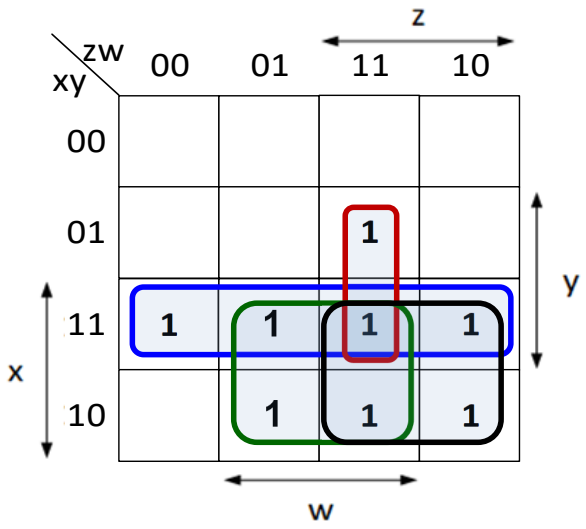
Παράδειγμα 4: F(x,y,z,w) = Σ(7, 9, 10, 11, 12, 13, 14, 15)



F = yzw + xy + xw +

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα 4: F(x,y,z,w) = Σ(7, 9, 10, 11, 12, 13, 14, 15)



F = yzw + xy + xw + xz

Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Όπως προαναφέρθηκε, για την περιγραφή συναρτήσεων 5 μεταβλητών χρησιμοποιούνται δύο χάρτες Karnaugh 4 μεταβλητών.

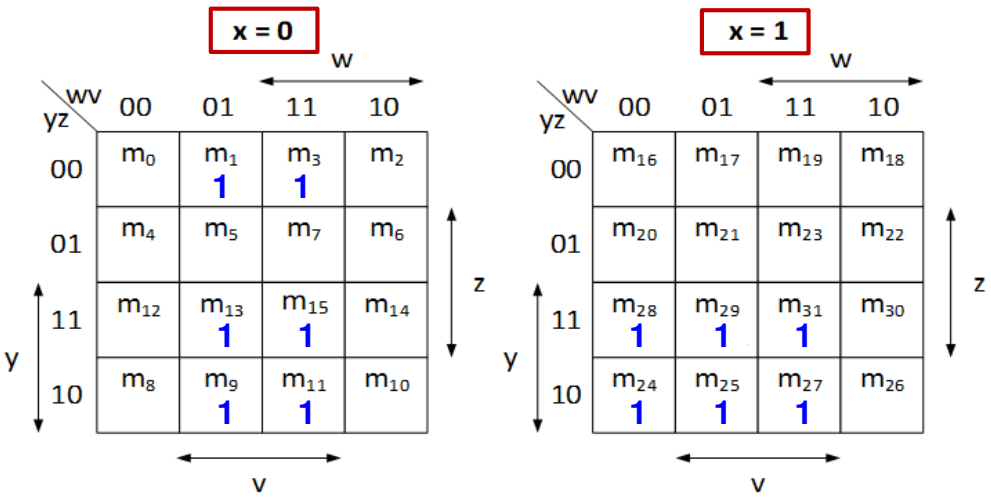
Για τον πρώτο χάρτη, σε μία από τις πέντε μεταβλητές τίθεται η λογική τιμή 0, ενώ για το δεύτερο χάρτη τίθεται στην ίδια μεταβλητή η λογική τιμή 1.

Για την ελαχιστοποίηση λογικών συναρτήσεων 5 μεταβλητών ακολουθούνται οι ίδιες αρχές με εκείνες που αναφέρθηκαν για τις συναρτήσεις 4 μεταβλητών.

Το νέο στοιχείο είναι ότι κάθε τετράγωνο του πρώτου χάρτη θεωρείται γειτονικό του αντίστοιχου τετραγώνου του δεύτερου χάρτη, αφού τα τετράγωνα αυτά διαφέρουν μόνο κατά τη μεταβλητή που λαμβάνει λογική τιμή 0 στον πρώτο χάρτη και λογική τιμή 1 στον δεύτερο χάρτη.

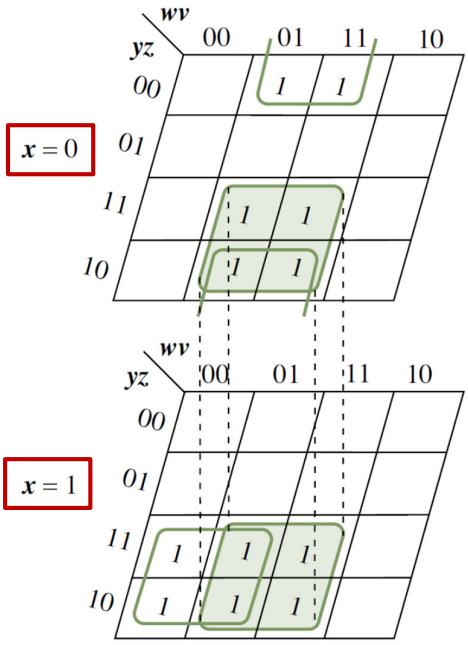
Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα: F(x,y,z,w,v) = Σ(1, 3, 9, 11, 13, 15, 24, 25, 27, 28, 29, 31)



Ελαχιστοποίηση συναρτήσεων με χάρτες Karnaugh

Παράδειγμα: F(x,y,z,w,v) = Σ(1, 3, 9, 11, 13, 15, 24, 25, 27, 28, 29, 31)



F = x′z′v + yv + xyw′

Βασικές οδηγίες για τη μέθοδο του χάρτη Karnaugh

Κάθε τετράγωνο του χάρτη Karnaugh που περιέχει μονάδα αντιστοιχεί σε έναν ελαχιστόρο της λογικής συνάρτησης.

Για κάθε τετράγωνο ενός χάρτη Karnaugh n μεταβλητών υπάρχουν n γειτονικά τετράγωνα και κάθε ζεύγος τετραγώνων αντιστοιχεί σε ελαχιστόρους που διαφέρουν κατά μία μόνο μεταβλητή.

Κατά την ελαχιστοποίηση μιας λογικής συνάρτησης, ο αριθμός των τετραγώνων που ομαδοποιείται είναι πάντα δύναμη του 2.

Η δημιουργία ζεύγους γειτονικών τετραγώνων που περιέχουν μονάδες οδηγεί στην απαλοιφή μιας μεταβλητής, η δημιουργία τετράδας τετραγώνων οδηγεί στην απαλοιφή δύο μεταβλητών και γενικότερα η ομαδοποίηση 2^n τετραγώνων οδηγεί στην απαλοιφή n μεταβλητών.

Βασικές οδηγίες για τη μέθοδο του χάρτη Karnaugh

Επιδιώκεται η δημιουργία ομάδων με το μεγαλύτερο δυνατό πλήθος τετραγώνων που περιέχουν μονάδες, έτσι ώστε η μορφή της συνάρτησης που θα προκύψει να περιλαμβάνει λογικά γινόμενα με όσο το δυνατόν λιγότερες μεταβλητές.

Επιδιώκεται η δημιουργία του μικρότερου δυνατού πλήθους ομάδων τετραγώνων, έτσι ώστε η μορφή της συνάρτησης που θα προκύψει να περιλαμβάνει όσο το δυνατόν λιγότερα λογικά γινόμενα.

Κάθε τετράγωνο που περιέχει μονάδα και αντιστοιχεί σε έναν ελαχιστόρο της συνάρτησης θα πρέπει να περιλαμβάνεται σε μία τουλάχιστον ομάδα.

Κάθε τετράγωνο που περιέχει μονάδα μπορεί να περιλαμβάνεται σε περισσότερες από μία ομάδες, εάν αυτό εξυπηρετεί τους στόχους του μικρότερου δυνατού πλήθους λογικών γινομένων και του μικρότερου δυνατού πλήθους μεταβλητών ανά λογικό γινόμενο.

Βασικές οδηγίες για τη μέθοδο του χάρτη Karnaugh

Ωστόσο, θα πρέπει να επιλέγονται ομάδες τετραγώνων με τις λιγότερες δυνατές επικαλύψεις.

Η ομαδοποίηση των τετραγώνων του χάρτη που περιέχουν μονάδες είναι προτιμότερο να ξεκινά με τα «μοναχικά» τετράγωνα (δηλαδή εκείνα που παρουσιάζουν περιορισμένη γειτνίαση με άλλα τετράγωνα που περιέχουν μονάδες).

Στη συνέχεια, η ομαδοποίηση επεκτείνεται στις περιοχές του χάρτη όπου υπάρχει συγκέντρωση τετραγώνων που περιέχουν μονάδες.

Υλοποίηση ελαχιστοποιημένων συναρτήσεων

Η ελαχιστοποιημένη μορφή μιας συνάρτησης που προκύπτει από κατάλληλη ομαδοποίηση των τετραγώνων του χάρτη Karnaugh που περιέχουν μονάδες, εξάγεται σε μορφή αθροίσματος γινομένων.

Έτσι, μπορούν εύκολα να προκύψουν λογικά κυκλώματα αποτελούμενα από δύο επίπεδα πυλών σε διάταξη AND-OR ή μόνο από πύλες NAND, που υλοποιούν την ελαχιστοποιημένη μορφή της συνάρτησης.

Υλοποίηση ελαχιστοποιημένων συναρτήσεων

Τα τετράγωνα του χάρτη που περιγράφει μια λογική συνάρτηση, τα οποία δεν περιέχουν μονάδες, αντιστοιχούν στους ελαχιστόρους που δεν περιλαμβάνονται στη συνάρτηση.

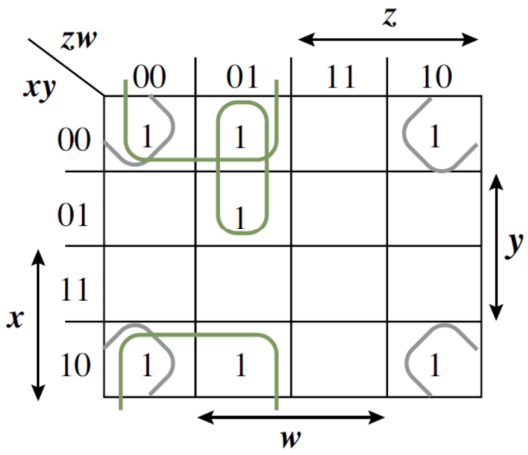
Το άθροισμα των ελαχιστόρων αυτών συνιστά τη συμπληρωματική συνάρτηση.

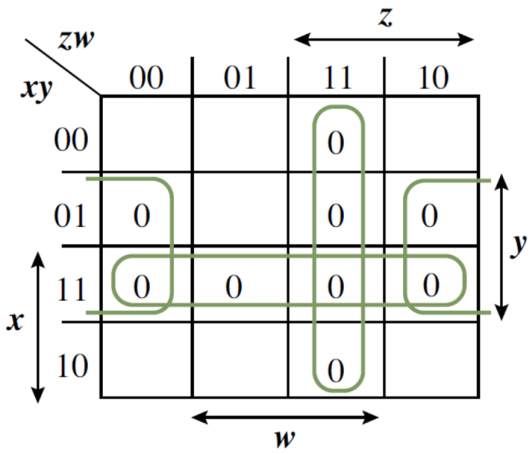
Εάν, λοιπόν, τοποθετήσουμε μηδενικά στα άδεια τετράγωνα του χάρτη και τα ομαδοποιήσουμε με βάση τη διαδικασία που παρουσιάστηκε προηγουμένως, μπορούμε να καταλήξουμε στην ελαχιστοποιημένη μορφή της συμπληρωματικής συνάρτησης, σε μορφή αθροίσματος γινομένων.

Στη συνέχεια, εάν εφαρμόσουμε σε αυτήν το θεώρημα De Morgan, μπορούμε να εξαγάγουμε την ελαχιστοποιημένη αρχική συνάρτηση σε μορφή γινομένου αθροισμάτων, ώστε να προκύψουν λογικά κυκλώματα με δύο επίπεδα πυλών, σε διάταξη OR-AND ή μόνο με πύλες NOR.

Υλοποίηση ελαχιστοποιημένων συναρτήσεων

Παράδειγμα: F(x,y,z,w) = Σ(0, 1, 2, 5, 8, 9, 10)





F(x,y,z,w) = y′z′ + y′w′ + x′z′w

F′(x,y,z,w) = xy + zw + yw′

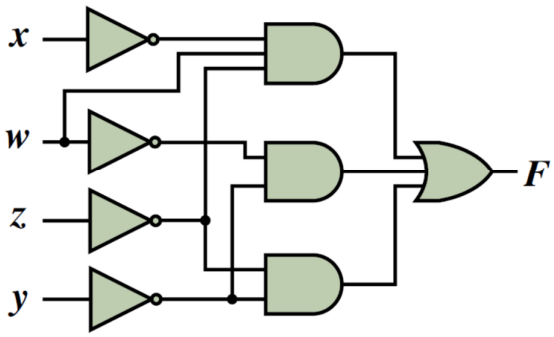
Υλοποίηση ελαχιστοποιημένων συναρτήσεων

Παράδειγμα: F(x,y,z,w) = Σ(0, 1, 2, 5, 8, 9, 10)

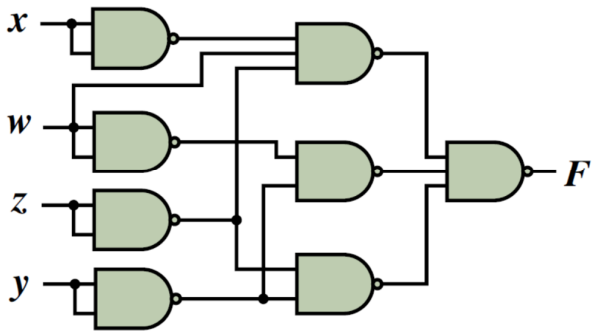
F(x,y,z,w) = y′z′ + y′w′ + x′z′w

Θεώρημα διπλής άρνησης & Θεώρημα De Morgan

⇒ F(x,y,z,w) = [(y′z′)′(y′w′)′(x′z′w)′]′



Υλοποίηση AND-OR



Υλοποίηση NAND-NAND

Υλοποίηση ελαχιστοποιημένων συναρτήσεων

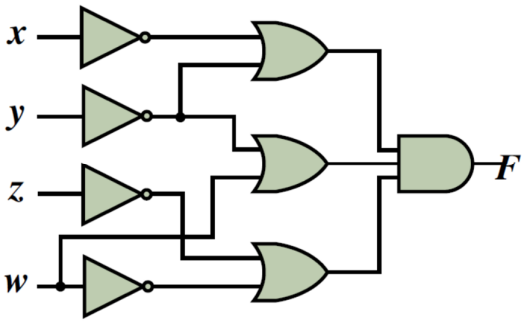
Παράδειγμα: F(x,y,z,w) = Σ(0, 1, 2, 5, 8, 9, 10)

F(x,y,z,w) = (xy + zw + yw')'

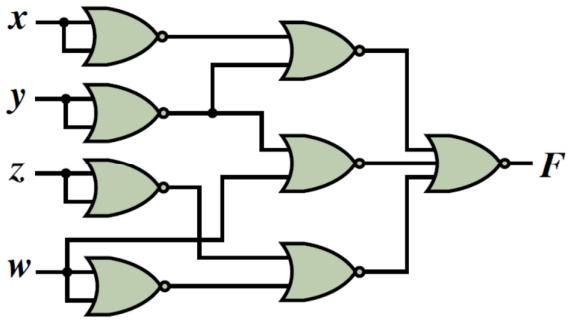
Θεώρημα De Morgan ⇒ F(x,y,z,w) = (x' + y')(z' +w')(y' + w)

Θεώρημα διπλής άρνησης & θεώρημα De Morgan ⇒

F(x,y,z,w) = [(x' + y')' + (z' + w')' + (y' + w)']'



Υλοποίηση OR-AND



Υλοποίηση NOR-NOR

Μερικώς καθορισμένες λογικές συναρτήσεις

Σε αρκετές περιπτώσεις λογικών κυκλωμάτων υπάρχουν συνδυασμοί εισόδων οι οποίοι δεν μπορούν να συμβούν ή δεν είναι επιτρεπτοί.

Για παράδειγμα, υποθέτουμε ότι 3 μεταβλητές x, y και z αντιστοιχούν στους ισάριθμους λαμπτήρες (κόκκινο, πορτοκαλί, πράσινο) ενός φαναριού κυκλοφορίας και ότι η καθεμία από αυτές λαμβάνει λογική τιμή 0 ή 1, όταν ο αντίστοιχος λαμπτήρας είναι σβηστός ή ανοιχτός, αντίστοιχα.

Οι επιτρεπτοί συνδυασμοί τιμών που μπορούν να λάβουν οι μεταβλητές αυτές είναι εκείνοι στους οποίους μόνο μία μεταβλητή έχει λογική τιμή 1 και οι υπόλοιπες δύο έχουν λογική τιμή 0, αφού για την ορθή λειτουργία του φαναριού, μόνο ένας λαμπτήρας μπορεί να είναι ανοιχτός.

Οι συνδυασμοί, δηλαδή, τιμών των μεταβλητών (x,y,z) = 000, 011, 101, 110 111 δεν επιτρέπονται ή δε χρησιμοποιούνται και αναφέρονται ως αδιάφορες λογικές συνθήκες (don't care logic conditions).

Μερικώς καθορισμένες λογικές συναρτήσεις

Ένα λογικό κύκλωμα με εισόδους τις μεταβλητές x, y και z μπορεί να σχεδιαστεί αγνοώντας τις καταστάσεις αυτές.

Οι ελαχιστόροι που αντιστοιχούν στις αδιάφορες λογικές συνθήκες, αναφέρονται ως αδιάφοροι όροι (don't care terms) και μια λογική συνάρτηση που περιλαμβάνει αδιάφορους όρους αναφέρεται ως μερικώς καθορισμένη (incompletely specified).

Οι αδιάφοροι όροι μπορούν να χρησιμοποιηθούν, ώστε να προκύψουν λογικά κυκλώματα με μικρότερο αριθμό πυλών.

Η λογική τιμή των συναρτήσεων που αντιστοιχεί στους αδιάφορους όρους μπορεί να θεωρηθεί ότι είναι 0 ή 1, ανάλογα με το ποια από τις δύο λογικές τιμές είναι κατάλληλη για ευρύτερη απλοποίηση της λογικής συνάρτησης.

Μερικώς καθορισμένες λογικές συναρτήσεις

Στον πίνακα αλήθειας ή στο χάρτη Karnaugh που περιγράφουν μια λογική συνάρτηση, για να διακρίνουμε έναν αδιάφορο όρο από τις τιμές της συνάρτησης που αντιστοιχούν σε επιτρεπτούς συνδυασμούς εισόδων, χρησιμοποιούμε το σύμβολο ×.

Κατά την ομαδοποίηση γειτονικών τετραγώνων του χάρτη Karnaugh, μπορούμε να επιλέξουμε να χρησιμοποιήσουμε για κάθε αδιάφορο όρο (×) λογική τιμή 0 ή 1, ανάλογα με το ποια από τις δύο τιμές του μπορεί να μας οδηγήσει στην απλούστερη μορφή της συνάρτησης.

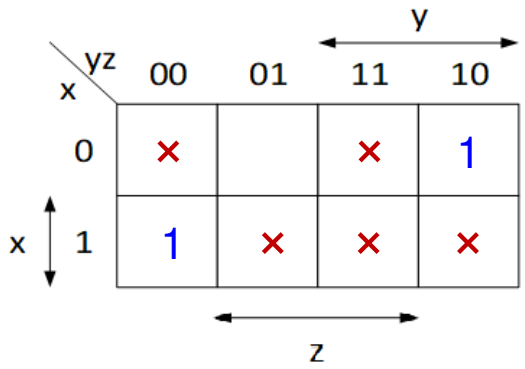
Στην έκφραση μιας λογικής συνάρτησης που περιλαμβάνει αδιάφορους όρους, θα πρέπει να περιλαμβάνονται και οι αδιάφοροι όροι.

Στο προαναφερόμενο παράδειγμα, η συνάρτηση που δηλώνει ότι τα οχήματα πρέπει να σταματήσουν (κόκκινο, πορτοκαλί) $F(x,y,z) = \Sigma(2, 4) + d(0, 3, 5, 6, 7)$ περιλαμβάνει τους ελαχιστόρους m_2, m_4 και ως αδιάφορους όρους τα λογικά γινόμενα x'y'z', x'yz, xy'z, xyz', xyz.

Μερικώς καθορισμένες λογικές συναρτήσεις

Παράδειγμα: F(x,y,z) = Σ(2, 4) + d(0, 3, 5, 6, 7)

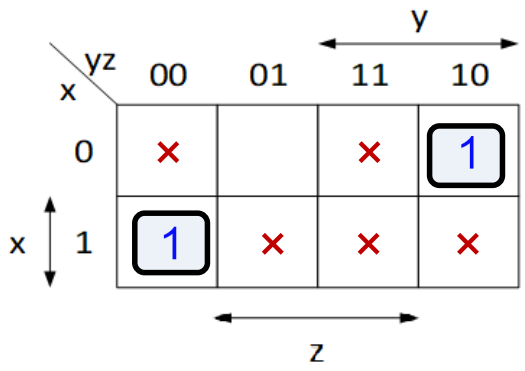
x	y	z	F(x, y, z)
0	0	0	×
0	0	1	0
0	1	0	1
0	1	1	×
1	0	0	1
1	0	1	×
1	1	0	×
1	1	1	×



Μερικώς καθορισμένες λογικές συναρτήσεις

Παράδειγμα: F(x,y,z) = Σ(2, 4) + d(0, 3, 5, 6, 7)

x	y	z	F(x, y, z)
0	0	0	×
0	0	1	0
0	1	0	1
0	1	1	×
1	0	0	1
1	0	1	×
1	1	0	×
1	1	1	×

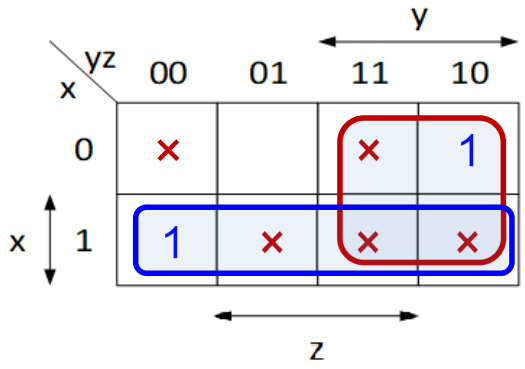


F(x,y,z) = xy′z′ + x′yz′

Μερικώς καθορισμένες λογικές συναρτήσεις

Παράδειγμα: F(x,y,z) = Σ(2, 4) + d(0, 3, 5, 6, 7)

x	y	z	F(x, y, z)
0	0	0	×
0	0	1	0
0	1	0	1
0	1	1	×
1	0	0	1
1	0	1	×
1	1	0	×
1	1	1	×



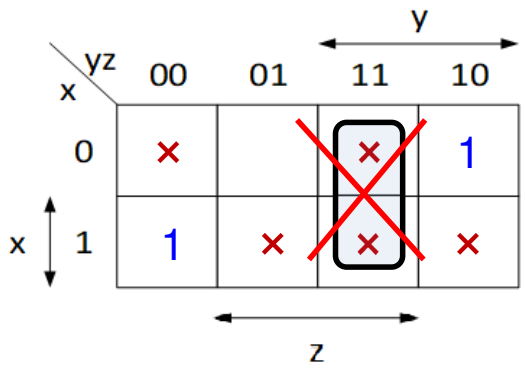
F(x,y,z) = x + y

Η αξιοποίηση των αδιάφορων όρων οδηγεί σε απλούστερη μορφή της λογικής συνάρτησης και επομένως σε οικονομικότερη υλοποίηση

Μερικώς καθορισμένες λογικές συναρτήσεις

Παράδειγμα: F(x,y,z) = Σ(2, 4) + d(0, 3, 5, 6, 7)

x	y	z	F(x, y, z)
0	0	0	×
0	0	1	0
0	1	0	1
0	1	1	×
1	0	0	1
1	0	1	×
1	1	0	×
1	1	1	×



Δεν σχηματίζουμε ομάδες μόνο με αδιάφορους όρους

5. Σύνθεση & ανάλυση συνδυαστικών κυκλωμάτων και τυποποιημένα συνδυαστικά κυκλώματα

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Συνδυαστικά κυκλώματα

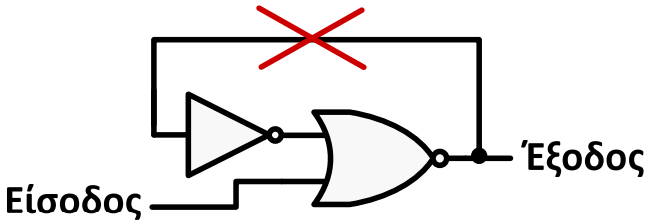
Τα λογικά κυκλώματα που έχουμε μελετήσει, αναφέρονται ως συνδυαστικά κυκλώματα (combinational circuits).

