4. ΝΟΜΟΙ ΔΥΑΔΙΚΗΣ ΑΛΓΕΒΡΑΣ

4.1 Βασικές έννοιες – Εισαγωγή

Η δυαδική άλγεβρα ή άλγεβρα Boole θεμελιώθηκε από τον Άγγλο μαθηματικό George Boole. Είναι μία "**Λογική Άλγεβρα**" για τη σχεδίαση κυκλωμάτων διακοπτών. Η άλγεβρα Boole έχει καθιερωθεί σαν την πλέον κατάλληλη λογική στις λειτουργίες των ψηφιακών συστημάτων. Οι βασικές αρχές της πραγματοποιούνται με τη χρήση των ηλεκτρονικών κυκλωμάτων, τις λεγόμενες "πύλες" (**gates**) που μελετήσαμε στο $3^{ο}$ κεφάλαιο, Οι λογικές μεταβλητές που χρησιμοποιούμε στις εφαρμογές μπορούν να πάρουν μόνο δυο τιμές το μηδέν-‘**0’** και το ένα-‘**1’** (το πεδίο ορισμού είναι το δίτιμο σύνολο {0,1} και το τελικό αποτέλεσμα σε μια πράξη θα είναι επίσης μόνο **0 ή 1**. Μια συνάρτηση άλγεβρας Boole είναι μια Λογική Συνάρτηση

4.2 Βασικοί Νόμοι της Άλγεβρας BOOLE.

Οι νόμοι της άλγεβρας Boole ικανοποιούν τις λογικές πράξεις πρόσθεσης, πολλαπλασιασμού και συμπληρώματος καθώς και την ιδιότητα των μεταβλητών να έχουν μόνο δύο τιμές το 0 και το 1. Οι βασικότεροι από τους νόμους είναι:

α/α	Νόμος		
1	**0 + X = X**	Πράξεις με το 0 και 1	
2	**1 + X = 1**	>>	
3	**X + X = X**	Πράξεις με τον εαυτό τους	
4	$X+\overline{X}=1$	>>	
5	**0 * X = 0**	Πράξεις με το 0 και 1	
6	**1 * X = X**	>>	
7	**X * X = X**	Πράξεις με τον εαυτό τους	
8	$X*\overline{X}=0$	>>	
9	$\overline{\overline{X}}=X$	Διπλό συμπλήρωμα	
10	**X + Y = Y + X**	Νόμοι αντιμετάθεσης	
11	**X * Y = Y * X**	>>	
12	**X + (Y + P) = (X + Y) + P**	Νόμοι προσεταιρισμού	
13	**X * (Y * P) = (X * Y) * P**	>>	
14	**X * (Y + P) = X * Y + X * P**	Νόμοι επιμερισμού	
15	**X + (Y * P) = (X+Y)*(X+P)**	>>	
16	**X * (X + Y) = X**	Νόμοι απορρόφησης	
17	**X+ X * Y = X**	>>	
18	$X+\overline{X}.Y$ **= X + Y**		
19	$\overline{X+Y}=\overline{X}*\overline{Y}$ **και**	Νόμοι De Morgan	
20	$\overline{X*Y}=\overline{X}+\overline{Y}$	>>	

Οι νόμοι (1-9) προκύπτουν από τις αντίστοιχες λογικές πράξεις (OR & AND).

Σημείωση: Όπου X,Y = οποιαδήποτε άλλη μεταβλητή χρησιμοποιούμε π.χ. A,B,C,D,E,F, …

Οι υπόλοιποι νόμοι (10-18) μπορούν να αποδειχτούν είτε με την εφαρμογή των προηγούμενων Νόμων είτε με τέλεια επαγωγή δίνοντας στις μεταβλητές που περιέχονται στο νόμο όλους τους δυνατούς συνδυασμούς τιμών τους και εφαρμόζοντας τις σχέσεις του νόμου μέχρι το τελικό αποτέλεσμα. Η απόδειξη γίνεται με το "**Πίνακα αληθείας**" (**Truth table**) αρκετά εύκολα.

4.3 Εφαρμογές - Ασκήσεις.

Παράδειγμα 1^{o}. Να αποδειχθεί ότι: $A+\overline{A}\,B=A+B$ (Νόμος18). Γράφουμε τον πίνακα και παρατηρούμε ότι ισχύει $A+\overline{A}\,B=A+B$ για κάθε συνδυασμό των A και B, άρα ισχύει και γενικά.

A	B	$\overline{A}$	$\overline{A}$.B	A+$\overline{A}$B	A+B
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

Με τους πίνακες αληθείας η απόδειξη είναι σχετικά εύκολη και οδηγεί στη λύση του προβλήματος.

<u>Η προηγούμενη απόδειξη μπορεί να γίνει με τη χρήση των σχέσεων της άλγεβρας Boole.</u>

Π.χ. 1. A+A*B=A*(1+B)=A*1=A ⇒ A+A*B=A

2. A+B)(A+C)=AA+AC+BA+BC=A+AC+AB+BC=(A+AB)+BC=A+BC⇒ (A+B)*(A+C)=A+B*C

3. $A+\overline{A}B=A*1+\overline{A}B=A(1+B)+\overline{A}B=A+AB+\overline{A}B=A+B(A+\overline{A})=A+B*1=A+B$.

Λύση των ασκήσεων 1γ, 1κ.

1γ. $\overline{A}.(B.\overline{C}+AB+\overline{A}.B)=\overline{A}.B.\overline{C}+\overline{A}.A.B+\overline{A}.\overline{A}.B=\overline{A}.B.\overline{C}+0+\overline{A}.B=\overline{A}.B(\overline{C}+1)=\overline{A}.B$

1κ. $A\overline{B}+A.B+\overline{A}.\overline{B}=A.(\overline{B}+B)+\overline{A}.\overline{B}=A.1+\overline{A}.\overline{B}=A+\overline{A}.\overline{B}=A+\overline{B}$

4.4 Νόμοι DE MORGAN

Οι νόμοι De Morgan για δύο μεταβλητές $\overline{(A.B)}=\overline{A}+\overline{B}$ **&** $\overline{(A+B)}=\overline{A}.\overline{B}$ επεκτείνονται και για περισσότερες μεταβλητές. Είναι βασικοί νόμοι για την απλοποίηση των λογικών συναρτήσεων και η απόδειξη γίνεται με τους πίνακες αληθείας (Truth table).

4.5 Δυϊκότητα της άλγεβρας Boole (Duality)

Σε κάθε σχέση της άλγεβρας Boole οι πράξεις "**+**" και "**.**" καθώς και τα ουδέτερα στοιχεία "**0**" & "**1**" μπορούν να αντιμετατεθούν χωρίς να αλλάξει το αποτέλεσμα.

π.χ A.(B+C)= A.B+A.C <=> A+B.C=(A+B).(A+C)

Έτσι για να αποδείξουμε μία σχέση άλγεβρας Boole είναι αρκετό να αποδείξουμε τη δυϊκή της. Π.χ. Για να αποδείξουμε ότι $A(\overline{A}+B)=AB$ κάνουμε τις πράξεις στο 1^{o} μέρος $A.\overline{A}+A.B$ ή $A.B$ αφού $A.\overline{A}=0$ οπότε έχουμε το 2^{o} μέρος.

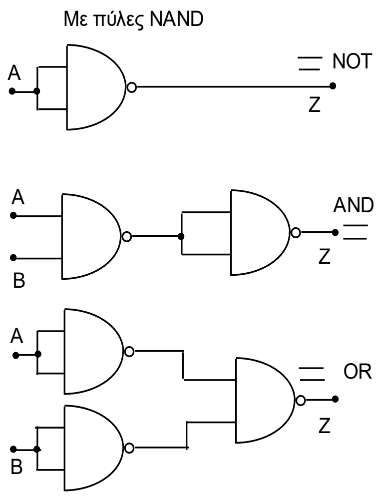
4.6. Ισοδύναμα κυκλώματα

4.6.1 Εισαγωγή

Είναι πολύ μεγάλη η σπουδαιότητα των πυλών NAND και NOR, αφού **μόνο** με αυτές μπορούμε να σχεδιάσουμε οποιοδήποτε κύκλωμα, αρκεί να αντικαταστήσουμε οπωσδήποτε όλες τις πύλες AND, OR, NOT μόνο με NAND ή μόνο με NOR. Δηλαδή να αντικαταστήσουμε τρία διαφορετικά IC που πρέπει να χρησιμοποιήσουμε σε ένα απλό κύκλωμα που απαιτεί έστω και μια πύλη μόνο πύλη από τις AND, OR, NOT, βάζοντας στην θέση τους ένα μόνο IC (NAND ή NOR) και χρησιμοποιώντας όλες τις πύλες του στη θέση των αντίστοιχων απλών πυλών.

