Συνδυαστικά Λογικά Κυκλώματα

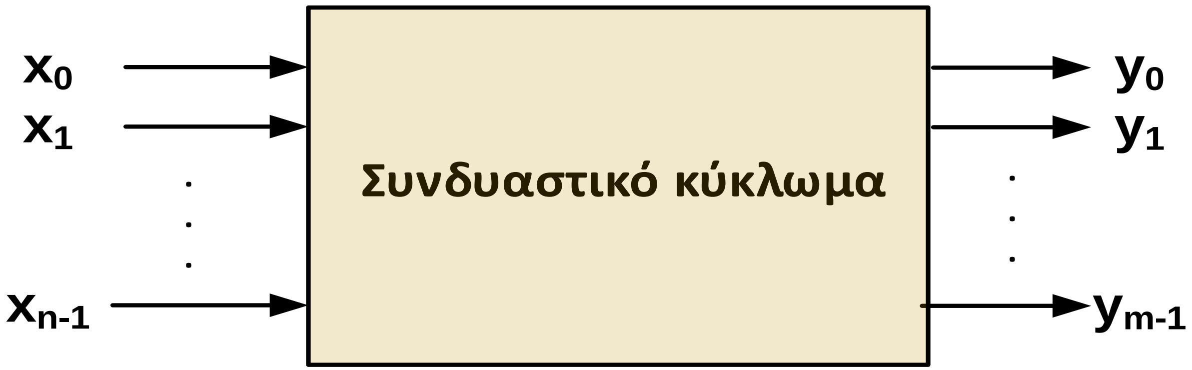
Στις προηγούμενες ενότητες μελετήσαμε την υλοποίηση λογικών συναρτήσεων σε λογικά κυκλώματα με χρήση βασικών ή/και παράγωγων πυλών και παρουσιάσαμε τις διαδικασίες ελαχιστοποίησης των λογικών συναρτήσεων, ώστε να υλοποιούμε τα απλούστερα κατά το δυνατό λογικά κυκλώματα.

Σε αυτά τα λογικά κυκλώματα, όπως και σε αυτά που θα εξετάσουμε στη συνέχεια, η λογική τιμή της εξόδου ή των εξόδων, κάθε χρονική στιγμή, καθορίζονται μόνο από τον τρέχοντα συνδυασμό τιμών των εισόδων και για το λόγο αυτό αναφέρονται ως **συνδυαστικά κυκλώματα**.

Στα κυκλώματα αυτά, για κάθε ***n*** μεταβλητές εισόδου, υπάρχουν $\mathbf{2^{n}}$ δυαδικοί συνδυασμοί τιμών των εισόδων και για κάθε συνδυασμό τιμών των εισόδων, η έξοδος ή οι έξοδοι λαμβάνουν συγκεκριμένες λογικές τιμές. Η λειτουργία ενός συνδυαστικού κυκλώματος καθορίζεται από τον 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 (μία συνάρτηση για κάθε μία έξοδο). Κάθε λογική συνάρτηση εξόδου (εξαρτημένη μεταβλητή) έχει ως ανεξάρτητες μεταβλητές τις ***n*** (ανεξάρτητες) μεταβλητές εισόδου.

Στο Σχήμα 5.1 παρουσιάζεται το συνοπτικό (ή “χονδρικό”) διάγραμμα ενός συνδυαστικού κυκλώματος με **n** εισόδους (ανεξάρτητες μεταβλητές) και **m** εξόδους (εξαρτημένες μεταβλητές). Για κάθε ένα συνδυασμό λογικών τιμών των μεταβλητών εισόδου, $\mathbf{x_0}$ έως $\mathbf{x_{n-1}}$, παράγεται ένας συνδυασμός λογικών τιμών των μεταβλητών εξόδου, $\mathbf{y_0}$ έως $\mathbf{y_{m-1}}$.



Σχήμα 5.1   Χονδρικό διάγραμμα συνδυαστικού κυκλώματος

Όπως έχουμε ήδη αναφέρει σε προηγούμενη ενότητα, ο σχεδιαστής ψηφιακών συστημάτων αντιμετωπίζει δύο βασικές περιπτώσεις, την **ανάλυση** ενός υπάρχοντος λογικού κυκλώματος, για να προσδιορίσει τη λειτουργία που επιτελεί, και τη **σύνθεση** (**σχεδίαση**) ενός λογικού κυκλώματος, το οποίο επιτελεί κάποια επιθυμητή λειτουργία.

Στην παρούσα και στις επόμενες ενότητες θα παρουσιάσουμε συστηματικές διαδικασίες ανάλυσης και σχεδίασης συνδυαστικών κυκλωμάτων, τα οποία επιτελούν λειτουργίες που συναντώνται συχνά στα ψηφιακά συστήματα.

Ανάλυση συνδυαστικών κυκλωμάτων

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

Συνίσταται στην **εξαγωγή των λογικών συναρτήσεων των εξόδων** του κυκλώματος ή/και **τη συμπλήρωση του πίνακα αλήθειας** και, πιθανόν, σε μια **λεκτική περιγραφή της λειτουργίας** του.

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

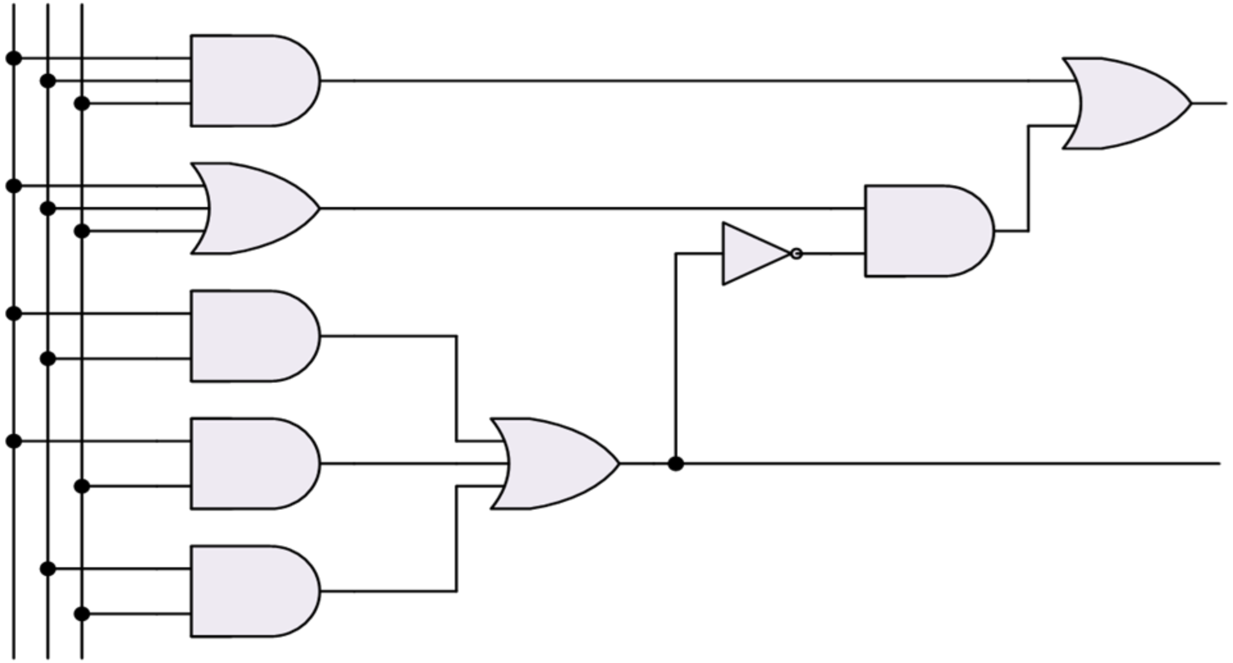
1. Ονομάζουμε τις εισόδους και τις εξόδους του κυκλώματος, καθώς και τις ενδιάμεσες εξόδους όλων των λογικών πυλών του κυκλώματος.