Ο χαρακτηρισμός αυτός αφορά λογικά κυκλώματα των οποίων η λογική τιμή της εξόδου ή των εξόδων, κάθε χρονική στιγμή, εξαρτάται μόνο από τη λογική τιμή των εισόδων που εφαρμόζεται σε αυτά την ίδια χρονική στιγμή.

Συνδυαστικά κυκλώματα

Οι λογικές πύλες που συνθέτουν ένα συνδυαστικό κύκλωμα διασυνδέονται με τέτοιο τρόπο, ώστε να μη δημιουργείται ανατροφοδότηση.

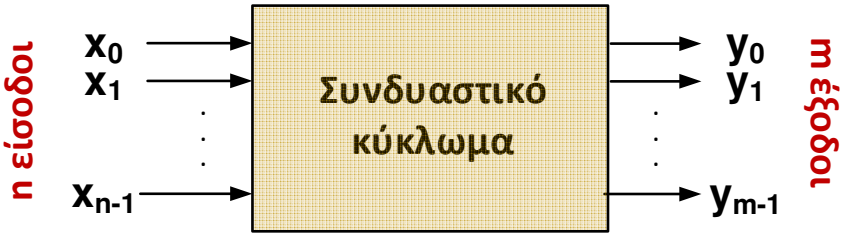
Με την ανατροφοδότηση, δημιουργείται εξάρτηση της εξόδου από λογικές τιμές που λαμβάνει η ανατροφοδοτούμενη είσοδος σε προηγούμενες χρονικές στιγμές, δηλαδή προσδίδεται μνήμη στο κύκλωμα, κάτι το οποίο είναι χαρακτηριστικό των ακολουθιακών κυκλωμάτων (sequential circuits).



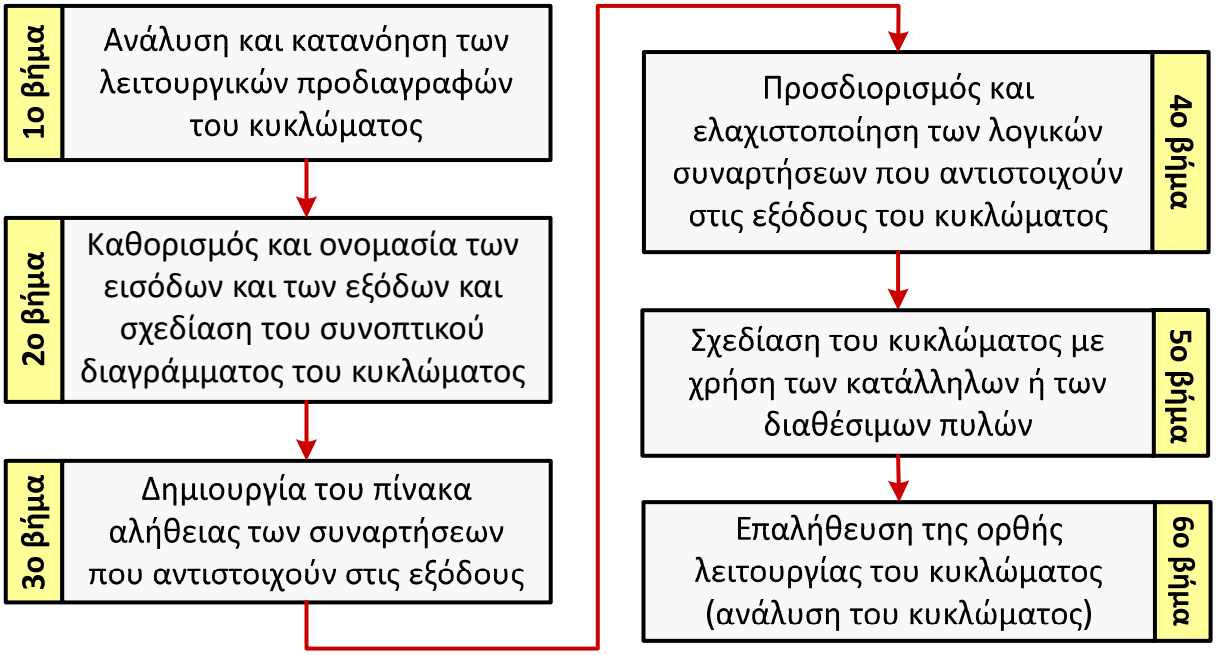
Συνδυαστικά κυκλώματα

Η λειτουργία ενός συνδυαστικού κυκλώματος μπορεί να περιγραφεί με μοναδικό τρόπο από έναν πίνακα αλήθειας με 2^n γραμμές (που αντιστοιχούν στους δυνατούς συνδυασμούς λογικών τιμών των μεταβλητών εισόδου) και n + m στήλες (που αντιστοιχούν στο πλήθος των μεταβλητών εισόδου και εξόδου).

Επίσης, η λειτουργία του μπορεί να περιγραφεί από m λογικές συναρτήσεις, μία για κάθε μεταβλητή εξόδου.



Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων



Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Παράδειγμα 1: Σύνθεση συνδυαστικού κυκλώματος που υλοποιεί τη συνάρτηση πλειοψηφίας 3 εισόδων α) με πύλες NOT, AND, OR και β) μόνο με πύλες NAND.

Ως συνάρτηση πλειοψηφίας 3 εισόδων ορίζεται η λογική συνάρτηση που αναγνωρίζει πότε μεταξύ 3 εισόδων πλειοψηφούν οι μονάδες έναντι των μηδενικών.

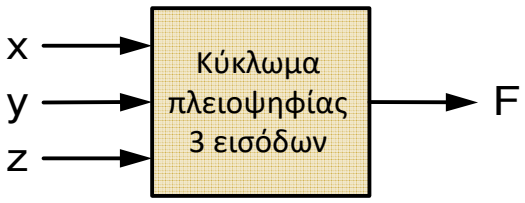
Από την περιγραφή της λειτουργίας του κυκλώματος είναι προφανές ότι το ζητούμενο συνδυαστικό κύκλωμα διαθέτει 3 εισόδους (x, y, z).

Επιπλέον, διαθέτει μία έξοδο (F), αφού το πλήθος των εισόδων είναι περιττός αριθμός (3) και επομένως θα έχουμε περισσότερες μονάδες ή περισσότερα μηδενικά (αποκλείεται να έχουμε ίδιο 1 και 0).

Η έξοδος λαμβάνει τιμή 1, όταν έχουμε περισσότερες μονάδες από μηδενικά, διαφορετικά λαμβάνει τιμή 0.

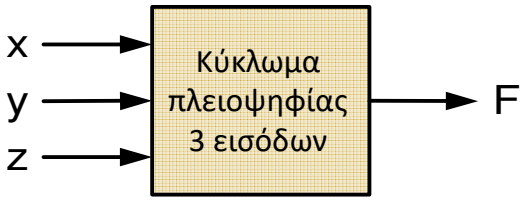
Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση συνοπτικού διαγράμματος:



Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση συνοπτικού διαγράμματος:

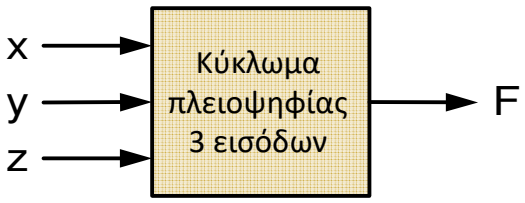


Πίνακας αλήθειας :

x	y	z	F(x, y, z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

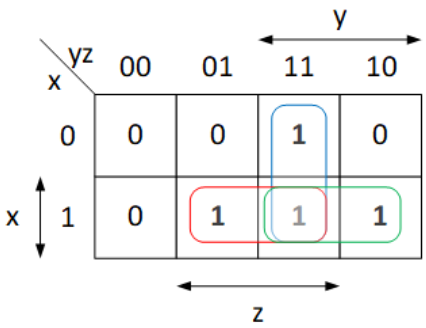
Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση συνοπτικού διαγράμματος:



Πίνακας αλήθειας και χάρτης Karnaugh:

x	y	z	F(x, y, z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

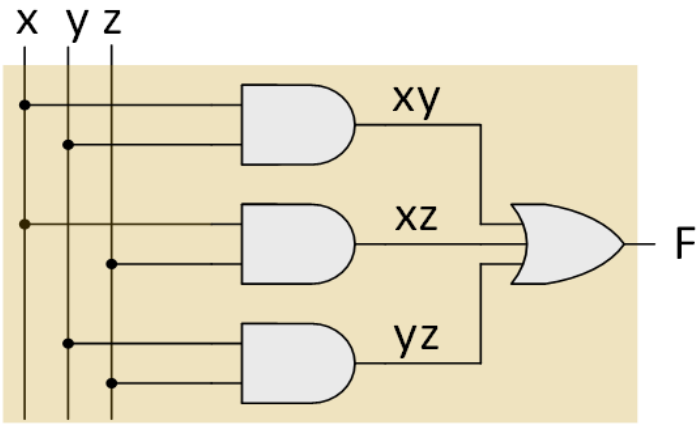


$F = \Sigma(3,5,6,7) = xy + xz + yz$

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

$F = \Sigma(3,5,6,7) = xy + xz + yz$

Σχεδίαση λογικού διαγράμματος του κυκλώματος με πύλες NOT, AND, OR:

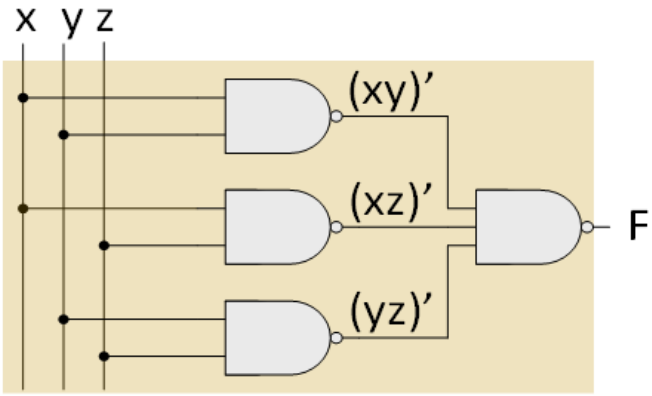


Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση λογικού διαγράμματος του κυκλώματος μόνο με πύλες NAND:

Εφαρμόζουμε κατά σειρά τα θεωρήματα διπλής άρνησης και De Morgan στην ελαχιστοποιημένη μορφή αθροίσματος γινομένων της συνάρτησης εξόδου:

$F = [(xy + xz + yz)']' = [(xy)'(xz)'(yz)']'$



Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Παράδειγμα 2: Σύνθεση συνδυαστικού κυκλώματος που να ανιχνεύει την εσφαλμένη λειτουργία ενός φωτεινού σηματοδότη κυκλοφορίας, με πύλες NOT, AND και OR.

Σε κανονική λειτουργία, μόνο μια από τις 3 λάμπες του σηματοδότη (πράσινη, πορτοκαλί, κόκκινη) είναι αναμμένη και οποιοσδήποτε άλλος συνδυασμός οδηγεί σε εσφαλμένη λειτουργία.

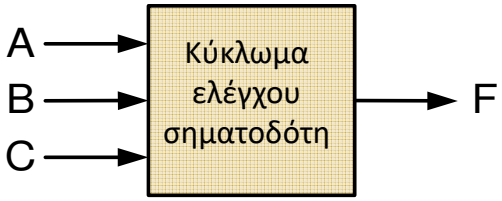
Από την παραπάνω ανάλυση της λειτουργίας του σηματοδότη προκύπτει ότι το συνδυαστικό κύκλωμα διαθέτει 3 εισόδους (A, B, C), μία από κάθε λάμπα (πράσινη, πορτοκαλί, κόκκινη) και μία έξοδο (F), η οποία όταν ανιχνεύεται εσφαλμένη λειτουργία λαμβάνει λογική τιμή 1, ενώ όταν η λειτουργία του σηματοδότη είναι η προβλεπόμενη, λαμβάνει λογική τιμή 0.

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση συνοπτικού διαγράμματος:

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση συνοπτικού διαγράμματος:



Πίνακας αλήθειας :

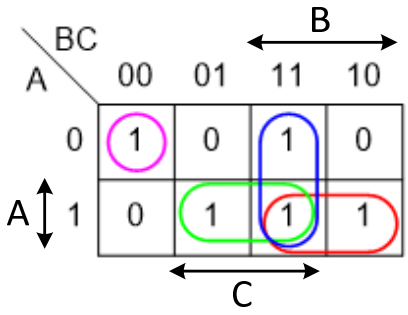
A	B	C	F
0	0	0	**1**
0	0	1	0
0	1	0	0
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση συνοπτικού διαγράμματος:

Πίνακας αλήθειας και χάρτης Karnaugh:

A	B	C	F
0	0	0	**1**
0	0	1	0
0	1	0	0
0	1	1	**1**
1	0	0	0
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

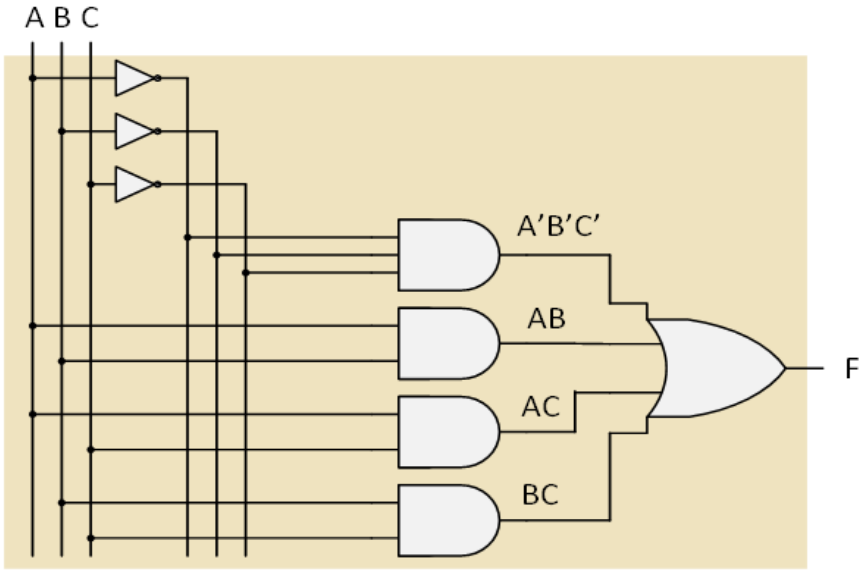


F = Σ(1,3,5,6,7) = A'B'C' + □B + □C + BC

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

F = Σ(1,3,5,6,7) = A'B'C' + □B + □C + BC

Σχεδίαση λογικού διαγράμματος του κυκλώματος με πύλες NOT, □ND, OR:



Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Παράδειγμα 3: Σύνθεση συνδυαστικού κυκλώματος για τη μετατροπή δυαδικά κωδικοποιημένων δεκαδικών ψηφίων (BCD) σε δυαδικά κωδικοποιημένα ψηφία σύμφωνα με τον κώδικα Gray, με χρήση του ελάχιστου δυνατού πλήθους λογικών πυλών.

Ο κώδικας BCD κωδικοποιεί με 4 δυαδικά ψηφία καθένα από τα 10 δεκαδικά ψηφία. Συνεπώς, το κύκλωμα διαθέτει 4 εισόδους (A_3, A_2, A_1, A_0, όπου $A_3A_2A_1A_0$ είναι το δυαδικά κωδικοποιημένο δεκαδικό ψηφίο).

Ο κώδικας Gray είναι κώδικας παράστασης αριθμών με δυαδικά ψηφία κατά τέτοιο τρόπο ώστε, κατά τη μετάβαση μεταξύ δύο διαδοχικών αριθμών, να αλλάζει μόνο ένα ψηφίο.

Το αντίστοιχο ψηφίο ενός δεκαδικού ψηφίου BCD σε κώδικα Gray συνίσταται από 4 ψηφία, επομένως το κύκλωμα διαθέτει 4 εξόδους (G_3, G_2, G_1, G_0, όπου $G_3G_2G_1G_0$ είναι το δεκαδικό ψηφίο κωδικοποιημένο σύμφωνα με τον κώδικα Gray).

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση συνοπτικού διαγράμματος:

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση συνοπτικού διαγράμματος:

Παρατηρούμε ότι οι τελευταίοι 6 συνδυασμοί των εισόδων δεν είναι επιτρεπτοί (αφού δεν αποτελούν δυαδικά κωδικοποιημένα δεκαδικά ψηφία BCD) και αποτελούν αδιάφορες λογικές συνθήκες. Οι ελαχιστόροι που αντιστοιχούν σε αυτούς αποτελούν αδιάφορους όρους και οι 4 συναρτήσεις εξόδου είναι μερικώς καθορισμένες.

Πίνακας αλήθειας:

A_3	A_2	A_1	A_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	×	×	×	×
1	0	1	1	×	×	×	×
1	1	0	0	×	×	×	×
1	1	0	1	×	×	×	×
1	1	1	0	×	×	×	×
1	1	1	1	×	×	×	×

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Λογικές συναρτήσεις των 4 εξόδων σε μορφή αθροίσματος ελαχιστόρων:

$$G_3 = \Sigma(8,9) + d(10,11,12,13,14,15)$$

$$G_2 = \Sigma(4,5,6,7,8,9) + d(10,11,12,13,14,15)$$

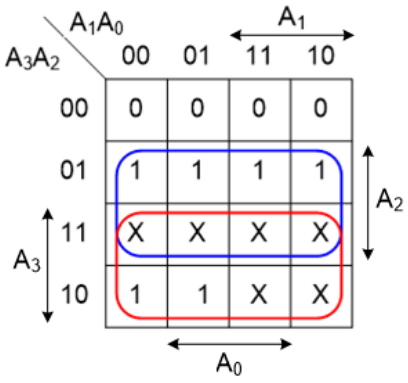
$$G_1 = \Sigma(2,3,4,5) + d(10,11,12,13,14,15)$$

$$G_0 = \Sigma(1,2,5,6,9) + d(10,11,12,13,14,15)$$

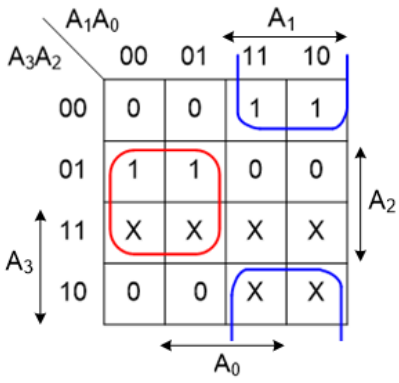
Από τον πίνακα αλήθειας μπορούμε εύκολα να συμπεράνουμε ότι $G_3 = A_3$, επομένως για τη συνάρτηση της εξόδου G_3 δεν απαιτείται να διενεργήσουμε ελαχιστοποίηση, ούτε να χρησιμοποιήσουμε λογικές πύλες για την υλοποίησή της.

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

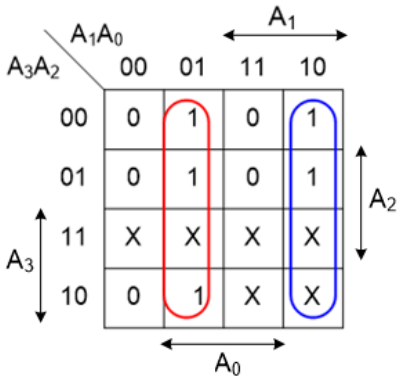
Ελαχιστοποίηση των λογικών συναρτήσεων των εξόδων G_2, G_1 και G_0 με χάρτες Karnaugh:



$G_2 = A_2 + A_3$



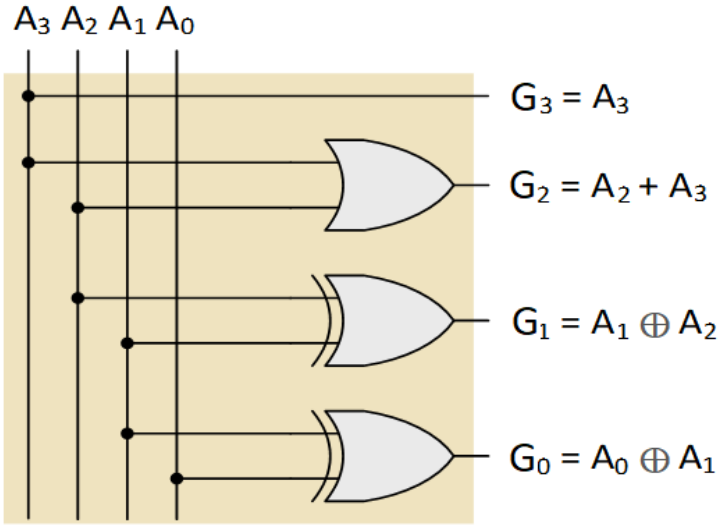
$G_1 = A'_1A_2 + A_1A'_2$
$= A_1 \oplus A_2$



$G_0 = A'_0A_1 + A_0A'_1$
$= A_0 \oplus A_1$

Σύνθεση (σχεδίαση) συνδυαστικών κυκλωμάτων

Σχεδίαση λογικού διαγράμματος με το ελάχιστο πλήθος λογικών πυλών:



Ανάλυση συνδυαστικών κυκλωμάτων

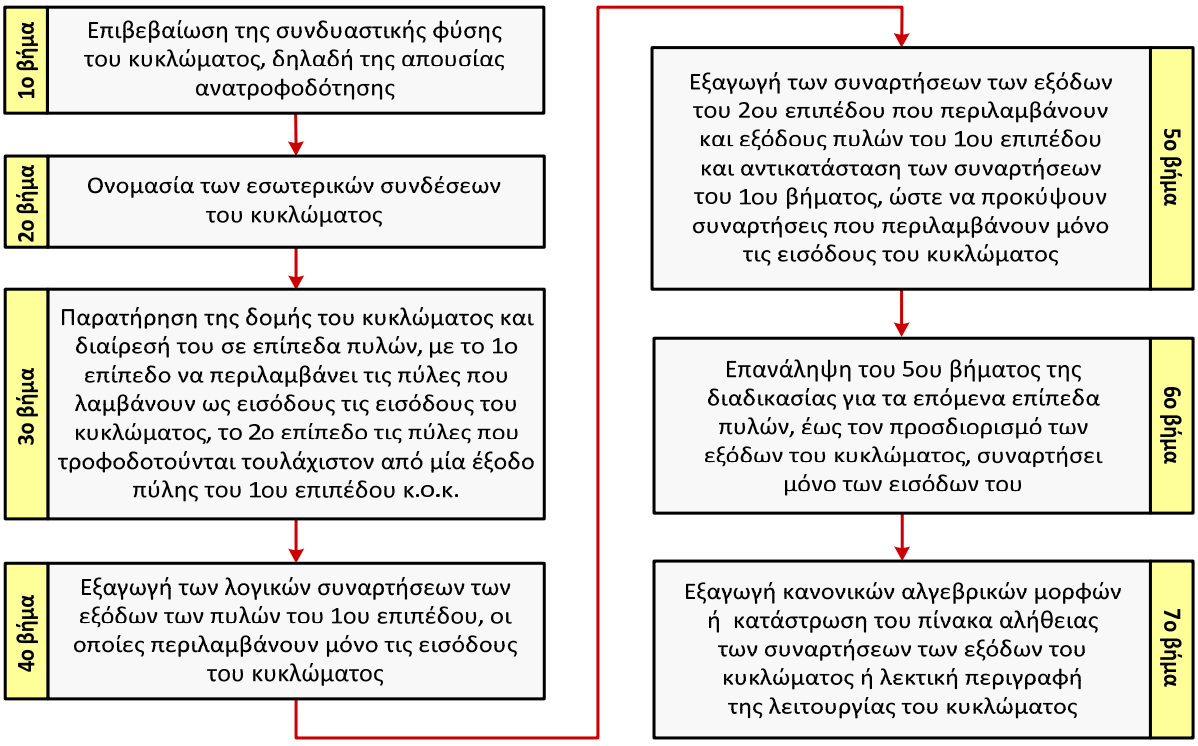
Για να επαληθευτεί η ορθή λειτουργία ενός συνδυαστικού κυκλώματος, δηλαδή για να επιβεβαιωθεί ότι λειτουργεί σύμφωνα με τις προδιαγραφές που σχεδιάστηκε, ακολουθείται μια διαδικασία που αναφέρεται ως ανάλυση συνδυαστικού κυκλώματος.