4.6.2. Ισοδύναμο κύκλωμα με πύλες NAND

Τα ισοδύναμα κυκλώματα των πυλών NOT, AND, OR σχεδιασμένα μόνο με πύλες NAND και η απόδειξη της ισοδυναμίας φαίνονται παρακάτω:



Απόδειξη

$$Z = \overline{A.A} = \overline{A}$$

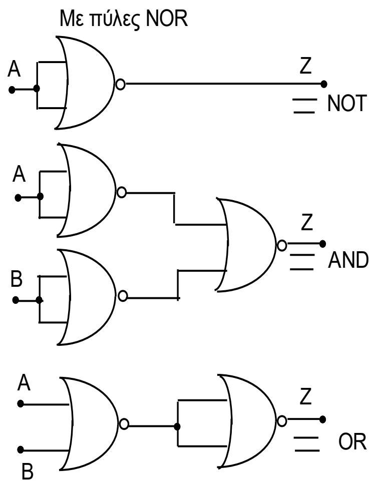
$$Z = \overline{A+A} = \overline{A}$$

$$Z = \overline{\overline{A.B}} = A.B$$

$$Z = \overline{\overline{A}+\overline{B}} = A.B$$

$$Z = \overline{\overline{A}.\overline{B}} = A+B$$

$$Z = \overline{\overline{A+B}} = A+B$$



4.6.3. Ισοδύναμο κύκλωμα με NOR

Τα ισοδύναμα κυκλώματα των πυλών NOT, OR, AND σχεδιασμένα μόνο με πύλες NOR και η απόδειξη της ισοδυναμίας φαίνονται παραπάνω.

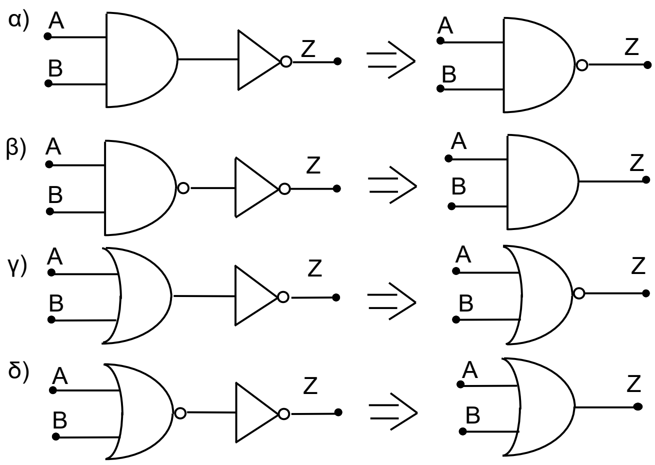
4.7. Αντικατάσταση πυλών με ισοδύναμα κυκλώματα

4.7.1. Εισαγωγή

Μερικές αντικαταστάσεις κυκλωμάτων περιγράφονται παρακάτω με την βοήθεια των οποίων μπορούμε να πάρουμε κυκλώματα ισοδύναμα με κάποια από τις πύλες. Πρακτικά χρησιμοποιούμε ένα είδος πύλης για τα κυκλώματά μας, όμως σε κάποιες περιπτώσεις πιθανόν να χρειαστούμε κάποιες πύλες που δεν έχουμε ενώ διαθέτουμε κάποιες άλλες, τότε οι συνδυασμοί αυτοί είναι χρήσιμοι.

4.7.2. Αντιστροφή των εξόδων των πυλών

Αν αντιστρέψουμε την έξοδο μιας πύλης AND παίρνουμε μια πύλη NAND, ενώ αν αντιστρέψουμε την έξοδο μιας πύλης NAND θα πάρουμε μια πύλη AND κ.ο.κ.
Ας δούμε τα κυκλώματα και τις αποδείξεις.



1ο κύκλωμα | 2ο κύκλωμα

$A.B = \overline{A.B}$ ≈ NAND

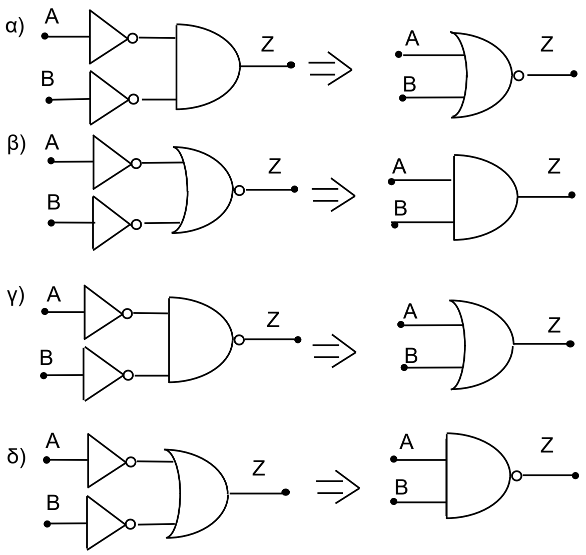
$\overline{AB} = \overline{\overline{A.B}} = A.B$ ≈ AND

$A + B = \overline{A + B}$ ≈ NOR

$\overline{A + B} = \overline{\overline{A + B}} = A + B$ ≈ OR

4.7.3. Αντιστροφή των εισόδων των πυλών

Αν αντιστρέψουμε της εισόδους μιας πύλης AND παίρνουμε μια πύλη NOR, ενώ αν αντιστρέψουμε τις εισόδους μιας πύλης NOR θα πάρουμε μια πύλη AND κ.ο.κ.
Ας δούμε τα κυκλώματα και τις αποδείξεις.



1ο κύκλωμα | 2ο κύκλωμα

$\overline{A}.\overline{B}$ ≈ $\overline{A + B} = \overline{A}.\overline{B}$ NOR

$\overline{\overline{A} + \overline{B}} = \overline{\overline{A}}.\overline{\overline{B}} = A.B$ ≈ AND

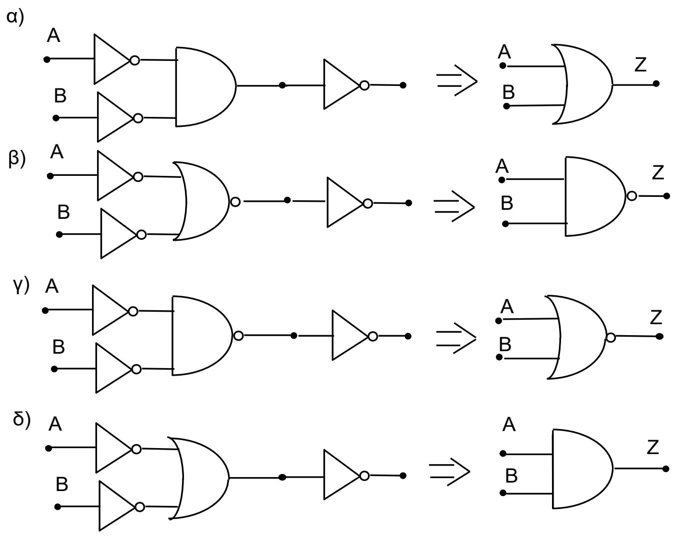
$\overline{\overline{A}.\overline{B}} = \overline{\overline{A}} + \overline{\overline{B}} = A + B$ ≈ OR

$\overline{A} + \overline{B}$ ≈ $\overline{A.B} = \overline{A}. + \overline{B}$ NAND

4.7.4. Αντιστροφή των εισόδων και των εξόδων των πυλών

Αν τώρα αντιστρέψουμε της εισόδους αλλά και την έξοδο μιας πύλης AND παίρνουμε μια πύλη OR, ενώ αν αντιστρέψουμε τις εισόδους και τη έξοδο μιας πύλης OR θα πάρουμε μια πύλη AND κ.ο.κ.

Ας δούμε τα κυκλώματα και τις αποδείξεις.