2. Ξεκινώντας από τις λογικές πύλες που δέχονται ως είσοδο τις εισόδους του κυκλώματος (τις ανεξάρτητες μεταβλητές εισόδου), προσδιορίζουμε σταδιακά τις εξόδους όλων των λογικών πυλών ως συναρτήσεις των εισόδων που δέχονται (ενδιάμεσες μεταβλητές ή συναρτήσεις).

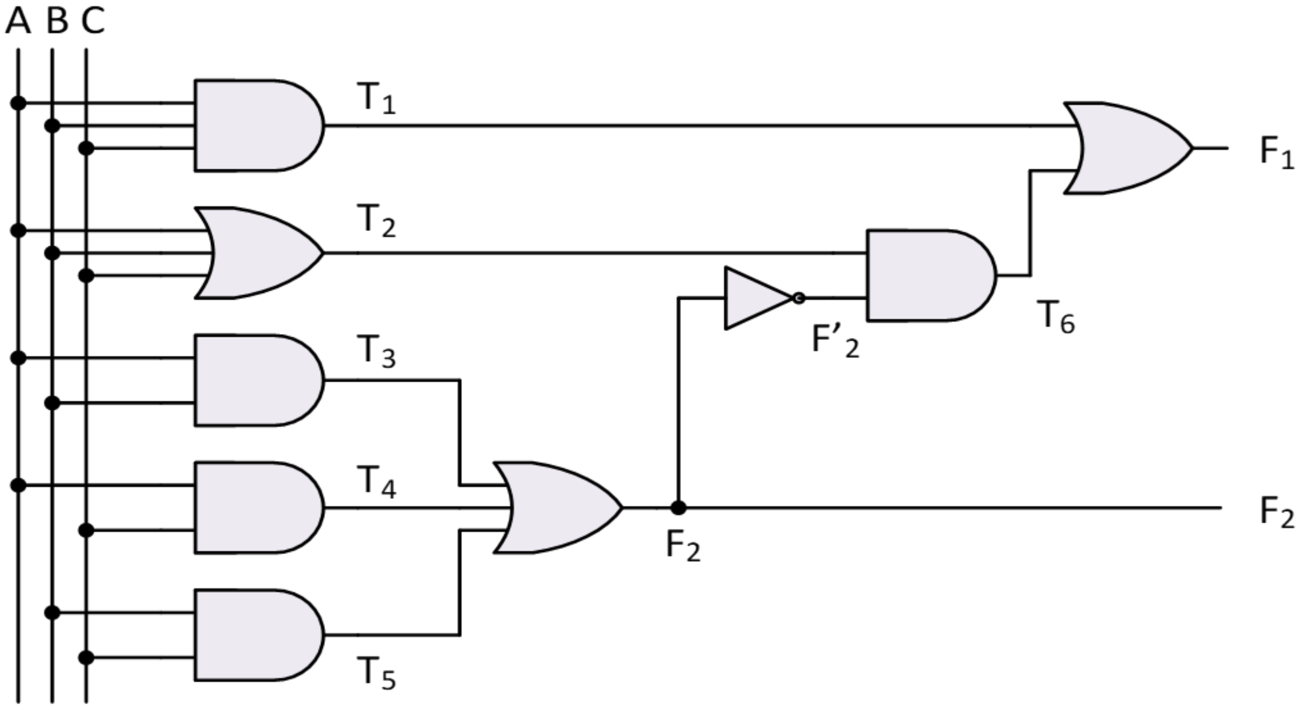
3. Επαναλαμβάνουμε τη διαδικασία του προηγούμενου βήματος μέχρι να προσδιορίσουμε τις λογικές συναρτήσεις όλων των εξόδων του κυκλώματος.

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 μόνο.