Επίσης, προκειμένου να διαπιστωθεί η λειτουργική ισοδυναμία δύο ή περισσότερων συνδυαστικών κυκλωμάτων, θα πρέπει αυτά να αναλυθούν.

Λειτουργικά ισοδύναμα είναι δύο συνδυαστικά κυκλώματα που έχουν το ίδιο πλήθος εισόδων και εξόδων και αντιστοιχούν στον ίδιο πίνακα αλήθειας ή στις ίδιες κανονικές αλγεβρικές μορφές.

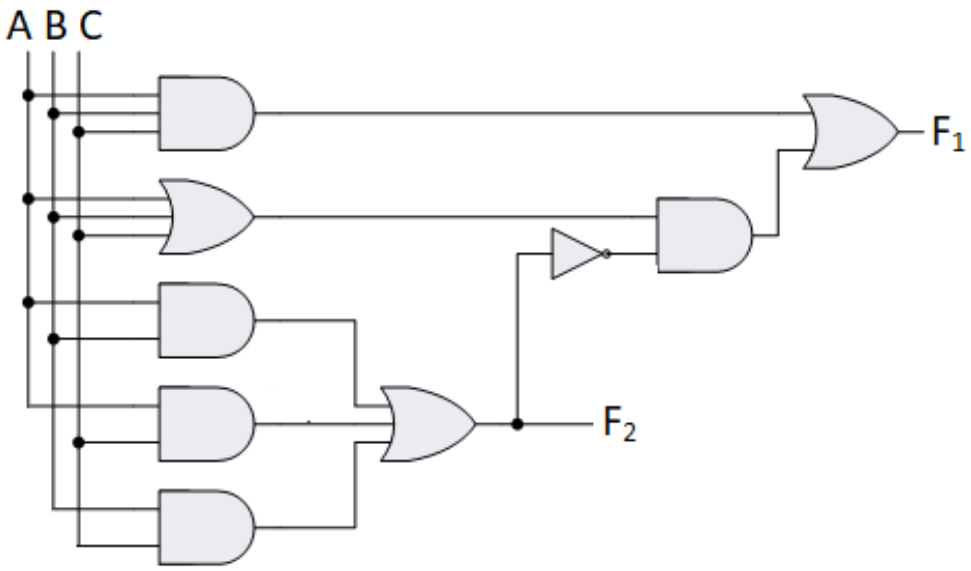
Ανάλυση συνδυαστικού κυκλώματος είναι λοιπόν ο προσδιορισμός της λειτουργίας ή των λειτουργιών που εκτελεί και συνίσταται στην εξαγωγή των λογικών συναρτήσεων ή του πίνακα αλήθειας ή ακόμη και μιας λεκτικής περιγραφής της λειτουργίας ή των λειτουργιών που εκτελούνται.

Ανάλυση συνδυαστικών κυκλωμάτων

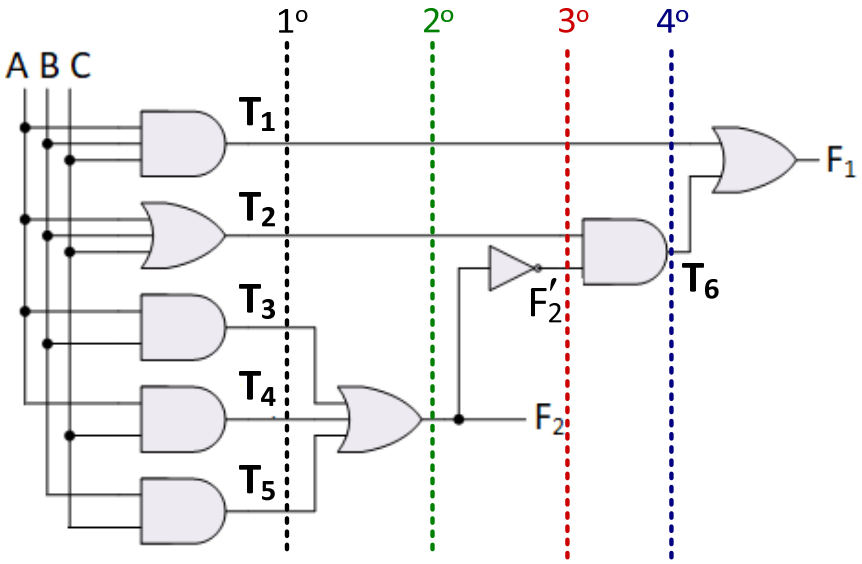


Ανάλυση συνδυαστικών κυκλωμάτων

Παράδειγμα: Ανάλυση συνδυαστικού κυκλώματος ξεκινώντας από το λογικό διάγραμμά του:

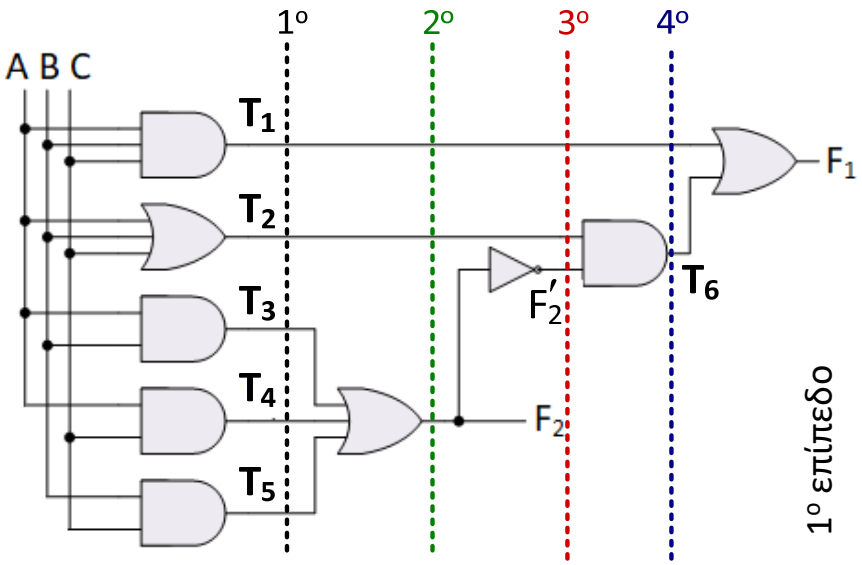


Ανάλυση συνδυαστικών κυκλωμάτων



Διαίρεση του κυκλώματος σε επίπεδα πυλών

Ανάλυση συνδυαστικών κυκλωμάτων

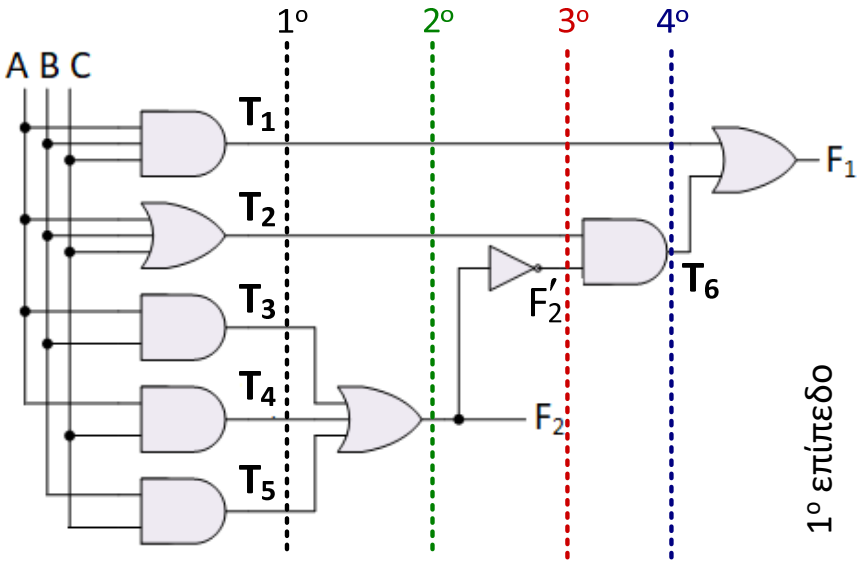


Διαίρεση του κυκλώματος σε επίπεδα πυλών

1ο επίπεδο

$T_1 = ABC$
$T_2 = A + B + C$
$T_3 = AB$
$T_4 = AC$
$T_5 = BC$

Ανάλυση συνδυαστικών κυκλωμάτων



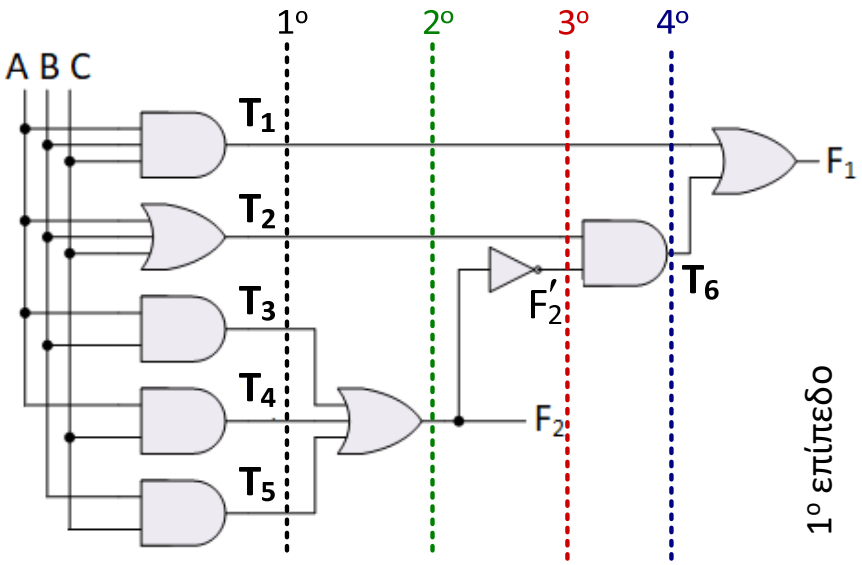
Διαίρεση του κυκλώματος σε επίπεδα πυλών

1ο επίπεδο

$T_1 = ABC$
$T_2 = A + B + C$
$T_3 = AB$
$T_4 = AC$
$T_5 = BC$

2ο επίπεδο: $F_2 = T_3 + T_4 + T_5 = AB + AC + BC$

Ανάλυση συνδυαστικών κυκλωμάτων



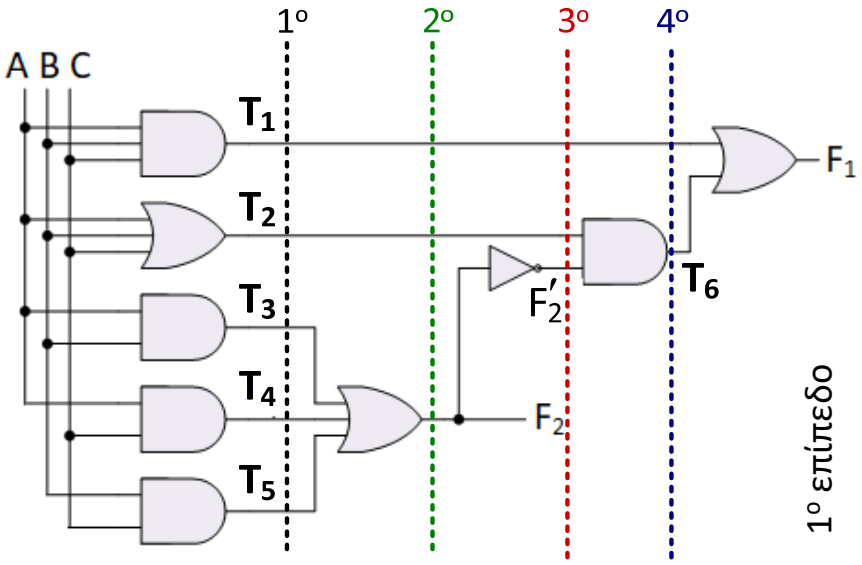
Διαίρεση του κυκλώματος σε επίπεδα πυλών

1ο επίπεδο:
$T_1 = ABC$
$T_2 = A + B + C$
$T_3 = AB$
$T_4 = AC$
$T_5 = BC$

2ο επίπεδο: $F_2 = T_3 + T_4 + T_5 = AB + AC + BC$

3ο & 4ο επίπεδο: $T_6 = T_2 F'_2 = (A + B + C)(AB + AC + BC)'$

Ανάλυση συνδυαστικών κυκλωμάτων



Διαίρεση του κυκλώματος σε επίπεδα πυλών

1ο επίπεδο:
$T_1 = ABC$
$T_2 = A + B + C$
$T_3 = AB$
$T_4 = AC$
$T_5 = BC$

2ο επίπεδο: $F_2 = T_3 + T_4 + T_5 = AB + AC + BC$

3ο & 4ο επίπεδο: $T_6 = T_2 F'_2 = (A + B + C)(AB + AC + BC)'$

$F_1 = T_1 + T_6 = ABC + (A + B + C)(AB + AC + BC)'$

Ανάλυση συνδυαστικών κυκλωμάτων

Εξαγωγή κανονικής μορφής (άθροισμα ελαχιστόρων) των 2 εξόδων:

F_2 = AB + AC + BC = AB(C + C′) + A(B + B′)C + (A + A′)BC
= ABC + ABC′ + ABC + AB′C + ABC + A′BC
= ABC + ABC′ + AB′C + A′BC = Σ(7,6,5,3) = **Σ(3,5,6,7)**

F_1 = ABC + (A + B + C)(AB + AC + BC)′
= ABC + (A + B + C)(AB)′(AC)′(BC)′
= ABC + (A + B + C)(A′ + B′)(A′ + C′)(B′ + C′)
= ABC + (AA′ + AB′ +A′B + BB′ + A′C + B′C)(A′B′ +A′C′ + B′C′ + C′C′)
= ABC + (AB′ +A′B + A′C + B′C)(A′B′ + A′C′ + B′C′ + C′)
= ABC + (AB′ +A′B + A′C + B′C)[A′B′ + C′(A′ + B′ + 1)]
= ABC + (AB′ +A′B + A′C + B′C)(A′B′ + C′)
= ABC + (AA′B′B′ + AB′C′ + A′A′BB′ + A′BC′ + A′A′B′C + A′CC′ + A′B′B′C + B′CC′)
= ABC + AB′C′ + A′BC′ + A′B′C = Σ(7,4,2,1) = **Σ(1,2,4,7)**

Ανάλυση συνδυαστικών κυκλωμάτων

Κατάστρωση πινάκων αλήθειας των 2 εξόδων:

F_2 = Σ(3,5,6,7)

F_1 = Σ(1,2,4,7)

A	B	C	F_2	F_1
0	0	0	**0**	**0**
0	0	1	**0**	**1**
0	1	0	**0**	**1**
0	1	1	**1**	**0**
1	0	0	**0**	**1**
1	0	1	**1**	**0**
1	1	0	**1**	**0**
1	1	1	**1**	**1**

Λεκτική περιγραφή λειτουργίας:

Αν και από τις κανονικές αλγεβρικές μορφές δεν μπορούμε να εξάγουμε μια λεκτική περιγραφή της λειτουργίας του κυκλώματος, από τον πίνακα αλήθειας προκύπτει ότι οι 2 έξοδοι του κυκλώματος σχηματίζουν διψήφιο δυαδικό αριθμό που ισούται με το πλήθος των μονάδων που περιλαμβάνονται στις 3 εισόδους.

Ανάλυση συνδυαστικών κυκλωμάτων

Εναλλακτικά, μπορούμε να καταστρώσουμε τον πίνακα αλήθειας των εξόδων, χωρίς προηγουμένως να εξάγουμε τις αλγεβρικές τους εκφράσεις.

Η διαδικασία συνίσταται στη σταδιακή εξαγωγή των στηλών του πίνακα αλήθειας που αντιστοιχούν στις εσωτερικές συνδέσεις του κυκλώματος, ώστε από αυτές να προκύψουν οι στήλες του πίνακα που αντιστοιχούν στις εξόδους του κυκλώματος, με χρήση μόνο των ορισμών των βασικών λογικών πράξεων.

ABC|A+B+C|AB|AC|BC

A	B	C	T_1	T_2	T_3	T_4	T_5	$F_2=T_3+T_4+T_5$	F'_2	$T_6=T_2F'_2$	$F_1=T_1+T_6$
0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	0	1	0	0	0	0	1	1	**1**
0	1	0	0	1	0	0	0	0	1	1	**1**
0	1	1	0	1	0	0	1	**1**	0	0	0
1	0	0	0	1	0	0	0	0	1	1	**1**
1	0	1	0	1	0	1	0	**1**	0	0	0
1	1	0	0	1	1	0	0	**1**	0	0	0
1	1	1	1	1	1	1	1	**1**	0	0	**1**

Τυποποιημένα συνδυαστικά κυκλώματα

Τα ψηφιακά συστήματα σχεδιάζονται και υλοποιούνται κυρίως για να μετασχηματίζουν δεδομένα.

Κάθε μετασχηματισμός μπορεί να παρασταθεί από μία ή περισσότερες λογικές συναρτήσεις.

Ορισμένες κατηγορίες μετασχηματισμών εμφανίζονται αρκετά συχνά σε πρακτικά προβλήματα, όπως για παράδειγμα οι αριθμητικές πράξεις, η κωδικοποίηση και η αποκωδικοποίηση δεδομένων, η επιλογή και μεταφορά δεδομένων.

Είναι λοιπόν χρήσιμο να υπάρχουν τα κυκλώματα αυτά προσχεδιασμένα και τυποποιημένα, έτσι ώστε να είναι εύκολη η χρησιμοποίησή τους στα ψηφιακά συστήματα.

Αριθμητικά συνδυαστικά κυκλώματα

Στα ψηφιακά συστήματα χρησιμοποιούνται ευρέως συνδυαστικά κυκλώματα που επιτελούν βασικές αριθμητικές λειτουργίες, όπως πρόσθεση, αφαίρεση, πολλαπλασιασμό, σύγκριση δυαδικών αριθμών κ.ά.

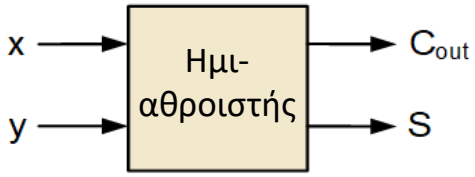
Η πρόσθεση δύο δυαδικών ψηφίων υλοποιείται από ένα συνδυαστικό κύκλωμα δύο εισόδων, δηλαδή των ψηφίων που προστίθενται, και δύο εξόδων, δηλαδή του αθροίσματος και του κρατουμένου.

Ωστόσο, κατά την πρόσθεση δύο αριθμών με περισσότερα ψηφία, για την υλοποίηση της πρόσθεσης των ψηφίων μιας θέσης απαιτείται συνδυαστικό κύκλωμα με τρεις εισόδους, έτσι ώστε να συμμετέχει στην πράξη και το κρατούμενο της πρόσθεσης των ψηφίων της προηγούμενης θέσης.

Το πρώτο κύκλωμα αναφέρεται ως ημιαθροιστής (half adder, HA), ενώ το δεύτερο ως πλήρης αθροιστής (full adder, FA).

Ημιαθροιστής (half adder)

Ημιαθροιστής είναι το συνδυαστικό κύκλωμα 2 εισόδων (x, y), δηλαδή των ψηφίων που προστίθενται, και 2 εξόδων, δηλαδή του αθροίσματος (S) και του κρατουμένου (C_{out}).



x	y	C_{out}	S
0	0	0	0
0	1	0	**1**
1	0	0	**1**
1	1	**1**	0

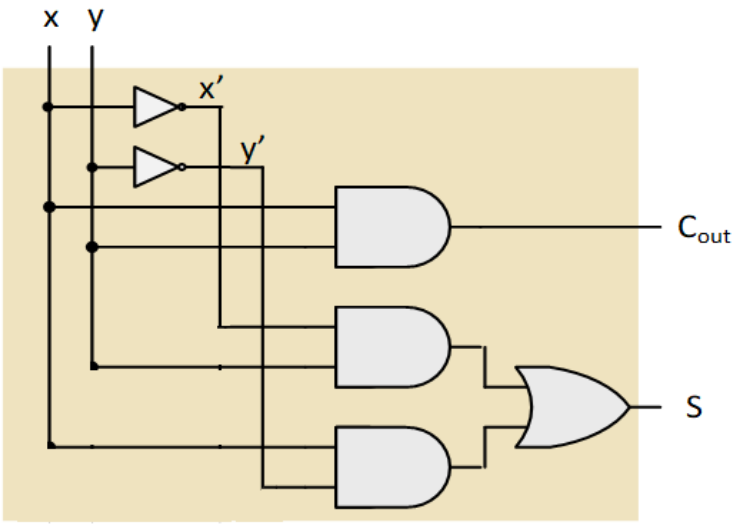
$$C_{out} = xy$$

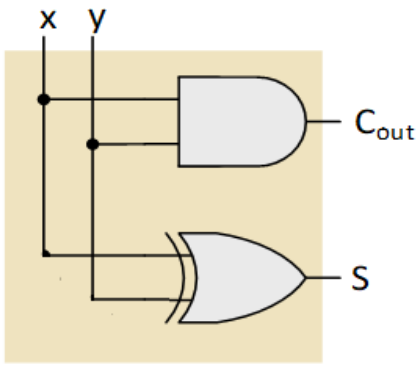
$$S = x'y + xy' = x \oplus y$$

Ημιαθροιστής (half adder)

$C_{out} = xy$

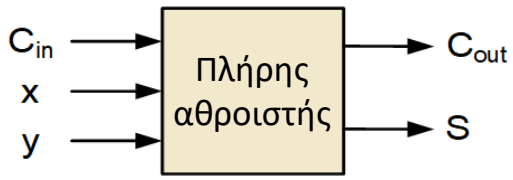
$S = x'y + xy' = x \oplus y$





Πλήρης αθροιστής (full adder)

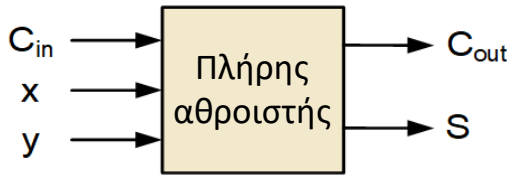
Πλήρης αθροιστής είναι το συνδυαστικό κύκλωμα 3 εισόδων που προσθέτει 2 δυαδικά ψηφία (x, y) και το τυχόν κρατούμενο (C_{in}) που έχει προκύψει από πρόσθεση σε προηγούμενη θέση και δίνει ως αποτέλεσμα το άθροισμα (S) και το κρατούμενο (C_{out}) της πρόσθεσης.