1ο κύκλωμα 2ο κύκλωμα

$\overline{A}.\overline{B} = \overline{\overline{A}}.\overline{\overline{B}} = \overline{\overline{A}} + \overline{\overline{B}} = A + B$ ≈ OR

$\overline{\overline{A} + \overline{B}} = \overline{\overline{\overline{A} + \overline{B}}} = \overline{A}. + \overline{B}$ ≈ NAND

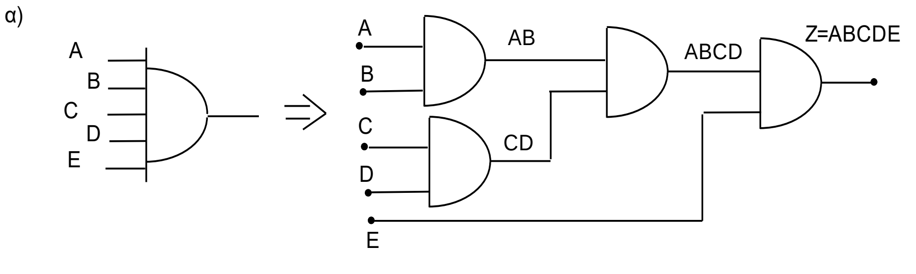
$\overline{\overline{A}}.\overline{\overline{B}} = \overline{\overline{\overline{A}}}.\overline{\overline{\overline{B}}} = \overline{A}.\overline{B} \approx \overline{A + B} = \overline{A}.\overline{B}$ NOR

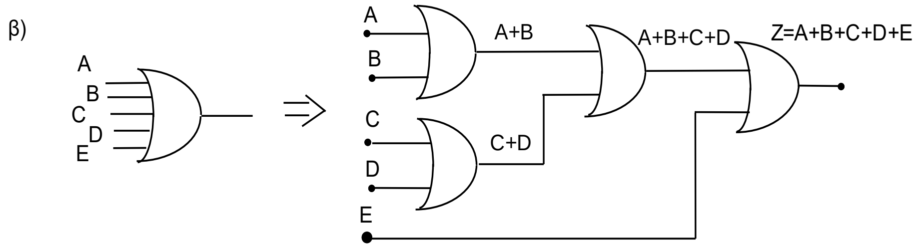
$\overline{\overline{A} + \overline{B}} = \overline{\overline{A}}.\overline{\overline{B}} = A.B$ ≈ AND

4.7.5 Αντικατάσταση πύλης πολλών εισόδων.

Η αντικατάσταση μιας πύλης πολλών εισόδων που προκύπτει από την θεωρητική μελέτη κάποιου προβλήματος με άλλες πύλες ίδιου τύπου αλλά με λιγότερες εισόδους επειδή είτε δεν υπάρχουν στο Εμπόριο οι συγκεκριμένες πύλες είτε δεν είναι διαθέσιμες σε αρκετή ποσότητα κ.λ.π. είναι μια πρακτική διαδικασία. Με την χρήση της ιδιότητας του προσεταιρισμού (στο άθροισμα και το γινόμενο) μπορούμε να αντικαταστήσουμε μια πύλη OR ή AND πολλών εισόδων με πύλες OR ή AND με λιγότερες εισόδους π.χ. μια πύλη AND έξι εισόδων να αντικατασταθεί από πύλες AND δυο εισόδων. (**Προσοχή δεν ισχύει άμεσα στις πύλες NAND,NOR γιατί είναι σύνθετες πύλες**)

Μερικά τέτοια κυκλώματα και οι αντίστοιχες αποδείξεις φαίνονται στα σχήματα α & β.





4.8 Κανονική μορφή μιας λογικής συνάρτησης

Σε μία λογική συνάρτηση Z=f(A,B,C,...) στην άλγεβρα Boole υπάρχουν εκφράσεις που είναι βολικές για τη κατασκευή του αντίστοιχου κυκλώματος που πραγματοποιεί τη συνάρτηση. Τέτοιες εκφράσεις ή μορφές είναι το «**άθροισμα γινομένων**» ή Ελάχιστοι όροι (minterms, Ελαχιστόροι) που πραγματοποιούνται με συνδυασμούς πυλών AND-OR και το «**γινόμενο αθροισμάτων**» ή Μέγιστοι όροι (Maxterms, Μεγιστόροι) Mi που πραγματοποιούνται με συνδυασμούς OR-AND.

Οι ελαχιστόροι είναι το γινόμενο όλων των μεταβλητών σε κανονική ή συμπληρωματική μορφή. Οι μεγιστόροι είναι το συμπλήρωμα των αντίστοιχων ελαχιστόρων με τον ίδιο δείκτη.

Στον πίνακα αληθείας φαίνονται οι ελάχιστοι και οι μέγιστοι όροι μιας συνάρτησης δυο μεταβλητών της Z=f(A,B).

Σημείωση: Ο πίνακας αληθείας είναι μοναδικός ενώ η μορφή της συνάρτησης δεν είναι.

α/α	Μεταβλητές A B	Ελάχιστοι Όροι mi	Μέγιστοι Όροι Mi
0	$\overline{A}$ $\overline{B}$	$\overline{A}*\overline{B} = m_0$	$A+B = M_0$
1	$\overline{A}$ B	$\overline{A}*B = m_1$	$A+\overline{B} = M_1$
2	A $\overline{B}$	$A*\overline{B} = m_2$	$\overline{A}+B = M_2$
3	A B	$A*B = m_3$	$\overline{A}+\overline{B} = M_3$

Αν μία συνάρτηση είναι εκφρασμένη σε μία από τις δύο αυτές μορφές λέμε ότι βρίσκεται σε **"Κανονική Μορφή"**.

Εδώ ισχύουν τα παρακάτω θεωρήματα:

Θ1. Για **n** μεταβλητές της Boole υπάρχουν ακριβώς **2^n min** & **2^n Max** όροι.

Θ2. Κάθε ελάχιστος **min** όρος είναι το συμπλήρωμα ενός μέγιστου **Max** όρου και αντίστροφα. Δηλ. $\overline{m_i} = M_i$ και $\overline{M_i} = m_i$.

Θ3. Το λογικό άθροισμα όλων των **min** όρων μιας λογικής συνάρτησης είναι ίσο με ένα-'**1**' $\sum_{i=0}^{2^n-1}(m_i) = 1$ όπου : **n**=αριθμός μεταβλητών.

Θ4. Το λογικό γινόμενο όλων των **Max** όρων μιας λογικής συνάρτησης είναι ίσο με μηδέν-'**0**' $\prod_{i=0}^{2^n-1}(M_i) = 0$ όπου : **n**=αριθμός μεταβλητών.

Αν τώρα οι τιμές της συνάρτησης είναι γνωστές τότε είναι εύκολο να εκφράσουμε τη συνάρτηση στην "ΚΑΝΟΝΙΚΗ" της μορφή σε μια από τις δυο εκφράσεις.

Π.χ: Έστω η $Z_{(A,B,C)}$ με τιμές που δίνονται στο πίνακα που ακολουθεί.

Μεταβλητές A B C	Ελάχιστοι Όροι mi	Μέγιστοι Όροι Mi	Z
$\overline{A}\ \overline{B}\ \overline{C}$	$\overline{A}*\overline{B}*\overline{C}$	$A+B+C$	0
$\overline{A}\ \overline{B}\ C$	$\overline{A}*\overline{B}*C$	$A+B+\overline{C}$	0
$\overline{A}\ B\ \overline{C}$	$\overline{A}*B*\overline{C}$	$A+\overline{B}+C$	1
$\overline{A}\ B\ C$	$\overline{A}*B*C$	$A+\overline{B}+\overline{C}$	1
$A\ \overline{B}\ \overline{C}$	$A*\overline{B}*\overline{C}$	$\overline{A}+B+C$	0
$A\ \overline{B}\ C$	$A*\overline{B}*C$	$\overline{A}+B+\overline{C}$	0
$A\ B\ \overline{C}$	$A*B*\overline{C}$	$\overline{A}+\overline{B}+C$	1
$A\ B\ C$	$A*B*C$	$\overline{A}+\overline{B}+\overline{C}$	0

Από τον πίνακα παρατηρούμε ότι για κάθε συνδυασμό των τιμών των μεταβλητών A,B,C σχηματίζουμε ένα "**Ελάχιστο**" όρο και ένα "**Μέγιστο**" όρο.

Ο ελάχιστος όρος (Ελαχιστόρος) αποτελείται από το γινόμενο των όρων σε κανονική και συμπληρωματική μορφή όταν η αντίστοιχη τιμή της συνάρτησης είναι ένα-"1", π.χ. για A=0, B=1, C=1 ο Min όρος είναι $\overline{A}BC$.

Ο μέγιστος όρος (Μεγιστόρος) προκύπτει από την θεώρηση του συμπληρώματος των αντίστοιχων ελάχιστων όρων για τις τιμές τις συνάρτησης που είναι μηδέν-"0".

