Λογικές συναρτήσεις

Μια **λογική συνάρτηση** περιγράφεται από μια αλγεβρική έκφραση που περιλαμβάνει:

- δυαδικές μεταβλητές,
- τις σταθερές 0 και 1,
- τους τελεστές των λογικών πράξεων,
- παρενθέσεις και αγκύλες.

Εκφράζει τη λογική σχέση ανάμεσα σε μια εξαρτημένη δυαδική μεταβλητή και έναν αριθμό ανεξάρτητων δυαδικών μεταβλητών.

Οι τιμές που μπορεί να πάρει η λογική συνάρτηση (δηλαδή η εξαρτημένη μεταβλητή) προσδιορίζονται με τον υπολογισμό της τιμής της αλγεβρικής έκφρασης για καθένα από τους δυνατούς συνδυασμούς τιμών των ανεξάρτητων μεταβλητών και αποτυπώνονται στον πίνακα αλήθειας της λογικής συνάρτησης.

Είναι προφανές ότι **μια λογική συνάρτηση** μπορεί να περιγραφεί με **έναν και μοναδικό πίνακα αλήθειας**, ενώ μπορεί να έχει **πολλές διαφορετικές αλγεβρικές εκφράσεις**, οι οποίες όμως **παράγουν τον ίδιο πίνακα αλήθειας**.

Αντιστρόφως, στην περίπτωση που **δύο ή περισσότερες αλγεβρικές εκφράσεις** παράγουν **τον ίδιο πίνακα αλήθειας**, οι εκφράσεις αυτές **είναι ισοδύναμες** και αντιστοιχούν στην **ίδια λογική συνάρτηση**.

Παράδειγμα 3.3: Δίνονται οι λογικές συναρτήσεις:

$$\mathbf{F_1(A, B, C) = AC' + B + AB'C'} \quad \text{και} \quad \mathbf{F_2(A, B, C) = B + AC'}$$

Να συμπληρωθεί ο πίνακας αλήθειας για τις δύο αυτές λογικές συναρτήσεις και να επιβεβαιωθεί η ισοδυναμία των δύο λογικών συναρτήσεων με αλγεβρικούς μετασχηματισμούς.

Στο πρώτο ερώτημα, για τη συμπλήρωση του πίνακα αλήθειας των συναρτήσεων, θα κάνουμε χρήση του θεωρήματος **Θ2: x + 1 = 1**.

Σύμφωνα με το αυτό το θεώρημα, **εάν ένας οποιοδήποτε όρος του λογικού αθροίσματος λάβει την τιμή 1, η συνάρτηση θα έχει και αυτή τιμή 1, ανεξάρτητα από τι τιμές που έχουν οι άλλοι όροι.**

Επομένως, για να συμπληρώσουμε τον πίνακα αλήθειας μιας λογικής συνάρτησης, χρειάζεται να προσδιορίσουμε τους συνδυασμούς τιμών των ανεξάρτητων μεταβλητών για τους οποίους κάθε όρος (ανεξάρτητα από τους υπόλοιπους) του λογικού αθροίσματος λαμβάνει την τιμή 1 και να τοποθετήσουμε αυτή την τιμή στον πίνακα αλήθειας στις θέσεις που αντιστοιχούν στο συνδυασμό τιμών των ανεξάρτητων μεταβλητών. Για όλους τους υπόλοιπους συνδυασμούς τιμών των ανεξάρτητων μεταβλητών στον πίνακα αλήθειας θέτουμε την τιμή 0.

Στην περίπτωση που η τιμή 1 εμφανίζεται στον πίνακα αλήθειας σε ένα συγκεκριμένο συνδυασμό τιμών των ανεξάρτητων μεταβλητών για δύο ή περισσότερους όρους του λογικού αθροίσματος, κάνουμε χρήση του ορισμού της λογικής πράξης **OR**, σύμφωνα με τον οποίο **1 + 1 = 1**.

				F_1= AC'+B+AB'C'					F_2=B+AC'		
A	**B**	**C**		**AC'**	**B**	**AB'C'**	**F_1**		**B**	**AC'**	**F_2**
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**

Στο δεύτερο ερώτημα, η ισοδυναμία των δύο λογικών συναρτήσεων αποδεικνύεται με αλγεβρικούς μετασχηματισμούς ως ακολούθως:

$$
\begin{aligned}
F_1(A, B, C) &= AC' + B + AB'C' \\
&= A \cdot 1 \cdot C' + B + AB'C' \\
&= AC'(1 + B') + B \\
&= AC' \cdot 1 + B \\
&= AC' + B = B + AC' = F_2(A, B, C)
\end{aligned}
$$

Να σημειωθεί ότι υπάρχουν και άλλες λογικές συναρτήσεις που έχουν τον ίδιο πίνακα αλήθειας με τις F_1 και F_2, όπως, για παράδειγμα, οι ακόλουθες:

$$F_3(A, B, C) = A'BC' + A'BC + AB'C' + ABC' + ABC$$

$$F_4(A, B, C) = B + AB'C'$$

Συμπλήρωμα Λογικής συνάρτησης

Το συμπλήρωμα F’ μιας λογικής συνάρτησης F, προκύπτει από τον πίνακα αλήθειας της F, εάν εναλλάξουμε τα 0 και τα 1, όπως παρουσιάζεται στον πίνακα που ακολουθεί, για μια συνάρτηση τριών μεταβλητών.

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

Με αλγεβρικό τρόπο, μπορούμε να προσδιορίσουμε το συμπλήρωμα μιας λογικής συνάρτησης με χρήση των δύο μορφών του θεωρήματος De Morgan, που εξετάσαμε σε προηγούμενη ενότητα για δύο μεταβλητές:

$$(A + B)' = A' \cdot B' \qquad \text{και} \qquad (A \cdot B)' = A' + B'$$

Το θεώρημα De Morgan μπορεί να επεκταθεί, ώστε να ισχύει για περισσότερες από δύο μεταβλητές.

Για παράδειγμα, η πρώτη μορφή του θεωρήματος De Morgan για τρεις μεταβλητές προκύπτει ως εξής:

$$\begin{aligned}
(A + B + C)' &= (A + X)' && [\text{ όπου } X = B + C \text{ }] \\
&= A' \cdot X' && [\text{ από το θεώρημα De Morgan δύο μεταβλητών }] \\
&= A' \cdot (B + C)' && [\text{ αντικατάσταση του } X \text{ με το } B + C \text{ }] \\
&= A' \cdot (B' \cdot C') && [\text{ από το θεώρημα De Morgan δύο μεταβλητών }] \\
&= A' \cdot B' \cdot C' && [\text{ από την προσεταιριστική ιδιότητα }]
\end{aligned}$$

Αυτό επιβεβαιώνεται και με τη συμπλήρωση των αντίστοιχων πινάκων αλήθειας:

A	B	C	A+B+C	(A+B+C)’	A’	B’	C’	A’B’C’
0	0	0	0	1	1	1	1	1
0	0	1	1	0	1	1	0	0
0	1	0	1	0	1	0	1	0
0	1	1	1	0	1	0	0	0
1	0	0	1	0	0	1	1	0
1	0	1	1	0	0	1	0	0
1	1	0	1	0	0	0	1	0
1	1	1	1	0	0	0	0	0

Οι **γενικευμένες μορφές του θεωρήματος De Morgan** για οποιοδήποτε αριθμό μεταβλητών είναι οι εξής:

$$\mathbf{(x_1 + x_2 + x_3 + \ldots + x_n)' = x_1'\cdot x_2'\cdot x_3'\cdot\ldots\cdot x_n'}$$

$$\mathbf{(x_1\cdot x_2\cdot x_3\cdot\ldots\cdot x_n)' = x_1' + x_2' + x_3' + \ldots + x_n'}$$

Ένας πρακτικός τρόπος για τον προδιορισμό του συμπληρώματος μιας λογικής συνάρτησης είναι να εναλλάξουμε τους τελεστές AND και OR και να πάρουμε το συμπλήρωμα κάθε μεταβλητής.

Αυτό, προφανώς, ισοδυναμεί με το **να πάρουμε τη δυϊκή μορφή της συνάρτησης και να “συμπληρώσουμε” κάθε μεταβλητή**.