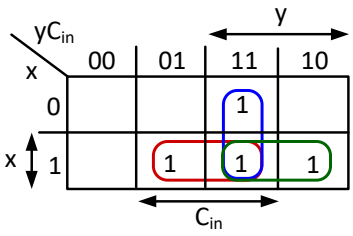


x	y	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Πλήρης αθροιστής (full adder)

Πλήρης αθροιστής είναι το συνδυαστικό κύκλωμα 3 εισόδων που προσθέτει 2 δυαδικά ψηφία (x, y) και το τυχόν κρατούμενο (C_{in}) που έχει προκύψει από πρόσθεση σε προηγούμενη θέση και δίνει ως αποτέλεσμα το άθροισμα (S) και το κρατούμενο (C_{out}) της πρόσθεσης.





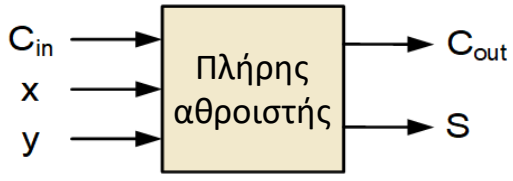
x	y	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

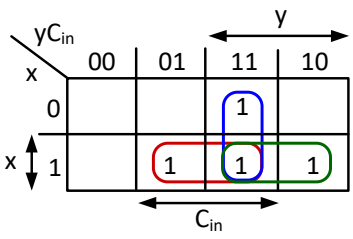
$C_{out} = x'yC_{in} + xy'C_{in} + xyC'_{in} + xyC_{in}$

$= \mathbf{xy + yC_{in} + xC_{in}}$

Πλήρης αθροιστής (full adder)

Πλήρης αθροιστής είναι το συνδυαστικό κύκλωμα 3 εισόδων που προσθέτει 2 δυαδικά ψηφία (x, y) και το τυχόν κρατούμενο (C_{in}) που έχει προκύψει από πρόσθεση σε προηγούμενη θέση και δίνει ως αποτέλεσμα το άθροισμα (S) και το κρατούμενο (C_{out}) της πρόσθεσης.

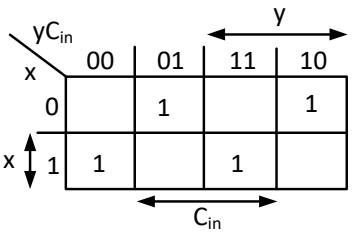




x	y	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$C_{out} = x'yC_{in} + xy'C_{in} + xyC'_{in} + xyC_{in}$

$= \mathbf{xy + yC_{in} + xC_{in}}$



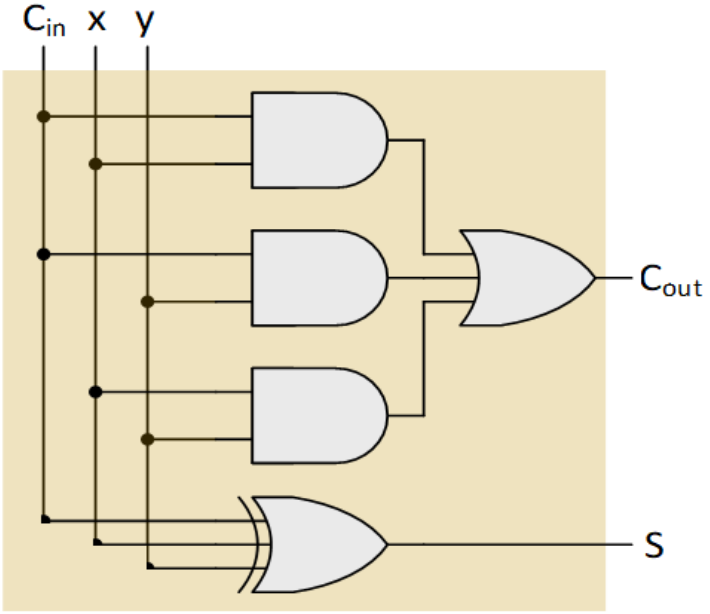
$\mathbf{S} = x'y'C_{in} + x'yC'_{in} + xy'C'_{in} + xyC_{in}$

$= (xy' + x'y')C'_{in} + (xy + x'y')C_{in}$

$= (x \oplus y)C'_{in} + (x \oplus y)'C_{in} = \mathbf{x \oplus y \oplus C_{in}}$ (περιττή συνάρτηση)

Ο χάρτης Karnaugh του S δεν οδηγεί σε απλοποίηση.

Πλήρης αθροιστής (full adder)



$$C_{out} = xy + yC_{in} + xC_{in}$$

$$S = x \oplus y \oplus C_{in}$$

Πλήρης αθροιστής (full adder)

Παράδειγμα: Υλοποίηση ενός πλήρους αθροιστή με 2 ημιαθροιστές και μία πύλη OR 2 εισόδων.

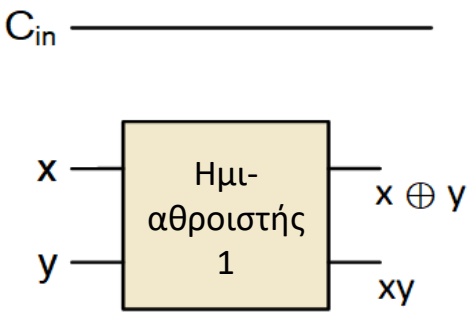
Η λογική συνάρτηση του αθροίσματος του πλήρους αθροιστή μπορεί να εκφραστεί ως εξής:

$$S = (x \oplus y) \oplus C_{in}.$$

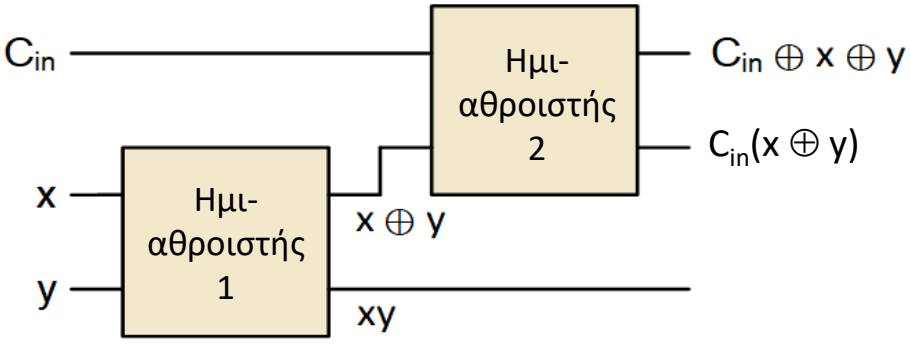
Η λογική συνάρτηση του κρατούμενου εξόδου του πλήρους αθροιστή μπορεί να εκφραστεί ως εξής:

$$\begin{aligned} C_{out} &= x'yC_{in} + xy'C_{in} + xyC'_{in} + xyC_{in} \\ &= C'_{in}xy + C_{in}x'y + C_{in}xy' + C_{in}xy \\ &= (C'_{in} + C_{in})xy + C_{in}(x'y + xy') \\ &= xy + (x \oplus y)C_{in} \end{aligned}$$

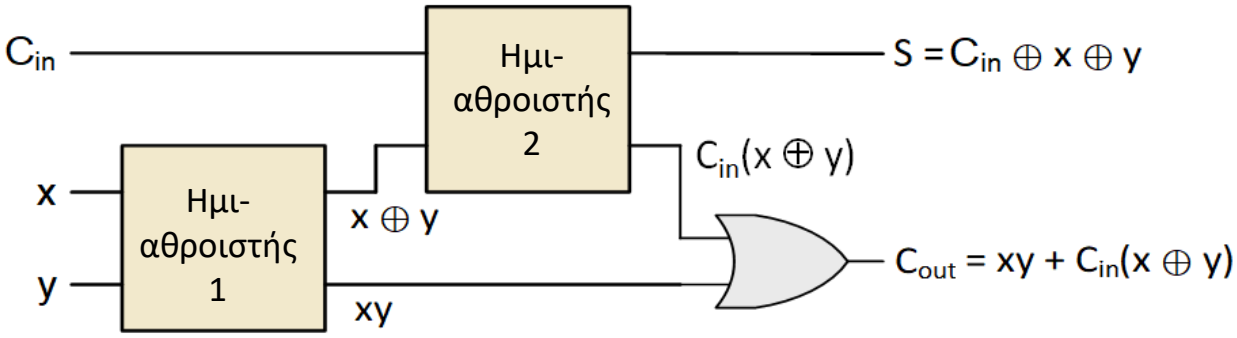
Πλήρης αθροιστής (full adder)



Πλήρης αθροιστής (full adder)



Πλήρης αθροιστής (full adder)



Πλήρης αφαιρέτης (full subtractor)

Ο πλήρης αφαιρέτης διαθέτει 3 εισόδους, x (μειωτέος), y (αφαιρετέος) και B_{in} (δανειζόμενο ψηφίο από προηγούμενη αφαίρεση) και 2 εξόδους D (διαφορά) και B_{out} (δανειζόμενο ψηφίο που προκύπτει από την αφαίρεση).

Κατά την αφαίρεση 2 δυαδικών ψηφίων προσθέτουμε το δανειζόμενο ψηφίο από την προηγούμενη αφαίρεση στον αφαιρετέο και το άθροισμά τους αφαιρείται από τον μειωτέο: $D = x - (y + B_{in})$.

Πίνακας αλήθειας πλήρους αφαιρέτη

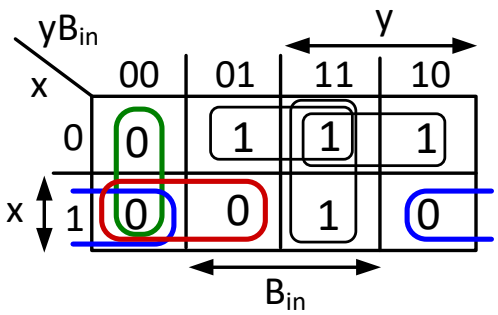
x	y	B_{in}	B_{out}	D
0	0	0	0	0
0	0	1	**1**	**1**
0	1	0	**1**	**1**
0	1	1	**1**	0
1	0	0	0	**1**
1	0	1	0	0
1	1	0	0	0
1	1	1	**1**	**1**

Παρατηρούμε ότι η διαφορά D είναι περιττή συνάρτηση των 3 μεταβλητών x, y, B_{in}, συνεπώς ισχύει ότι και για το άθροισμα S στον πλήρη αθροιστή:

$$D = x \oplus y \oplus B_{in}$$

Επίσης, προκύπτει ότι: $B_{out} = \Sigma(1,2,3,7)$

Πλήρης αφαιρέτης (full subtractor)



$B_{out} = x'y + yB_{in} + x'B_{in}$

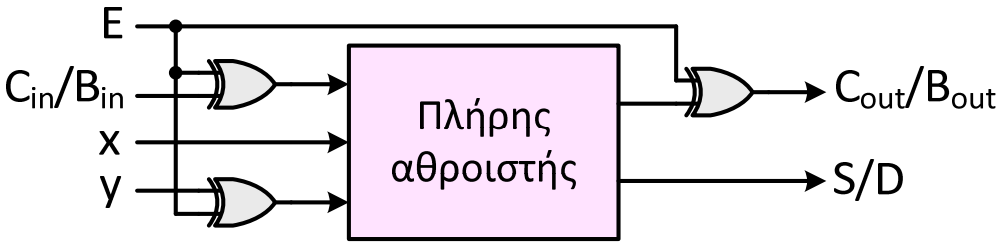
$B'_{out} = xy' + y'B'_{in} + xB'_{in}$

$C_{out} = xy + yC_{in} + xC_{in}$

Συγκρίνοντας τη συμπληρωματική συνάρτηση του B_{out} του πλήρους αφαιρέτη με τη συνάρτηση του C_{out} του πλήρους αθροιστή και λαμβάνοντας υπόψη ότι $S = D = x \oplus y \oplus B_{in} = x \oplus y' \oplus B'_{in}$, προκύπτει ότι ο πλήρης αφαιρέτης υλοποιείται με έναν πλήρη αθροιστή και 3 αντιστροφείς:

Πλήρης αθροιστής / αφαιρέτης

Με βάση τα προαναφερθέντα, και αφού ισχύει ότι $A \oplus 1 = A'$ και $A \oplus 0 = A$, μπορούμε να υλοποιήσουμε ένα κύκλωμα πλήρους αθροιστή / αφαιρέτη με έναν πλήρη αθροιστή και 3 πύλες XOR 2 εισόδων.



Το παραπάνω κύκλωμα λειτουργεί ως πλήρης αθροιστής όταν E = 0 και ως πλήρης αφαιρέτης όταν E = 1.

Παράλληλος αθροιστής

Χρησιμοποιώντας ως δομικό στοιχείο τον πλήρη αθροιστή, μπορούμε να συνθέσουμε συνδυαστικά κυκλώματα παράλληλων αθροιστών για αριθμούς με περισσότερα ψηφία.

Το αποτέλεσμα της πρόσθεσης δύο μη προσημασμένων δυαδικών αριθμών N ψηφίων αποτελείται από N + 1 ψηφία, συμπεριλαμβανομένου του τελικού κρατουμένου, συνεπώς το συνδυαστικό κύκλωμα του παράλληλου αθροιστή διαθέτει N + 1 εξόδους.

Η πρόσθεση 2 δυαδικών αριθμών εκτελείται ανά ζεύγη ψηφίων της ίδιας θέσης, ξεκινώντας από το ζεύγος των λιγότερο σημαντικών ψηφίων των αριθμών και αν προκύπτει κρατούμενο ψηφίο μετά την πρόσθεση σε κάποια θέση, τότε αυτό προστίθεται στα ψηφία τής αμέσως πιο σημαντικής θέσης.

Έτσι, ο παράλληλος αθροιστής δύο δυαδικών αριθμών N ψηφίων προκύπτει εύκολα, εάν συνδέσουμε σειριακά N πλήρεις αθροιστές.

Παράλληλος αθροιστής

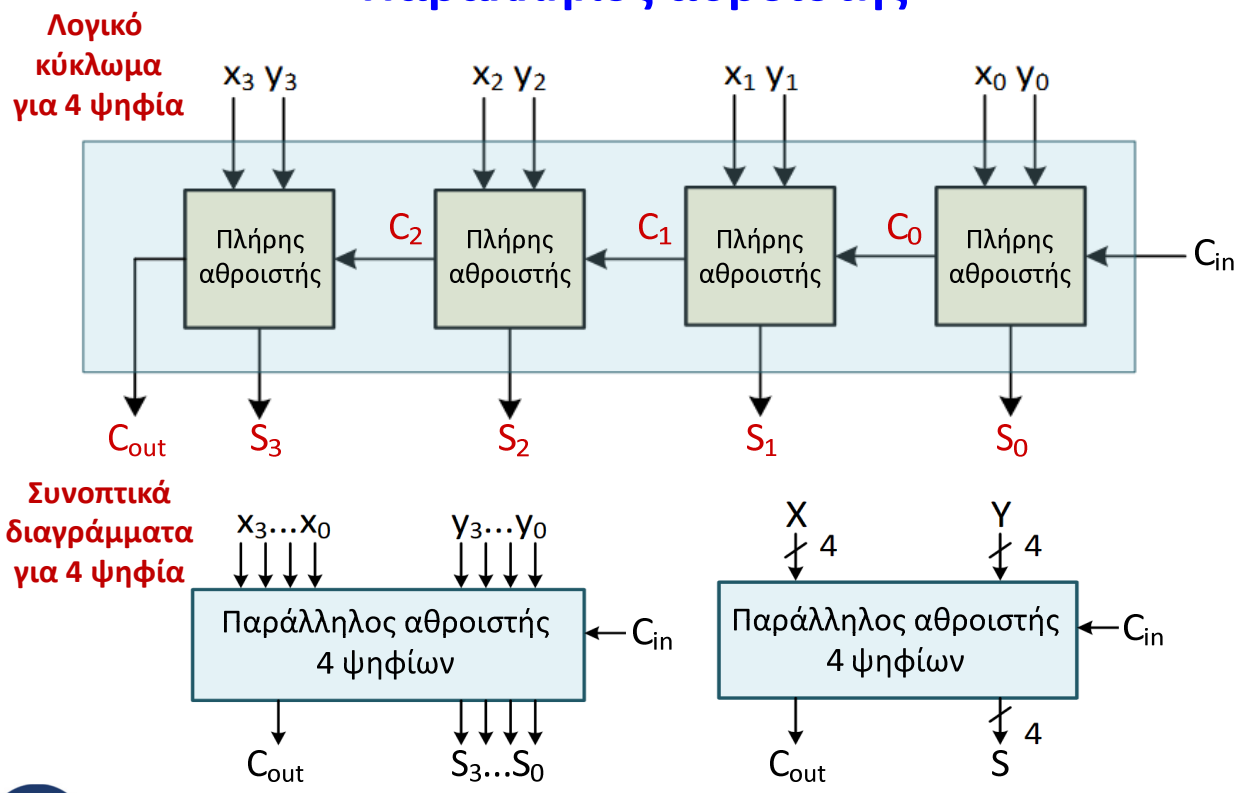
Κάθε πλήρης αθροιστής δέχεται ένα ζεύγος ψηφίων των αριθμών εισόδου και παράγει ένα ψηφίο αθροίσματος και ένα ψηφίο κρατουμένου, το οποίο διαδίδεται στον πλήρη αθροιστή της επόμενης (πιο σημαντικής) θέσης.

Το τελικό κρατούμενο της πρόσθεσης των δύο αριθμών λαμβάνεται στην έξοδο κρατουμένου του πλήρους αθροιστή της πιο σημαντικής θέσης.

Στην λιγότερο σημαντική θέση δεν υφίσταται κρατούμενο από πρόσθεση σε προηγούμενη θέση ($C_{in} = 0$).

Οι παράλληλοι αθροιστές αυτής της δομής αναφέρονται ως αθροιστές διάδοσης κρατουμένου ή κυματικοί αθροιστές (ripple carry adders).

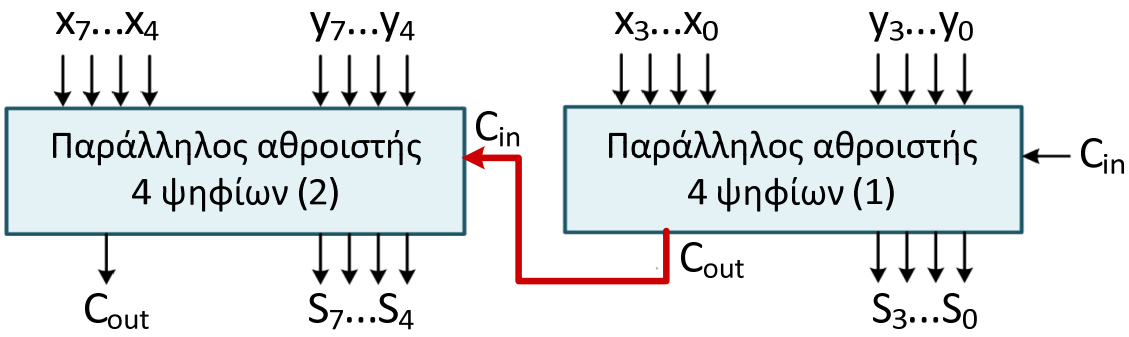
Παράλληλος αθροιστής



Παράλληλος αθροιστής

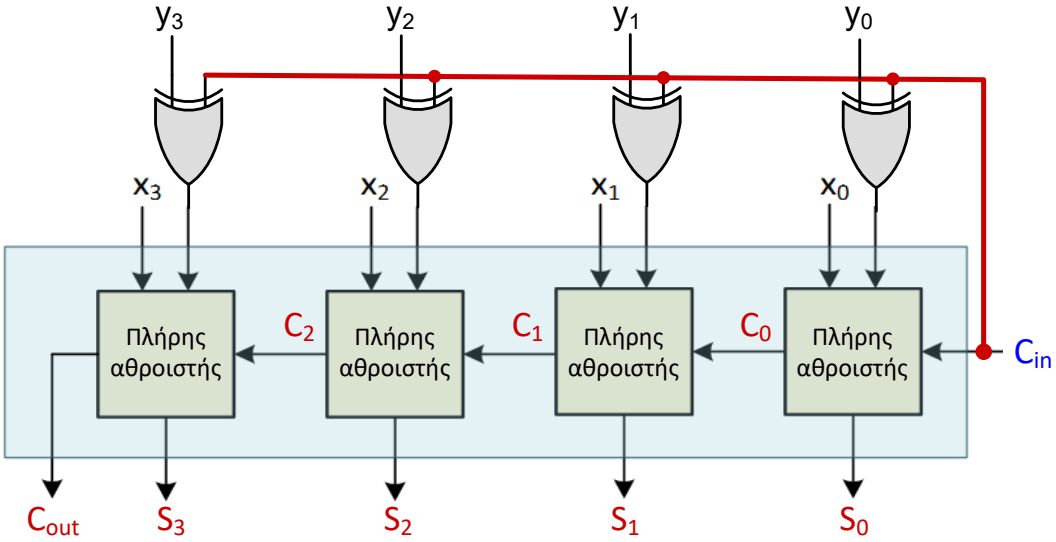
Παράδειγμα: Υλοποίηση παράλληλου αθροιστή 8 ψηφίων με χρήση δύο παράλληλων αθροιστών 4 ψηφίων.

Η ζητούμενη υλοποίηση επιτυγχάνεται με την τροφοδότηση του κρατουμένου εισόδου του δεύτερου παράλληλου αθροιστή 4 ψηφίων με το κρατούμενο εξόδου του πρώτου παράλληλου αθροιστή 4 ψηφίων.

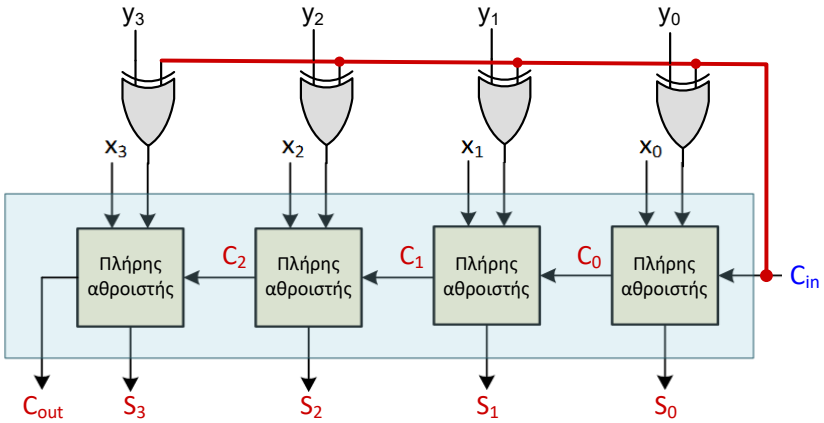


Παράλληλος αθροιστής / αφαιρέτης

Ένας παράλληλος αθροιστής / αφαιρέτης μπορεί να σχεδιαστεί με προσθήκη σε έναν παράλληλο αθροιστή ισάριθμων πυλών XOR με το πλήθος των ψηφίων των αριθμών που προστίθενται ή αφαιρούνται.



