Συνδυαστικά Λογικά Κυκλώματα

Ένα συνδυαστικό λογικό κύκλωμα συντίθεται από λογικές πύλες, δέχεται εισόδους και παράγει μία ή περισσότερες εξόδους.

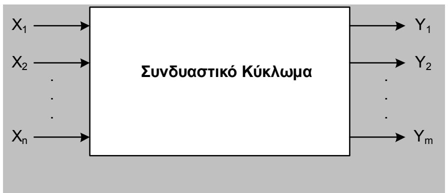
Στα συνδυαστικά λογικά κυκλώματα οι έξοδοι σε κάθε χρονική στιγμή καθορίζονται μόνο από τον τρέχοντα συνδυασμό τιμών των εισόδων.

Για κάθε ***ν*** μεταβλητές εισόδου, υπάρχουν $\mathbf{2}^{\nu}$ δυαδικοί συνδυασμοί τιμών των εισόδων.

Σε κάθε συνδυαστικό κύκλωμα, για κάθε συνδυασμό τιμών των εισόδων, παράγεται ένας αντίστοιχος συνδυασμός τιμών των εξόδων.

Η λειτουργία ενός συνδυαστικού κυκλώματος καθορίζεται από ένα πίνακα αλήθειας, ο οποίος δίνει τις τιμές των εξόδων για κάθε ένα συνδυασμό τιμών των εισόδων.

Η λειτουργία ενός συνδυαστικού κυκλώματος μπορεί επίσης να περιγραφεί από λογικές συναρτήσεις Boole (μία συνάρτηση για κάθε μία έξοδο). Κάθε συνάρτηση εξόδου έχει ως ανεξάρτητες μεταβλητές τις ***ν*** μεταβλητές εισόδου.



Χονδρικό Διάγραμμα (Block Diagram)

Ανάλυση συνδυαστικών κυκλωμάτων

Η ανάλυση ενός κυκλώματος έχει σκοπό τον προσδιορισμό της λειτουργίας που εκτελεί.

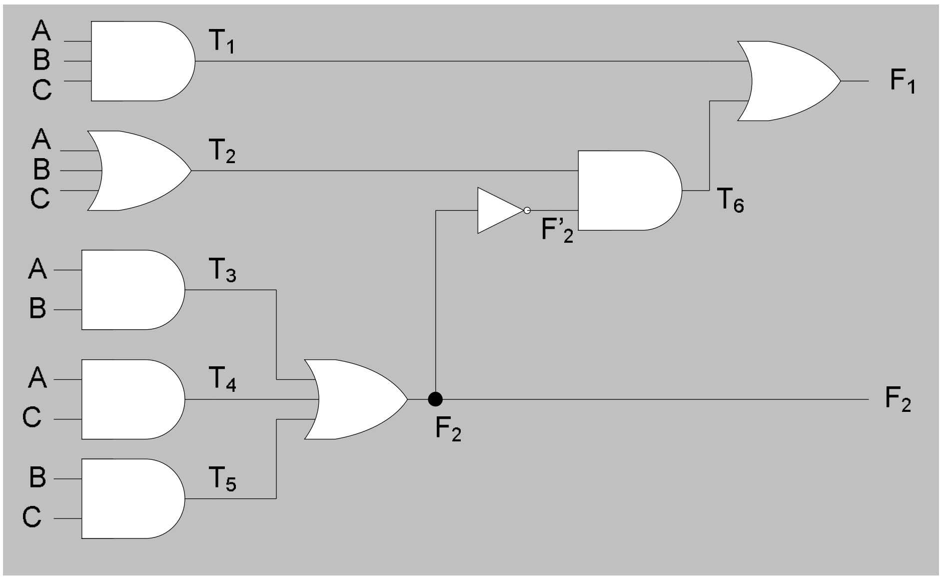
Στην ανάλυση συνήθως ξεκινάμε από το λογικό κύκλωμα και καταλήγουμε σε ένα σύνολο λογικών συναρτήσεων, στον πίνακα αλήθειας του κυκλώματος και, πιθανώς, σε μια λεκτική περιγραφή της λειτουργίας του κυκλώματος.