Παράδειγμα 3.4: Να προσδιοριστούν τα συμπληρώματα των λογικών συναρτήσεων:

$$\mathbf{F_1 = x'yz' + x'y'z} \text{ και } \mathbf{F_2 = x(y'z' + yz)}.$$

Για τη συνάρτηση F_1 θα έχουμε:

$F_1' = (x'yz' + x'y'z)' = (x'yz')'\cdot(x'y'z)'$ [θεώρημα De Morgan]

$= [(x')' + y' + (z')']\cdot[(x')' + (y')' + z']$ [θεώρημα De Morgan]

$= (x + y' + z)\cdot(x + y + z')$ [θεώρημα διπλής άρνησης]

Εναλλακτικά, με εφαρμογή της αρχής του δυϊσμού, έχουμε:

Η δυϊκή της F_1 είναι: $(x' + y + z')(x' + y' + z)$

και η συμπλήρωση κάθε μεταβλητής δίνει: $(x + y' + z)(x + y + z') = F_1'$

Για τη συνάρτηση F_2 θα έχουμε αντίστοιχα:

$$F_2' = [x \cdot (y'z' + yz)]' = x' + (y'z' + yz)'$$ [θεώρημα De Morgan]

Για τη συνάρτηση F_2 θα έχουμε αντίστοιχα:

$$\begin{aligned} F_2' &= [x(y'z' + yz)]' = x' + (y'z' + yz)' && [\text{ θεώρημα De Morgan }] \\ &= x' + (y'z')'(yz)' && [\text{ θεώρημα De Morgan }] \\ &= x' + (y + z)(y' + z') && [\text{ επιμεριστική ιδιότητα }] \\ &= x' + yy' + yz' + y'z + zz' && [\ yy' = 0,\ zz' = 0\ ] \\ &= x' + yz' + y'z \end{aligned}$$

Εναλλακτικά, με εφαρμογή της αρχής του δυϊσμού, έχουμε:

Η δυϊκή της F_2 είναι: $x + (y' + z')(y + z)$

Η συμπλήρωση κάθε μεταβλητής δίνει:

$$\begin{aligned} & x' + (y + z)(y' + z') && [\text{ επιμεριστική ιδιότητα }] \\ & = x' + y'z + yz' = F_2' \end{aligned}$$

Γενικευμένο θεώρημα απορρόφησης $[\,x + x \cdot y = x\,]$

Η γενίκευση του θεωρήματος απορρόφησης εκφράζεται ως ακολούθως:

$$x + F(\mathbf{x}, y, \ldots, w) = x + F(\mathbf{0}, y, \ldots, w)$$

όπου, στη συνάρτηση F, στο δεξί μέλος της ισότητας, η μεταβλητή **x** λαμβάνει την τιμή **0**.

Για παράδειγμα, εάν $F(x, y, z, w) = z \cdot (x \cdot y + w)$, θα έχουμε:

$$x + F(x, y, z, w) = x + z \cdot (\mathbf{x} \cdot y + w) = x + z \cdot (\mathbf{0} \cdot y + w) = x + z \cdot w$$

Επίσης, με εφαρμογή της αρχής του δυϊσμού θα έχουμε:

$$x \cdot F(\mathbf{x}, y, \ldots, w) = x \cdot F(\mathbf{1}, y, \ldots, w)$$

όπου, στη συνάρτηση F, στο δεξί μέλος της ισότητας, η μεταβλητή **x** λαμβάνει την τιμή **1**.

Θεώρημα επέκτασης ή ανάπτυξης συναρτήσεων του Shannon

Ο Claude Elwood Shannon πρότεινε το ακόλουθο θεώρημα:

Κάθε λογική συνάρτηση μπορεί να αναπτυχθεί ως προς μία από τις μεταβλητές που συμμετέχουν σε αυτήν, ως εξής:

$$F(\mathbf{x}, y, \ldots, w) = \mathbf{x} \cdot F(\mathbf{1}, y, \ldots, w) + \mathbf{x'} \cdot F(\mathbf{0}, y, \ldots, w)$$

και με εφαρμογή της αρχής του δυϊσμού:

$$F(\mathbf{x}, y, \ldots, w) = [\mathbf{x} + F(\mathbf{0}, y, \ldots, w)] \cdot [\mathbf{x'} + F(\mathbf{1}, y, \ldots, w)]$$

όπου οι συναρτήσεις $F(\mathbf{1}, y, \ldots, w)$ και $F(\mathbf{0}, y, \ldots, w)$ προκύπτουν από τη συνάρτηση $F(\mathbf{x}, y, \ldots, w)$ για $\mathbf{x = 1}$ και $\mathbf{x = 0}$, αντίστοιχα.

Παράδειγμα 3.5: Να γίνει ανάπτυξη σύμφωνα με το θεώρημα Shannon της λογικής συνάρτησης **F(x, y, z) = x’y’ + xz’ + yz** , ως προς τη μεταβλητή **x**, και, ακολούθως, ως προς τη μεταβλητή **y**.

Ανάπτυξη ως προς τη μεταβλητή x:

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

$$= x \cdot F(1, y, \ldots, w) + x' \cdot F(0, y, \ldots, w)$$

$$= x \cdot (0 \cdot y' + 1 \cdot z' + yz) + x' \cdot (1 \cdot y' + 0 \cdot z' + yz)$$

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

Οι εκφράσεις που βρίσκονται μέσα στις παρενθέσεις είναι συναρτήσεις των μεταβλητών y και z και μπορούν με όμοιο τρόπο να αναπτυχθούν ως προς μία από τις δύο αυτές μεταβλητές.

Η ανάπτυξη αυτής της έκφρασης της F(x, y, z) ως προς τη μεταβλητή y, θα μας δώσει:

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

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

$$= x[y(z' + z) + y'(z' + 0)] + x'[y(0 + z) + y'(1 + 0)]$$

$$= x(y\cdot 1 + y'z') + x'(yz + y'\cdot 1)$$

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

$$= xy + xy'z' + x'yz + x'y'$$

Λογικά κυκλώματα

Ένα λογικό κύκλωμα αποτελείται από **γραμμές διασύνδεσης** λογικών πυλών, που **αντιστοιχούν στις διαδρομές των δυαδικών σημάτων**, και από **λογικές πύλες**, που **επιτελούν την επεξεργασία των σημάτων**, η οποία περιγράφεται με τη λογική συνάρτηση που επιτελείται από κάθε πύλη.

Ένα **λογικό κύκλωμα** είναι το **σχηματικό μοντέλο ενός ηλεκτρονικού κυκλώματος** (:φυσικό κύκλωμα) και **περιγράφει σχηματικά** (: γραφικά σύμβολα πυλών) τις **λογικές πράξεις** που εκτελούνται **μεταξύ των σημάτων** εισόδου και παράγουν **σήματα εξόδου**, τα οποία περιγράφονται μέ **λογικές συναρτήσεις**.

Τα απλούστερα λογικά κυκλώματα είναι οι λογικές πύλες, οι οποίες μπορούν να διασυνδεθούν κατάλληλα και να σχηματίσουν πιο σύνθετα λογικά κυκλώματα.

Γνωρίζουμε ότι **μια λογική συνάρτηση**:

- μπορεί να περιγραφεί με **έναν και μοναδικό πίνακα αλήθειας**
- μπορεί να έχει **πολλές διαφορετικές αλγεβρικές εκφράσεις**, οι οποίες όμως παράγουν τον **ίδιο πίνακα αλήθειας**.

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.

Αυτή ακριβώς είναι και η έννοια της **ψηφιακής λογικής σχεδίασης**:

ένα κύκλωμα μπορεί να υλοποιηθεί με πολλούς διαφορετικούς τρόπους.

Το ερώτημα που τίθεται είναι "ποια από το σύνολο των υλοποιήσεων είναι η προτιμότερη κατά τη σχεδίαση ενός ψηφιακού συστήματος".

Κάποια από τα λογικά κυκλώματα που υλοποιούν μια συνάρτηση είναι απλούστερα από άλλα και σημαντικός στόχος κατά τη σχεδίαση είναι η μείωση του κόστους υλοποίησης.