Π.χ. Για τον ελάχιστο όρο $\overline{A}BC$ αντιστοιχεί ο μέγιστος όρος $\overline{\overline{A}BC}=\overline{\overline{A}}+\overline{B}+\overline{C}=A+\overline{B}+\overline{C}$

Επομένως :

α) Η **ΚΑΝΟΝΙΚΗ** μορφή της συνάρτησης με **άθροισμα γινομένων** (**Α.Γ**) προκύπτει αν αθροίσουμε τους ελάχιστους όρους που αντιστοιχούν στις μονάδες της συνάρτησης. Π.χ. Για τη Z έχουμε από το πίνακα $Z_{AEO}=\overline{A}.B\ \overline{C}+\overline{A}.BC+AB\ \overline{C}$

β) Η **ΚΑΝΟΝΙΚΗ** μορφή της συνάρτησης με **γινόμενο αθροισμάτων** (**Γ.Α**) προκύπτει αν πολλαπλασιάσουμε τους μέγιστους όρους που αντιστοιχούν στα μηδενικά της συνάρτησης. Π.χ. Για τη Z(A,B,C) έχουμε από το πίνακα

$$Z_{ΓΜΟ}=(A+B+C)(A+B+\overline{C})(\overline{A}+B+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+\overline{C})$$

Η κανονική μορφή, μιας συνάρτησης, **<u>δεν είναι</u>** <u>και η</u> **<u>πιο απλή</u>** <u>μορφή</u> της. Υπάρχει απλούστερη που προκύπτει από τη σχετική απλοποίηση

Π.χ. Για τις δύο εκφράσεις των Min & Max όρων έχουμε

α) $Z=\overline{A}B\overline{C}+\overline{A}BC+AB\overline{C}=\overline{A}B(\overline{C}+C)+AB\overline{C}=\overline{A}B+AB\overline{C}=B(\overline{A}+A\overline{C})=B(\overline{A}+\overline{C})=\overline{A}B+B\overline{C}$

Σημείωση: Ένας άλλος τρόπος για να εκφράσουμε τη συνάρτηση Z=f(A,B,C) από το πίνακα, σαν άθροισμα των ελαχίστων όρων της (ΑΓ) είναι να χρησιμοποιήσουμε τη σχέση : **$Z_{(A,B,C)}$=** $\sum_{i=0}^{2^n-1}(a_i.m_i)=1$ (1.4) όπου: **n** = αριθ. Μεταβλητών και

a_i=συντελεστές με τιμή που αντιστοιχεί στο ελάχιστο όρο, **m_i** του Π.Α.

ή σαν γινόμενο μέγιστων όρων από τη σχέση:

$Z_{(A,B,C)}$= $\prod_{i=0}^{2^n-1}(a_i.M_i)=0$ (1.5). (ίδιο νόημα για τα **n, a_i)**

Παράδειγμα: Για το πίνακα αληθείας του XOR και τη σχέση (1.4)

$Z_{(A,B)}=a_1*m_1+a_2*m_2+a_3*m_3+a_4*m_4=0*\overline{A}.\overline{B}+1.\overline{A}.B+1.A\overline{B}+0.A.B=\overline{A}.B+A.\overline{B}$ άρα η συνάρτηση του αποκλειστικού-OR=XOR είναι: $Z(A,B)=A\oplus B=\overline{A}B+A\overline{B}$

Αν χρησιμοποιήσουμε τη σχέση (1.5) έχουμε: $Z_{(A,B)}=(a_1+M_1)(a_2+M_2)(a_3+M_3)(a_4+M_4)=$

$=[0+(A+B)][1+(A+\overline{B})][1+(\overline{A}+B)][0+(\overline{A}+\overline{B})]=(A+B)*(1)*(1)*(\overline{A}+\overline{B})=(A+B)*(\overline{A}+\overline{B})$

που είναι ισοδύναμη αυτής που βρήκαμε από τη σχέση (1.4), αν κάνουμε τις πράξεις

δηλαδή $\underbrace{A\overline{A}}_{\mathbf{0}}+A\overline{B}+\overline{A}B+\underbrace{B.\overline{B}}_{\mathbf{0}}=A\overline{B}+\overline{A}B$

γ) Ένας άλλος ορισμός για τη κανονική μορφή μιας συνάρτησης είναι ο παρακάτω: **<u>ΚΑΝΟΝΙΚΗ</u>** λέμε τη συνάρτηση στην οποία οι μεταβλητές **<u>ή</u>** το *συμπλήρωμα* τους εμφανίζεται **μόνο μία φορά** σε **κάθε όρο** της στη *<u>μορφή αθροίσματος</u>* <u>των</u> *<u>ελαχίστων όρων</u>* **ή** το *<u>γινόμενο μέγιστων όρων.</u>*

Π.χ. **α) Στην μορφή ΑΕΟ (ή Α.Γ).** Για το XOR η μορφή της $Z_{(A,B)}=\overline{A}B+A\overline{B}$ είναι κανονική ενώ η F(A,B,C)=A*C+B δεν είναι κανονική. Για να τη φέρουμε στη κανονική μορφή πολλαπλασιάζουμε κάθε όρο με τη μεταβλητή **X** που λείπει από αυτόν, με τη μορφή $(X+\overline{X})$, η οποία από τις σχέσεις Boole είναι ένα. Δηλαδή:

$Z_{(A,B,C)}=AC+B=AC(B+\overline{B})+B(A+\overline{A})(C+\overline{C})=ABC+A\overline{B}C+AB\overline{C}+\overline{A}BC+\overline{A}B\overline{C}+ABC$

β) Στην μορφή ΓΜΟ (ή Γ.Α). Η μορφή της $Z_{(A,B)}=(\overline{A}+B)(A+\overline{B})$ είναι κανονική ενώ η $Z_{(A,B,C)}=(\overline{A}+B)(B+\overline{C})$ δεν είναι κανονική. Για να τη φέρουμε στη κανονική μορφή προσθέτουμε σε κάθε όρο με τη μεταβλητή **X** που λείπει από αυτόν, με τη μορφή $(X\overline{X})$, η οποία από τις σχέσεις Boole είναι μηδέν.

Δηλαδή: $Z=(\overline{A}+\overline{B})(B+\overline{C})=(\overline{A}+\overline{B}+C\overline{C})(A\overline{A}+B+\overline{C})=(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C})(A+B+\overline{C})(\overline{A}+B+\overline{C})$

(Στις πράξεις κάναμε χρήση του νόμου 15 της άλγεβρας Boole).

4.9. Έκφραση Λογικής Συνάρτησης με τα βάρη της.

Μια Λογική Συνάρτηση παριστάνεται, σε πιο απλή μορφή, με τα "**βάρη**" της εφόσον κάθε όρο της τον αντικαταστήσουμε με το δεκαδικό ισοδύναμό του, αν για κάθε μεταβλητή, με την σειρά που γράφεται, θέσουμε ένα βάρος σύμφωνα με τον βασικό κώδικα BCD-8421.

Π.χ. **α) Στην μορφή Α.Γ** Για την αντίστοιχη συνάρτηση της α περίπτωσης τα βάρη είναι: $A=2^2=4$, $B=2^1=2$, $C=2^0=1$ και η έκφραση της λογικής συνάρτησης είναι:

$$
\begin{aligned}
Z_{(A,B,C)} &= ABC + A\overline{B}C + AB\overline{C} + \overline{A}BC + \overline{A}B\overline{C} \\
&= 111 + 101 + 110 + 011 + 010 \\
&= 4+2+1 \quad 4+0+1 \quad 4+2+0 \quad 0+2+1 \quad 0+2+0 \\
&= 7 \quad 5 \quad 6 \quad 3 \quad 2 \\
&= m_7 \quad m_5 \quad m_6 \quad m_3 \quad m_2
\end{aligned}
$$

ή $\mathbf{Z}_{(A,B,C)} = \Sigma(\mathbf{2,3,5,6,7})$.

Π.χ. **β) Στην μορφή Γ.Α** Επίσης για τη προηγούμενη συνάρτηση της β περίπτωσης τα βάρη είναι: $A=2^2=4$, $B=2^1=2$, $C=2^0=1$ και η έκφραση της λογικής συνάρτησης είναι:

$$
\begin{aligned}
Z_{(A,B,C)} &= (\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C})(A+B+\overline{C})(\overline{A}+B+\overline{C}) \\
&= 1\ 1\ 0\ \ 1\ 1\ 1\ \ 0\ 0\ 1\ \ 1\ 0\ 1 \\
&= 6 \quad 7 \quad 1 \quad 5 \\
&= M_6 \quad M_7 \quad M_1 \quad M_5
\end{aligned}
$$

ή $\mathbf{Z}_{(A,B,C)} = \Pi(\mathbf{1,5,6,7})$ ή $\overline{Z} = \Sigma(1,5,6,7)$ το οποίο προκύπτει από το συμπλήρωμα των όρων.

Δηλαδή $\overline{Z} = \Sigma(1,5,6,7)$ ή ($\overline{(m_1 + m_5 + m_6 + m_7)} = M1 + M_5 + M_6 + M_7$ $=\Pi(\mathbf{1,5,6,7})$

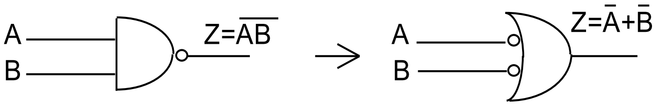
4.10 Τεχνικές Σχεδίασης Λογικών Κυκλωμάτων

4.10.1 Εισαγωγή

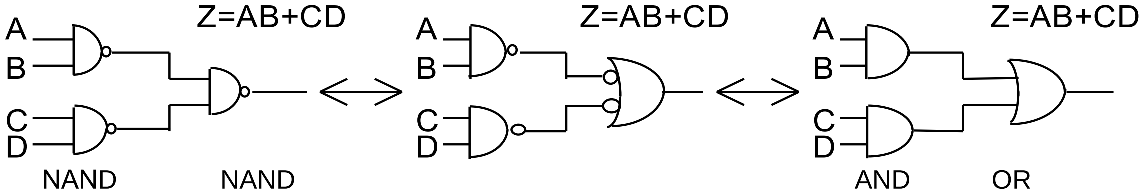
Η σχεδίαση κυκλωμάτων με την χρήση ενός είδους πύλης, δηλαδή μόνο πύλες NOR ή μόνο πύλες NAND είναι μια συνήθης πρακτική. Στο εμπόριο υπάρχουν συνήθως αυτοτελή τέτοια λογικά κυκλώματα τα οποία παρέχουν μια πολύ καλή ενίσχυση, οπότε προκύπτει η ανάγκη χρησιμοποίησής τους στην σχεδίαση κυκλωμάτων με μια **τεχνική** (ή λογική) όπως λέγεται σχεδίασης.