Για να προσδιορίσουμε τις λογικές συναρτήσεις των εξόδων ενός συνδυαστικού λογικού κυκλώματος από το λογικό κύκλωμά του, ακολουθούμε τα εξής βήματα:

1. Ονομάζουμε τις εισόδους και τις εξόδους του κυκλώματος, καθώς και τις εξόδους όλων των λογικών πυλών του κυκλώματος.
2. Ξεκινώντας από τις λογικές πύλες που δέχονται ως είσοδο τις εισόδους του κυκλώματος (τις μεταβλητές εισόδου), προσδιορίζουμε σταδιακά τις εξόδους όλων των λογικών πυλών ως συναρτήσεις των εισόδων που δέχονται (ενδιάμεσες μεταβλητές ή συναρτήσεις).
3. Επαναλαμβάνουμε τη διαδικασία του προηγούμενου βήματος μέχρι να προσδιορίσουμε τις λογικές συναρτήσεις όλων των εξόδων του κυκλώματος.

4. Με επανειλημμένες αντικαταστάσεις, από το τέλος προς την αρχή (από τις εξόδους προς τις εισόδους του κυκλώματος), των ενδιάμεσων μεταβλητών που ορίστηκαν προηγουμένως με τις αλγεβρικές μορφές τους, προσδιορίζουμε τις εξόδους του κυκλώματος ως συναρτήσεις των μεταβλητών εισόδου.

Παράδειγμα 1. *Να γίνει ανάλυση του λογικού κυκλώματος του σχήματος.*



Προσδιορίζουμε τις ενδιάμεσες μεταβλητές (ή συναρτήσεις) και τις συναρτήσεις των εξόδων (εξαρτημένων μεταβλητών) του λογικού κυκλώματος, ακολουθώντας τα προαναφερθέντα βήματα:

$T_1 = ABC$ $T_2 = A + B + C$ $T_3 = AB$ $T_4 = AC$ $T_5 = BC$

$\mathbf{F_2 = AB + AC + BC}$

$T_6 = T_2 F'_2 = (A + B + C)(AB + AC + BC)'$

$\mathbf{F_1} = T_1 + T_6 = ABC + (A + B + C)(AB + AC + BC)' =$

$= ABC + (A + B + C)[AB)'(AC)'(BC)'] = ABC + (A + B + C)(A' + B')(A' + C')(B' + C')$

$= ABC + (AA' + AB' +A'B + BB' + A'C + CC') (A'B' +A' C' + B'C' + C'C') =$

$= ABC + (AB' +A'B + A'C) (A'B' +A' C' + B'C' + C') =$

$= ABC + (AB' +A'B + A'C) (A'B' + C'(A' + B' + 1) = ABC + (AB' +A'B + A'C) (A'B' +C') =$

$= ABC + (AA'B'B' + AB'C' + A'A'BB' + A'BC' + A'A'B'C + A'CC') =$

$\mathbf{= ABC + AB'C' + A'BC' + A'B'C}$

Η συμπλήρωση του πίνακα αλήθειας του λογικού κυκλώματος γίνεται εύκολα από τις λογικές συναρτήσεις εξόδων, εάν τις εκφράσουμε στη μορφή αθροίσματος ελαχιστόρων:

$\mathbf{F_1 = ABC + AB'C' + A'BC' + A'B'C = \Sigma(m_7, m_4, m_2, m_1)}$

$F_2 = AB + AC + BC$ $= AB(C + C') + A(B + B')C + (A + A')BC$

$= ABC + ABC' + ABC + AB'C + ABC + A'BC$

$= ABC + ABC' + AB'C + A'BC = \Sigma(m_7, m_6, m_5, m_3)$

A	B	C	AB	AC	BC	F_2	ABC	AB'C'	A'BC'	A'B'C	F_1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	1
0	1	0	0	0	0	0	0	0	1	0	1
0	1	1	0	0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	1
1	0	1	0	1	0	1	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	0	1

Εναλλακτικά, μπορούμε να συμπληρώσουμε τον πίνακα αλήθειας απ' ευθείας από το λογικό κύκλωμα, ακολουθώντας τα εξής βήματα:

1. *Ονομάζουμε τις εισόδους και τις εξόδους του κυκλώματος, καθώς και τις εξόδους όλων των λογικών πυλών του κυκλώματος.*
2. *Ξεκινώντας από τις λογικές πύλες που δέχονται ως είσοδο τις εισόδους του κυκλώματος (μεταβλητές εισόδου), προσδιορίζουμε τις τιμές που παίρνουν αυτές οι έξοδοι για κάθε συνδυασμό τιμών των εισόδων του κυκλώματος και συμπληρώνουμε αντίστοιχα τον πίνακα αλήθειας.*
3. *Προχωράμε στην συμπλήρωση του πίνακα αλήθειας για εκείνες τις εξόδους πυλών, οι οποίες δέχονται ως είσοδο ενδιάμεσες μεταβλητές που καθορίστηκαν προηγουμένως, δηλαδή δέχονται είσοδο από τις εξόδους προηγούμενων πυλών.*
4. *Επαναλαμβάνουμε τη διαδικασία του προηγούμενου βήματος μέχρι να προσδιορίσουμε πλήρως τις στήλες τιμών όλων των εξόδων.*