Το κόστος υλοποίησης συνδέεται, προφανώς, με το πλήθος των λογικών πυλών που χρησιμοποιούνται.

Από την άλλη πλευρά, οι λογικές πύλες NAND και NOR υλοποιούνται, ως ηλεκτρονικά κυκλώματα, ευκολότερα και με μικρότερο κόστος συγκριτικά με τις πύλες AND και OR.

Επιπλέον, οι πύλες δύο εισόδων παρουσιάζουν καλύτερα χαρακτηριστικά λειτουργίας συγκριτικά με τις πύλες πολλών εισόδων, όπως μικρότερη καθυστέρηση απόκρισης και μεγαλύτερη ανεκτικότητα στην παρουσία θορύβου.

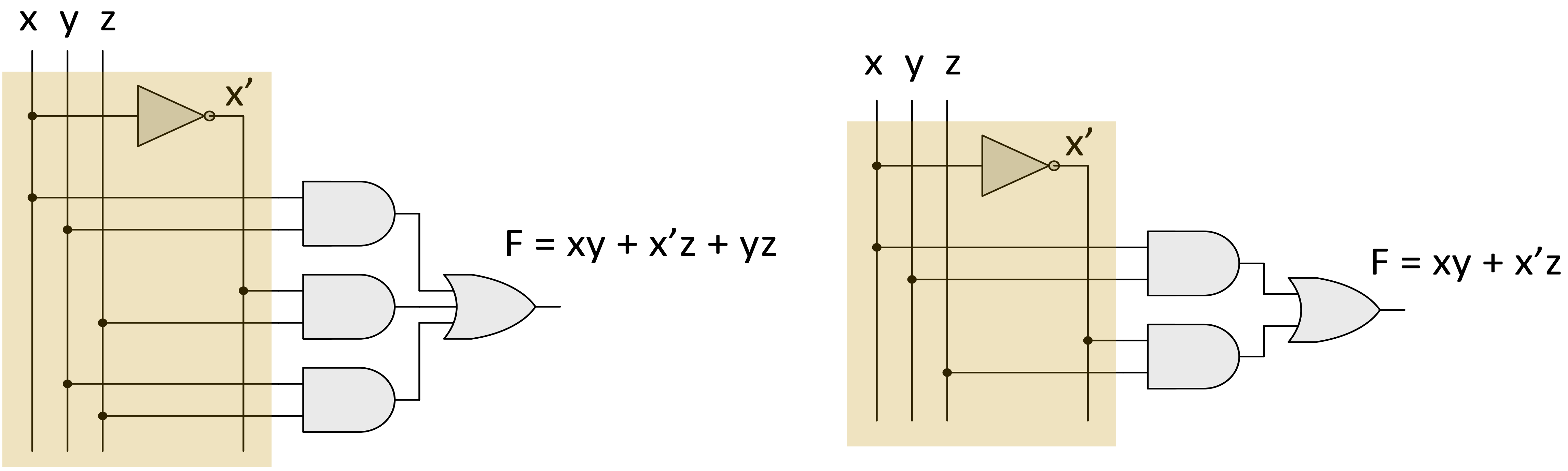
Όλα τα παραπάνω πρέπει να λαμβάνονται υπόψη κατά τη σχεδίαση των κυκλωμάτων.

Σύνθεση λογικών κυκλωμάτων με λογικές πύλες

Στο Παράδειγμα 3.1 αποδείξαμε ότι η λογική συνάρτηση **F(x, y, z) = xy + x’z + yz** απλοποιείται στην **F(x, y, z) = xy + x’z**.

Τα λογικά κυκλώματα που υλοποιούν τις δύο ισοδύναμες μορφές αυτής της λογικής συνάρτησης παρουσιάζονται στο Σχήμα 3.12 και περιλαμβάνουν δύο **επίπεδα πυλών** (εάν δεν συνυπολογίσουμε τον αντιστροφέα που απαιτείται για την παραγωγή της συμπληρωματικής μορφής μιας από τις εισόδους).

Ως **επίπεδο πυλών** ορίζεται **το ελάχιστο πλήθος πυλών που πρέπει να περάσουν τα σήματα εισόδου για να φτάσουν στην έξοδο**, χωρίς να προσμετράμε τις πύλες NOT που πιθανώς απαιτούνται για την εξαγωγή των συμπληρωματικών μορφών των μεταβλητών εισόδου.



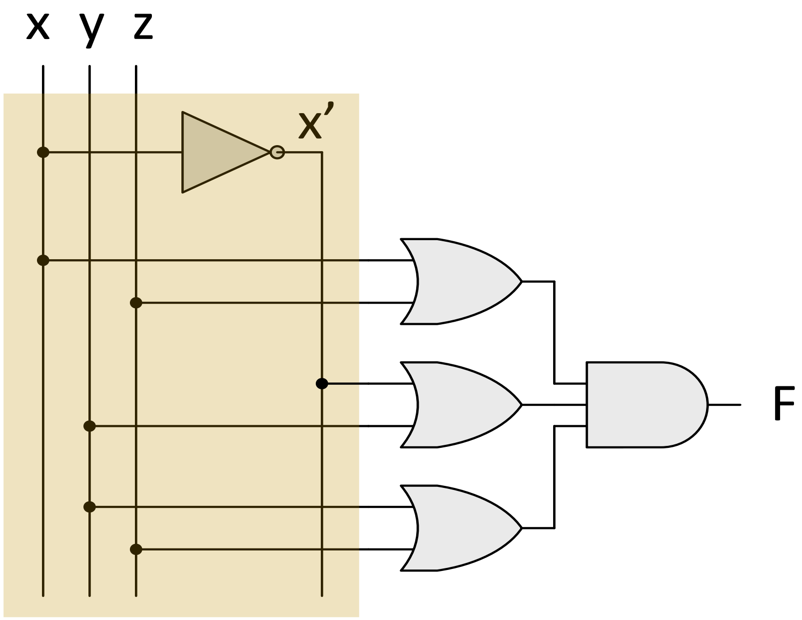
(α) (β)

Σχήμα 3.12 Λογικά κυκλώματα που υλοποιούν τις συναρτήσεις:
(α) F(x, y, z) = xy + x'z + yz και (β) F(x, y, z) = xy + x'z

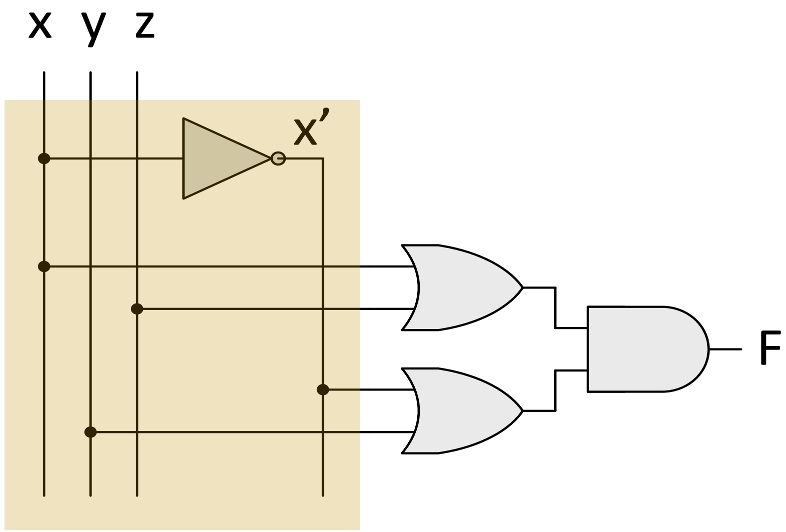
Παρατηρούμε ότι, η απλοποιημένη μορφή της F(x, y, z) μπορεί να υλοποιηθεί με μία πύλη AND λιγότερη και απαιτεί μια πύλη OR δύο ειδόδων, αντί τριών εισόδων που έχουμε στην υλοποίηση της αρχικής μορφής της συνάρτησης.

Για τους λόγους αυτούς, η δεύτερη υλοποίηση είναι προτιμότερη.