Η πύλη **NAND** (NOT-AND) μπορεί να αντικαταστήσει οποιαδήποτε από τις πύλες AND, OR, NOT, όπως είδαμε στη σχετική παράγραφο. Εδώ θα δούμε τώρα πως θα σχεδιάσουμε ένα κύκλωμα δυο επιπέδων μόνο με πύλες NAND.

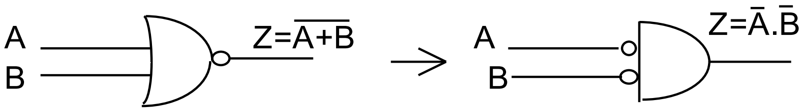
Η πύλη NAND έχει ένα ισοδύναμο σύμβολο από το νόμο De Morgan $\overline{AB} = \overline{A} + \overline{B}$



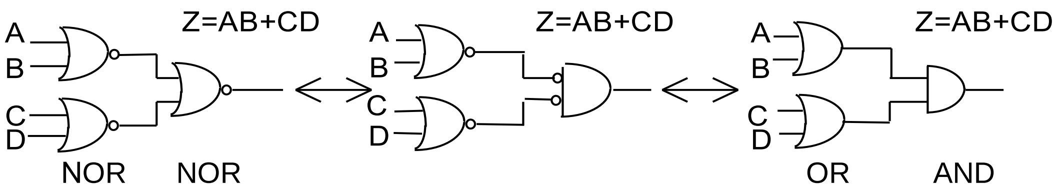
Οπότε από το <u>ισοδύναμο</u> σύμβολο προκύπτει ότι ένα κύκλωμα δυο επιπέδων NAND-NAND, ισοδυναμεί με ένα κύκλωμα δυο επιπέδων AND-OR.



Το κύκλωμα AND-OR, που προέκυψε, πραγματοποιεί εκφράσεις που αντιστοιχούν σε **αθροίσματα γινομένων.** Παρόμοια με την προηγούμενη διαδικασία για την πύλη NAND και η πύλη NOR έχει ένα συναρτησιακά ισοδύναμο κύκλωμα που δίδεται από τον De Morgan $\overline{A+B} = \overline{A}.\overline{B}$ με σύμβολο



Οπότε από το <u>ισοδύναμο</u> σύμβολο προκύπτει ότι ένα κύκλωμα δυο επιπέδων NOR-NOR, ισοδυναμεί με ένα κύκλωμα δυο επιπέδων OR-AND.



Το κύκλωμα OR-AND, που προέκυψε, πραγματοποιεί εκφράσεις που αντιστοιχούν σε **γινόμενα αθροισμάτων.**

Οι δυο τεχνικές σχεδίασης ταιριάζουν με τις δυο κατηγορίες πυλών γενικής χρήσης τις πύλες NAND και τις πύλες NOR **α) Με λογική NAND (μόνο** πύλες **NAND)** ή

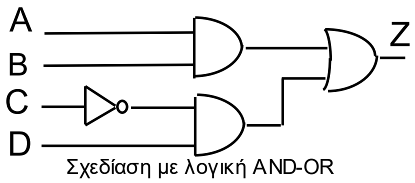
β) Με λογική NOR (μόνο πύλες **NOR).**

4.10.2 Τεχνική (ή λογική) σχεδίασης NAND

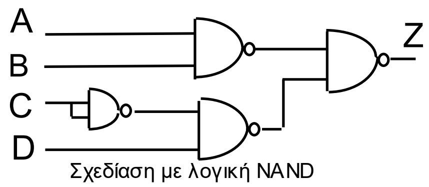
Για την σχεδίαση των κυκλωμάτων που η συνάρτησή τους είναι γραμμένη στη μορφή Α.Γ χρησιμοποιούμε μόνο πύλες NAND σε όλα τα επίπεδα σχεδίασης.

Π.χ. Δίδεται η λογική συνάρτηση $Z=AB+\bar{C}D$.

Το λογικό κύκλωμα σχεδιασμένο με πύλες NOT, AND, OR φαίνεται στο σχήμα.



Σχεδίαση με λογική AND-OR



Σχεδίαση με λογική NAND

Για την σχεδίαση με πύλες NAND μετατρέπουμε την συνάρτηση με την βοήθεια του θεωρήματος De Morgan οπότε αυτή γράφεται $Z=(\overline{\overline{Z}})=\overline{\overline{(AB+\bar{C}D)}}=\overline{(\overline{AB})(\overline{\bar{C}D})}$

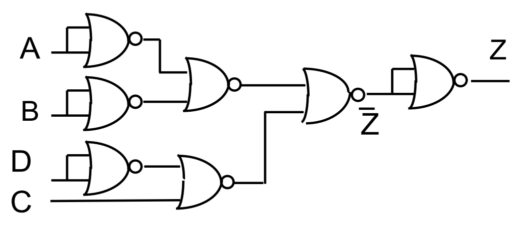
Από την γραφή της βλέπουμε ότι απαιτούνται δυο πύλες NAND δυο εισόδων, μια για κάθε έκφραση από τις $\overline{(AB)}$ και $\overline{(\bar{C}D)}$, μια πύλη NAND για την αντιστροφή του C και μια πύλη NAND για το συμπλήρωμα της συνάρτησης $(\overline{....})$.

Όλη η παραπάνω διαδικασία **δεν είναι απαραίτητη** και το κύκλωμα σχεδιάζεται **αμέσως** πολύ **απλά** αν θεωρήσουμε για κάθε **ένα όρο** της συνάρτησης μια πύλη NAND, με αριθμό εισόδων όσο το πλήθος των μεταβλητών του όρου (ακόμα και για την περίπτωση μιας μεταβλητής μόνο - δηλ. αντιστροφέας) και μια πύλη NAND για το τελικό άθροισμα.

Η έκφραση που προκύπτει σχεδιάζεται με μόνο με πύλες NAND ή όπως λέμε με την τεχνική σχεδίασης NAND. (δείτε το σχήμα παραπάνω)

Σημείωση: Αν υπάρχει ανάγκη (έλειψη πυλών κ.λ.π) την παραπάνω συνάρτηση μπορούμε να την σχεδιάσουμε μόνο με πύλες NOR αν συνεχίσουμε να εφαρμόζουμε τον κανόνα De Morgan οπότε έχουμε $Z=(\overline{\overline{Z}})=\overline{\overline{(AB+\bar{C}D)}}=\overline{(\bar{A}+\bar{B})(C+\bar{D})}$ και καταλήγουμε στη μορφή Γ.Α οπότε απαιτούνται δυο πύλες NOR, των δυο εισόδων κάθε μιά, μια για κάθε έκφραση από τις $\overline{(\bar{A}+\bar{B})}$, $\overline{(C+\bar{D})}$, τρεις πύλες NOR για την αντιστροφή των A,B,D και μια πύλη NOR για το συμπλήρωμα της συνάρτησης $(\overline{....})$.

Η έκφραση που προκύπτει σχεδιάζεται με πύλες NOR

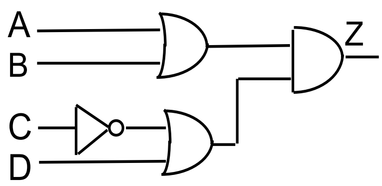


4.10.2 Τεχνική (ή λογική) σχεδίασης NOR

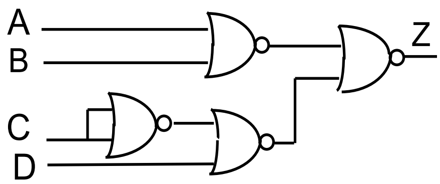
Για την σχεδίαση των κυκλωμάτων που η συνάρτησή τους είναι γραμμένη στη μορφή Γ.Α χρησιμοποιούμε μόνο πύλες NOR σε όλα τα επίπεδα σχεδίασης.

Π.χ. Δίδεται η λογική συνάρτηση $Z=(A+B)(\overline{C}+D)$.

Το λογικό κύκλωμα σχεδιασμένο με πύλες NOT, OR, AND φαίνεται στο σχήμα.



Σχεδίαση με λογική OR-AND



Σχεδίαση με λογική NOR

Για την σχεδίαση με πύλες NOR μετατρέπουμε την συνάρτηση με την βοήθεια του θεωρήματος De Morgan οπότε αυτή γράφεται $Z=(\overline{\overline{Z}})=\overline{\overline{(A+B)(\overline{C}+D)}}=\overline{\overline{(A+B)}\,\overline{(\overline{\overline{C}}+D)}}$

Από την γραφή της βλέπουμε ότι απαιτούνται δυο πύλες NOR δυο εισόδων, μια για κάθε έκφραση από τις $\overline{(A+B)}$ και $\overline{(\overline{C}+D)}$, μια πύλη NOR για την αντιστροφή του C και μια πύλη NOR για το συμπλήρωμα της συνάρτησης $\overline{(....)}$.

Όλη η παραπάνω διαδικασία **δεν είναι απαραίτητη** και το κύκλωμα σχεδιάζεται **αμέσως** πολύ **απλά** αν θεωρήσουμε για κάθε **ένα όρο** της συνάρτησης μια πύλη NOR με αριθμό εισόδων όσο το πλήθος των μεταβλητών του όρου (ακόμα και για την περίπτωση μιας μεταβλητής μόνο - δηλ. αντιστροφέας) και μια πύλη NOR για το τελικό άθροισμα.

Η έκφραση που προκύπτει σχεδιάζεται με μόνο με πύλες NOR ή όπως λέμε με την τεχνική σχεδίασης NOR. (δείτε το σχήμα παραπάνω)

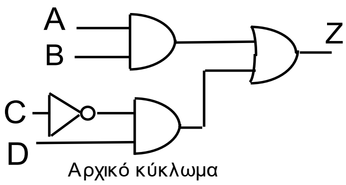
4.10.3 Τεχνική (ή λογική) σχεδίασης ήδη σχεδιασμένων κυκλωμάτων

Εάν έχουμε σχεδιασμένο ένα κύκλωμα με λογική AND-OR ή OR-AND αλλά δεν είναι διαθέσιμες οι πύλες που είναι σημειωμένες μπορούμε να αλλάξουμε την σχεδίασή του με την βοήθεια των παραπάνω τεχνικών σχεδίασης.

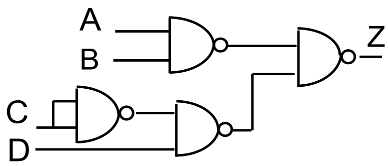
Αν το σχεδιασμένο κύκλωμα είναι της **λογικής** AND-OR τότε μπορούμε για να σχεδιάσουμε το νέο κύκλωμα με λογική **NAND** αν **αντικαταστήσουμε** όλες τις απλές πύλες NOT,AND,OR, του κυκλώματος με πύλες NAND.

Π.χ. Να επανασχεδιαστεί το λογικό κύκλωμα του σχήματος.

Παρατηρούμε ότι είναι σχεδιασμένο σε τρία επίπεδα σχεδίασης λογικής AND-OR οπότε αντικαθιστούμε όλες τις πύλες του με πύλες NAND.



Αρχικό κύκλωμα

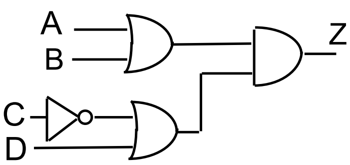


Σχεδίαση με λογική NAND

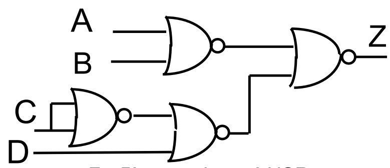
Αν το σχεδιασμένο κύκλωμα είναι της **λογικής** OR-AND τότε μπορούμε για να σχεδιάσουμε το νέο κύκλωμα με λογική **NOR** αν **αντικαταστήσουμε** όλες τις απλές πύλες NOT,AND,OR, του κυκλώματος με πύλες NOR.

Π.χ. Να επανασχεδιαστεί το λογικό κύκλωμα του σχήματος.

Παρατηρούμε ότι είναι σχεδιασμένο σε τρία επίπεδα σχεδίασης λογικής OR-AND οπότε αντικαθιστούμε όλες τις πύλες του με πύλες NOR



Αρχικό κύκλωμα



Σχεδίαση με λογική NOR

Αν το σχεδιασμένο κύκλωμα δεν είναι της **λογικής** AND-OR ή της **λογικής** OR-AND τότε **δεν μπορούμε** να το επανασχεδιάσουμε **αμέσως**.

Πρέπει να εξαγάγουμε τη λογική συνάρτηση που υλοποιεί να την απλοποιήσουμε οπότε να προκύψει μια από τις δυο μορφές της Α.Γ ή Γ.Α και στη συνέχεια να σχεδιάσουμε το νέο κύκλωμα με λογική **NAND** ή **NOR** κατά τα γνωστά.

4.11 Σχεδίαση και Ανάλυση Συνδυαστικών Κυκλωμάτων

4.11.1 Σχεδίαση Συνδυαστικών Κυκλωμάτων

Σε πολλές εφαρμογές δεν είναι γνωστή η λογική συνάρτηση και πρέπει να την εκφράσουμε από τα δεδομένα του προβλήματος. Η διαδικασία για την εξαγωγή της λογικής συνάρτησης είναι:

1) Πίνακα καταστάσεων (Π.Κ). Γράφουμε σε αυτόν την έξοδο του κυκλώματος για όλους τους συνδυασμούς του πλήθους των μεταβλητών της εισόδου.

2) Εξαγωγή της λογικής συνάρτησης από τον Π.Κ. Εξάγουμε τη λογική συνάρτηση είτα στη μορφή Α.Γ είτε στη μορφή Γ.Α .

3) Απλοποίηση της λογικής συνάρτησης (Α.Λ..Σ). Απλοποιούμε την εξαχθείσα λογική συνάρτηση με τη βοήθεια των κανόνων της άλγεβρας Boole.

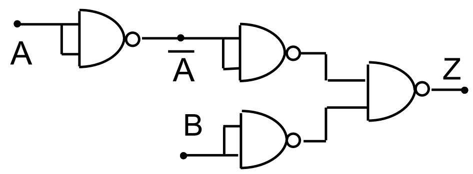
4) Σχεδίαση της του λογικού κυκλώματος της Α.Λ.Σ. Σχεδιάζουμε την απλοποιημένη έκφραση τις λογικής συνάρτησης με μια από τις τεχνικές (λογικές) σχεδίασης

1ο Παράδειγμα: Να σχεδιαστεί το λογικό κύκλωμα του οποίου η έξοδος είναι ένα στις περιπτώσεις που φαίνονται στον Π.Κ με λογική σχεδίασης NAND.

Λύση: α. Από το Π.Κ γράφουμε τη λογική συνάρτηση στη μορφή Α.Γ δηλαδή για τις μονάδες

A	B	Z
0	0	**1**
0	1	**1**
1	0	**0**
1	1	**1**

του πίνακα οπότε έχουμε $Z = \overline{A}.\overline{B} + \overline{A}B + A.B$



β. Απλοποιούμε τη συνάρτηση οπότε έχουμε με τη βοήθεια της Boole και έχουμε : $Z = \overline{A}.(\overline{B} + B) + A.B = \overline{A} + A.B = \overline{A} + B$ ή τελικά $Z = \overline{A} + B$

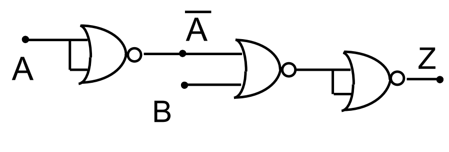
γ. Σχεδιάζουμε την τελική έκφραση ή αλλιώς την έκφραση της Α.Λ.Σ με την λογική σχεδίασης NAND.

2ο Παράδειγμα: Να σχεδιαστεί το λογικό κύκλωμα του οποίου η έξοδος μηδέν στις περιπτώσεις που φαίνονται στον Π.Κ με λογική σχεδίασης NOR.