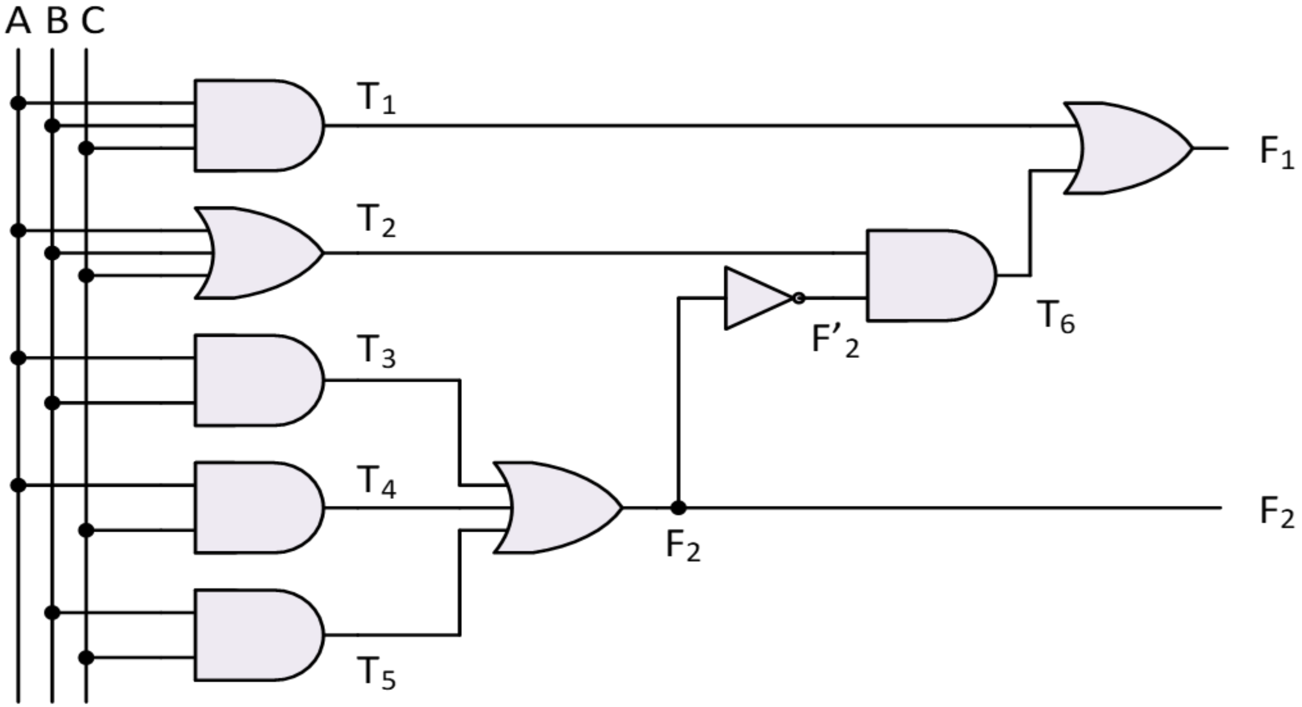
Παράδειγμα 5.1: Να γίνει ανάλυση του λογικού κυκλώματος του σχήματος.



Παράδειγμα 5.1: Να γίνει ανάλυση του λογικού κυκλώματος του σχήματος.

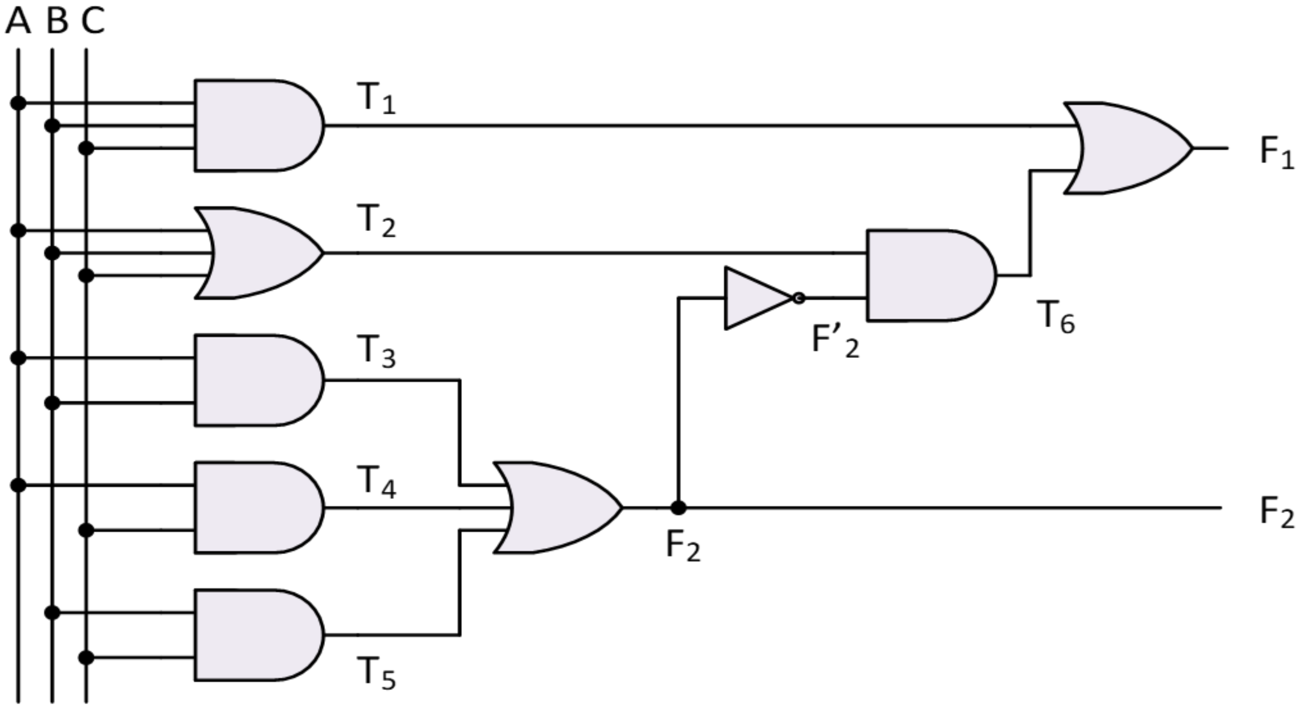


Παράδειγμα 5.1: Να γίνει ανάλυση του λογικού κυκλώματος του σχήματος.