Παράλληλος αθροιστής / αφαιρέτης



$X = x_3x_2x_1x_0$

$Y = y_3y_2y_1y_0$

$C_{in} = 0 \Rightarrow S = X + Y$

$C_{in} = 1 \Rightarrow S = X + Y' + 1$

$S = S_3S_2S_1S_0$

Τροφοδοτούμε τη μία είσοδο κάθε πύλης XOR με το αντίστοιχο ψηφίο του δεύτερου αριθμού (y_i), ενώ την άλλη είσοδό της την τροφοδοτούμε με το κρατούμενο εισόδου (C_{in}) της λιγότερο σημαντικής θέσης.

Όταν $C_{in} = 0$, η έξοδος κάθε πύλης XOR είναι y_i ($S = X + Y$), ενώ όταν $C_{in} = 1$, η έξοδος κάθε πύλης XOR είναι y'_i και το αποτέλεσμα της πράξης είναι το άθροισμα του X με το συμπλήρωμα ως προς 2 του Y ($S = X + Y' + 1 = X + \Sigma_2 Y$).

Παράλληλος αθροιστής / αφαιρέτης

Όταν $C_{in} = 0$, το κύκλωμα λειτουργεί ως παράλληλος αθροιστής και εκτελεί την πρόσθεση $X + Y = x_3x_2x_1x_0 + y_3y_2y_1y_0$.

Για μη προσημασμένους αριθμούς (X, Y), το κύκλωμα έχει ως αποτέλεσμα $S = S_3S_2S_1S_0 = X + Y$ και κρατούμενο C_{out}.

Για προσημασμένους αριθμούς, το κύκλωμα έχει ως αποτέλεσμα $S = S_3S_2S_1S_0 = X + Y$ εφόσον δε συμβαίνει υπερχείλιση (overflow).

Όταν $C_{in} = 1$, το κύκλωμα λειτουργεί ως παράλληλος αφαιρέτης, αφού εκτελεί την πρόσθεση $X + Y' + 1 = x_3x_2x_1x_0 + y'_3y'_2y'_1y'_0 + 1$, στην οποία ανάγεται η πράξη της αφαίρεσης.

Για μη προσημασμένους αριθμούς το κύκλωμα έχει ως αποτέλεσμα $S = S_3S_2S_1S_0 = X - Y$ όταν $X \geq Y$ και $S = S_3S_2S_1S_0 = \Sigma_2(Y - X)$ όταν $X < Y$.

Για προσημασμένους αριθμούς, το κύκλωμα έχει ως αποτέλεσμα $S = S_3S_2S_1S_0 = X - Y$ εφόσον δε συμβαίνει υπερχείλιση.

Παράλληλος αθροιστής / αφαιρέτης

Υπερχείλιση κατά την πρόσθεση ή την αφαίρεση δύο προσημασμένων αριθμών, σημαίνει ότι οι αριθμοί αποτελούνται από n ψηφία και για την παράσταση του αποτελέσματος της πράξης απαιτούνται n + 1 ψηφία.

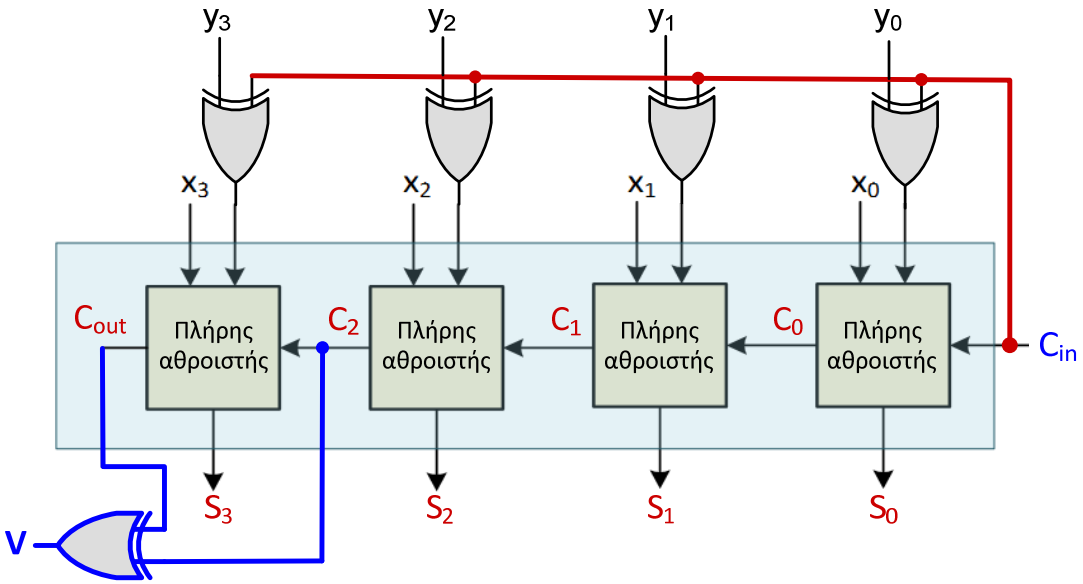
Στην περίπτωση αυτή μετατοπίζεται το ψηφίο-πρόσημο από την αναμενόμενη θέση.

Υπερχείλιση κατά την πρόσθεση (X + Y) δύο προσημασμένων αριθμών συμβαίνει όταν οι X και Y είναι ομόσημοι και το αποτέλεσμα της πράξης έχει διαφορετικό πρόσημο από τους προσθετέους X και Y.
Παράδειγμα: (+7) 0111 + (+5) 0101 = (−4) 1100 αντί για (+12) 01100.

Υπερχείλιση κατά την αφαίρεση (X – Y) δύο προσημασμένων αριθμών συμβαίνει όταν ο X είναι θετικός, ο Y είναι αρνητικός και το αποτέλεσμα της πράξης είναι αρνητικό, καθώς και όταν ο X είναι αρνητικός, ο Y είναι θετικός και το αποτέλεσμα της πράξης είναι θετικό.
Παράδειγμα: (−7) 1001 − (+5) 0101 = (−7) 1001 + (−5) 1011 = (+4) 0100 αντί για (−12) 10100.

Παράλληλος αθροιστής / αφαιρέτης

Η υπερχείλιση ανιχνεύεται με την προσθήκη μιας πύλης XOR δύο εισόδων στο κύκλωμα του παράλληλου αθροιστή / αφαιρέτη.



Παράλληλος αθροιστής / αφαιρέτης

Η ανίχνευση της υπερχείλισης γίνεται μέσω της διαπίστωσης ότι τα κρατούμενα ψηφία που προκύπτουν κατά την πράξη στις δύο πιο σημαντικές θέσεις λαμβάνουν διαφορετική τιμή ($V = C_2 \oplus C_{out} = 1$).

Αυτό προκύπτει από το αποτέλεσμα της πρόσθεσης των ψηφίων-προσήμων των δύο προσθετέων που εκτελείται στον πλήρη αθροιστή της πιο σημαντικής θέσης, όταν οι δύο προσθετέοι είναι ομόσημοι και το άθροισμα έχει διαφορετικό πρόσημο από τους προσθετέους:

$$0\ (x_3) + 0\ (y_3) + 1\ (C_2) = 1\ (S_3) + 0\ (C_{out})$$
$$1\ (x_3) + 1\ (y_3) + 0\ (C_2) = 0\ (S_3) + 1\ (C_{out})$$

Εάν V = 0 το αποτέλεσμα $S_3S_2S_1S_0$ είναι σωστό, ενώ εάν V = 1 συμβαίνει υπερχείλιση και το αποτέλεσμα της πράξης χρειάζεται 5 ψηφία για να παρασταθεί.

Στην δεύτερη περίπτωση (V = 1), η τιμή του C_{out} εκφράζει το πρόσημο του αποτελέσματος, το οποίο έχει μετατοπιστεί.

Δυαδικός πολλαπλασιαστής

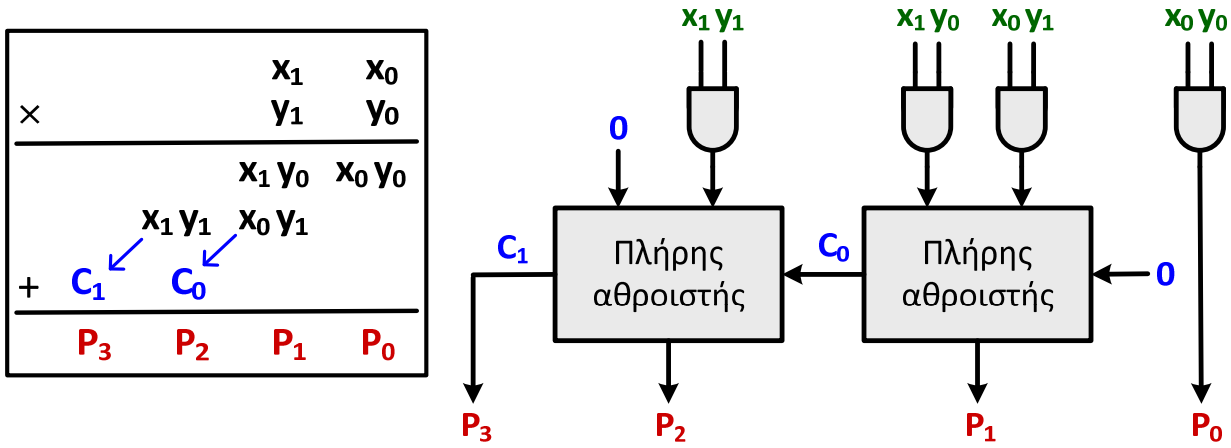
Το αριθμητικό γινόμενο δύο δυαδικών ψηφίων ταυτίζεται με το λογικό γινόμενό τους, επομένως υλοποιείται εύκολα με μία πύλη AND 2 εισόδων.

Κατά τον πολλαπλασιασμό αριθμών με περισσότερα δυαδικά ψηφία, απαιτείται η πρόσθεση των μερικών γινομένων 2 ψηφίων.

Για τους παραπάνω λόγους, η σύνθεση ενός πολλαπλασιαστή δυαδικών αριθμών, επιτυγχάνεται με χρήση πυλών AND 2 εισόδων και παράλληλων αθροιστών.

Για την υλοποίηση ενός πολλαπλασιαστή δύο μη προσημασμένων δυαδικών αριθμών 2 ψηφίων, απαιτούνται 4 πύλες AND 2 εισόδων για την εξαγωγή των ισάριθμων μερικών γινομένων και ένας παράλληλος αθροιστής 2 ψηφίων για την πρόσθεσή τους και την εξαγωγή του τελικού γινομένου.

Δυαδικός πολλαπλασιαστής



Οι πλήρεις αθροιστές του παράλληλου αθροιστή που χρησιμοποιήθηκε, στον συγκεκριμένο πολλαπλασιαστή μπορούν να αντικατασταθούν με ημιαθροιστές, αφού εκτελούν πρόσθεση μεταξύ 2 ψηφίων.

Συγκριτής μεγέθους αριθμών

Η σύγκριση δύο αριθμών είναι η πράξη που προσδιορίζει εάν ένας αριθμός είναι μεγαλύτερος από έναν άλλο αριθμό, μικρότερος από αυτόν ή ίσος με αυτόν.

Ο συγκριτής μεγέθους (magnitude comparator) αριθμών είναι συνδυαστικό κύκλωμα, το οποίο συγκρίνει δύο αριθμούς A και B και προσδιορίζει τη σχέση των μεγεθών τους.

Το αποτέλεσμα της σύγκρισης προκύπτει από την τιμή των τριών εξόδων του κυκλώματος G (A > B), E (A = B) και L (A < B), με τα ονόματα των τριών εξόδων να προκύπτουν από τα αρχικά γράμματα των λέξεων Greater, Equal και Lesser, αντίστοιχα.

Ο πίνακας αλήθειας του κυκλώματος σύγκρισης δύο αριθμών με n ψηφία ο καθένας, περιλαμβάνει $2^{2 \cdot n}$ γραμμές, με αποτέλεσμα να είναι δύσκολο να χρησιμοποιηθεί για τη σύνθεση του κυκλώματος σύγκρισης (παράδειγμα: για n = 3 προκύπτουν $2^6 = 64$ γραμμές).

Συγκριτής μεγέθους αριθμών

Για το λόγο αυτό, ακολουθούμε μια πιο αποδοτική μέθοδο για τη σύνθεση του κυκλώματος σύγκρισης.

Έστω δύο αριθμοί A και B με πλήθος ψηφίων n = 4:

$$A = A_3\, A_2\, A_1\, A_0$$
$$B = B_3\, B_2\, B_1\, B_0$$

Οι δύο αριθμοί είναι ίσοι όταν:

$$A_3 = B_3, A_2 = B_2, A_1 = B_1, A_0 = B_0$$

Η σχέση ισότητας δύο ψηφίων μπορεί να εκφραστεί λογικά με τη συνάρτηση ισοδυναμίας (XNOR): $x_i = A_i B_i + A_i' B_i'$

x_i = 1 όταν τα ψηφία αντίστοιχων θέσεων των δύο αριθμών είναι ίσα.

Για να είναι ίσοι οι αριθμοί A και B (δηλαδή για να ισχύει E = 1) πρέπει να όλες οι μεταβλητές x_i να έχουν τιμή 1, δηλαδή προκύπτει ότι:

$$\boxed{E = x_3 x_2 x_1 x_0}$$

Συγκριτής μεγέθους αριθμών

Για να διαπιστώσουμε εάν ο Α είναι μεγαλύτερος ή μικρότερος από τον Β, ελέγχουμε την τιμή των ψηφίων των αντίστοιχων θέσεων, ξεκινώντας από τα ψηφία της πιο σημαντικής θέσης.

Εάν τα δύο πιο σημαντικά ψηφία είναι ίσα, ελέγχουμε το ζεύγος ψηφίων της αμέσως λιγότερο σημαντικής θέσης.

Εάν το ψηφίο του Α είναι 1 και το αντίστοιχο του Β είναι 0, συμπεραίνουμε ότι A > B, ενώ εάν το ψηφίο του Α είναι 0 και το αντίστοιχο ψηφίο του Β είναι 1, συμπεραίνουμε ότι A < B.

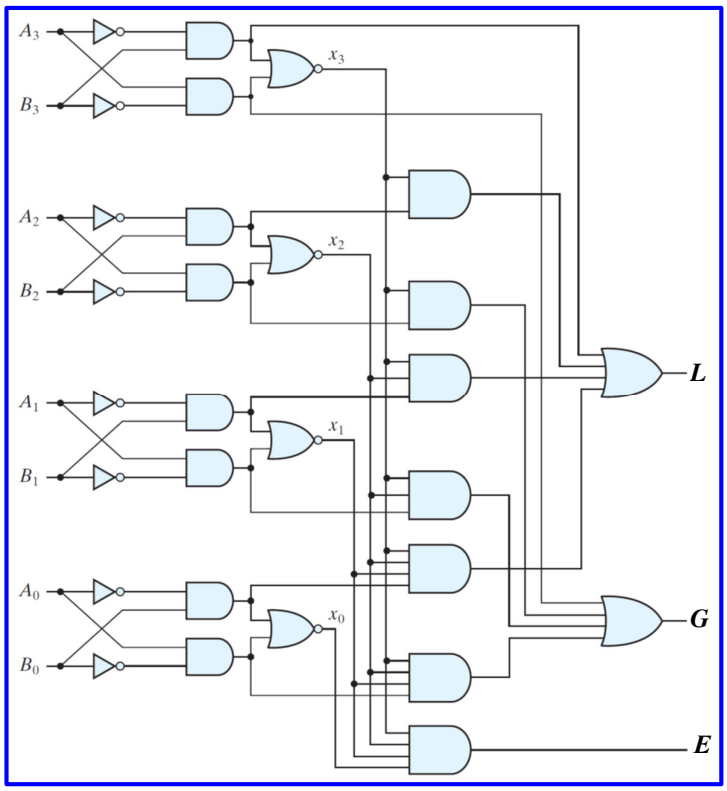
Η ακολουθιακή αυτή διαδικασία σύγκρισης των ψηφίων, εκφράζεται με δύο λογικές συναρτήσεις:

$$G = A_3B_3' + x_3A_2B_2' + x_3x_2A_1B_1' + x_3x_2x_1A_0B_0'$$
$$L = A_3'B_3 + x_3A_2'B_2 + x_3x_2A_1'B_1 + x_3x_2x_1A_0'B_0$$

Συγκριτής μεγέθους αριθμών

Κατά την υλοποίηση των συναρτήσεων E, G και L γίνεται επαναχρησιμοποίηση λογικών γινομένων, ώστε να καταλήξουμε σε κύκλωμα μειωμένου κόστους.

Η σύγκριση μη προσημασμένων, αλλά και προσημασμένων αριθμών μπορεί να επιτευχθεί και μέσω της της αφαίρεσής τους



Κωδικοποιητές

Τα ψηφιακά συστήματα έχουν τη δυνατότητα να χειρίζονται διακριτά στοιχεία πληροφορίας που ανήκουν σε ένα πεπερασμένο σύνολο, αφού το καθένα από αυτά μπορεί να παρασταθεί με μία ακολουθία δυαδικών ψηφίων, ακολουθώντας τους κανόνες ενός δυαδικού κώδικα.

Κάθε διαφορετικό στοιχείο πληροφορίας αντιστοιχίζεται σε μία μοναδική ακολουθία ή συνδυασμό δυαδικών ψηφίων.

Για την παράσταση ενός συνόλου 2^N στοιχείων πληροφορίας, όπως, για παράδειγμα, αριθμών ή γραμμάτων, απαιτείται δυαδικός κώδικας που να χρησιμοποιεί τουλάχιστον Ν δυαδικά ψηφία.

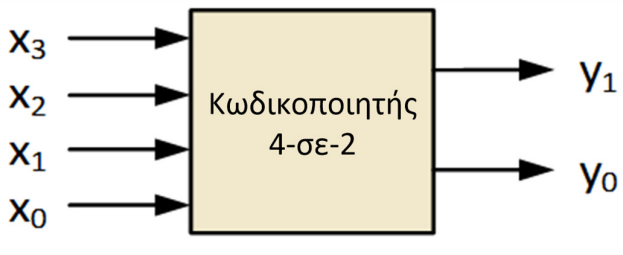
Οι κωδικοποιητές (encoders) είναι συνδυαστικά κυκλώματα τα οποία παράγουν στην έξοδό τους ψηφιακά δεδομένα σε πιο συμπαγή μορφή από εκείνη των δεδομένων που λαμβάνουν στην είσοδό τους.

Κωδικοποιητές

Ένας κωδικοποιητής, λαμβάνει στην είσοδό του **έως** 2^N δυαδικά ψηφία και παράγει στην έξοδό του Ν δυαδικά ψηφία.

Λόγω του ότι το πλήθος των ψηφίων εξόδου δεν επαρκεί για την παράσταση του μεγαλύτερου πλήθους των δεδομένων εισόδου, μόνο ένα από τα ψηφία εισόδου μπορεί να έχει λογική τιμή 1, έτσι ώστε το κύκλωμα να παράγει στην έξοδό του ένα δυαδικό αριθμό Ν ψηφίων, που αντιστοιχεί στο ψηφίο της εισόδου με λογική τιμή 1.

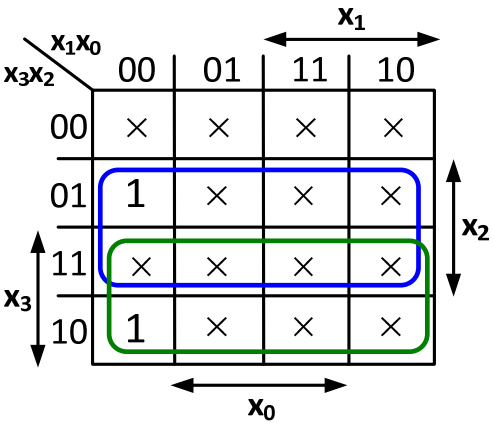
Απλός κωδικοποιητής 4-σε-2



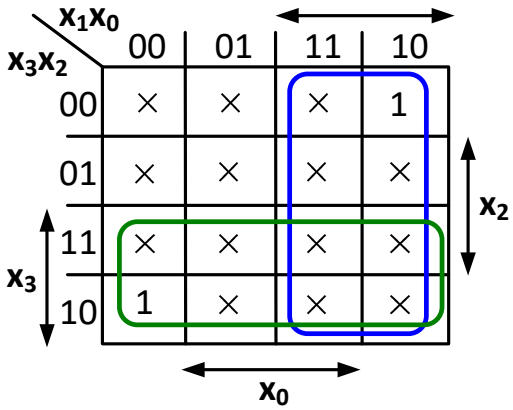
x_3	x_2	x_1	x_0	y_1	y_0
0	0	0	0	×	×
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	×	×
0	1	0	0	1	0
0	1	0	1	×	×
0	1	1	0	×	×
0	1	1	1	×	×
1	0	0	0	1	1
1	0	0	1	×	×
1	0	1	0	×	×
1	0	1	1	×	×
1	1	0	0	×	×
1	1	0	1	×	×
1	1	1	0	×	×
1	1	1	1	×	×

Αφού μόνο ένα από τα ψηφία εισόδου μπορεί να έχει λογική τιμή 1, 12 από τους συνολικά 16 συνδυασμούς εισόδων δεν επιτρέπονται και οι αντίστοιχοι ελαχιστόροι αποτελούν αδιάφορους όρους.

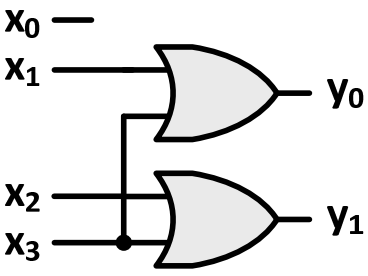
Απλός κωδικοποιητής 4-σε-2



$y_1 = x_2 + x_3$



$y_0 = x_1 + x_3$



Απλός κωδικοποιητής 4-σε-2

Στη λειτουργία του απλού κωδικοποιητή εντοπίζονται 2 προβλήματα.

Το πρώτο αφορά το γεγονός ότι οι έξοδοι του κυκλώματος που υλοποιήθηκε μηδενίζονται, όταν όλες οι είσοδοι έχουν λογική τιμή 0, με αποτέλεσμα αυτός ο συνδυασμός τιμών των εισόδων να μην μπορεί να διακριθεί από την περίπτωση όπου μόνο η είσοδος x_0 λογική τιμή 1.

Το δεύτερο πρόβλημα οφείλεται στο ότι δεν προβλέπονται οι περιπτώσεις όπου περισσότερες από μία εισόδους έχουν τιμή 1.

Το πρώτο πρόβλημα αντιμετωπίζεται με την προσθήκη μιας επιπλέον εξόδου εγκυρότητας (V) που λαμβάνει τιμή 0 όταν όλες οι είσοδοι είναι 0 και τιμή 1 σε όλες τις υπόλοιπες περιπτώσεις.

Οι υπόλοιπες δύο έξοδοι δεν ορίζονται όταν μηδενίζεται η V, συνεπώς ο συνδυασμός μηδενικών εισόδων αποτελεί αδιάφορη λογική συνθήκη για τις εξόδους αυτές.

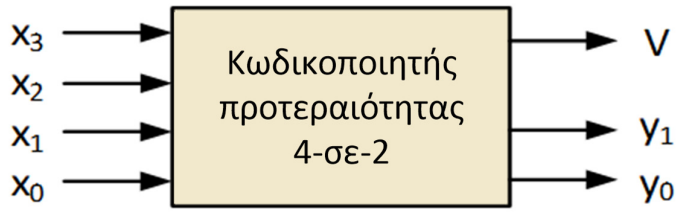
Κωδικοποιητής προτεραιότητας 4-σε-2

Το δεύτερο πρόβλημα λύνεται εάν ο κωδικοποιητής τροποποιηθεί ώστε να υποστηρίζει προκαθορισμένη προτεραιότητα εισόδων και τότε αναφέρεται ως κωδικοποιητής προτεραιότητας (priority encoder).

Έτσι όταν η τιμή περισσότερων εισόδων είναι 1, η έξοδος του κυκλώματος καθορίζεται από την είσοδο με τη μεγαλύτερη προτεραιότητα.

Για παράδειγμα, μπορεί να καθοριστεί ως είσοδος με τη μεγαλύτερη προτεραιότητα η είσοδος x_3 και να ακολουθούν σε σειρά προτεραιότητας οι είσοδοι x_2, x_1 και x_0.

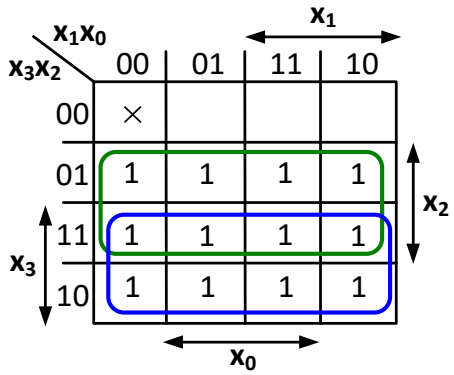
Κωδικοποιητής προτεραιότητας 4-σε-2



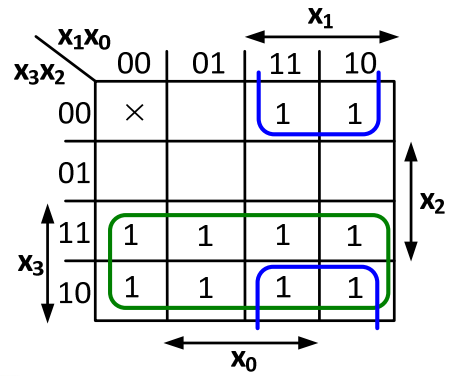
x_3	x_2	x_1	x_0	y_1	y_0	V
0	0	0	0	X	X	0
0	0	0	**1**	**0**	**0**	1
0	0	**1**	0	**0**	**1**	1
0	0	**1**	1	0	1	1
0	**1**	0	0	**1**	**0**	1
0	**1**	0	1	1	0	1
0	**1**	1	0	1	0	1
0	**1**	1	1	1	0	1
1	0	0	0	**1**	**1**	1
1	0	0	1	1	1	1
1	0	1	0	1	1	1
1	0	1	1	1	1	1
1	1	0	0	1	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

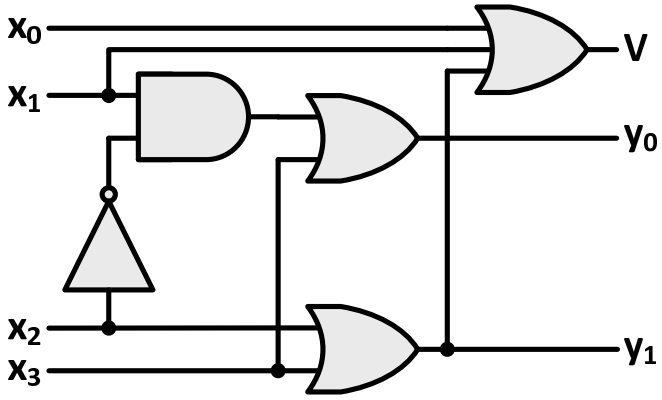
$$V' = x'_3x'_2x'_1x'_0 \Rightarrow V = x_3 + x_2 + x_1 + x_0$$

Κωδικοποιητής προτεραιότητας 4-σε-2



$$y_1 = x_2 + x_3$$





$$y_0 = x_3 + x_1x'_2$$

Παράδειγμα απλού κωδικοποιητή

Σύνθεση κωδικοποιητή των δεκαδικών ψηφίων σε δυαδικούς αριθμούς

Ο ζητούμενος κωδικοποιητής διαθέτει 10 εισόδους, μία για καθένα από τα 10 ψηφία (0 έως 9) και 4 εξόδους στις οποίες λαμβάνονται οι αντίστοιχοι δυαδικοί αριθμοί. Επομένως, πρόκειται για απλό κωδικοποιητή 10-σε-4.

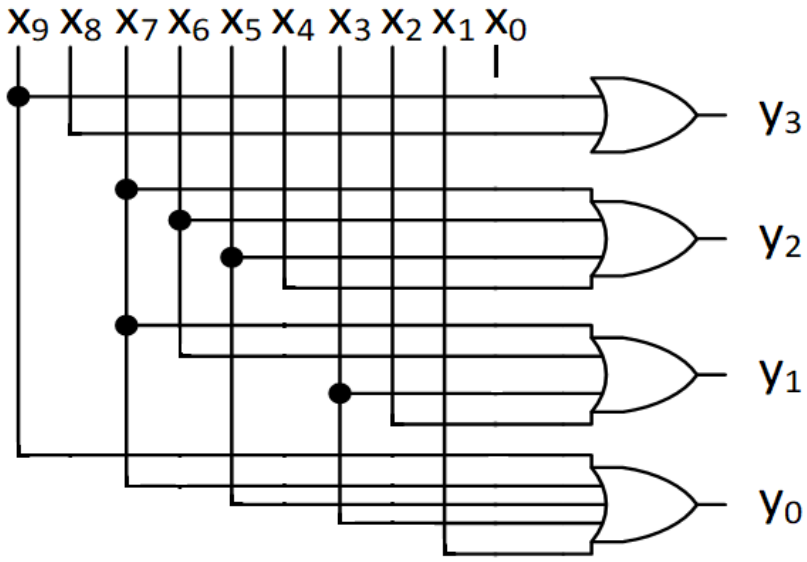
Λόγω του ότι το πλήθος των ψηφίων εξόδου δεν επαρκεί για την παράσταση του μεγαλύτερου πλήθους των δεδομένων εισόδου, μόνο ένα από τα ψηφία εισόδου μπορεί να έχει λογική τιμή 1, έτσι ώστε το κύκλωμα να παράγει στην έξοδό του ένα δυαδικό αριθμό 4 ψηφίων, που αντιστοιχεί στο δεκαδικό ψηφίο της εισόδου με λογική τιμή 1.

Παράδειγμα απλού κωδικοποιητή

X_9	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0	y_3	y_2	y_1	y_0	
0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	$y_3 = x_8 + x_9$
0	0	0	0	0	0	0	0	**1**	0	0	0	0	1	
0	0	0	0	0	0	0	**1**	0	0	0	0	1	0	
0	0	0	0	0	0	**1**	0	0	0	0	0	1	1	$y_2 = x_4 + x_5 + x_6 + x_7$
0	0	0	0	0	**1**	0	0	0	0	0	1	0	0	
0	0	0	0	**1**	0	0	0	0	0	0	1	0	1	
0	0	0	**1**	0	0	0	0	0	0	0	1	1	0	$y_1 = x_2 + x_3 + x_6 + x_7$
0	0	**1**	0	0	0	0	0	0	0	0	1	1	1	
0	**1**	0	0	0	0	0	0	0	0	1	0	0	0	
1	0	0	0	0	0	0	0	0	0	1	0	0	1	$y_0 = x_1 + x_3 + x_5 + x_7 + x_9$

Αφού μόνο ένα από τα ψηφία εισόδου μπορεί να έχει λογική τιμή 1, 1014 από τους συνολικά 1024 (= 2^{10}) συνδυασμούς εισόδων δεν επιτρέπονται και οι αντίστοιχοι ελαχιστόροι είναι αδιάφοροι όροι.

Παράδειγμα απλού κωδικοποιητή



Αποκωδικοποιητές

Οι αποκωδικοποιητές (decoders) είναι συνδυαστικά κυκλώματα τα οποία εκτελούν λειτουργία αντίστροφη από εκείνη των κωδικοποιητών.

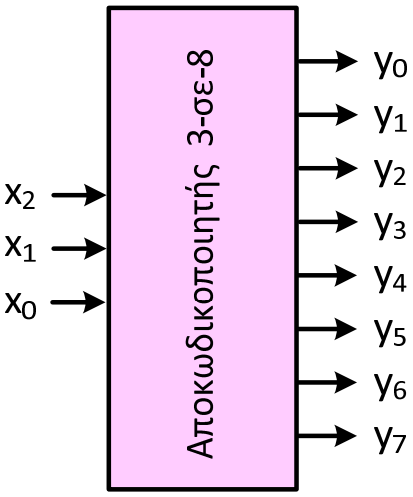
Λαμβάνουν στην είσοδό τους N δυαδικά ψηφία και παράγουν στην έξοδό τους **έως** 2^N δυαδικά ψηφία.

Για κάθε συνδυασμό λογικών τιμών εισόδου, μόνο το ψηφίο εξόδου που αντιστοιχεί σε αυτόν λαμβάνει λογική τιμή 1.

Ουσιαστικά, ένας αποκωδικοποιητής N-σε-2^N αποτελεί γεννήτρια ελαχιστόρων, αφού κάθε έξοδος αντιστοιχεί σε έναν ελαχιστόρο N μεταβλητών εισόδου.

Το λογικό κύκλωμα του αποκωδικοποιητή σχεδιάζεται με βάση τον πίνακα αλήθειας, χρησιμοποιώντας αντιστροφείς για την παραγωγή των συμπληρωματικών μορφών των εισόδων και λογικές πύλες AND για την παραγωγή των ελαχιστόρων.

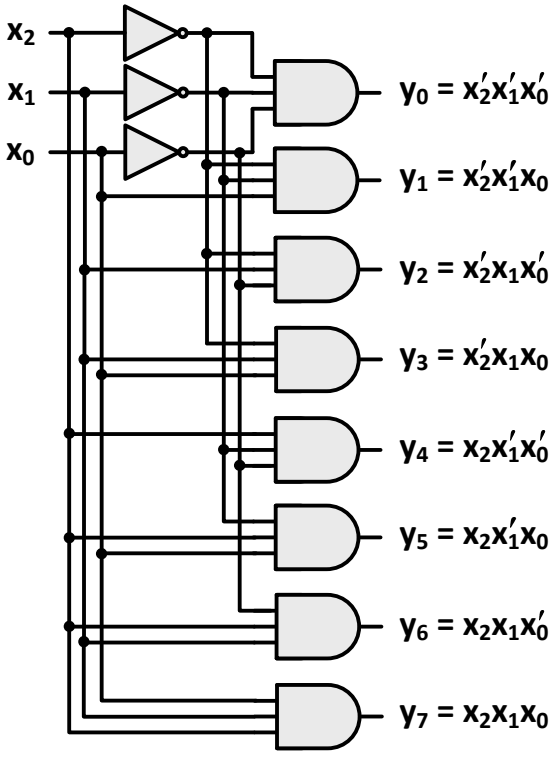
Αποκωδικοποιητής 3-σε-8



x_2	x_1	x_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	**1**	0	0	0	0	0	0	0
0	0	1	0	**1**	0	0	0	0	0	0
0	1	0	0	0	**1**	0	0	0	0	0
0	1	1	0	0	0	**1**	0	0	0	0
1	0	0	0	0	0	0	**1**	0	0	0
1	0	1	0	0	0	0	0	**1**	0	0
1	1	0	0	0	0	0	0	0	**1**	0
1	1	1	0	0	0	0	0	0	0	**1**

Για κάθε συνδυασμό τιμών των μεταβλητών εισόδου, μόνο μία έξοδος λαμβάνει λογική τιμή 1 και παράγει τον αντίστοιχο ελαχιστόρο.

Αποκωδικοποιητής 3-σε-8



Υλοποίηση λογικών συναρτήσεων με αποκωδικοποιητή

Αφού ένας αποκωδικοποιητής N-σε-2^N αποτελεί και γεννήτρια ελαχιστόρων, προκύπτει ότι συνδυάζοντάς τον με μία λογική πύλη OR, η οποία παράγει το λογικό άθροισμα κατάλληλων εξόδων του, μπορούμε να υλοποιήσουμε οποιαδήποτε λογική συνάρτηση σε μορφή αθροίσματος ελαχιστόρων.

Για να γίνει αυτό, πρέπει το πλήθος των εισόδων του αποκωδικοποιητή να ισούται με το πλήθος των μεταβλητών της συνάρτησης και το πλήθος των εισόδων της πύλης OR να ισούται με το πλήθος των ελαχιστόρων που συμμετέχουν στη συνάρτηση.

Υλοποίηση λογικών συναρτήσεων με αποκωδικοποιητή

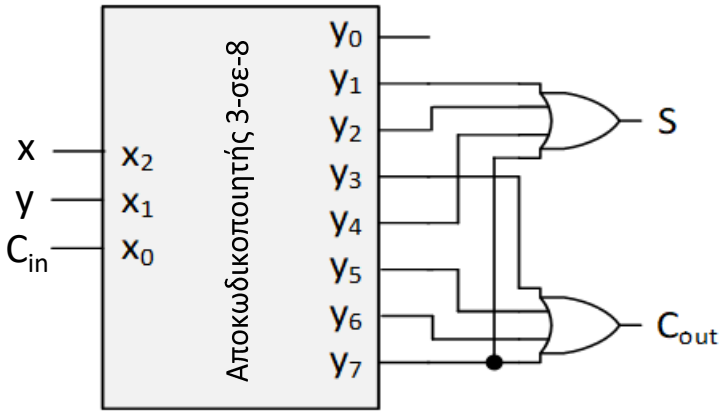
Παράδειγμα: Υλοποίηση πλήρους αθροιστή (συνάρτηση αθροίσματος και συνάρτηση κρατουμένου εξόδου) με κατάλληλο αποκωδικοποιητή και πύλες OR.

Ο πλήρης αθροιστής έχει 3 εισόδους (x, y, C_{in}) και 2 εξόδους (C_{out}, S).

Επομένως για την υλοποίησή του απαιτούνται ένας αποκωδικοποιητής 3-σε-8 και 2 πύλες OR.

Λογικές συναρτήσεις των 2 εξόδων του πλήρους αθροιστή:

$S = \Sigma(1,2,4,7)$ και $C_{out} = \Sigma(3,5,6,7)$



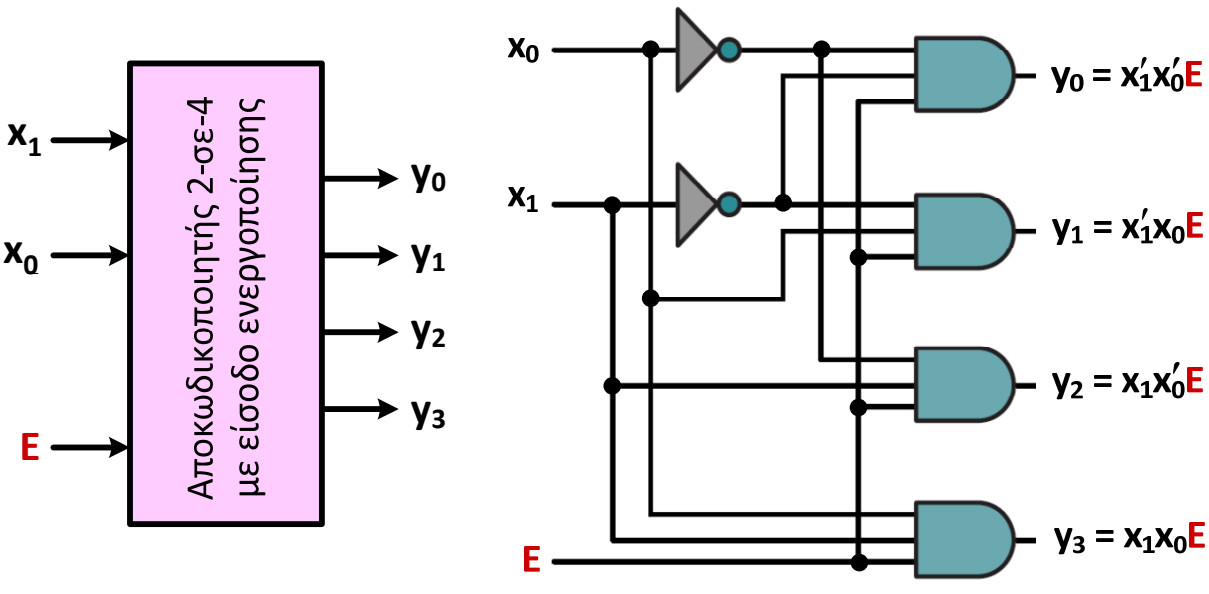
Αποκωδικοποιητής με είσοδο ενεργοποίησης

Εκτός από τα ψηφία εισόδου που πρόκειται να αποκωδικοποιηθούν, το κύκλωμα ενός αποκωδικοποιητή μπορεί να περιλαμβάνει μία ακόμη είσοδο ενεργοποίησης (enable input) E, ανάλογα με την τιμή της οποίας να επιτρέπεται ή όχι η αποκωδικοποίηση.

Η προσθήκη αυτή προϋποθέτει τη χρήση πυλών AND με μία επιπλέον είσοδο.

Όταν E = 0 οι έξοδοι του κυκλώματος μηδενίζονται, ενώ όταν E = 1 οι είσοδοί του αποκωδικοποιούνται με βάση τον πίνακα αλήθειας που περιγράφει τη λειτουργία του.

Αποκωδικοποιητής με είσοδο ενεργοποίησης

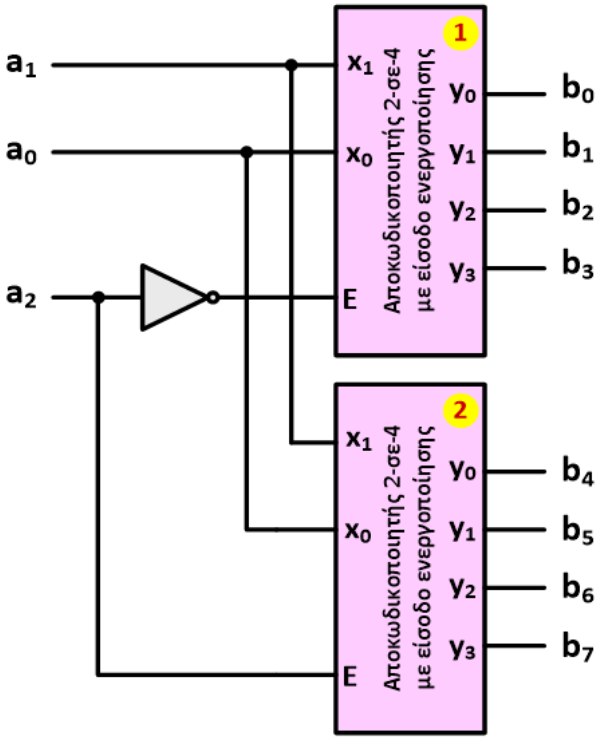


Σύνθεση πολύπλοκων αποκωδικοποιητών

Η είσοδος E δίνει τη δυνατότητα συνδυασμού αποκωδικοποιητών για τη σύνθεση αποκωδικοποιητών με περισσότερες εισόδους.

Στη σχεδίαση αποκωδικοποιητή 3-σε-8 με 2 αποκωδικοποιητές 2-σε-4, όταν $a_2 = 0$, ενεργοποιείται ο πρώτος αποκωδικοποιητής 2-σε-4 και παράγει στις εξόδους b_0 έως b_3 τους ελαχιστόρους m_0 έως m_3.

Όταν $a_2 = 1$, ενεργοποιείται ο δεύτερος αποκωδικοποιητής 2-σε-4 και παράγει στις εξόδους b_4 έως b_7 τους ελαχιστόρους m_4 έως m_7.



Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Μια χρήσιμη εφαρμογή των αποκωδικοποιητών είναι η μετατροπή αλφαριθμητικών χαρακτήρων σε μορφή κατάλληλη για προβολή σε ηλεκτρονικές ψηφιακές συσκευές, όπως είναι οι αριθμομηχανές και τα ψηφιακά ρολόγια.

Σε τέτοιες συσκευές χρησιμοποιούνται συνήθως ενδείκτες 7 τμημάτων (7-segment displays) που αποτελούνται από 7 διόδους φωτοεκπομπής.

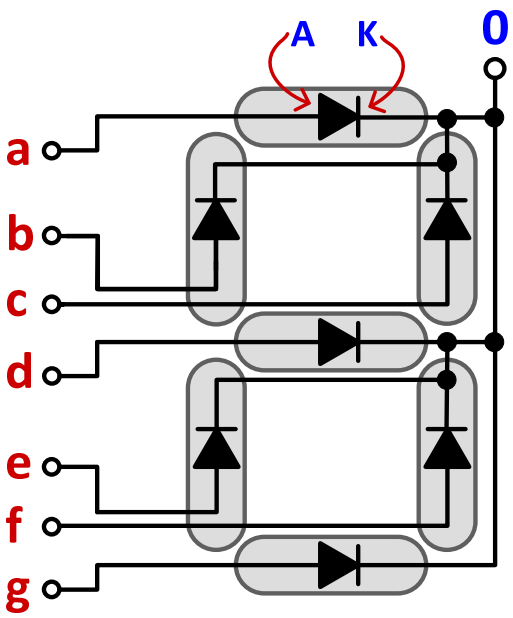
Κάθε τμήμα του ενδείκτη ενεργοποιείται (φωτοβολεί) ανεξάρτητα και ο ενδείκτης απεικονίζει δεκαδικά ψηφία ή/και άλλους χαρακτήρες.

Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Καθεμία από τις διόδους φωτοβολεί, όταν στην άνοδό (Α) εφαρμόζεται επαρκώς μεγαλύτερη τάση από εκείνη που εφαρμόζεται στην κάθοδο (Κ).

Οι κάθοδοι των διόδων του ενδείκτη 7 τμημάτων τροφοδοτούνται με λογική τιμή 0, που αντιστοιχεί στη χαμηλή στάθμη τάσης (γείωση).

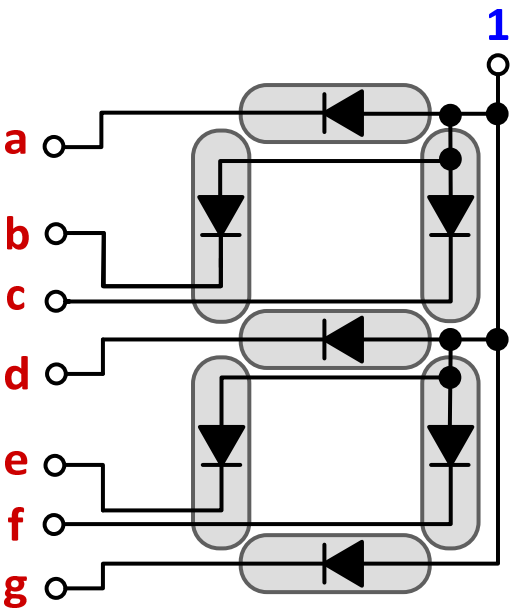
Αυτό έχει ως αποτέλεσμα, όταν η άνοδος μίας διόδου τροφοδοτηθεί με λογική τιμή 1 (που αντιστοιχεί στην υψηλή στάθμη τάσης), αυτή να φωτοβολεί και το αντίστοιχο τμήμα του ενδείκτη να «ανάβει».



Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Εναλλακτικά, ο ενδείκτης μπορεί να δομηθεί έτσι ώστε οι άνοδοι των διόδων των τμημάτων του να τροφοδοτούνται μόνιμα με λογικό 1, που αντιστοιχεί στην υψηλή στάθμη τάσης.

Αυτό έχει ως αποτέλεσμα, όταν η κάθοδος μίας διόδου τροφοδοτηθεί με λογική τιμή 0 (που αντιστοιχεί στη χαμηλή στάθμη τάσης), αυτή να φωτοβολεί και το αντίστοιχο τμήμα του ενδείκτη να «ανάβει».



Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Παράδειγμα 1: Κατάστρωση του πίνακα αλήθειας αποκωδικοποιητή των δεκαδικών ψηφίων BCD σε ενδείκτη 7 τμημάτων και μελέτη επιλογών υλοποίησής του.

Ο κώδικας BCD κωδικοποιεί τα δεκαδικά ψηφία 0 έως 9 σε δυαδική μορφή, χρησιμοποιώντας 4 δυαδικά ψηφία για κάθε δεκαδικό ψηφίο.

Επομένως, ο ζητούμενος αποκωδικοποιητής έχει 4 εισόδους και 7 εξόδους (για την τροφοδότηση των 7 τμημάτων του ενδείκτη)

Δεκαδικός	x_3	x_2	x_1	x_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	1	0	0	1	0
2	0	0	1	0	1	0	1	1	1	0	1
3	0	0	1	1	1	0	1	1	0	1	1
4	0	1	0	0	0	1	1	1	0	1	0
5	0	1	0	1	1	1	0	1	0	1	1
6	0	1	1	0	1	1	0	1	1	1	1
7	0	1	1	1	1	0	1	0	0	1	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Οι τιμές των εξόδων για τους υπόλοιπους 6 συνδυασμούς των εισόδων είναι αδιάφορες.

Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Για την υλοποίηση του ζητούμενου αποκωδικοποιητή, έχουμε 2 επιλογές.

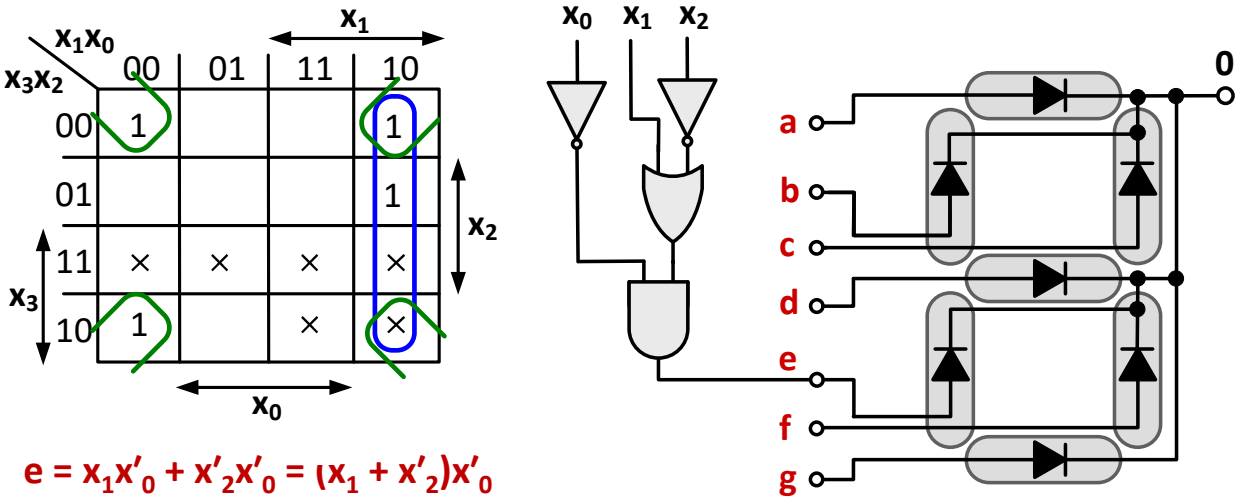
Η πρώτη επιλογή είναι να καταστρώσουμε για τις 7 εξόδους (a, b, c, d, e, f, g) ισάριθμους χάρτες Karnaugh, να εξάγουμε τις ελαχιστοποιημένες λογικές συναρτήσεις τους και να τις υλοποιήσουμε με τον μικρότερο δυνατό αριθμό λογικών πυλών.

Η δεύτερη επιλογή είναι να χρησιμοποιήσουμε έναν τυποποιημένο αποκωδικοποιητή 4-σε-16 (ή να συνθέσουμε έναν αποκωδικοποιητή 4-σε-10) στις εξόδους του οποίου λαμβάνουμε τους ελαχιστόρους που αντιστοιχούν στους 10 συνδυασμούς των 4 εισόδων που κωδικοποιούν τα 10 δεκαδικά ψηφία.

Στη συνέχεια, για την παραγωγή καθεμιάς από τις 7 εξόδους (a, b, c, d, e, f, g) χρησιμοποιούμε μια πύλη OR, ώστε να λάβουμε το άθροισμα ελαχιστόρων που αντιστοιχεί σε κάθε έξοδο, σύμφωνα με τον πίνακα αλήθειας.

Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Υλοποίηση της συνάρτησης της εξόδου e με την πρώτη επιλογή.



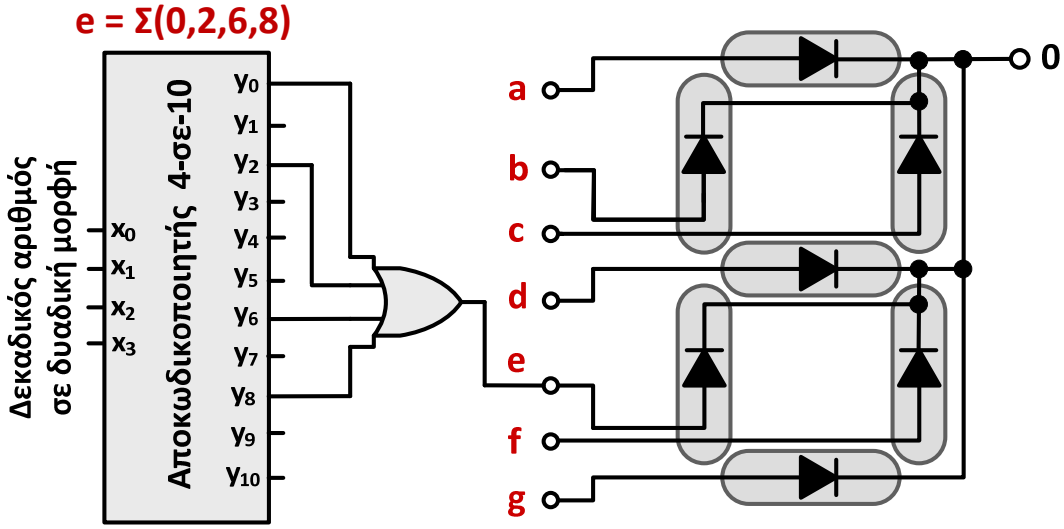
$e = x_1x'_0 + x'_2x'_0 = (x_1 + x'_2)x'_0$

Με όμοιο τρόπο υλοποιούνται οι συναρτήσεις των υπόλοιπων εξόδων.

Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Υλοποίηση της συνάρτησης της εξόδου e με την δεύτερη επιλογή.

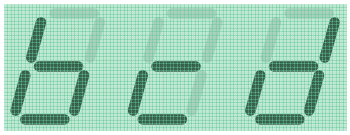
$e = \Sigma(0,2,6,8)$



Με όμοιο τρόπο υλοποιούνται οι συναρτήσεις των υπόλοιπων εξόδων και επειδή στον πίνακα αλήθειας οι μονάδες είναι περισσότερες από τα μηδενικά, μπορούμε να χρησιμοποιήσουμε την εναλλακτική δομή του ενδείκτη, στην οποία οι δίοδοι φωτοβολούν όταν εφαρμόζεται λογική τιμή 0 στην κάθοδό τους.

Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Παράδειγμα 2: Υλοποίηση με τις προαναφερόμενες δύο επιλογές ενός αποκωδικοποιητή που να εμφανίζει σε ενδείκτη 7 τμημάτων τους παρακάτω χαρακτήρες:



Αφού έχουμε 3 χαρακτήρες, απαιτούνται 2 μεταβλητές εισόδου, από τις οποίες προκύπτουν 4 συνδυασμοί.

Έτσι, ο ζητούμενος αποκωδικοποιητής διαθέτει 2 εισόδους και 7 εξόδους (a, b, c, d, e, f, g).

Για την υλοποίηση του αποκωδικοποιητή, θα χρησιμοποιήσουμε τους 3 συνδυασμούς (έναν για κάθε χαρακτήρα) και οι τιμές των εξόδων που αντιστοιχούν στον τέταρτο συνδυασμό είναι αδιάφορες.

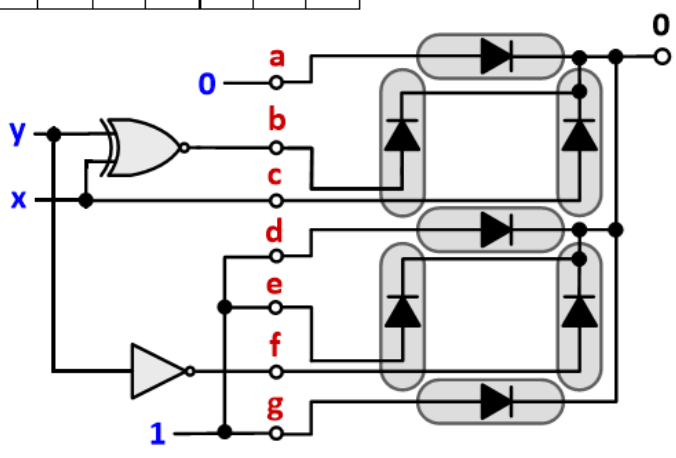
Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Υλοποίηση των συναρτήσεων των 7 εξόδων με την πρώτη επιλογή.

x	y		a	b	c	d	e	f	g
0	0	**b**	0	1	0	1	1	1	1
0	1	**c**	0	0	0	1	1	0	1
1	0	**d**	0	0	1	1	1	1	1
1	1		×	×	×	×	×	×	×

a = 0, b = (x ⊕ y)', c = x,

d = e = g = 1, f = y'

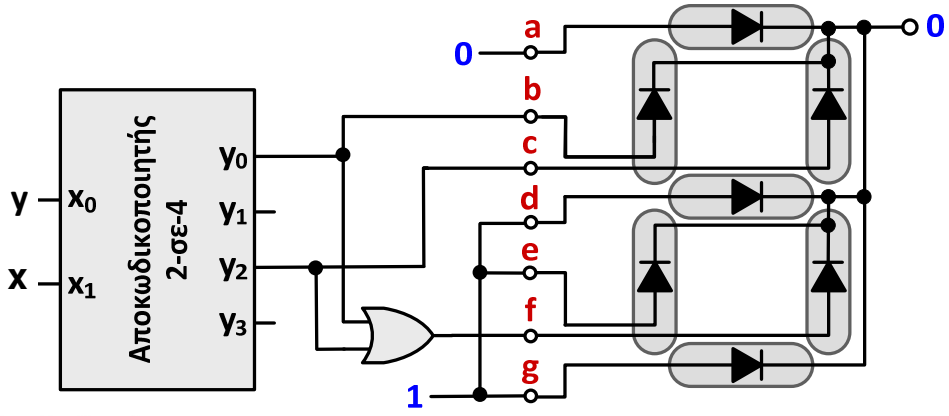


Αποκωδικοποιητές χαρακτήρων σε ενδείκτη 7 τμημάτων

Υλοποίηση των συναρτήσεων των 7 εξόδων με τη δεύτερη επιλογή.

Για την υλοποίηση του ζητούμενου αποκωδικοποιητή θα χρησιμοποιήσουμε έναν τυποποιημένο αποκωδικοποιητή 2-σε-4, λαμβάνοντας υπόψη ότι οι συναρτήσεις των εξόδων όπως προκύπτουν από τον πίνακα αλήθειας είναι οι εξής:

$$a = 0,\ b = m_0,\ c = m_2,\ d = e = g = 1,\ f = m_0 + m_2$$



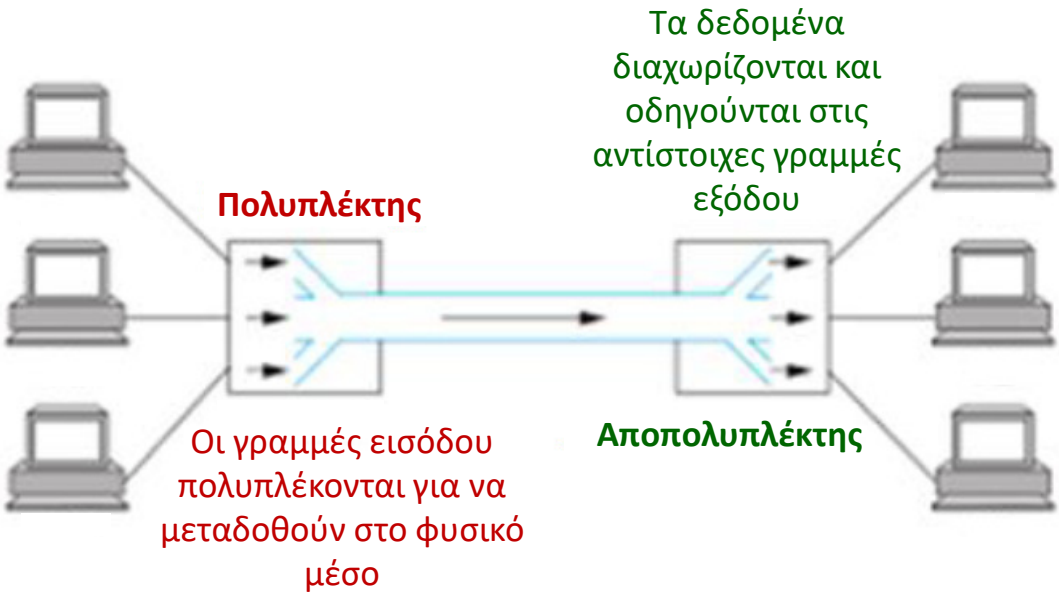
Πολυπλεξία και αποπολυπλεξία

Στις τηλεπικοινωνίες και τα δίκτυα υπολογιστών πολυπλεξία (multiplexing) είναι η τεχνική που επιτρέπει σε ψηφιακά δεδομένα από διαφορετικές πηγές, να μεταδοθούν μέσα από το ίδιο φυσικό μέσο (π.χ. καλώδιο επικοινωνίας), το οποίο έτσι διαμοιράζεται σε πολλαπλούς χρήστες.

Για παράδειγμα, στην πολυπλεξία διαίρεσης χρόνου (time division multiplexing, TDM), ο χρόνος μετάδοσης χωρίζεται σε χρονοθυρίδες (time slots) και κάθε πακέτο δεδομένων μεταδίδεται στο φυσικό μέσο σε συγκεκριμένη χρονοθυρίδα.

Η αντίστροφη διαδικασία (αποπολυπλεξία, demultiplexing) διενεργείται από κάθε δέκτη, για να απομονωθεί και να ληφθεί, το ζητούμενο πακέτο δεδομένων.

Πολυπλεξία και αποπολυπλεξία



Πολυπλέκτες

Οι πολυπλέκτες (multiplexers) ως συνδυαστικά κυκλώματα, λαμβάνουν στις εισόδους τους 2^N δυαδικά ψηφία και μεταφέρουν στη μοναδική έξοδό τους την τιμή ενός από τα ψηφία εισόδου.

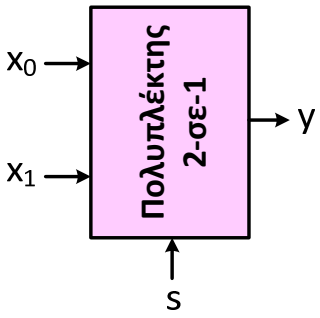
Επομένως, είναι κυκλώματα με 2^N εισόδους και μία έξοδο (πολυπλέκτες 2^N-σε-1).

Για την επιλογή του ψηφίου εισόδου που θα μεταφερθεί στην έξοδό τους, οι πολυπλέκτες διαθέτουν N πρόσθετες εισόδους, οι οποίες αναφέρονται ως είσοδοι επιλογής (select inputs), έτσι ώστε να διακρίνονται από τις 2^N εισόδους δεδομένων (data inputs).

Πολυπλέκτης 2-σε-1

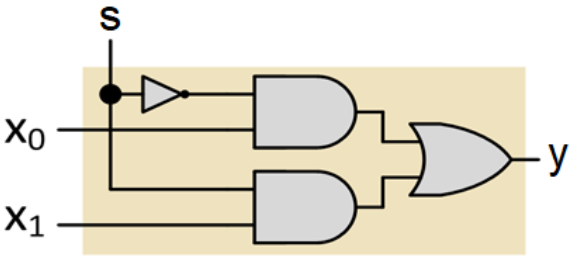
Ο πολυπλέκτης 2-σε-1 είναι ο απλούστερος δυνατός και διαθέτει 2 γραμμές εισόδου δεδομένων (x_0, x_1), μία έξοδο **y** και μια γραμμή επιλογής **s**.

Όταν **s = 0**, μεταφέρεται στην έξοδο y η είσοδος x_0, ενώ όταν **s = 1**, μεταφέρεται στην έξοδο y η είσοδος x_1.



s	y
0	x_0
1	x_1

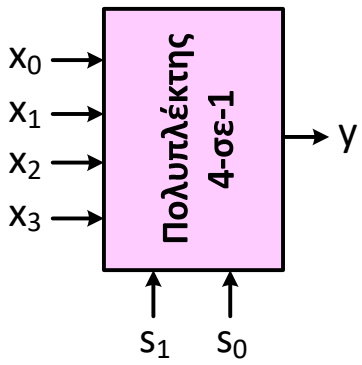
$y = s'x_0 + sx_1$



Πολυπλέκτης 4-σε-1

Ο πολυπλέκτης 4-σε-1 διαθέτει 4 γραμμές εισόδου δεδομένων (x_0, x_1, x_2, x_3) μία έξοδο **y** και δύο γραμμές (s_0, s_1).

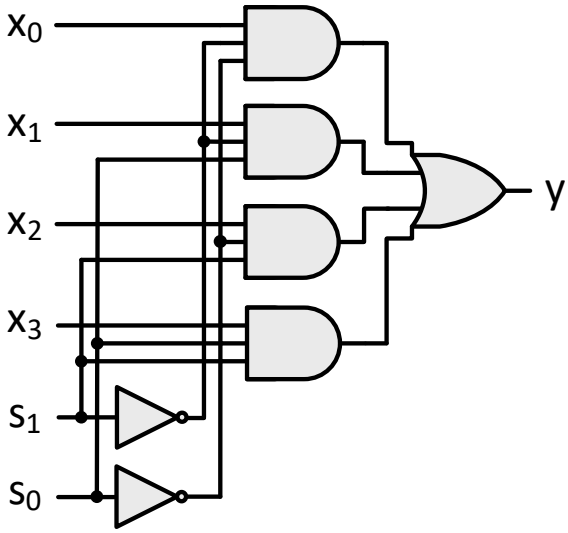
Για καθέναν από τους 4 δυνατούς συνδυασμούς τιμών των 2 εισόδων επιλογής (s_0, s_1), μεταφέρεται στην έξοδο y η αντίστοιχη γραμμή εισόδου.



s_1	s_0	y
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3

$y = s'_1s'_0x_0 + s'_1s_0x_1 + s_1s'_0x_2 + s_1s_0x_3$

Πολυπλέκτης 4-σε-1



Με όμοιο τρόπο μπορούμε να συνθέσουμε πολυπλέκτες με μεγαλύτερο πλήθος εισόδων δεδομένων (1-σε-8, 1-σε-16 κ.ο.κ.), τηρώντας πάντα την αναλογία 2^N / N για τις εισόδους δεδομένων και επιλογής, αντίστοιχα.

Σύνθεση πολυπλεκτών με απλούστερους πολυπλέκτες

Είναι δυνατό να συνθέσουμε πολυπλέκτες πολλών εισόδων με μικρότερους πολυπλέκτες.

Για παράδειγμα, μπορούμε να υλοποιήσουμε έναν πολυπλέκτη 4-σε-1 με πολυπλέκτες 2-σε-1, αρκεί να μετασχηματίσουμε κατάλληλα τη λογική έκφραση της εξόδου ενός πολυπλέκτη 4-σε-1:

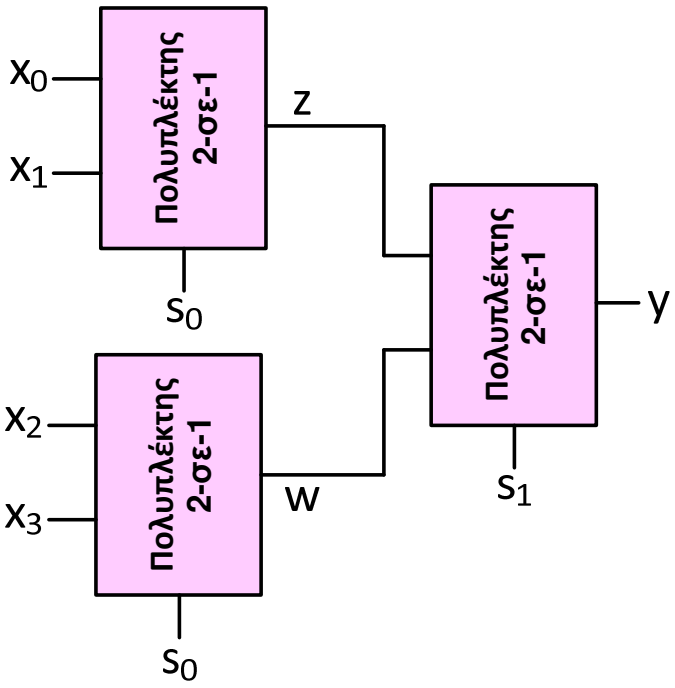
$$y = s'_1 s'_0 x_0 + s'_1 s_0 x_1 + s_1 s'_0 x_2 + s_1 s_0 x_3 = s'_1 (s'_0 x_0 + s_0 x_1) + s_1 (s'_0 x_2 + s_0 x_3)$$
$$= s'_1 z + s_1 w, \text{ όπου: } z = s'_0 x_0 + s_0 x_1 \text{ και } w = s'_0 x_2 + s_0 x_3$$

Παρατηρούμε ότι **z** είναι η έξοδος ενός πολυπλέκτη 2-σε-1 με εισόδους δεδομένων x_0 και x_1 και είσοδο επιλογής s_0.

Ομοίως, **w** είναι η έξοδος ενός πολυπλέκτη 2-σε-1 με εισόδους δεδομένων x_2 και x_3 και είσοδο επιλογής s_0.

Τέλος, **y** είναι η έξοδος ενός πολυπλέκτη 2-σε-1 με εισόδους δεδομένων **z** και **w** (που είναι έξοδοι των 2 προηγούμενων) και είσοδο επιλογής s_1.

Σύνθεση πολυπλεκτών με απλούστερους πολυπλέκτες



Υλοποίηση λογικών συναρτήσεων με πολυπλέκτη

Από τη σχέση που διέπει τη λειτουργία ενός πολυπλέκτη 2^N-σε-1, προκύπτει ότι μπορούμε να παράγουμε στην έξοδό του, οποιοδήποτε άθροισμα ελαχιστόρων των N μεταβλητών που αντιστοιχούν στις εισόδους επιλογής, θέτοντας λογική τιμή 1 στις εισόδους δεδομένων που αντιστοιχούν στους ελαχιστόρους που συμμετέχουν στο επιθυμητό άθροισμα και λογική τομή 0 στους ελαχιστόρους που δε συμμετέχουν.

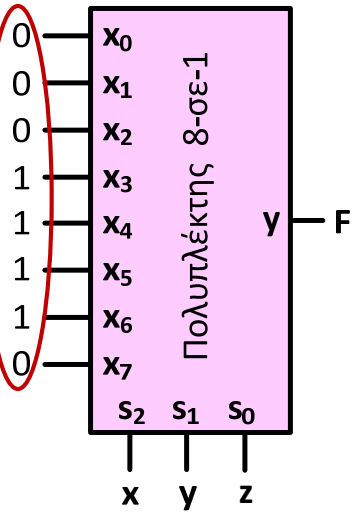
Με τον τρόπο αυτό, χρησιμοποιώντας έναν πολυπλέκτη 2^N-σε-1, μπορούμε να υλοποιήσουμε οποιαδήποτε λογική συνάρτηση N μεταβλητών, αρκεί να τροφοδοτήσουμε με τις μεταβλητές αυτές τις εισόδους επιλογής του πολυπλέκτη και να θέσουμε τις κατάλληλες λογικές τιμές στις εισόδους δεδομένων του πολυπλέκτη.

Υλοποίηση λογικών συναρτήσεων με πολυπλέκτη

Παράδειγμα 1: Υλοποίηση της λογικής συνάρτησης F(x,y,z) = Σ(3,4,5,6) με πολυπλέκτη 8-σε-1.

F(x,y,z) = Σ(3,4,5,6)

x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



Πρόκειται για συνάρτηση με N = 3 μεταβλητές, επομένως με βάση τα προαναφερόμενα υλοποιείται με έναν πολυπλέκτη 2^3-σε 1 (8-σε-1).

Υλοποίηση λογικών συναρτήσεων με πολυπλέκτη

Ωστόσο, προκύπτει, ότι είναι πιο αποδοτικό να υλοποιήσουμε οποιαδήποτε λογική συνάρτηση N μεταβλητών, με έναν μικρότερο πολυπλέκτη 2^{N-1}-σε-1, τροφοδοτώντας τις (N – 1) εισόδους επιλογής του πολυπλέκτη με (N – 1) μεταβλητές της συνάρτησης και χρησιμοποιώντας τη μεταβλητή της συνάρτησης που απομένει για την τροφοδότηση εισόδου ή εισόδων δεδομένων του πολυπλέκτη.

Για να επιτευχθεί η υλοποίηση λογικών συναρτήσεων, ενδέχεται να απαιτηθεί η χρήση ενός αντιστροφέα, έτσι ώστε να παράγεται η συμπληρωματική μορφή της μεταβλητής που συμμετέχει στην τροφοδότηση των εισόδων δεδομένων του πολυπλέκτη.

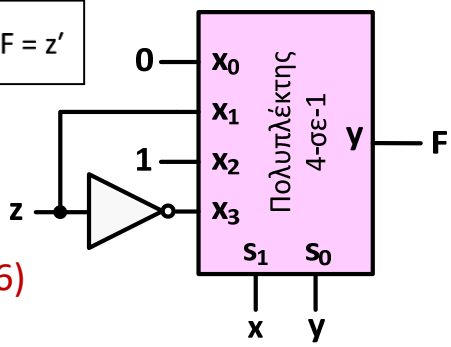
Υλοποίηση λογικών συναρτήσεων με πολυπλέκτη

Παράδειγμα 2: Υλοποίηση της λογικής συνάρτησης F(x,y,z) = Σ(3,4,5,6) με πολυπλέκτη 4-σε-1 και έναν αντιστροφέα (αν απαιτείται).

Τροφοδοτούμε με τις 2 πρώτες μεταβλητές (x, y) τις εισόδους επιλογής του πολυπλέκτη, ενώ τις εισόδους δεδομένων με την κατάλληλη λογική τιμή ή με την κατάλληλη μορφή (κανονική ή συμπληρωματική) της τελευταίας μεταβλητής (z), ανάλογα με την τιμή που λαμβάνει η συνάρτηση για καθέναν από τους 4 συνδυασμούς τιμών των μεταβλητών x και y.

x	y	z	F	
0	0	0	0	F = 0
0	0	1	0	
0	1	0	0	F = z
0	1	1	1	
1	0	0	1	F = 1
1	0	1	1	
1	1	0	1	F = z'
1	1	1	0	

Πίνακας προγραμματισμού του πολυπλέκτη



F(x,y,z) = Σ(3,4,5,6)

Θεώρημα ανάπτυξης λογικών συναρτήσεων (Shannon)

Κάθε λογική συνάρτηση μπορεί να αναπτυχθεί ως προς μία από τις μεταβλητές που συμμετέχουν σε αυτήν, ως εξής:

$$F(x, y, \ldots, w) = xF(1, y, \ldots, w) + x'F(0, y, \ldots, w)$$

Οι συναρτήσεις F(1, y,..., w) και F(0, y,..., w) προκύπτουν από τη συνάρτηση F(x, y..., w) για x = 1 και x = 0, αντίστοιχα.

Οι συναρτήσεις αυτές, ως συναρτήσεις πλέον των μεταβλητών y, ..., w, μπορούν με όμοιο τρόπο να αναπτυχθούν ως προς τη μεταβλητή y κ.ο.κ.

Ανάπτυξη συνάρτησης 3 μεταβλητών:

$$F(x, y, z) = xF(1, y, z) + x'F(0, y, z) =$$
$$x[yF(1, 1, z) + y'F(1, 0, z)] + x'[yF(0, 1, z) + y'F(0,0,z)]$$

Υλοποίηση συναρτήσεων με πολυπλέκτες 2-σε-1

Από την ανάπτυξη μιας λογικής συνάρτησης σύμφωνα με το θεώρημα Shannon, προκύπτει μια μορφή της συνάρτησης που είναι άμεσα υλοποιήσιμη με πολυπλέκτες 2-σε-1:

$$F(x, y, z) = x[yF(1, 1, z) + y'F(1, 0, z)] + x'[yF(0, 1, z) + y'F(0, 0, z)]$$

Οι συναρτήσεις στις αγκύλες υλοποιούνται με έναν πολυπλέκτη 2-σε-1 η καθεμία.

Η είσοδος επιλογής των δύο πολυπλεκτών τροφοδοτείται με τη μεταβλητή y.

Οι είσοδοι δεδομένων των δύο πολυπλεκτών τροφοδοτούνται με τις συναρτήσεις F(1, 1, z), F(1, 0, z), F(0, 1, z), F(0, 0, z), οι οποίες ισούνται με 0 ή 1 ή z ή z'. Οι συναρτήσεις αυτές προκύπτουν εύκολα από τον πίνακα αλήθειας της συνάρτησης F(x, y, z).

Οι έξοδοι των δύο πολυπλεκτών τροφοδοτούν τις εισόδους δεδομένων ενός τρίτου πολυπλέκτη 2-σε-1 με είσοδο επιλογής τη μεταβλητή x.

Υλοποίηση συναρτήσεων με πολυπλέκτες 2-σε-1

Παράδειγμα 3: Υλοποίηση της λογικής συνάρτησης F(x,y,z) = Σ(3,4,5,6) με πολυπλέκτες 2-σε-1 και έναν αντιστροφέα (αν απαιτείται).

$$F(x, y, z) = x[yF(1, 1, z) + y'F(1, 0, z)] + x'[yF(0, 1, z) + y'F(0, 0, z)]$$

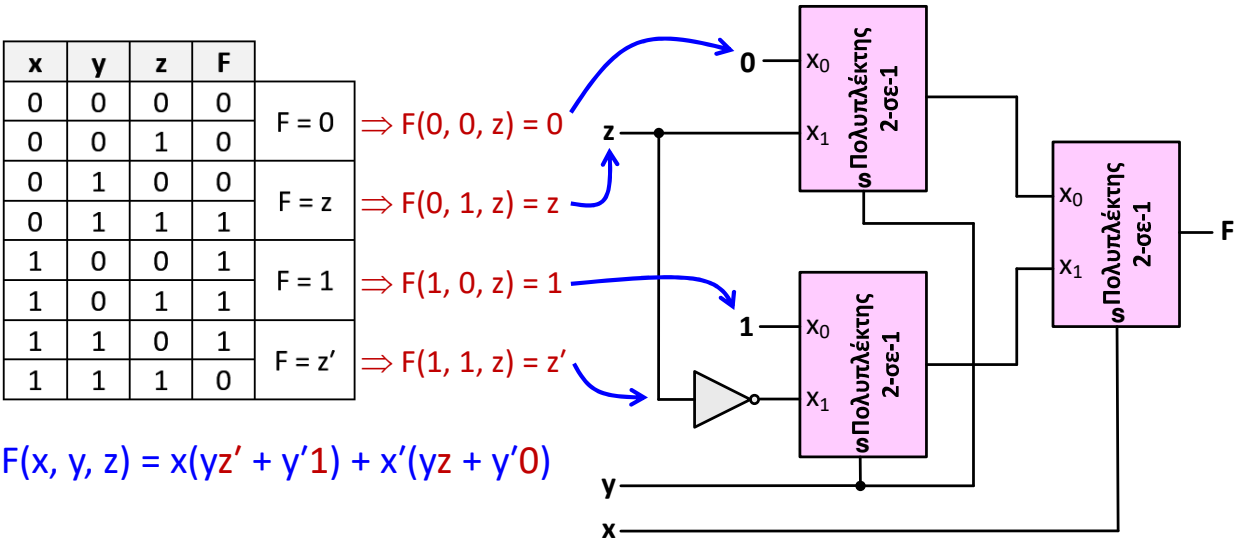
x	y	z	F	
0	0	0	0	F = 0
0	0	1	0	
0	1	0	0	F = z
0	1	1	1	
1	0	0	1	F = 1
1	0	1	1	
1	1	0	1	F = z'
1	1	1	0	

⇒ F(0, 0, z) = 0

⇒ F(0, 1, z) = z

⇒ F(1, 0, z) = 1

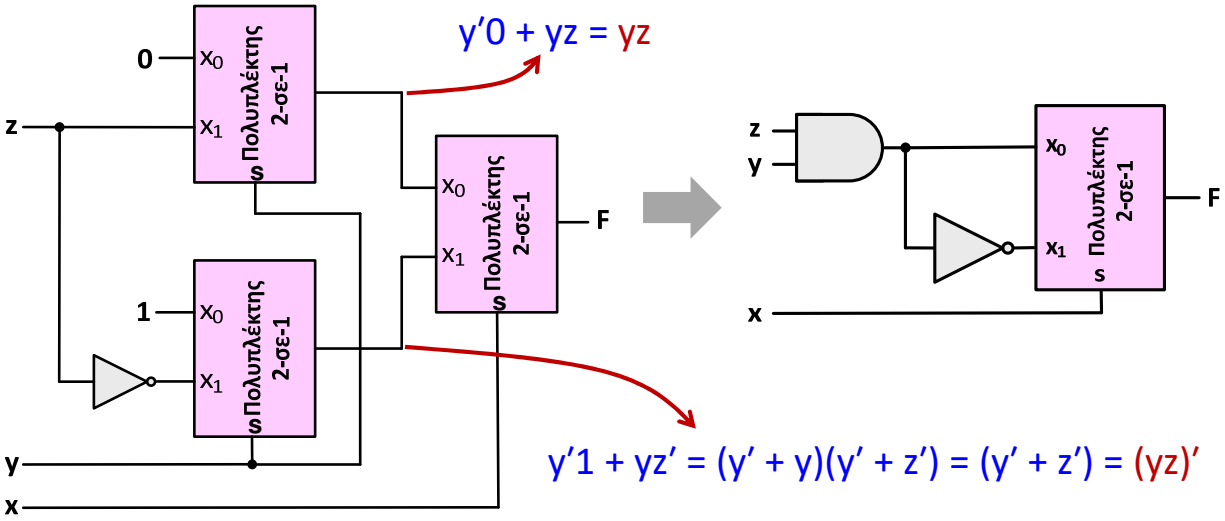
⇒ F(1, 1, z) = z'



$$F(x, y, z) = x(yz' + y'1) + x'(yz + y'0)$$

Υλοποίηση συναρτήσεων με πολυπλέκτες 2-σε-1

Εάν είναι διαθέσιμες πρόσθετες λογικές πύλες, όταν είσοδοι δεδομένων των πολυπλεκτών του πρώτου επιπέδου τροφοδοτούνται με λογικές τιμές 0 ή 1, αντικαθιστώντας πολυπλέκτες με λογικές πύλες, μπορούμε να πετύχουμε απλούστερη υλοποίηση με μικρότερο πλήθος πολυπλεκτών:



Υλοποίηση συναρτήσεων με πολυπλέκτες 2-σε-1

Με την προαναφερόμενη μεθοδολογία, μπορεί να υλοποιηθεί οποιαδήποτε λογική συνάρτηση N μεταβλητών, χρησιμοποιώντας N – 1 επίπεδα πολυπλεκτών 2-σε-1.

Στο πρώτο επίπεδο συμμετέχουν έως 2^{N-2} πολυπλέκτες 2-σε-1, με το πλήθος τους να υποδιπλασιάζεται σε κάθε επόμενο επίπεδο, έως το τελευταίο επίπεδο, το οποίο περιλαμβάνει έναν πολυπλέκτη 2-σε-1.

Στην περίπτωση όπου από την ανάπτυξη μιας λογικής συνάρτησης προκύπτει ότι οι είσοδοι δεδομένων ενός πολυπλέκτη 2-σε-1 τροφοδοτούνται με την ίδια λογική τιμή, μεταβλητή ή συμπληρωματική μορφή μεταβλητής, ο αντίστοιχος πολυπλέκτης απαλείφεται, οδηγώντας σε απλούστερο κύκλωμα υλοποίησης.

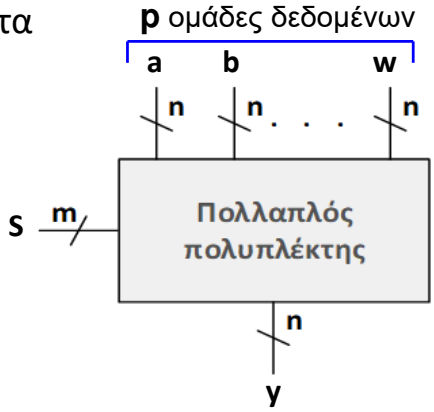
Πολυπλέκτες επιλογής πολλαπλών εισόδων

Σε αρκετές εφαρμογές, απαιτούνται κυκλώματα για την επιλογή ομάδας δεδομένων.

Στα κυκλώματα αυτά, που αναφέρονται ως πολυπλέκτες επιλογής πολλαπλών εισόδων (multiple-input selection multiplexers) ή απλούστερα ως πολλαπλοί πολυπλέκτες, συνδυάζονται πολυπλέκτες που χρησιμοποιούν κοινές εισόδους επιλογής.

Για τη σύνθεσή τους απαιτείται ο συνδυασμός τόσων πολυπλεκτών όσες είναι οι είσοδοι δεδομένων κάθε ομάδας.

Το πλήθος των εισόδων δεδομένων κάθε πολυπλέκτη ταυτίζεται με το πλήθος των ομάδων δεδομένων.



Πλήθος ψηφίων ομάδας δεδομένων = Πλήθος πολυπλεκτών = **n**

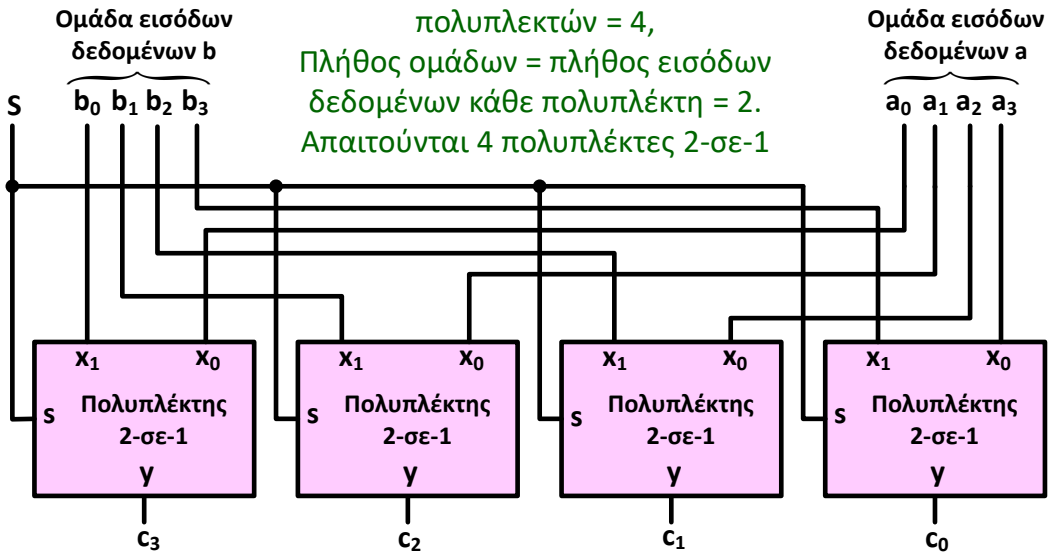
Πλήθος ομάδων = Πλήθος εισόδων δεδομένων κάθε πολυπλέκτη = **p**

Πλήθος εισόδων επιλογής κάθε πολυπλέκτη = **m**, $\mathbf{p \le 2^m}$.

Πολυπλέκτες επιλογής πολλαπλών εισόδων

Παράδειγμα: Σύνθεση ενός τετραπλού πολυπλέκτη 2-σε-1, δηλαδή ενός πολυπλέκτη που να επιλέγει μία από 2 ομάδες ψηφίων που η καθεμία περιλαμβάνει 4 ψηφία και να την μεταφέρει σε 4 εξόδους.

Πλήθος ψηφίων ομάδας = πλήθος πολυπλεκτών = 4,
Πλήθος ομάδων = πλήθος εισόδων δεδομένων κάθε πολυπλέκτη = 2.
Απαιτούνται 4 πολυπλέκτες 2-σε-1



Αποπολυπλέκτες

Οι αποπολυπλέκτες (demultiplexers) επιτελούν την αντίστροφη λειτουργία από εκείνη των πολυπλεκτών.

Διαθέτουν μία είσοδο δεδομένων και 2^N εξόδους δεδομένων, σε μία από τις οποίες μεταφέρεται η είσοδος δεδομένων, ανάλογα με το συνδυασμό τιμών των N εισόδων επιλογής.

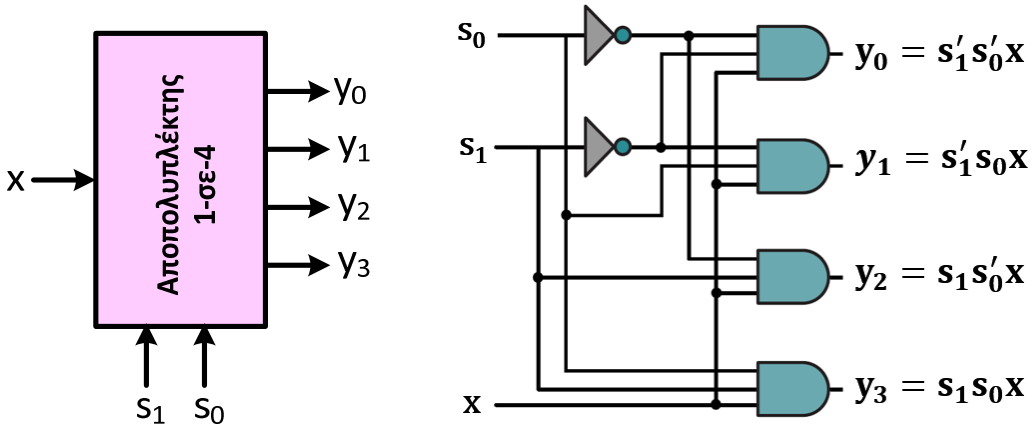
Ο αποκωδικοποιητής με είσοδο ενεργοποίησης που παρουσιάστηκε, γίνεται αποπολυπλέκτης, εάν οι είσοδοι x_1, x_0 είναι είσοδοι επιλογής και η είσοδος E είναι είσοδος δεδομένων.

Το πλήθος των εξόδων που μπορούν να ελεγχθούν καθορίζεται από το πλήθος των εισόδων επιλογής.

Έτσι, N είσοδοι επιλογής, μπορούν να ελέγξουν μέχρι 2^N εξόδους (αποπολυπλέκτης 1-σε-2^N).

Αποπολυπλέκτες

Ο αποκωδικοποιητής με είσοδο ενεργοποίησης που προαναφέρθηκε, γίνεται αποπολυπλέκτης, εάν οι είσοδοι x_1, x_0 είναι είσοδοι επιλογής και η είσοδος E είναι είσοδος δεδομένων.



Με όμοιο τρόπο μπορούμε να συνθέσουμε αποπολυπλέκτες με μικρότερο ή μεγαλύτερο πλήθος εξόδων (1-σε-2, 1-σε-8, 1-σε-16 κ.ο.κ.), τηρώντας πάντα την αναλογία 2^N / N για τις εξόδους και τις εισόδους επιλογής, αντίστοιχα.

Αποπολυπλέκτες

Παράδειγμα:

Σχεδίαση του συνοπτικού διαγράμματος διασύνδεσης ενός πολυπλέκτη 16-σε-1 και ενός αποπολυπλέκτη 1-σε-32 και προσδιορισμός του πλήθους των εισόδων επιλογής του πολυπλέκτη και του αποπολυπλέκτη.

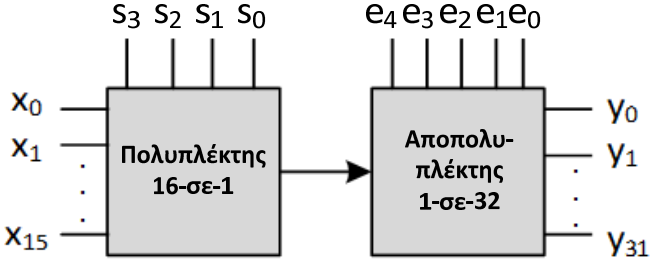
Προσδιορισμός των συνδυασμών τιμών των εισόδων επιλογής του πολυπλέκτη και του αποπολυπλέκτη, εάν είναι επιθυμητή η μεταφορά της τιμής της εισόδου δεδομένων x_{13} του πολυπλέκτη στην έξοδο y_{27} του αποπολυπλέκτη.

Αποπολυπλέκτες

Για τη διασύνδεση του πολυπλέκτη με τον αποπολυπλέκτη απαιτείται απλώς η σύνδεση της μοναδικής εξόδου του πολυπλέκτη στην μοναδική είσοδο του αποπολυπλέκτη.

Για να επιλεγεί μία από τις 16 εισόδους δεδομένων του πολυπλέκτη απαιτούνται 4 είσοδοι επιλογής, αφού $2^4 = 16$. Ομοίως, για να επιλεγεί μία από τις 32 εξόδους του αποπολυπλέκτη απαιτούνται 5 είσοδοι επιλογής, αφού $2^5 = 32$.

Για να επιλεγεί η είσοδος x_{13} του πολυπλέκτη πρέπει $s_3s_2s_1s_0 = 1101$ (= 13_{10}), ενώ για να επιλεγεί η έξοδος y_{27} του αποπολυπλέκτη πρέπει $e_4e_3e_2e_1e_0 = 11011$ (= 27_{10}).



Βιβλιογραφία

M. Mano, M.D. Ciletti, Ψηφιακή σχεδίαση (6η έκδοση), Εκδόσεις Παπασωτηρίου, 2018.

Μ. Ρουμελιώτης, Σ. Σουραβλάς, Ψηφιακή σχεδίαση: αρχές και εφαρμογές. Εκδόσεις Τζιόλα, 2013.

V.P. Nelson, H. Troy Nagle, J. David Irwin, B.D. Carrol, Ανάλυση και σχεδίαση κυκλωμάτων ψηφιακής λογικής, Εκδόσεις Επίκεντρο, 2007.

J.F. Wakerly, Ψηφιακή σχεδίαση: αρχές και πρακτικές (5η έκδοση), Εκδόσεις Κλειδάριθμος, 2019.

Λ. Μπισδούνης, Ψηφιακά συστήματα, Βασικές εξειδικεύσεις σε αρχιτεκτονική & δίκτυα υπολογιστών, Ελληνικό Ανοικτό Πανεπιστήμιο, 2015 (ανατύπωση 2017).