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

Η λειτουργία του κυκλώματος περιγράφεται από τις λογικές συναρτήσεις των εξόδων F_1 *και* F_2 *ή/και από τον πίνακα αλήθειας. Όσον αφορά στη λεκτική περιγραφή της λειτουργίας του, τόσο οι συναρτήσεις εξόδων όσο και ο πίνακας αλήθειας του κυκλώματος που αναλύθηκε, αντιστοιχούν πλήρως στη λειτουργία ενός τυπικού κυκλώματος που ονομάζουμε **πλήρη αθροιστή** και το οποίο θα περιγραφεί λεπτομερώς αργότερα, κατά την εξέταση των αριθμητικών λογικών κυκλωμάτων.*

Σχεδίαση συνδυαστικών κυκλωμάτων

Η διαδικασία σχεδίασης ξεκινά συνήθως με τη λεκτική περιγραφή της λειτουργίας του προς σχεδίαση συνδυαστικού κυκλώματος και καταλήγει σε ένα λογικό κύκλωμα. Για τη σχεδίαση ενός συνδυαστικού λογικού κυκλώματος ακολουθούμε τα εξής βήματα:

1. Από την περιγραφή της λειτουργίας του προς σχεδίαση κυκλώματος, προσδιορίζουμε το πλήθος των εισόδων και των εξόδων και δίνουμε κατάλληλα ονόματα σ' αυτές.

2. Συμπληρώνουμε τον πίνακα αλήθειας που περιγράφει την απαιτούμενη σχέση εισόδων και εξόδων.

3. Προσδιορίζουμε τις λογικές συναρτήσεις των εξόδων (συνήθως στη μορφή αθροίσματος ελαχιστόρων) και τις απλοποιούμε είτε με εφαρμογή των κανόνων της άλγεβρας Boole, είτε με χρήση των πινάκων Karnaugh.

4. Σχεδιάζουμε το λογικό κύκλωμα.

Παράδειγμα 2. ***Να σχεδιαστεί λογικό κύκλωμα που υλοποιεί τη συνάρτηση πλειοψηφίας τριών εισόδων (δηλαδή, το κύκλωμα που αναγνωρίζει πότε έχουμε περισσότερα "1" από "0" στις τρεις εισόδους).***

Από την περιγραφή της λειτουργίας του κυκλώματος είναι προφανές ότι έχουμε τρεις εισόδους. Επιπλέον απαιτείται μία έξοδος αφού το πλήθος των εισόδων είναι περιττός αριθμός (3) και επομένως είτε θα έχουμε περισσότερα "1", είτε περισσότερα "0" (αποκλείεται να έχουμε ίσο πλήθος από "1" και "0"). Η έξοδος θα παίρνει τιμή "1" όταν έχουμε περισσότερα "1" από "0" στις τρεις εισόδους, αλλιώς θα παίρνει τιμή "0".

Πίνακας αλήθειας:

A	*B*	*C*	*F*
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

Λογική συνάρτηση εξόδου:

$$F = \Sigma(m_3, m_5, m_6, m_7) = A'BC + AB'C + ABC' + ABC$$

Απλοποίηση λογικής συνάρτησης εξόδου με πίνακα Karnaugh:

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$F = AB + AC + BC$$

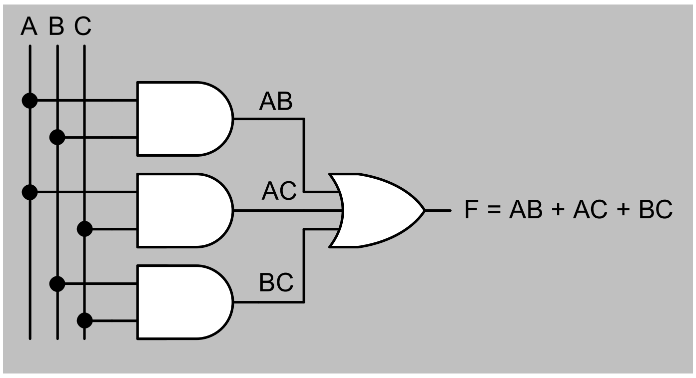
Απλοποίηση λογικής συνάρτησης εξόδου με πίνακα με άλγεβρα Boole:

$$F = A'BC + AB'C + ABC' + \underline{ABC} = A'BC + AB'C + ABC' + \underline{ABC + ABC + ABC} =$$

$$= (A'BC + ABC) + (AB'C + ABC) + (ABC' + ABC) =$$

$$= BC(A' + A) + AC(B' + B) + AB(C' + C) = BC + AC + AB$$

Λογικό κύκλωμα:



<u>Παράδειγμα 3.</u> Να σχεδιαστεί λογικό κύκλωμα που να ελέγχει τη λειτουργία ενός φωτεινού σηματοδότη ρύθμισης της κυκλοφορίας αυτοκινήτων. Το κύκλωμα αυτό θα πρέπει να ανιχνεύει οποιοδήποτε λάθος συνδυασμό. Να σημειωθεί ότι σε κανονική λειτουργία, μόνο ένα από τα τρία φώτα του σηματοδότη (πράσινο, πορτοκαλί ή κόκκινο), είναι αναμμένο και οποιοσδήποτε άλλος συνδυασμός αποτελεί σφάλμα λειτουργίας.

Από την παραπάνω περιγραφή λειτουργίας του σηματοδότη προκύπτει ότι το σύστημα ελέγχου θα πρέπει να δέχεται τρεις εισόδους, μία για κάθε φως (πράσινο, πορτοκαλί και κόκκινο) και μία έξοδο, η οποία ανιχνεύει την κατάσταση λάθους λειτουργίας (λογική κατάσταση "1").

Επιλέγουμε (αυθαίρετα) τις ακόλουθες αντιστοιχίες: A = πράσινο, B = πορτοκαλί, C = κόκκινο και έστω F η έξοδος του συστήματος ελέγχου.

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

Λογική συνάρτηση εξόδου:

$$F = \Sigma(m_1, m_3, m_5, m_6, m_7) = A'B'C' + A'BC + AB'C + ABC' + ABC$$

Απλοποίηση λογικής συνάρτησης εξόδου με πίνακα Karnaugh:

A \ BC	00	01	11	10
0	1	0	1	0
1	0	1	1	1

$$F = A'B'C' + AB + AC + BC$$

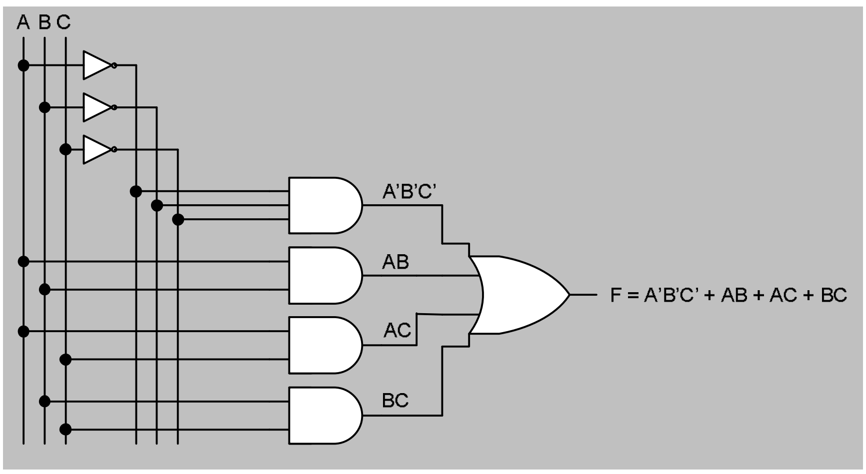
Απλοποίηση λογικής συνάρτησης εξόδου με πίνακα με άλγεβρα Boole:

$$F = A'B'C' + A'BC + AB'C + ABC' + \underline{ABC} = A'B'C' + A'BC + AB'C + ABC' + \underline{ABC + ABC + ABC} =$$

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

$$= A'B'C' + BC(A' + A) + AC(B' + B) + AB(C' + C) = A'B'C' + BC + AC + AB$$

Λογικό κύκλωμα:



<u>Παράδειγμα 4.</u> ***Να σχεδιαστεί λογικό κύκλωμα που να μετατρέπει ένα δυαδικά κωδικοποιημένο δεκαδικό ψηφίο (Binary Coded Decimal – BCD) σε κώδικα GRAY.***