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

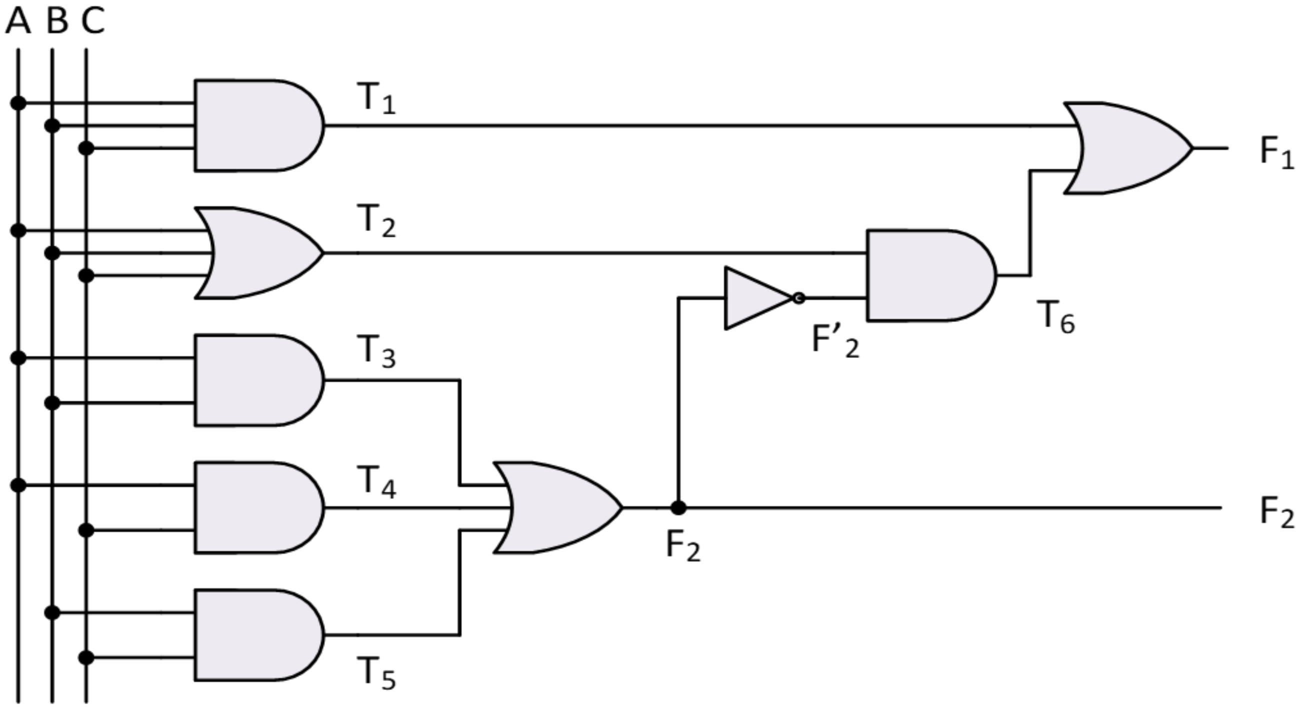
Παράδειγμα 5.1: Να γίνει ανάλυση του λογικού κυκλώματος του σχήματος.



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

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

Παράδειγμα 5.1: Να γίνει ανάλυση του λογικού κυκλώματος του σχήματος.

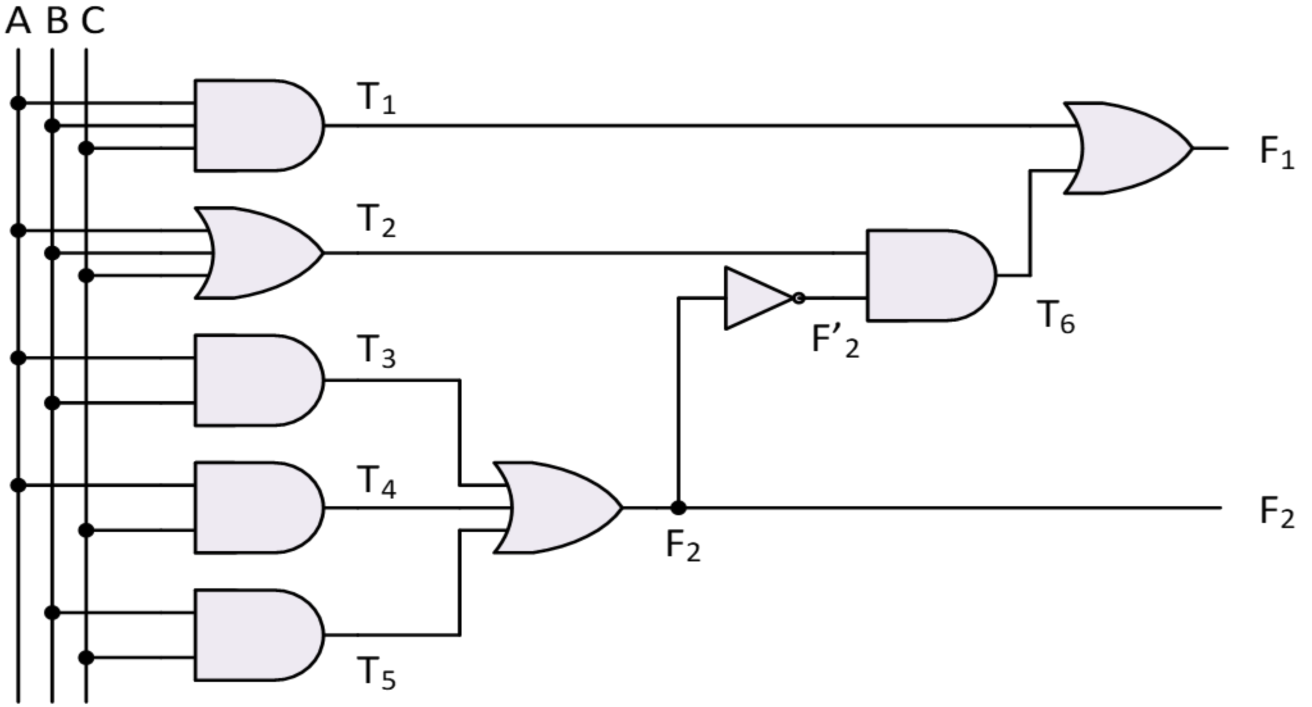


$T_1 = ABC$ $T_2 = A + B + C$ $T_3 = AB$ $T_4 = AC$ $T_5 = BC$

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

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

Παράδειγμα 5.1: Να γίνει ανάλυση του λογικού κυκλώματος του σχήματος.



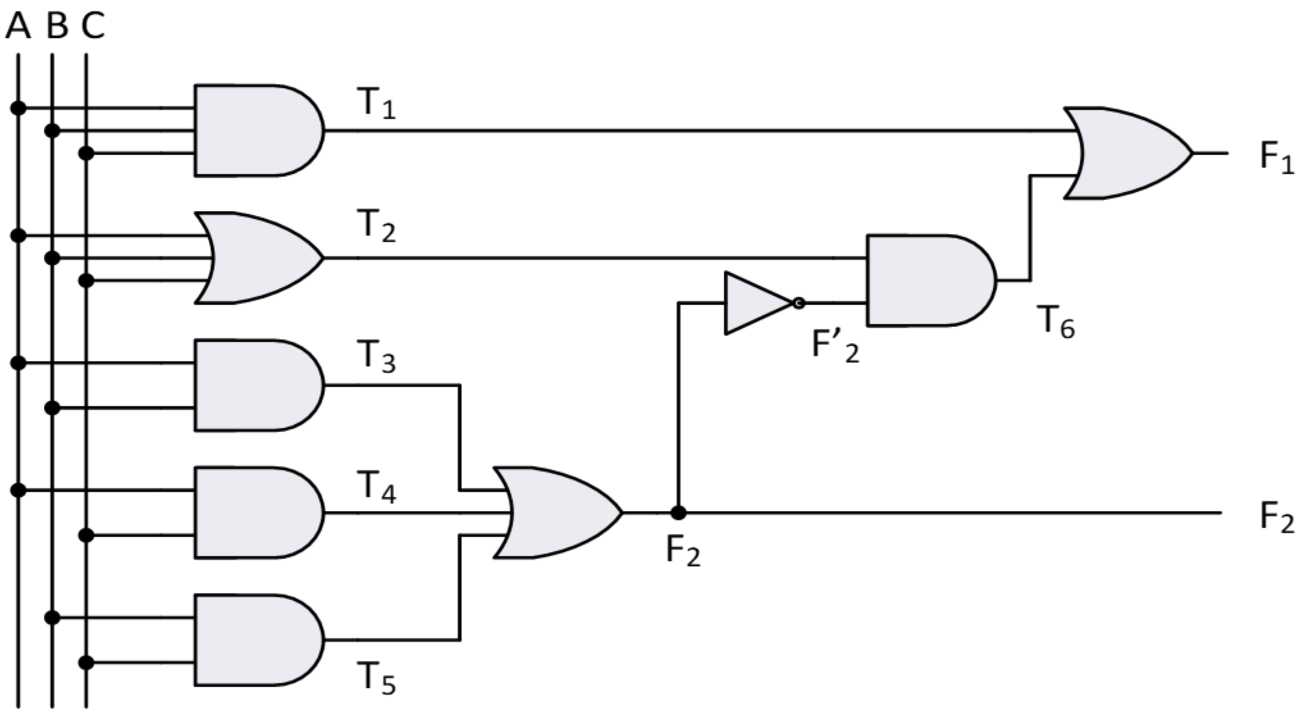
$T_1 = ABC$ $T_2 = A + B + C$ $T_3 = AB$ $T_4 = AC$ $T_5 = BC$

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

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

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

Παράδειγμα 5.1: Να γίνει ανάλυση του λογικού κυκλώματος του σχήματος.



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

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

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

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

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

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

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

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

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

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

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

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

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

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

$\mathbf{= \Sigma(7, 4, 2, 1)}$

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

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

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

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

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

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

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

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

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

$\mathbf{= \Sigma(7, 4, 2, 1)}$

$\mathbf{F_2 = AB + AC + BC} = AB(C + C') + A(B + B')C + (A + A')BC$

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

$\mathbf{= ABC + ABC' + AB'C + A'BC = \Sigma(7, 6, 5, 3)}$

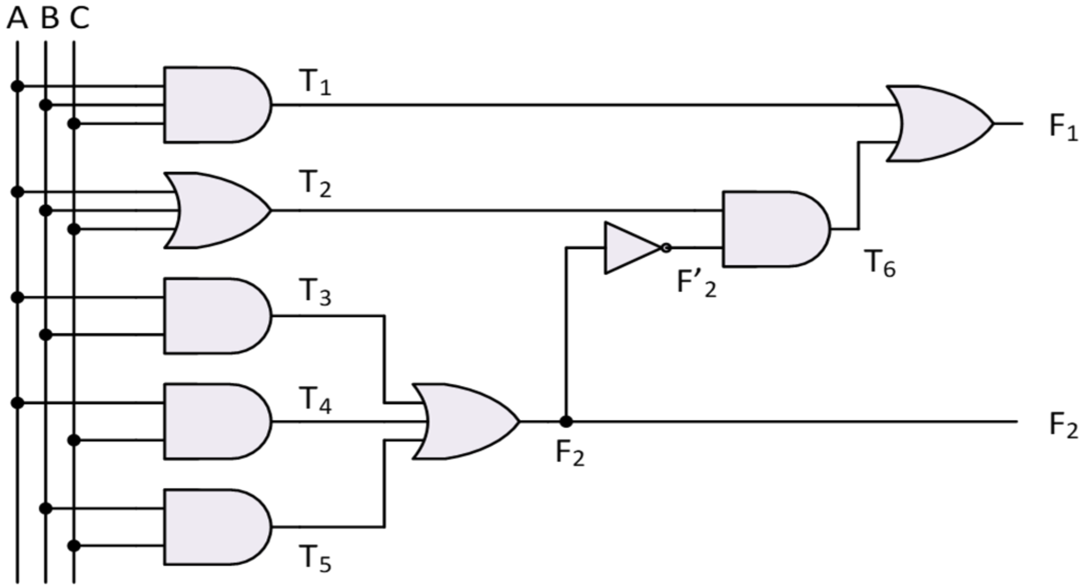
$F_1 = \Sigma(7, 4, 2, 1)$

$F_2 = = \Sigma(7, 6, 5, 3)$

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**

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**

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, δίνουμε κατάλληλα ονόματα σ’ αυτές και σχεδιάζουμε το χονδρικό διάγραμμα του κυκλώματος.
2. Συμπληρώνουμε τον πίνακα αλήθειας, που περιγράφει τη σχέση εισόδων και εξόδων.
3. Προσδιορίζουμε τις λογικές συναρτήσεις των εξόδων (συνήθως σε μορφή αθροίσματος ελαχιστόρων) και τις απλοποιούμε είτε με αλγεβρικούς μετασχηματισμούς, είτεμε εφαρμογή των κανόνων της άλγεβρας Boole, είτε με χρήση των πινάκων Karnaugh.
4. Σχεδιάζουμε το λογικό κύκλωμα με λογικές πύλες.
5. Επαληθεύουμε την ορθή λειτουργία του κυκλώματος, με την ανάλυση του κυκλώματος που συνθέσαμε.

Παράδειγμα 5.2: Να σχεδιάσετε λογικό κύκλωμα που υλοποιεί τη συνάρτηση πλειοψηφίας τριών εισόδων (Παραδείγματα 3.7 και 4.2):

(α) με βασικές πύλες και,

(β) με πύλες NAND μόνο.

Παράδειγμα 5.2: Να σχεδιάσετε λογικό κύκλωμα που υλοποιεί τη συνάρτηση πλειοψηφίας τριών εισόδων (Παραδείγματα 3.7 και 4.2):

(α) με βασικές πύλες και,

(β) με πύλες NAND μόνο.

(α) Ως συνάρτηση πλειοψηφίας τριών εισόδων ορίζεται η λογική συνάρτηση που "αναγνωρίζει" πότε σε τρεις εισόδους πλειοψηφούν τα 1 έναντι των 0.

Παράδειγμα 5.2: Να σχεδιάσετε λογικό κύκλωμα που υλοποιεί τη συνάρτηση πλειοψηφίας τριών εισόδων (Παραδείγματα 3.7 και 4.2):

(α) με βασικές πύλες και,

(β) με πύλες NAND μόνο.

(α) Ως συνάρτηση πλειοψηφίας τριών εισόδων ορίζεται η λογική συνάρτηση που “αναγνωρίζει” πότε σε τρεις εισόδους πλειοψηφούν τα 1 έναντι των 0.

Από την περιγραφή της λειτουργίας του κυκλώματος είναι προφανές ότι έχουμε τρεις εισόδους.

Παράδειγμα 5.2: Να σχεδιάσετε λογικό κύκλωμα που υλοποιεί τη συνάρτηση πλειοψηφίας τριών εισόδων (Παραδείγματα 3.7 και 4.2):

(α) με βασικές πύλες και,

(β) με πύλες NAND μόνο.

(α) Ως συνάρτηση πλειοψηφίας τριών εισόδων ορίζεται η λογική συνάρτηση που "αναγνωρίζει" πότε σε τρεις εισόδους πλειοψηφούν τα 1 έναντι των 0.

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").

Παράδειγμα 5.2: Να σχεδιάσετε λογικό κύκλωμα που υλοποιεί τη συνάρτηση πλειοψηφίας τριών εισόδων (Παραδείγματα 3.7 και 4.2):

(α) με βασικές πύλες και,

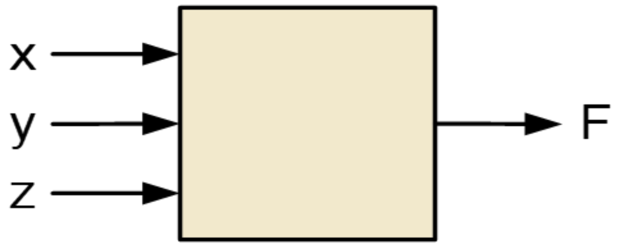
(β) με πύλες NAND μόνο.

(α) Ως συνάρτηση πλειοψηφίας τριών εισόδων ορίζεται η λογική συνάρτηση που “αναγνωρίζει” πότε σε τρεις εισόδους πλειοψηφούν τα 1 έναντι των 0.

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”).

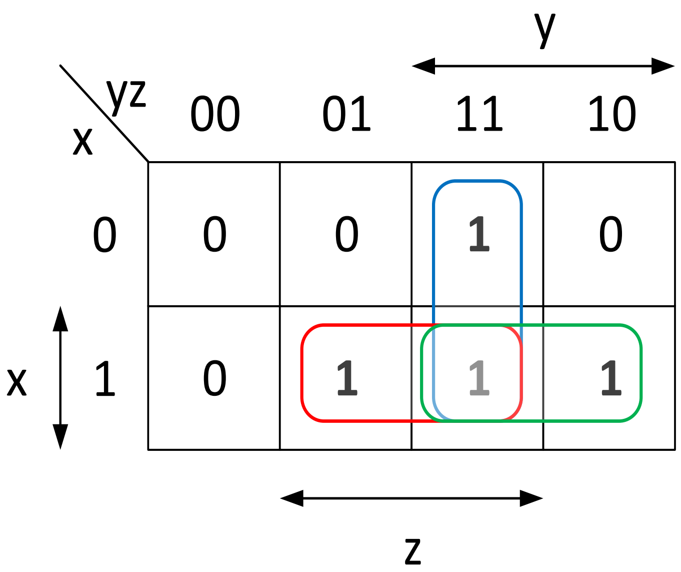
Επομένως, το χονδρικό διάγραμμα είναι το ακόλουθο:



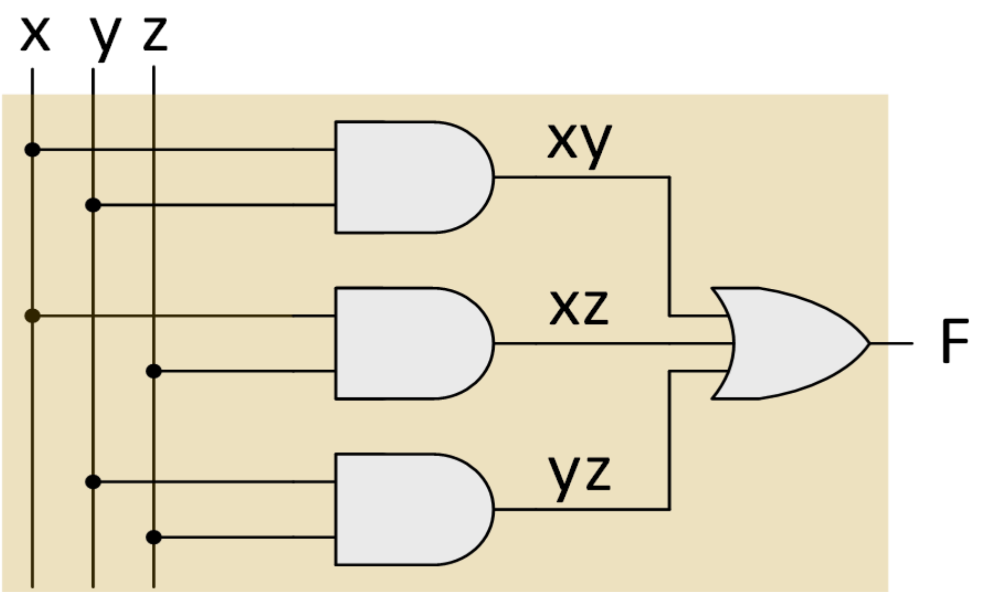
Η έξοδος θα παίρνει τιμή “1” όταν έχουμε περισσότερα “1” από “0” στις τρεις εισόδους, αλλιώς θα παίρνει τιμή “0”.