Λύση: α. Από το Π.Κ γράφουμε τη λογική συνάρτηση στη μορφή Γ.Α δηλαδή για τα μηδενικά

A	B	Z
0	0	**0**
0	1	**0**
1	0	**1**
1	1	**0**

του πίνακα οπότε έχουμε $\overline{Z} = \overline{A}.\overline{B} + \overline{A}B + A.B$



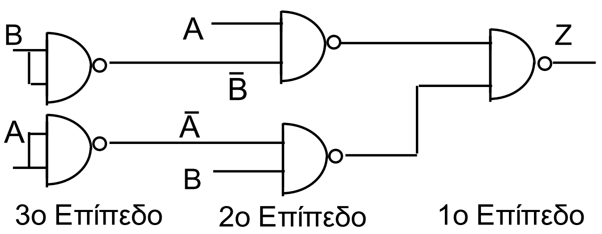
β. Απλοποιούμε τη συνάρτηση οπότε έχουμε με τη βοήθεια της Boole και έχουμε : $\overline{Z} = \overline{A}.\overline{B} + \overline{A}B + A.B = \overline{A} + A.B$ ή τελικά $Z = \overline{A} + B$

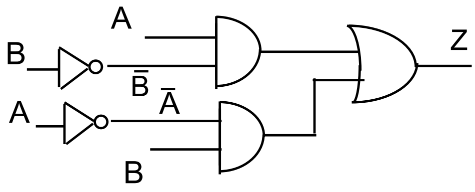
γ. Σχεδιάζουμε την τελική έκφραση (αλλιώς την Α.Λ.Σ) με την λογική σχεδίασης NOR.

4.11.2 Ανάλυση Σχεδίασης Συνδυαστικών Κυκλωμάτων

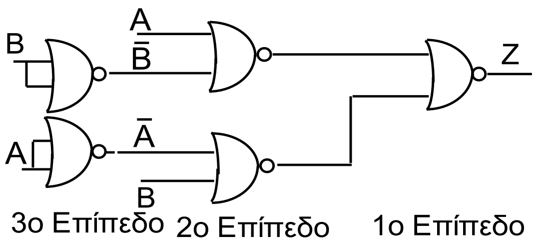
Ένα κύκλωμα που είναι σχεδιασμένο με μία από τις λογικές σχεδίασης α) NAND ή β) NOR αναλύεται, αν το χωρίσουμε σε επίπεδα (Levels), από τα οποία όσα αναλογούν σε άρτιο αριθμό αντιστοιχούν σε πύλες AND και όσα αναλογούν σε περιττό αριθμό σε OR, αν είναι σχεδιασμένο με τη Λογική NAND, ενώ αντίθετα όσα αναλογούν σε επίπεδο με περιττό αριθμό ανήκει σε AND και όσα αναλογούν σε επίπεδο με άρτιο αριθμό σε OR. αν είναι σχεδιασμένο με τη λογική NOR π.χ.

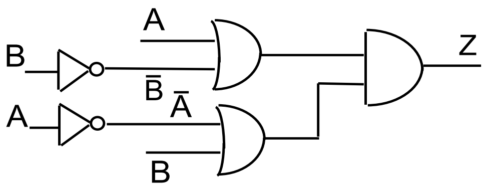
α) Ανάλυση Σχεδίασης λογικής NAND





β) Ανάλυση Σχεδίασης λογικής NOR



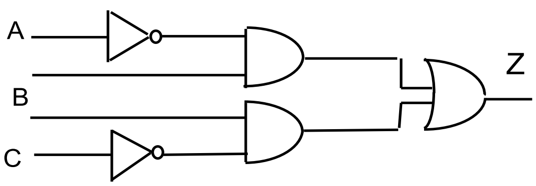


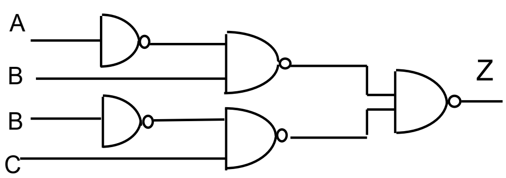
Παράδειγμα : Για τις δύο παρακάτω εκφράσεις α,β έχουμε:

α) $Z=\bar{A}B\bar{C}+\bar{A}BC+AB\bar{C}=\bar{A}B(\bar{C}+C)+AB\bar{C}=\bar{A}B+AB\bar{C}=B(\bar{A}+A\bar{C})=B(\bar{A}+\bar{C})=\bar{A}B+B\bar{C}$

Το κύκλωμα της συνάρτησης σχεδιασμένο με δυο τρόπους σχεδίασης φαίνεται στα σχήματα (επόμενη σελίδα)

α) Σχεδίαση με πύλες AND-OR και β) Σχεδίαση με πύλες NAND (μόνο)





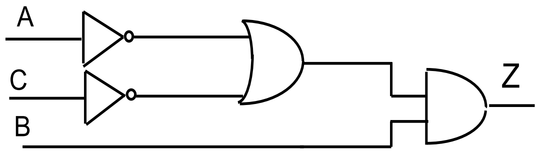
β). $Z=(A+B+C)(A+B+\bar{C})(\bar{A}+B+C)(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+\bar{C})$

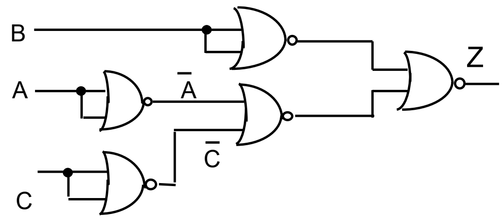
$=(A+B+C\bar{C})(\bar{A}+B+C\bar{C})(\bar{A}+\bar{B}+\bar{C})=(A+B)(\bar{A}+B)(\bar{A}+\bar{B}+\bar{C})=$

$Z=(A\bar{A}+\bar{A}B+AB+BB)(\bar{A}+\bar{B}+\bar{C})=B(\bar{A}+\bar{B}+\bar{C})=\bar{A}B+B\bar{B}+B\bar{C}=\bar{A}B+B\bar{C}=B(\bar{A}+\bar{C})$

και το αντίστοιχο κύκλωμα σχεδιασμένο με δυο τρόπους σχεδίασης φαίνεται στα σχήματα της επόμενης σελίδας.

α) Σχεδίαση με πύλες OR-AND   και   β) Σχεδίαση με πύλες NOR (μόνο)





Σημείωση: Μια παραλλαγή της (**β**) μεθόδου (ΓΑ) που χρησιμοποιείται στη κατασκευή κυκλώματος **μόνο** με **πύλες** NOR είναι η παρακάτω:

♦Χρησιμοποιούμε την έκφραση με το άθροισμα γινομένων για τους όρους όμως της συνάρτησης που είναι μηδέν-"**0**". Αυτό που θα προκύψει σαν έκφραση θα αντιστοιχεί στο ***συμπλήρωμα*** της ζητούμενης συνάρτησης.

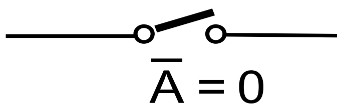
Παράδειγμα: Για την εφαρμογή μας έχουμε

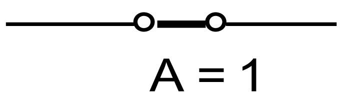
$$\overline{Z}_{(A,B,C)}= \overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}C+A\overline{B}.\overline{C} +A\overline{B}C+ABC= \overline{A}.\overline{B}(\overline{C} +C)+A\overline{B}(C+ \overline{C})+ABC =$$
$$= \overline{A}.\overline{B} +A\overline{B} +ABC= \overline{B}(\overline{A} +A)+ABC= \overline{B} +ABC= \overline{B} +B(AC)= \overline{B} +AC$$ και

$Z_{(A,B,C)}= \overline{(\overline{B}+AC)} = \overline{\overline{B}}*\overline{(A*C)} =B*(\overline{A} + \overline{C})$ δηλαδή το αποτέλεσμα της περίπτωσης β (μετά τις πράξεις). Το ποια από τις παραπάνω μεθόδους χρησιμοποιούμε εξαρτάται από την εφαρμογή και μόνο.

4.12. Άλγεβρα Διακοπτών

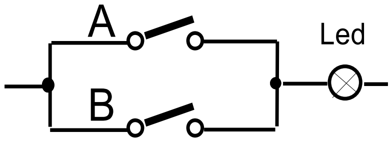
Τα θεωρήματα της άλγεβρας Boole μπορούν να αποδειχτούν μέσω κυκλωμάτων διακοπτών (μηχανικών για απλούστευση) όπου οι δύο τιμές **0,1** του δυαδικού σημαίνουν διακόπτες ανοιχτός (**0=OFF**)-κλειστός & (**1=ON**)-ανοικτός.

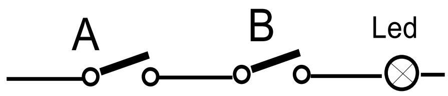
α. 

β. 

Έτσι οι λογικές πράξεις OR, NAND γίνονται με διακόπτες οπότε έχουμε και τη φυσική ερμηνεία των πράξεων OR-(Η) και AND-(ΚΑΙ) αφού στο (**α**) για να "ανάψει" η λυχνία (**Λ)** αρκεί <u>**ή**</u> ο **A** να είναι κλειστός <u>**ή**</u> ο **B** να είναι κλειστός <u>**ή**</u> αμφότεροι.

Στο (**β**) θα πρέπει **και** ο **A** να είναι κλειστός **και** ο **B** να είναι κλειστός, που είναι η

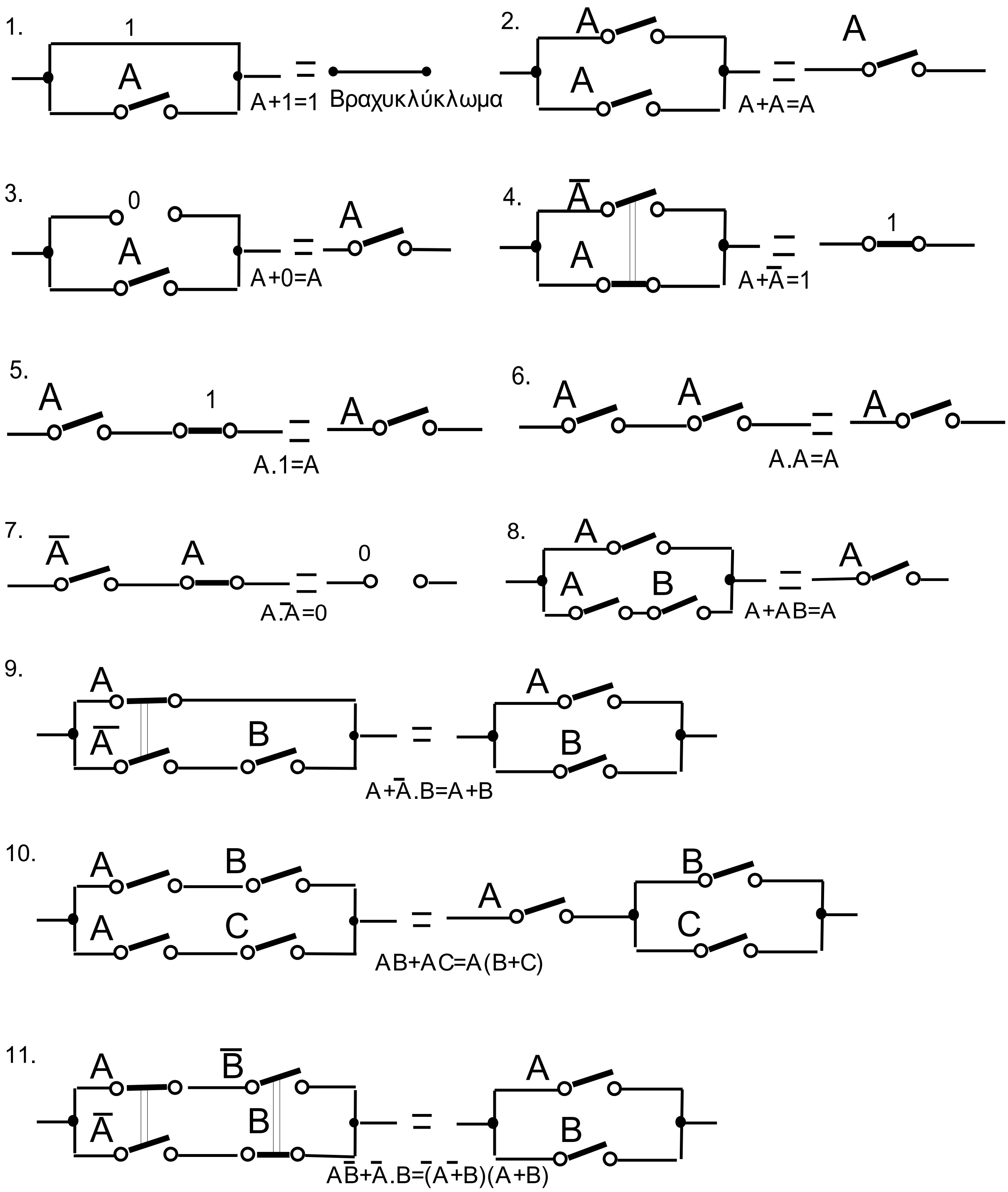




φυσική ερμηνεία των πράξεων OR-(Η) και AND-(ΚΑΙ) αφού στο (**α**) για να "ανάψει" η λυχνία (**Λ)** αρκεί <u>**ή**</u> ο **A** να είναι κλειστός <u>**ή**</u> ο **B** να είναι κλειστός <u>**ή**</u> αμφότεροι.

Στο (**β**) θα πρέπει **και** ο **A** να είναι κλειστός **και** ο **B** να είναι κλειστός

Από τα παραπάνω έχουμε ότι λογική πρόσθεση (**+**) σημαίνει διακόπτες παράλληλα (**¦¦**), ενώ λογικός πολλαπλασιασμός (*****) σημαίνει διακόπτες στη σειρά. Μερικά κυκλώματα διακοπτών που αποδεικνύουν τα θεωρήματα Boole είναι παρακάτω:



4.13. ΑΣΚΗΣΕΙΣ

1. Να σχεδιάσετε το κύκλωμα των λογικών συναρτήσεων με πύλες NAND

α. $Z_1 = A.B.C$ **β**. $Z_2 = A + B + C$ **γ**. $Z_3 = A.\overline{B} + \overline{A}.C$ **δ**. $Z_4 = A.\overline{B}.\overline{C} + \overline{A}.\overline{C}.D + \overline{A}.B$

2. Σ Να σχεδιάσετε το κύκλωμα των λογικών συναρτήσεων με πύλες NOR.

α. $Z_1 = A.B.C$ **β**. $Z_2 = A + B + C$ **γ**. $Z_3 = (A + \overline{B}).(\overline{A} + C)$ **δ**. $Z_4 = (A + B + C)(\overline{B} + \overline{C})(\overline{A})$

3. Να απλοποιηθούν οι παρακάτω λογικές συναρτήσεις

α. $Z_1 = A + \overline{B} + B.C + \overline{A}.C$ **β.** $Z_2 = \overline{A}.C.(BC + AB + \overline{A}.B)$ **γ.** $Z_3 = (\overline{A} + B.C).(\overline{A} + B)$

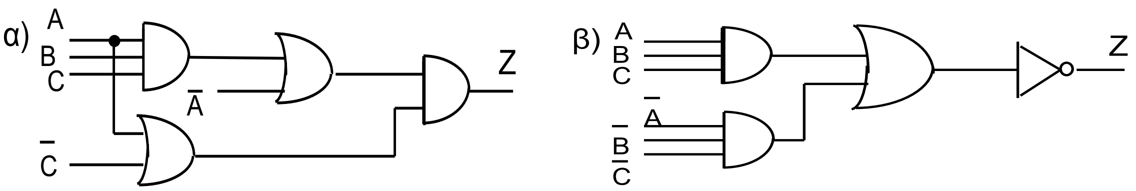
δ. $Z_4 = \overline{B} + .A.B.C + \overline{A}.B$ **ε.** $Z_5 = ABC + \overline{B}.\overline{C} + \overline{A}$

4. Να γραφεί ο πίνακας αληθείας για τη λύση του παρακάτω προβλήματος: Διαθέτουμε μία λυχνία και τέσσερις διακόπτες σε διαφορετικές θέσεις και ζητούμε τις δυνατές σχέσεις μεταξύ των A,B,C,D διακοπτών ώστε να ανάβει ή να σβήνει η λυχνία από οποιοδήποτε διακόπτη.(Αποκλείεται ταυτόχρονη χρήση δύο διακοπτών ή περισσοτέρων).

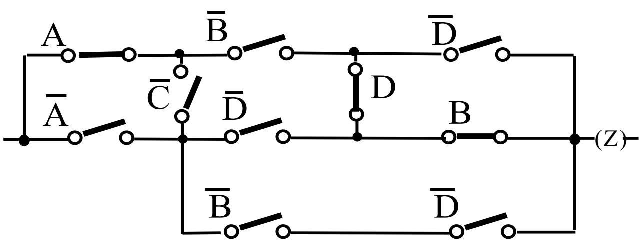
5. Να εκφραστούν οι $Z_1 = A.\overline{B}.\overline{C} + \overline{A}.\overline{C}.D + \overline{A}.B$ $Z_2 = (A + \overline{B} + \overline{C})(\overline{A} + \overline{C} + D)(\overline{A} + B)$

σε κανονική μορφή και άθροισμα των βαρών των.

6. Να Βρεθούν οι λογικές συναρτήσεις των κυκλωμάτων και να απλοποιηθούν.



7. Να βρεθεί η λογική συνάρτηση που υλοποιεί το διακοπτικό κύκλωμα. Να απλοποιηθεί και να σχεδιαστεί το κύκλωμα της Α.Λ.Σ με πύλες NAND.



8. Από το πίνακα αληθείας να δώσετε τη λογική συνάρτηση Z με τη μορφή αθροίσματος ελαχίστων όρων και με τη μορφή γινομένων μέγιστων όρων. Στη συνέχεια να σχεδιάσετε τα κυκλώματα πυλών για τις συναρτήσεις και συγκρίνετε με τα κυκλώματα που προκύπτουν.

A	B	C	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0