Επίσης, όπως παρουσιάζεται στο Σχήμα 3.13, δύο άλλες ισοδύναμες υλοποιήσεις προκύπτουν, εάν εκφράσουμε την αρχική και την απλοποιημένη συνάρτηση σε μορφές γινόμενου αθροισμάτων, εφαρμόζοντας το αξίωμα της επιμεριστικότητας στην αρχική συνάρτηση:
F(x, y, z) = xy + x’z + yz = (x + z)(x’ + y)(y + z), η οποία απλοποιείται στην **F(x, y, z) = (x + z)(x’ + y).**



(α)



(β)

Σχήμα 3.13 Λογικά κυκλώματα που υλοποιούν τις συναρτήσεις:
(α) F(x, y, z) = (x + z)(x’ + y)(y + z) και (β) F(x, y, z) = (x + z)(x’ + y)

Η χρήση παράγωγων πυλών, όπου αυτό είναι εφικτό, είναι συνηθισμένη και οδηγεί επίσης σε απλούστερα λογικά κυκλώματα.

Για παράδειγμα, ας θεωρήσουμε τη λογική συνάρτηση

$$\mathbf{F(x, y, z) = \Sigma(1, 2, 4, 7) = x'y'z + x'yz' + xy'z' + xyz}$$

Η συνάρτηση αυτή δεν μπορεί να απλοποιηθεί σε επίπεδο βασικών πυλών μόνο, όπως θα αποδειχθεί αργότερα.

Η υλοποίησή της απαιτεί:

- **4 πύλες AND τριών εισόδων και 1 πύλη OR τεσσάρων εισόδων**

σε δύο επίπεδα πυλών (χωρίς να λαμβάνουμε υπόψη τις πύλες NOT που απαιτούνται για την εξαγωγή των συμπληρωματικών μορφών των μεταβλητών εισόδου)

ενώ, αν την υλοποιήσουμε **με πύλες δύο εισόδων μόνο**, απαιτεί:

- **8 πύλες AND και 3 πύλες OR σε τέσσερα επίπεδα πυλών**.

Είναι δυνατό όμως να υλοποιηθεί **με δύο πύλες XOR δύο εισόδων σε δύο επίπεδα πυλών ή με μία μόνο πύλη XOR τριών εισόδων σε ένα επίπεδο πυλών**, όπως αποδεικνύεται με αλγεβρικούς μετασχηματισμούς:

$F(x, y, z) = x'y'z + x'yz' + xy'z' + xyz$ [εφαρμόζουμε την επιμεριστική ιδιότητα]

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

$= x'(y \oplus z) + x(y \oplus z)'$ [θέτουμε $y \oplus z = A$]

$= x'A + xA'$

$= x \oplus A$ [αντικαθιστούμε $A = y \oplus z$]

$= x \oplus (y \oplus z) = x \oplus y \oplus z$

Σχεδίαση λογικών κυκλωμάτων μόνο με πύλες NAND ή NOR

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

Για το λόγο αυτό, αποτελούν τις βασικές πύλες υλοποίησης σε όλες τις οικογένειες ψηφιακών ολοκληρωμένων κυκλωμάτων (Chips).

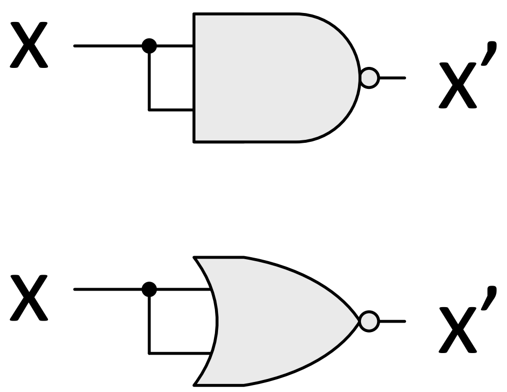
Λόγω αυτής της κυριαρχίας των πυλών NAND και NOR στη σχεδίαση ψηφιακών ολοκληρωμένων κυκλωμάτων, έχουν αναπτυχθεί κανόνες και διαδικασίες για τη μετατροπή λογικών συναρτήσεων που περιέχουν τελεστές AND, OR και NOT σε ισοδύναμα λογικά κυκλώματα είτε με πύλες NAND μόνο, είτε με πύλες NOR μόνο.

Οι πύλες NAND και NOR χαρακτηρίζονται ως **οικουμενικές πύλες**, επειδή, όπως θα αποδείξουμε στη συνέχεια, οι βασικές λογικές πύλες AND, OR και NOT και, κατ’ επέκταση, οποιοδήποτε ψηφιακό κύκλωμα μπορεί να υλοποιηθεί μόνο με αυτές.

Από τον πίνακα αλήθειας για τις πύλες NAND και NOR δύο εισόδων παρατηρούμε ότι, όταν και οι δύο είσοδοι των πυλών έχουν τιμή 1, η έξοδος λαμβάνει τιμή 0, και όταν και οι δύο είσοδοι των πυλών έχουν τιμή 0, η έξοδος λαμβάνει τιμή 1, δηλαδή αντιστρέφουν την τιμή που έχουν οι είσοδοι.

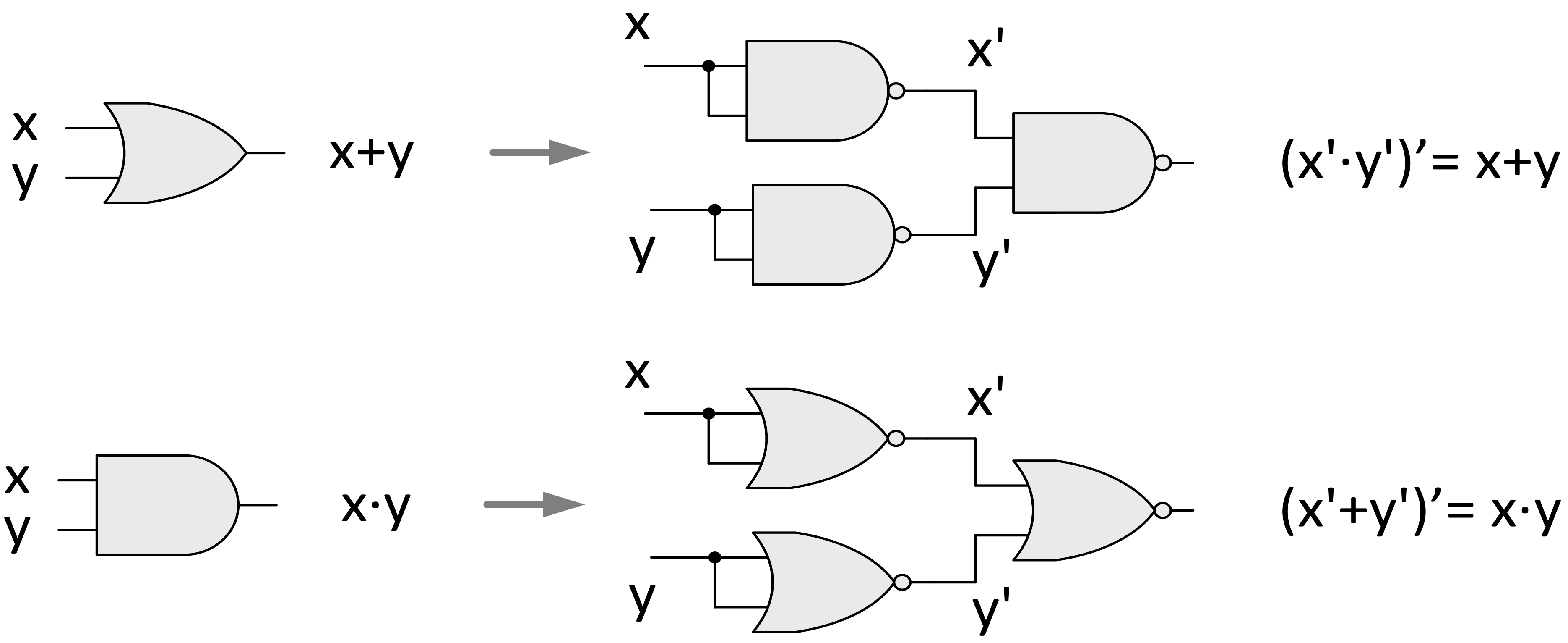
Άρα, μπορούμε να υλοποιήσουμε τη λογική πύλη NOT, **βραχυκυκλώνοντας (:συνδέοντας μεταξύ τους) τις εισόδους** μιας πύλης NAND ή μιας πύλης NOR, όπως παρουσιάζεται στο Σχήμα 3.14.

x	y	(x•y)’	(x+y)’
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0



Η δυνατότητα υλοποίησης των πυλών AND και OR μόνο με πύλες NOR και NAND αντίστοιχα, αποδεικνύονται με εφαρμογή του θεωρήματος διπλής άρνησης και του θεωρήματος De Morgan, ως ακολούθως:

$$\mathbf{x + y = [(x + y)']' = (x'y')'} \quad \text{και} \quad \mathbf{xy = [(xy)']' = (x' + y')'}$$



Σχήμα 3.15 Υλοποίηση σε λογικό κύκλωμα των λογικών πυλών OR και AND μόνο με πύλες NAND και NOR

Παράδειγμα 3.6: Δίνεται η λογική συνάρτηση **$F(x, y, z) = xy + x'z + yz$**.

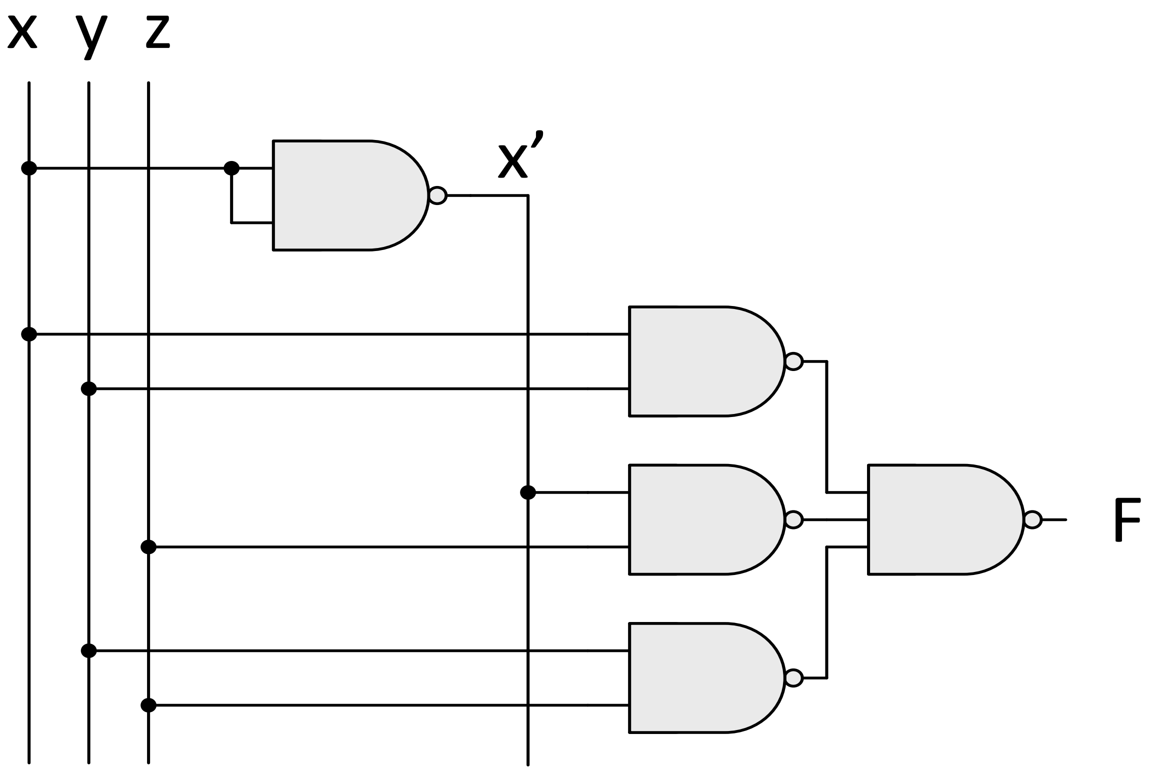
1. Να υλοποιηθεί η συνάρτηση F μόνο με πύλες NAND.
2. Να αποδειχθεί ότι η συνάρτηση μπορεί να εκφραστεί σε μορφή γινόμενου αθροισμάτων ως $F(x, y, z) = (x + z)(x' + y)(y + z)$ και να υλοποιηθεί η συνάρτηση μόνο με πύλες NOR.

1. Με εφαρμογή του θεωρήματος διπλής άρνησης και του θεωρήματος De Morgan, λαμβάνουμε:

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

Παρατηρούμε ότι, στην αλγεβρική αυτή έκφραση συμμετέχουν μόνο λογικές πράξεις NAND και NOT, η οποία όμως μπορεί να υλοποιηθεί με μια πύλη NAND με βραχυκλωμένες εισόδους (Σχήμα 3.14).

Το λογικό κύκλωμα παρουσιάζεται στο Σχήμα 3.16 (α).



(α)

Σχήμα 3.16   Υλοποίηση της λογικής συνάρτησης F(x, y, z) = xy + x’z + yz
(α) μόνο με πύλες NAND : F(x, y, z) = [(xy)’(x’z)’(yz)’]’

2. Ένας εύκολος τρόπος απόδειξης είναι με χρήση της αρχής του δυϊσμού, δηλαδή:

- να εξαγάγουμε τη δυϊκή έκφραση της αρχικής συνάρτησης,
- να εφαρμόσουμε την επιμεριστική ιδιότητα σε αυτή και, ακολούθως,
- να εφαρμόσουμε την αρχή του δυϊσμού στο αποτέλεσμα:

Δ[F(x, y, z)] = (x + y)(x’ + z)(y + z)	[Δυϊκή της αρχικής συνάρτησης]
= (xx’ + xz + x’y + yz)(y + z)	[Επιμεριστική ιδιότητα]
= xyz + x’y + yz + xz + x’yz	[xx’ = 0, xx = x, zz = z]
= xz(1 + y) + x’y(1 + z) + yz	[Επιμεριστική ιδιότητα]
= xz + x’y + yz	[1 + y = 1, 1 + z =1]

- Λαμβάνοντας τη δυϊκή έκφραση του αποτελέσματος, έχουμε:

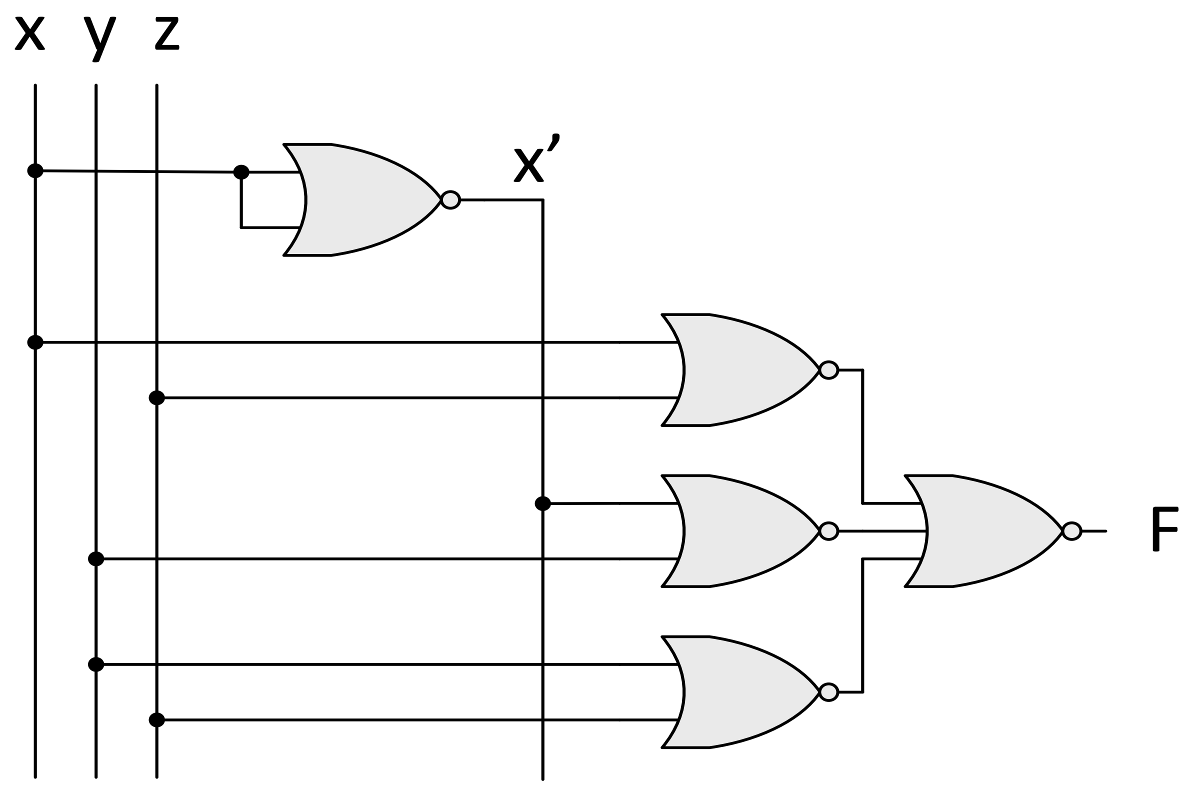
F(x, y, z) = (x + z)(x’ + y)(y + z)

Με εφαρμογή του θεωρήματος διπλής άρνησης και του θεωρήματος De Morgan, λαμβάνουμε:

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

Παρατηρούμε ότι, στην αλγεβρική αυτή έκφραση συμμετέχουν μόνο λογικές πράξεις NOR και NOT, η οποία όμως μπορεί να υλοποιηθεί με μια πύλη NOR με βραχυκλωμένες εισόδους (Σχήμα 3.14).

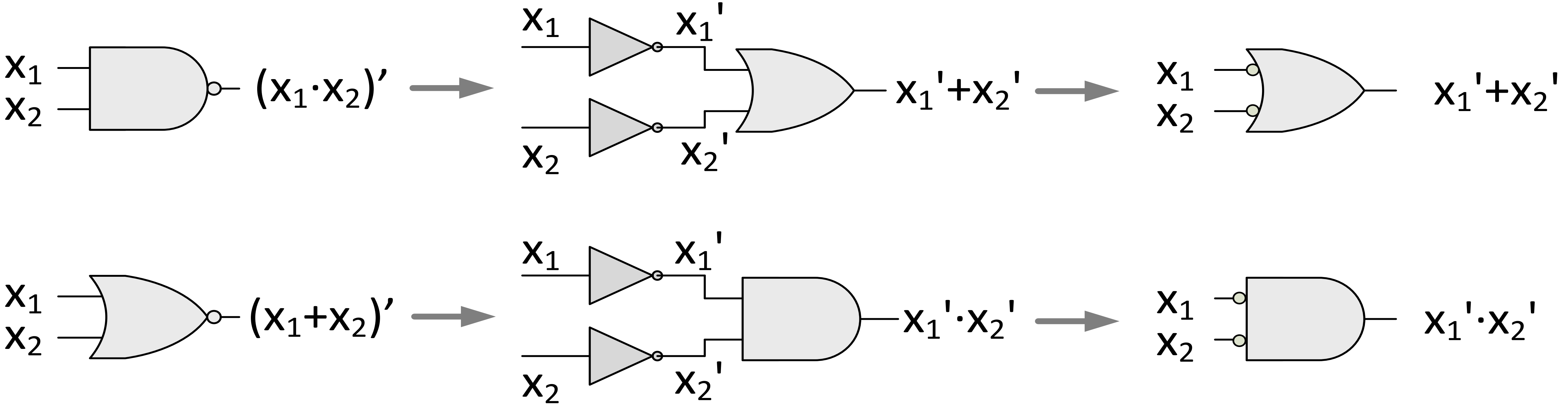
Το λογικό κύκλωμα παρουσιάζεται στο Σχήμα 3.16 (β).



(β)

Σχήμα 3.16   Υλοποίηση της λογικής συνάρτησης F(x, y, z) = xy + x’z + yz
(β) μόνο με πύλες NOR: F(x, y, z) = [(x + z)’ + (x’ + y)’ + (y + z)’]’

Κλείνοντας αυτή την ενότητα, θεωρούμε χρήσιμο να παραθέσουμε δύο **εναλλακτικά γραφικά σύμβολα για τις λογικές πύλες NAND και NOR**, τα οποία προκύπτουν από το θεώρημα De Morgan, $\mathbf{(x_1x_2)' = x_1' + x_2'}$ και $\mathbf{(x_1 + x_2)' = x_1'x_2'}$, και η υλοποίηση αυτών των σχέσεων σε λογικό κύκλωμα παρουσιάζεται στο Σχ. 3.17.



Σχήμα 3.17 Υλοποίηση σε λογικό κύκλωμα των σχέσεων του θεωρήματος De Morgan

Το “**κυκλάκι**” (bubble) στις εισόδους των πυλών AND και OR **συμβολίζει το συμπλήρωμα της λογικής τιμής** των εισόδων.

Τα δύο αυτά γραφικά σύμβολα είναι χρήσιμα στην ανάλυση και σχεδίαση λογικών κυκλωμάτων με πύλες NAND και NOR και όταν εμφανίζονται στο ίδιο λογικό κύκλωμα, λέμε ότι έχουμε περίπτωση μικτής σημειογραφίας.

Κανονικές και πρότυπες μορφές των λογικών συναρτήσεων

Ένα λογικό γινόμενο (πράξη AND), στο οποίο συμμετέχουν όλες οι ανεξάρτητες μεταβλητές μιας λογικής συνάρτησης στην κανονική ή τη συμπληρωματική τους μορφή μία φορά, ονομάζεται **ελαχιστόρος** (minterm).

Έτσι, σε κάθε δυνατό συνδυασμό τιμών των ανεξάρτητων μεταβλητών ενός πίνακα αλήθειας (εισόδων του συστήματος), αντιστοιχεί ένας ελαχιστόρος που σχηματίζεται από το λογικό γινόμενο των μεταβλητών, θέτοντας κάθε μεταβλητή σε κανονική μορφή, αν σε εκείνο το συνδυασμό τιμών του πίνακα αλήθειας έχει τιμή 1 και, αντίστοιχα, σε συμπληρωματική μορφή αν έχει τιμή 0.

Κάθε ελαχιστόρος συμβολίζεται ως $\mathbf{m_i}$, όπου ο δείκτης **i** συμπίπτει με τη δεκαδική αξία του συνδυασμού τιμών των μεταβλητών στον πίνακα αλήθειας, στον οποίο αντιστοιχεί ο ελαχιστόρος.

Ένα λογικό άθροισμα (πράξη OR), στο οποίο συμμετέχουν όλες οι ανεξάρτητες μεταβλητές μιας λογικής συνάρτησης, στην κανονική ή τη συμπληρωματική τους μορφή μία φορά, ονομάζεται **μεγιστόρος** (Maxterm).

Έτσι, σε κάθε δυνατό συνδυασμό τιμών των ανεξάρτητων μεταβλητών ενός πίνακα αλήθειας, αντιστοιχεί ένας μεγιστόρος που σχηματίζεται από το λογικό άθροισμα των μεταβλητών, θέτοντας τη μεταβλητή σε συμπληρωματική μορφή, αν σε εκείνο το συνδυασμό τιμών του πίνακα αλήθειας έχει τιμή 1 και, αντίστοιχα, σε κανονική μορφή αν έχει τιμή 0.

Κάθε μεγιστόρος συμβολίζεται ως $\mathbf{M_j}$, όπου ο δείκτης **j** συμπίπτει με τη δεκαδική αξία του συνδυασμού τιμών των ανεξάρτητων μεταβλητών στον οποίο αντιστοιχεί ο μεγιστόρος.

Προφανώς, το πλήθος των ελαχιστόρων και μεγιστόρων είναι ίσο με το πλήθος των δυνατών συνδυασμών τιμών των ανεξάρτητων μεταβλητών, δηλαδή $\mathbf{2^n}$.

x	y	Ελαχιστόροι		Μεγιστόροι	
0	0	x’y’ 0 0	**m_0**	x+y 0 0	**M_0**
0	1	x’y 0 1	**m_1**	x+y’ 0 1	**M_1**
1	0	x y’ 1 0	**m_2**	x’+y 1 0	**M_2**
1	1	x y 1 1	**m_3**	x’+y’ 1 1	**M_3**

Παρατηρούμε ότι, σύμφωνα με το θεώρημα De Morgan, κάθε μεγιστόρος ισούται με το συμπλήρωμα του αντίστοιχου ελαχιστόρο και αντιστρόφως.

Για παράδειγμα, ο μεγιστόρος $M_2 = x' + y$ ισούται με το συμπλήρωμα του αντίστοιχου ελαχιστόρου: $m_2' = (xy')' = x' + y'' = x' + y$.

Αντίστοιχα, ο ελαχιστόρος $m_0 = x'y'$ ισούται με το συμπλήρωμα του αντίστοιχου μεγιστόρου: $M_0' = (x + y)' = x'y'$.

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’yz’	m_2	x+y’+z	M_2
0	1	1	x’yz	m_3	x+y’+z’	M_3
1	0	0	xy’z’	m_4	x’+y+z	M_4
1	0	1	xy’z	m_5	x’+y+z’	M_5
1	1	0	xyz’	m_6	x’+y’+z	M_6
1	1	1	xyz	m_7	x’+y’+z’	M_7

Μπορούμε να προσδιορίσουμε και να διατυπώσουμε την αλγεβρική έκφραση μιας λογικής συνάρτησης F από τον πίνακα αλήθειας αυτής με δύο εναλλακτικούς τρόπους:

1. Εντοπίζουμε τους συνδυασμούς τιμών των μεταβλητών για για τους οποίους η συνάρτηση λαμβάνει τιμή 1 και συνθέτουμε το λογικό άθροισμα των αντίστοιχων ελαχιστόρων. Στην περίπτωση αυτή λέμε ότι η συνάρτηση F εκφράζεται σε κανονική μορφή ως άθροισμα ελαχιστόρων.
2. Εντοπίζουμε τους συνδυασμούς τιμών των μεταβλητών για για τους οποίους η συνάρτηση λαμβάνει τιμή 0 και συνθέτουμε το λογικό γινόμενο των αντίστοιχων μεγιστόρων. Στην περίπτωση αυτή λέμε ότι η συνάρτηση F εκφράζεται σε κανονική μορφή ως γινόμενο μεγιστόρων.

Παράδειγμα 3.7: Να συμπληρωθεί ο πίνακας αλήθειας της συνάρτησης πλειοψηφίας τριών εισόδων, F(x, y, z), και του συμπληρώματος αυτής και να εκφραστούν η λογική συνάρτηση F, καθώς και το συμπλήρωμα αυτής F’, ως άθροισμα ελαχιστόρων και ως γινόμενο μεγιστόρων.

Διευκρινίζεται ότι, ως συνάρτηση πλειοψηφίας τριών εισόδων F(x, y, z) ορίζεται η λογική συνάρτηση που “αναγνωρίζει” πότε σε τρεις εισόδους (: ανεξάρτητες μεταβλητές) πλειοψηφούν τα 1 έναντι των 0.

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

Παρατηρούμε ότι, ο πίνακας αλήθειας της F έχει τιμή 1 στους συνδυασμούς τιμών των ανεξάρτητων μεταβλητών ‘011’, ‘101’, ‘110’ και ‘111’, στους οποίους η τιμή 1 πλειοψηφεί, ενώ για τους υπόλοιπους συνδυασμούς λαμβάνει τιμή 0.

Επομένως, σε μορφή αθροίσματος ελαχιστόρων, η συνάρτηση F θα είναι:

$$F(x, y, z) = m_3 + m_5 + m_6 + m_7 = \Sigma(m_3, m_5, m_6, m_7) = \Sigma(3, 5, 6, 7)$$
$$= x'yz + xy'z + xyz' + xyz$$

Είναι προφανές ότι, το συμπλήρωμα F’ της συνάρτησης σε μορφή αθροίσματος ελαχιστόρων, θα περιλαμβάνει τους υπόλοιπους ελαχιστόρους, δηλαδή:

$$F'(x, y, z) = m_0 + m_1 + m_2 + m_4 = \Sigma(m_0, m_1, m_2, m_4) = \Sigma(0, 1, 2, 4)$$
$$= x'y'z' + x'y'z + x'yz' + xy'z'$$

Αντίστοιχα, ο πίνακας αλήθειας της F έχει τιμή 0 στους συνδυασμούς τιμών των ανεξάρτητων μεταβλητών ‘000’, ‘001’, ‘010’ και ‘100’, οπότε, σε μορφή γινόμενου αθροισμάτων, η συνάρτηση F θα είναι:

$$F(x, y, z) = M_0 \cdot M_1 \cdot M_2 \cdot M_4 = \Pi(M_0, M_1, M_2, M_4) = \Pi(0, 1, 2, 4)$$
$$= (x + y + z) \cdot (x + y + z') \cdot (x + y' + z) \cdot (x' + y + z)$$

και το συμπλήρωμα αυτής:

$$F(x, y, z) = M_3 \cdot M_5 \cdot M_6 \cdot M_7 = \Pi(M_3, M_5, M_6, M_7) = \Pi(3, 5, 6, 7)$$
$$= (x' + y + z) \cdot (x' + y + z') \cdot (x' + y' + z) \cdot (x' + y' + z')$$

Η έκφραση της συνάρτησης F σε μορφή γινόμενου αθροισμάτων μπορεί να προδιοριστεί και με την εφαρμογή του θεωρήματος διπλής άρνησης της άλγεβρας Boole, (F')' = F, λαμβάνοντας το συμπλήρωμα του συμπληρώματος της έκφρασης της F σε μορφή αθροίσματος γιμομένων:

$$
\begin{aligned}
F(x, y, z) &= [F'(x, y, z)]' \\
&= (m_0 + m_1 + m_2 + m_4)' \\
&= m_0' \cdot m_1' \cdot m_2' \cdot m_4' \\
&= M_0 \cdot M_1 \cdot M_2 \cdot M_4 \\
&= (x + y + z) \cdot (x + y + z') \cdot (x + y' + z) \cdot (x' + y + z)
\end{aligned}
$$

Οι αλγεβρικές εκφράσεις μιας λογικής συνάρτησης σε μορφή αθροίσματος γινομένων και γινόμενου αθροισμάτων, αναφέρονται ως **κανονικές μορφές**.

Οι μορφές αυτές συνήθως δεν είναι οι απλούστερες δυνατές όσον αφορά στο πλήθος των λογικών πυλών που απαιτούνται για την υλοποίηση κυκλωμάτων.

Για το λόγο αυτό είναι επιθυμητό οι λογικές συναρτήσεις που υλοποιούνται με ηλεκτρονικά κυκλώματα να εκφράζονται σε απλούστερες μορφές είτε σε μορφή αθροίσματος λογικών γινομένων, είτε σε μορφή γινόμενου λογικών αθροισμάτων, χωρίς όμως αυτά τα λογικά γινόμενα ή τα λογικά αθροίσματα να περιλαμβάνουν όλες τις ανεξάρτητες μεταβλητές.

Οι μορφές αυτές αναφέρονται ως **πρότυπες μορφές**.

Ας πάρουμε, ως παράδειγμα, τη λογική συνάρτηση τριών ανεξάρτητων μεταβλητών

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

Αυτή είναι εκφρασμένη σε πρότυπη μορφή και όχι σε κανονική μορφή, αφού δεν είναι σε μορφή αθροίσματος ελαχιστόρων.

Παράλληλα, έχουμε αποδείξει (Παράδειγμα 3.1) ότι η συγκεκριμένη συνάρτηση μπορεί να εκφραστεί στην ακόμα απλούστερη ισοδύναμη μορφή

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

Επειδή το ζητούμενο είναι να υλοποιούμε σε ηλεκτρονικό κύκλωμα την απλούστερη δυνατή ισοδύναμη μορφή μιας λογικής συνάρτησης, θα πρέπει πάντα να επιδιώκουμε την εύρεση αυτής της απλούστερης μορφής με απλοποίηση της αρχικής μορφής της λογικής συνάρτησης.

Έχουμε ήδη παρουσιάσει τη διαδικασία της απλοποίησης λογικών συναρτήσεων με αλγεβρικούς μετασχηματισμούς και έχουμε, επίσης, επισημάνει ότι η απλοποίηση με τη χρήση αλγεβρικών μετασχηματισμών δεν οδηγεί πάντοτε με βεβαιότητα στην απλούστερη δυνατή (ελαχιστοποιημένη) ισοδύναμη αλγεβρική έκφραση, επειδή οι αλγεβρικοί μετασχηματισμοί δεν μπορούν να συστήσουν συγκεκριμένη μεθοδολογία απλοποίησης.

Όπως θα δούμε σε επόμενη ενότητα, υπάρχουν συστηματικοί τρόποι ελαχιστοποίησης αλγεβρικών εκφράσεων, στους οποίους κατά κανόνα χρησιμοποιούμε τις κανονικές μορφές (άθροισμα ελαχιστόρων ή γινόμενο μεγιστόρων) των λογικών συναρτήσεων.

Για το λόγο αυτό θα παρουσιάσουμε τη διαδικασία εξαγωγής της κανονικής μορφής μια συνάρτησης από την πρότυπη μορφή αυτής.

Για να μετατρέψουμε μια πρότυπη μορφή αθροίσματος γινομένων σε κανονική, ακολουθούμε τα εξής βήματα:

1. Σε κάθε λογικό γινόμενο προσδιορίζουμε ποια ή ποιες μεταβλητές δεν συμμετέχουν.
2. Για κάθε μεταβλητή η οποία δεν συμμετέχει στο γινόμενο εισάγουμε ένα 1 στη θέση της μεταβλητής (αξίωμα Α4, $x \cdot 1 = x$).
3. Κάθε 1 που εισάγουμε σε ένα γινόμενο εκφράζεται στη μορφή λογικού αθροίσματος της μεταβλητής και του συμπληρώματος αυτής (αξίωμα Α5, $x + x' = 1$).
4. Εφαρμόζουμε την επιμεριστική ιδιότητα και λαμβάνουμε την επιθυμητή κανονική μορφή.
5. Σε περίπτωση που προκύψουν επαναλαμβανόμενοι ελαχιστόροι, λαμβάνουμε υπόψη τον ελαχιστόρο μόνο μια φορά (θεώρημα Θ1, $x + x = x$).

Παράδειγμα 3.8: Δίνεται σε πρότυπη μορφή ως άθροισμα γινομένων η συνάρτηση:

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

Να εκφραστεί η συνάρτηση F σε κανονική μορφή αθροίσματος ελαχιστόρων και γινόμενου αθροισμάτων και να συμπληρωθεί ο πίνακας αλήθειας.

Ακολουθούμε τα βήματα που περιγράψαμε αμέσως προηγούμενα:

$$\begin{aligned}
F(x, y, z) &= y + xz' + x'yz \\
&= \mathbf{1}\cdot y\cdot \mathbf{1} + x\cdot \mathbf{1}\cdot z' + x'yz \\
&= (\mathbf{x + x'})y(\mathbf{z + z'}) + x(\mathbf{y + y'})z' + x'yz \\
&= xy(z + z') + x'y(z + z') + xyz' + xy'z' + x'yz \\
&= xyz + xyz' + x'yz + x'yz' + \cancel{xyz'} + xy'z' + \cancel{x'yz} \\
&= xyz + xyz' + x'yz + x'yz' + xy'z' \\
&\quad (111)\ \ (110)\ \ (011)\ \ (010)\ \ (100) \\
&= m_7 + m_6 + m_3 + m_2 + m_4 \\
&= M_0\cdot M_1\cdot M_5 \\
&= (x + y + z)(x + y + z')(x' + y + z')
\end{aligned}$$

x	y	z	F(x, y, z)		
0	0	0	0		M_0
0	0	1	0		M_1
0	1	0	1	m_2	
0	1	1	1	m_3	
1	0	0	1	m_4	
1	0	1	0		M_5
1	1	0	1	m_6	
1	1	1	1	m_7	

Για να μετατρέψουμε μια πρότυπη μορφή γινόμενου αθροισμάτων σε κανονική, ακολουθούμε τα εξής βήματα:

1. Σε κάθε λογικό άθροισμα προσδιορίζουμε ποια ή ποιες μεταβλητές δεν συμμετέχουν.
2. Για κάθε μεταβλητή η οποία δεν συμμετέχει στο άθροισμα εισάγουμε ένα 0 στη θέση της μεταβλητής (αξίωμα Α4, $x + 0 = x$).
3. Κάθε 0 που εισάγουμε σε ένα άθροισμα εκφράζεται στη μορφή λογικού γινομένου της μεταβλητής και του συμπληρώματος αυτής (αξίωμα Α5, $x \cdot x' = 0$).
4. Εφαρμόζουμε την επιμεριστική ιδιότητα και λαμβάνουμε την επιθυμητή κανονική μορφή.
5. Σε περίπτωση που προκύψουν επαναλαμβανόμενοι μεγιστόροι, λαμβάνουμε υπόψη τον μεγιστόρο μόνο μια φορά (θεώρημα Θ1, $x \cdot x = x$)

Παράδειγμα 3.9: Δίνεται σε πρότυπη μορφή ως γινόμενο αθροισμάτων η συνάρτηση $F(x, y, z) = (x + y')(x' + z)$. Να εκφραστεί η συνάρτηση F σε κανονική μορφή γινόμενου μεγιστόρων και αθροίσματος ελαχιστόρων και να συμπληρωθεί ο πίνακας αλήθειας.

Ακολουθούμε τα βήματα που περιγράψαμε αμέσως προηγούμενα:

$$
\begin{aligned}
F(x, y, z) &= (x + y')(x' + z) \\
&= (x + y' + \mathbf{0})(x' + \mathbf{0} + z) \\
&= (x + y' + \mathbf{zz'})(x' + \mathbf{yy'} + z) \\
&= (x + y' + z)(x + y' + z')(x' + y + z)(x' + y' + z) \\
&\quad\ (0\ 1\ 0)\qquad (0\ 1\ 1)\qquad (1\ 0\ 0)\qquad (1\ 1\ 0) \\
&= M_2 \cdot M_3 \cdot M_4 \cdot M_6 = m_0 + m_1 + m_5 + m_7 \\
&= x'y'z' + x'y'z + xy'z + xyz
\end{aligned}
$$

x	y	z	F(x, y, z)		
0	0	0	1		m_0
0	0	1	1		m_1
0	1	0	0	M_2	
0	1	1	0	M_3	
1	0	0	0	M_4	
1	0	1	1		m_5
1	1	0	0	M_6	
1	1	1	1		m_7

Πηγές

Για τη δημιουργία των σημειώσεων των διαλέξεων του μαθήματος έχει χρησιμοποιηθεί υλικό από την παρακάτω βιβλιογραφία:

- Morris Mano, M. & Ciletti, M.D. (2013). Ψηφιακή Σχεδίαση (5η έκδοση). Εκδόσεις Παπασωτηρίου.
- Nelson, V.P., Troy Nagle, H., David Irwin, J. & Carrol, B.D. (2007). Ανάλυση και Σχεδίαση Κυκλωμάτων Ψηφιακής Λογικής. Εκδόσεις Επίκεντρο.
- Ρουμελιώτης, Μ. & Σουραβλάς, Στ. (2013). Ψηφιακή Σχεδίαση: Αρχές & Εφαρμογές. Εκδόσεις Τζιόλα.
- Wakerly, J.F. (2005). Ψηφιακή Σχεδίαση: Αρχές & Πρακτικές (3η έκδοση). Εκδόσεις Κλειδάριθμος.
- Brown, S., Vranezic, Z. (2001). Σχεδίαση Ψηφιακών Συστημάτων με τη Γλώσσα VHDL. Εκδόσεις Τζιόλα.
- Μπισδούνης, Λ. (2015). Βασικές Εξειδικεύσεις σε Αρχιτεκτονική και Δίκτυα Υπολογιστών – Τόμος Α’. Εκδόσεις Ελληνικό Ανοικτό Πανεπιστήμιο.
- Δ/νση Β’θμιας Εκπ/σης Φλώρινας, Κέντρο ΠΛΗ.ΝΕ.Τ. Μαθηματική Λογική. (http://dide.flo.sch.gr/Exercises/Math-Logic.pdf)