Ο πίνακας αλήθειας και ο πίνακας Karnaugh είναι ως ακολούθως:

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**

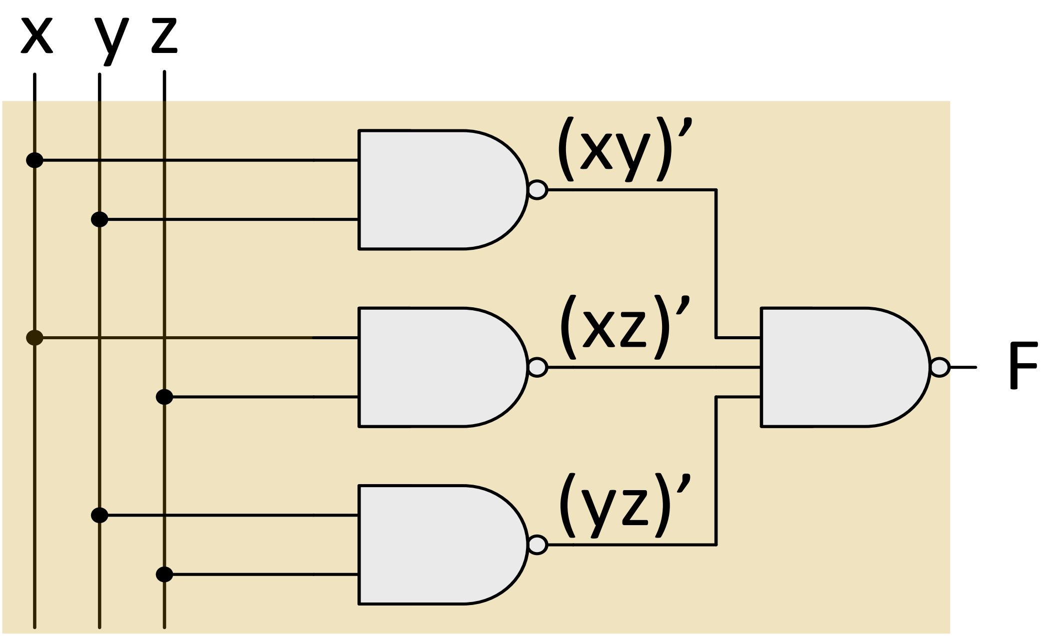


$F = \Sigma(3, 5, 6, 7) = xy + xz + yz$



(β) Εφαρμόζοντας το θεώρημα De Morgan στην απλοποιημένη μορφή της συνάρτησης εξόδου έχουμε:

$$F = [F']' = [(xy + xz + yz)']' = [(xy)' \cdot (xz)' \cdot (yz)']'$$

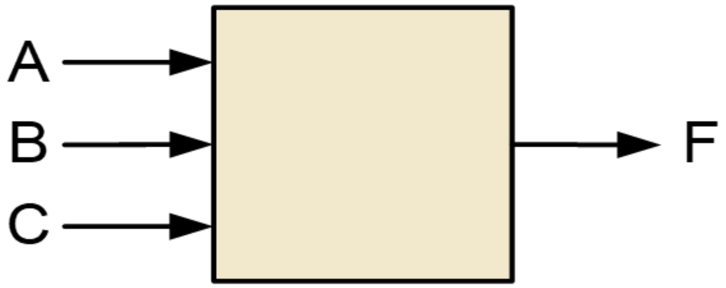


Παράδειγμα 5.3: Να σχεδιαστεί λογικό κύκλωμα που να αναγνωρίζει την εσφαλμένη λειτουργία ενός φωτεινού σηματοδότη ρύθμισης της κυκλοφορίας αυτοκινήτων. Το κύκλωμα αυτό θα πρέπει να ανιχνεύει οποιοδήποτε λάθος συνδυασμό.

Σε κανονική λειτουργία, μόνο ένας από τους τρεις φανούς 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 φανό (πράσινο, πορτοκαλί και κόκκινο) και **μία έξοδο**, η οποία **ανιχνεύει** την **κατάσταση εσφαλμένης λειτουργίας**, οπότε λαμβάνει τη λογική **τιμή 1**, ενώ σε **ομαλή λειτουργία** του σηματοδότη λαμβάνει την **τιμή 0**.

Το χονδρικό διάγραμμα του κυκλώματος είναι το ακόλουθο:

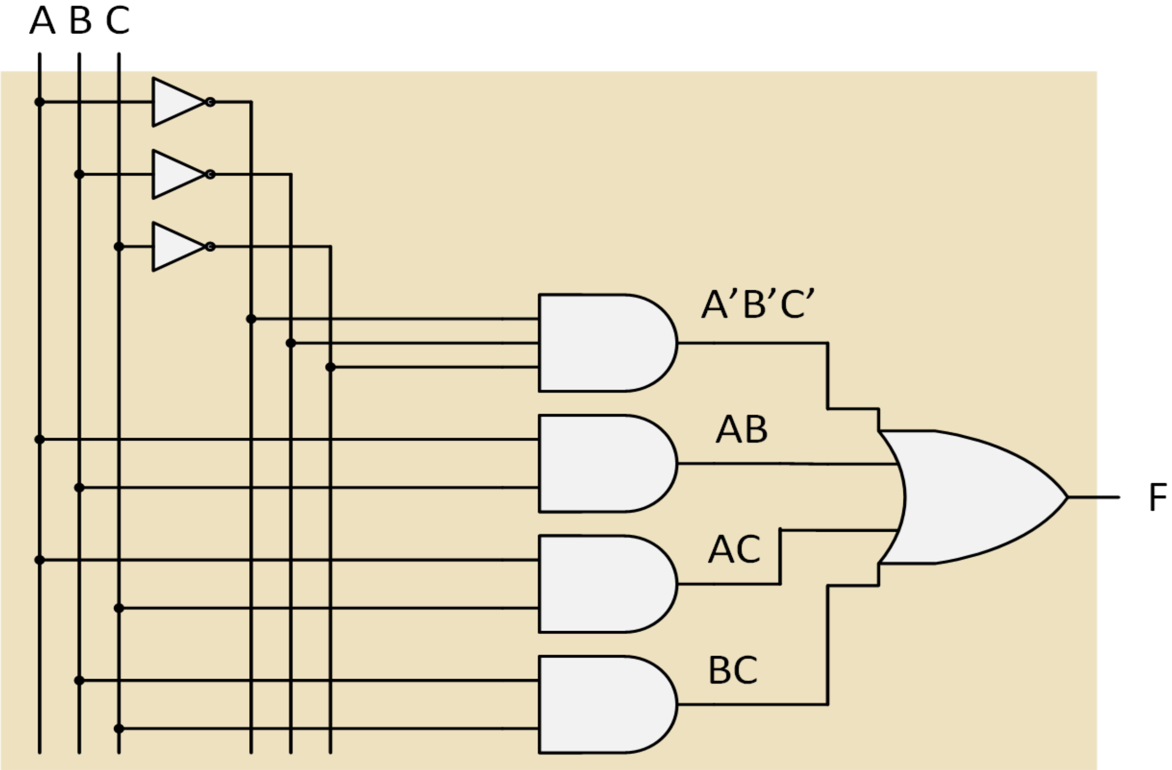


Πίνακας αλήθειας και πίνακας Karnaugh:

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**

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

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



Παράδειγμα 5.4: Να σχεδιαστεί λογικό κύκλωμα που να μετατρέπει ένα δυαδικά κωδικοποιημένο δεκαδικό ψηφίο (Binary Coded Decimal – BCD) σε κώδικα GRAY.

Ο αριθμητικός κώδικας BCD (Binary Coded Decimal, μετάφραση: δυαδικά κωδικοποιημένοι δεκαδικοί) είναι ένας κώδικας στον οποίο κάθε ψηφίο ενός δεκαδικού αριθμού εκφράζεται ξεχωριστά σε δυαδική μορφή.
Ένα δυαδικά κωδικοποιημένο δεκαδικό ψηφίο εκφράζεται με τέσσερα δυαδικά ψηφία:

$$\mathbf{A = A_3A_2A_1A_0}.$$

Ο κώδικας Gray είναι ένας κώδικας αναπαράστασης αριθμών με δυαδικά ψηφία κατά τέτοιο τρόπο ώστε, κατά τη μετάβαση μεταξύ δύο διαδοχικών αριθμών, να αλλάζει μόνο ένα ψηφίο κάθε φορά και παρουσιάστηκε αναλυτικά σε προηγούμενη ενότητα.

Ο ισοδύναμος αριθμός BCD ενός δεκαδικού ψηφίου εκφράζεται σε κώδικα Gray, επίσης, με τέσσερα δυαδικά ψηφία: $\mathbf{X = X_3X_2X_1X_0}$.

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**

Από τον πίνακα αλήθειας παρατηρούμε ότι οι λογικές συναρτήσεις των εξόδων είναι μερικώς καθορισμένες, αφού για τους συνδυασμούς τιμών των εισόδων από 1010 έως και 1111 δεν ορίζονται τιμές για τις εξόδους. **Επομένως θα έχουμε αδιάφορους όρους**.

Οι λογικές συναρτήσεις των τεσσάρων εξόδων του συστήματος σε μορφή αθροίσματος ελαχιστόρων είναι:

$$\mathbf{X_3 = \Sigma(}8, 9\mathbf{) + d(}10, 11, 12, 13, 14, 15\mathbf{)}$$

$$\mathbf{X_2 = \Sigma(}4, 5, 6, 7, 8, 9\mathbf{) + d(}10, 11, 12, 13, 14, 15\mathbf{)}$$

$$\mathbf{X_1 = \Sigma(}2, 3, 4, 5\mathbf{) + d(}10, 11, 12, 13, 14, 15\mathbf{)}$$

$$\mathbf{X_0 = \Sigma(}1, 2, 5, 6, 9\mathbf{) + d(}10, 11, 12, 13, 14, 15\mathbf{)}$$

Επίσης, από τον πίνακα αλήθειας παρατηρούμε ότι η έξοδος $\mathbf{X_3}$ λαμβάνει ακριβώς τις ίδιες τιμές με την είσοδο $\mathbf{A_3}$, δηλαδή $\mathbf{X_3 = A_3}$, και δεν χρειάζεται να χρησιμοποιήσουμε οποιαδήποτε μέθοδο για την απλοποίησή της.

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

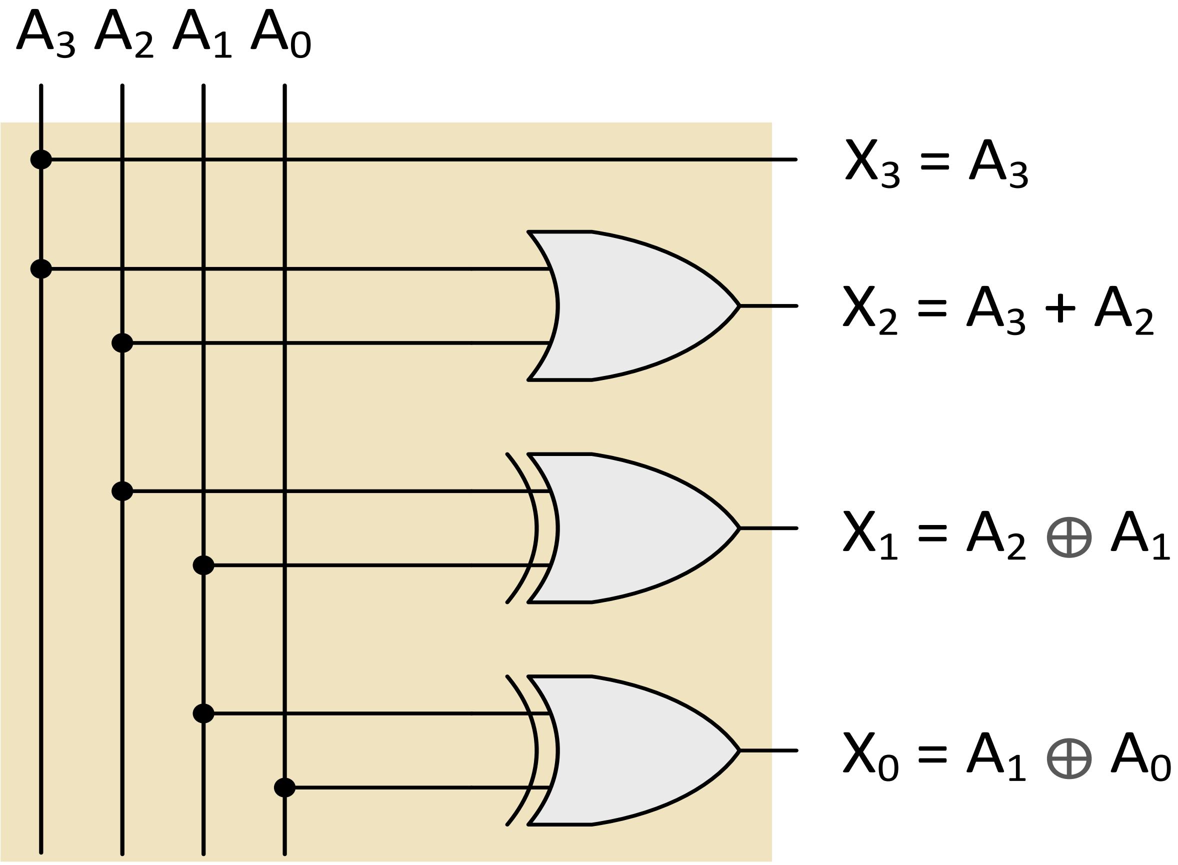
$X_2 = A_3 + A_2$

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

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



A_3 A_2 A_1 A_0
$X_3 = A_3$
$X_2 = A_3 + A_2$
$X_1 = A_2 \oplus A_1$
$X_0 = A_1 \oplus A_0$