Ο δυαδικός κώδικας BCD - Binary Coded Decimal αναπαριστάνει ("κωδικοποιεί") ένα δεκαδικό ψηφίο με το δυαδικό του ισοδύναμο. Λαμβάνοντας υπόψη ότι για να εκφράσουμε το μεγαλύτερο δεκαδικό ψηφίο, το εννιά (9), σε δυαδική μορφή απαιτούνται τέσσερα δυαδικά ψηφία (1001_2), θα πρέπει όλα τα δεκαδικά ψηφία να εκφράζονται με το ίδιο πλήθος δυαδικών ψηφίων. Για να κωδικοποιήσουμε σε BCD ένα δεκαδικό αριθμό με περισσότερα από ένα ψηφία, κωδικοποιούμε σε BCD κάθε δεκαδικό ψηφίο του αριθμού ανεξάρτητα, δηλαδή με μία τετράδα δυαδικών ψηφίων. Για παράδειγμα, η BCD κωδικοποίηση του αριθμού 293_{10}, θα είναι: $0010\ 1001\ 0011_{BCD}$

Ο δυαδικός κώδικας GRAY χρησιμοποιείται σε εφαρμογές στις οποίες κατά την παραγωγή ή τη μετάδοση μιας ακολουθίας δυαδικών αριθμών μπορεί να προκύψει σφάλμα ή ασάφεια, όταν μεταβαίνουμε από έναν αριθμό στον επόμενό του. Εάν χρησιμοποιούσαμε δυαδικούς αριθμούς, μία αλλαγή, για παράδειγμα, από το 0111 σε 1000, θα μπορούσε να προκαλέσει την εμφάνιση του εσφαλμένου ενδιάμεσου αριθμού 1001, αν η τιμή του τελευταίου προς τα δεξιά ψηφίου χρειαζόταν περισσότερο χρόνο για να αλλάξει σε σχέση με τις τιμές των άλλων τριών ψηφίων. Με τη χρήση του κώδικα GRAY μπορούμε να λύσουμε το πρόβλημα αυτό: το χαρακτηριστικό του κώδικα GRAY είναι ότι μόνο ένα ψηφίο της παράστασης ενός αριθμού στον κώδικα αυτό αλλάζει κατά τη μετάβαση από έναν αριθμό στον επόμενό του (δηλαδή δύο διαδοχικοί αριθμοί εκφρασμένοι σε κώδικα GRAY διαφέρουν μόνο σε ένα ψηφίο). Η μετατροπή ενός δυαδικού αριθμού σε κώδικα GRAY ακολουθεί την εξής διαδικασία: το μέγιστο σημαντικό ψηφίο παραμένει το ίδιο και το ψηφίο μιας άλλης οποιασδήποτε θέσης ν προκύπτει από το λογικό XOR του ψηφίου αυτού με το ψηφίο που βρίσκεται στην αμέσως επόμενη πιο σημαντική θέση ν+1. Για παράδειγμα, ο δυαδικός αριθμός $A_3A_2A_1A_0 = 1100_2$ ($=12_{10}$) στον κώδικα GRAY θα είναι $X_3X_2X_1X_0 = 1010$, αφού:
$X_3 = A_3 = 1$, $X_2 = A_3 \oplus A_2 = 1 \oplus 1 = 0$, $X_1 = A_2 \oplus A_1 = 1 \oplus 0 = 1$, $X_0 = A_1 \oplus A_0 = 0 \oplus 0 = 0$.

Από τα παραπάνω προκύπτει ότι ζητούμενο κύκλωμα θα έχει τέσσερις εισόδους, $A_3A_2A_1A_0$ και τέσσερις εξόδους, $X_3X_2X_1X_0$.

Πίνακας αλήθειας:

A_3	A_2	A_1	A_0	X_3	X_2	X_1	X_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

Είναι προφανές ότι αφού έχουμε τέσσερις ανεξάρτητες μεταβλητές εισόδους θα έχουμε συνολικά δεκάξι συνδυασμούς τιμών, από τους οποίους όμως χρησιμοποιούμε μόνο τους δέκα πρώτους. Οι υπόλοιποι έξι μη χρησιμοποιούμενοι συνδυασμοί θα αντιμετωπιστούν ως αδιάφορες καταστάσεις (δηλαδή ως αδιάφοροι όροι στον πίνακα Karnaugh).

Απλοποίηση λογικών συναρτήσεων εξόδων:

Από τον πίνακα αλήθειας παρατηρούμε ότι: $X_3 = A_3$.

A_3A_2 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

$X_2 = A_3 + A_2$

A_3A_2 \ A_1A_0	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	X	X	X	X
10	0	0	X	X

$X_1 = A_2A'_1 + A'_2A_1 = A_2 \oplus A_1$

A_3A_2 \ A_1A_0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	X	X	X	X
10	0	1	X	X

$X_0 = A'_1A_0 + A_1A'_0 = A_1 \oplus A_0$

Λογικό κύκλωμα: