Ψηφική Σχεδίαση

Ενότητα 4: Υλοποίηση Κυκλωμάτων με πύλες NOT – AND και NOR, περιττή συνάρτηση, συνάρτηση ισοτιμίας.

Δρ. Μηνάς Δασυγένης
@ieee.ormdasygg
Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών
http://arch.icte.uowm.gr/mdasyg

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σκοπός της ενότητας

- Να γίνει υλοποίηση κυκλωμάτων με πύλες NOT, OR , NOR.
- Να γίνει ανάλυση των περιττών και των άρτιων συναρτήσεων και των συναρτήσεων ισοτιμίας.
- Εισαγωγή στα κυκλώματα CMOS.

Βελτιστοποίηση Κυκλωμάτων Πολλαπλών επιπέδων (1)

- Μπορεί να προσφέρει μεγαλύτερη εξοικονόμηση στο κόστος ενός κυκλώματος.

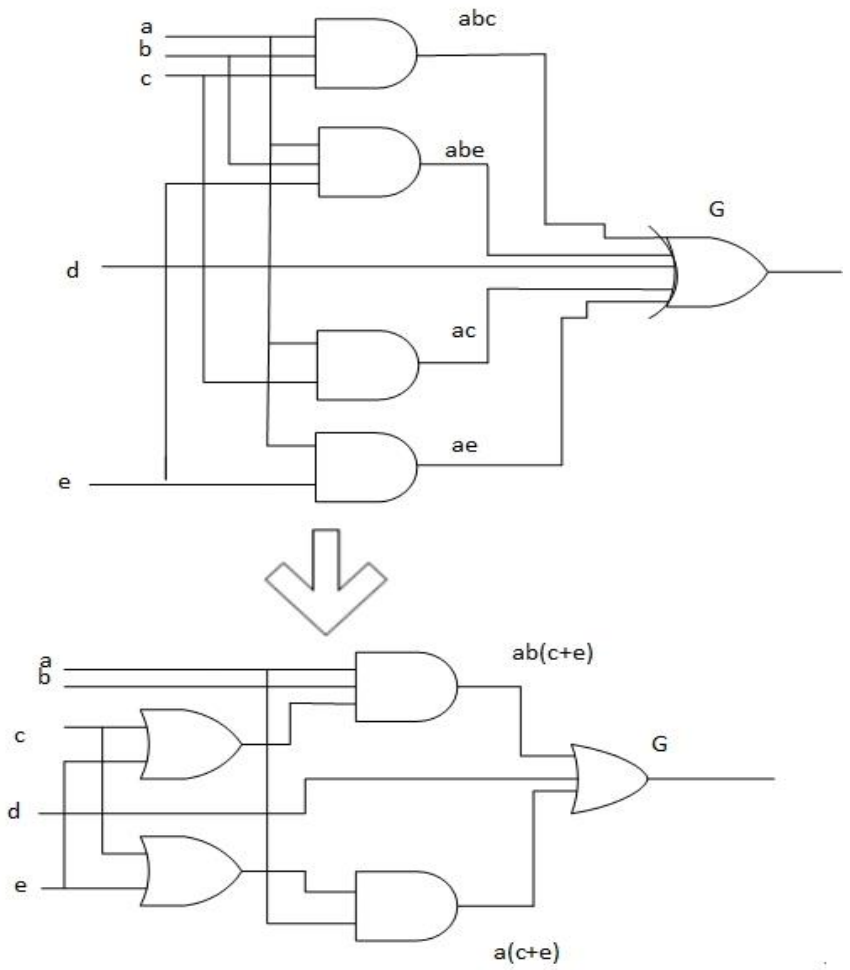
- Θεωρήστε:

  G = abc + abe + d + ac + ae

  κόστος = 5 πύλες + 15 διασυνδέσεις

- G = ab(c + e) + d a(c + e)

  κόστος = 5 πύλες + 12 διασυνδέσεις



Βελτιστοποίηση Κυκλωμάτων Πολλαπλών επιπέδων (2)

- G = ab(c + e) + d + a(c + e)

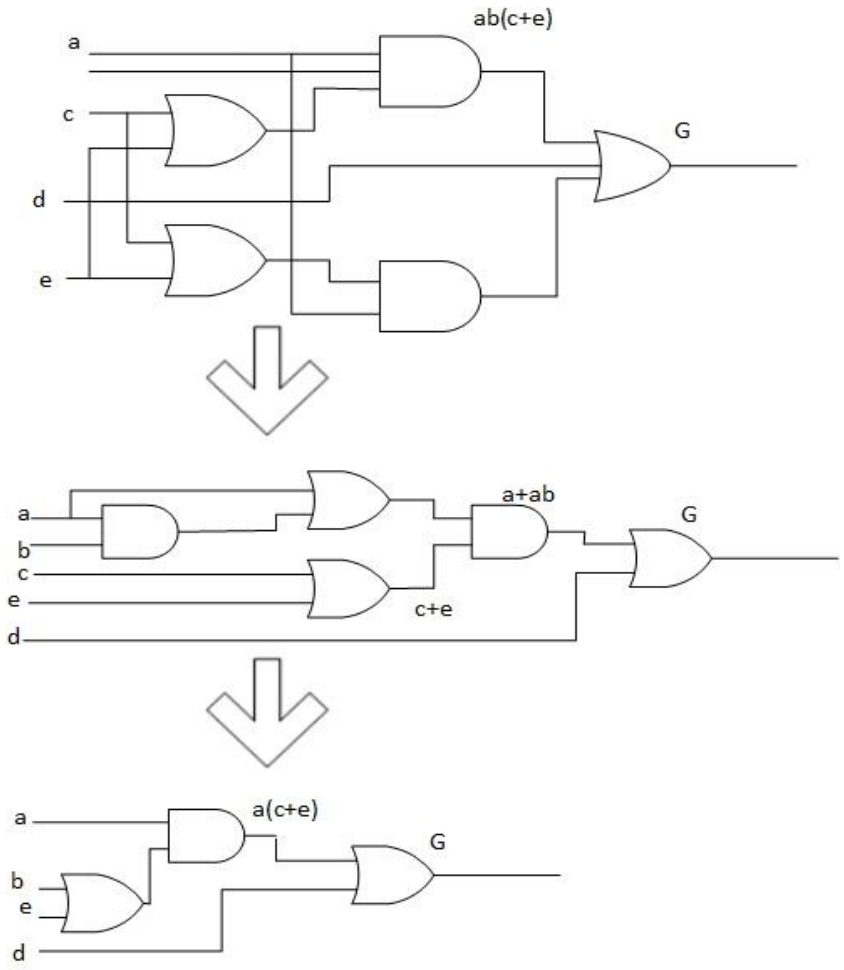
  κόστος = 5 πύλες + 12 διασυνδέσεις

- G = (ab + a)(c + e) + d

  κόστος = 5 πύλες + 9 διασυνδέσεις

- G = a(c + e) + d

  κόστος = 3 πύλες + 6 διασυνδέσεις

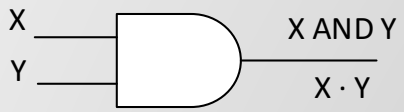


Βελτιστοποίηση Κυκλωμάτων Πολλαπλών επιπέδων (3)

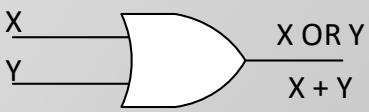
- Δεν υπάρχει συστηματική μέθοδος/ αλγόριθμος (όπως χάρτες-Karnaugh ή Quenn-McCluskey για διεπίπεδη ελαχιστοποίηση) για πολλαπλά επίπεδα.
- Βασιζόμαστε σε ένα σύολο βασικών λειτουργιών μετασχηματισμών, για να βρούμε μια καλή λύση, αλλά όχι απαραίτητα βέλτιστη (sub-optimal solution).
- Μετασχηματισμοί:
  - Παραγοντοποίηση (Factoring)
  - Αποσύνθεση (Decomposition)
  - Εξαγωγή (Extraction)
  - Αντικατάσταση (Substitution)
  - Απαλοιφή (Elimination ή Flattening ή Collapsing)

Λογικές Πύλες AND, Or, NOT

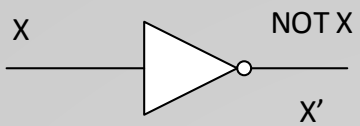
- Μπορούμε να κατασκευάσουμε οποιοδήποτε συνδυαστικό κύκλωμα με τις πύλες AND, OR και NOT.



X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1



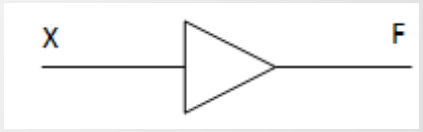
X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1



X	NOT X
0	1
1	0

- Επιπρόσθετες λογικές πύλες μπορούν να χρησιμοποιηθούν για πρακτικούς λόγους.

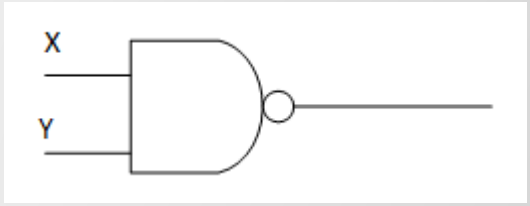
Οι πύλες Buffer, NAND, NOR



F = X

X	F
0	0
1	1

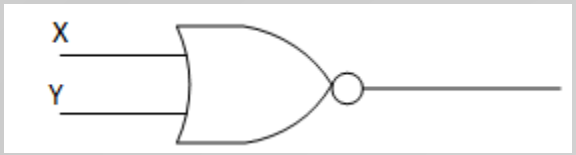
X NAND Y



$(X \cdot Y)'$

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

X NOR Y

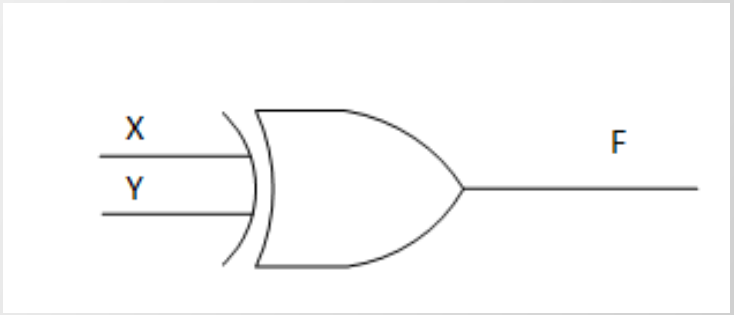


$(X + Y)$

X	Y	X OR Y
0	0	1
0	1	0
1	0	0
1	1	0

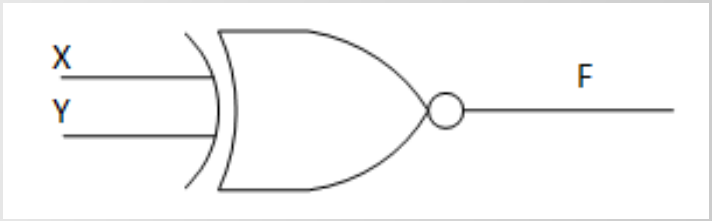
Οι πύλες XOR, XNOR

XOR: πύλη “μη-ισότητας”



X	Y	F = X ⊕ Y
0	0	0
0	1	1
1	0	1
1	1	0

XNOR: πύλη “ισότητας”



X	Y	F
0	0	1
0	1	0
1	0	0
1	1	1

Υλοποίηση με πύλες ΌΧΙ-ΚΑΙ, ΟΥΤΕ

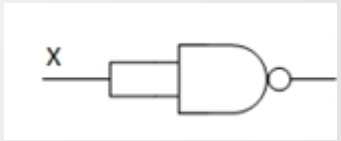
- Τα ψηφιακά κυκλώματακατασκευάζονται πολύ συχνότερα με πύλες ΟΧΙ-ΚΑΙ (NAND) ή ΟΥΤΕ (NOR) αντί για πύλες ΚΑΙ, Η΄.
- Είναι οι βασικές πύλες που χρησιμοποιούνται σε όλες τις οικογένειες ολοκληρωμενων κυκλωμάτων ψηφιακής λογικής.
- Υπάρχουν τρόποι μετατροπής των συναρτήσεων σε ισοδύναμους όρους NAND, NOR.

Η πύλη NAND (1)

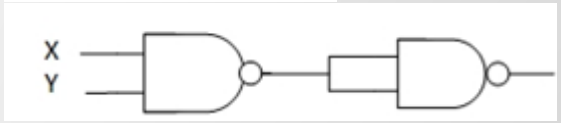
- Είναι γνωστή ως «οικουμενική» (“universal”) πύλη γιατί μπορούμε να υλοποιήσουμε οποιοδήποτε ψηφιακό κύκλωμα μόνο με αυτές τις πύλες.

- Για να αποδείξουμε το πιο πάνω χρειάζεται να δείξουμε ότι οι πύλες AND, OR και NOT μπορούν να εκφραστούν χρησιμοποιόντας μόνο πύλες NAND.

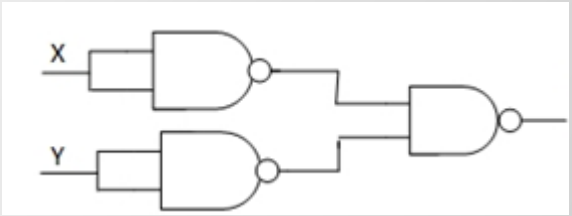
Η πύλη NAND (2)



$F = (X \cdot X)' = X' + X' = X'$

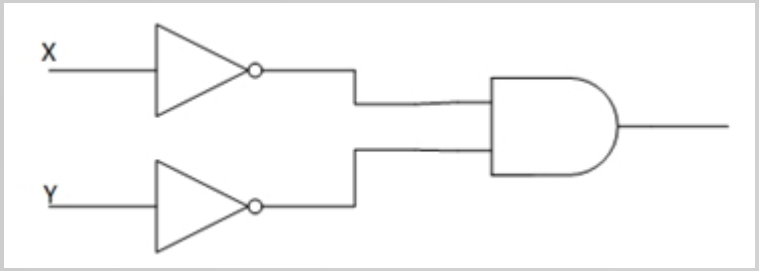


$F = ((X \cdot Y)')' = (X' + Y')' = X'' \cdot Y'' = X \cdot Y$



$F = (X' \cdot Y')' = X'' + Y'' = X + Y$

$F = X'$

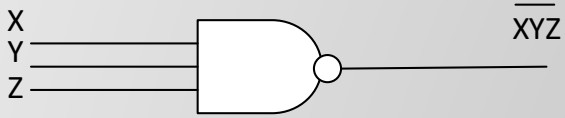


$F = X + Y$

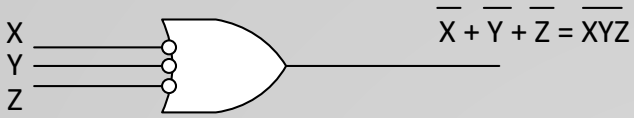
$F = X \cdot Y$

Κυκλώματα NAND

- Για να βρείτε την υλοποίηση ενός κυκλώματος χρησιμοποιώντας μόνο πύλες NAND ακολουθήστε τα πιο κάτω βήματα:
  - Βρέστε ένα απλοποιημένο SOP.
  - Το SOP είναι ένα AND-OR κύκλωμα.
  - Αλλάξτε το AND-OR κύκλωμα σε NAND κύκλωμα.
  - Χρησιμοποιήστε τα πιο κάτωεναλλακτικά σύμβολα:



(a) AND – NOT

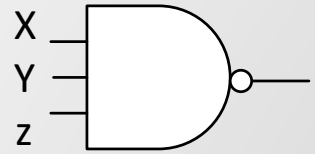


(b) NOT – OR

  - Το (a) είναι ισοδύναμο με το (b)

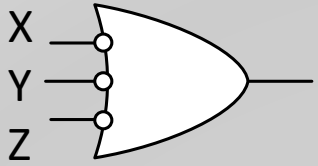
(c) NOT

Σχηματικά σύμβολα πυλών NAND



(a) AND – invert

Έξοδος: (xyz)’

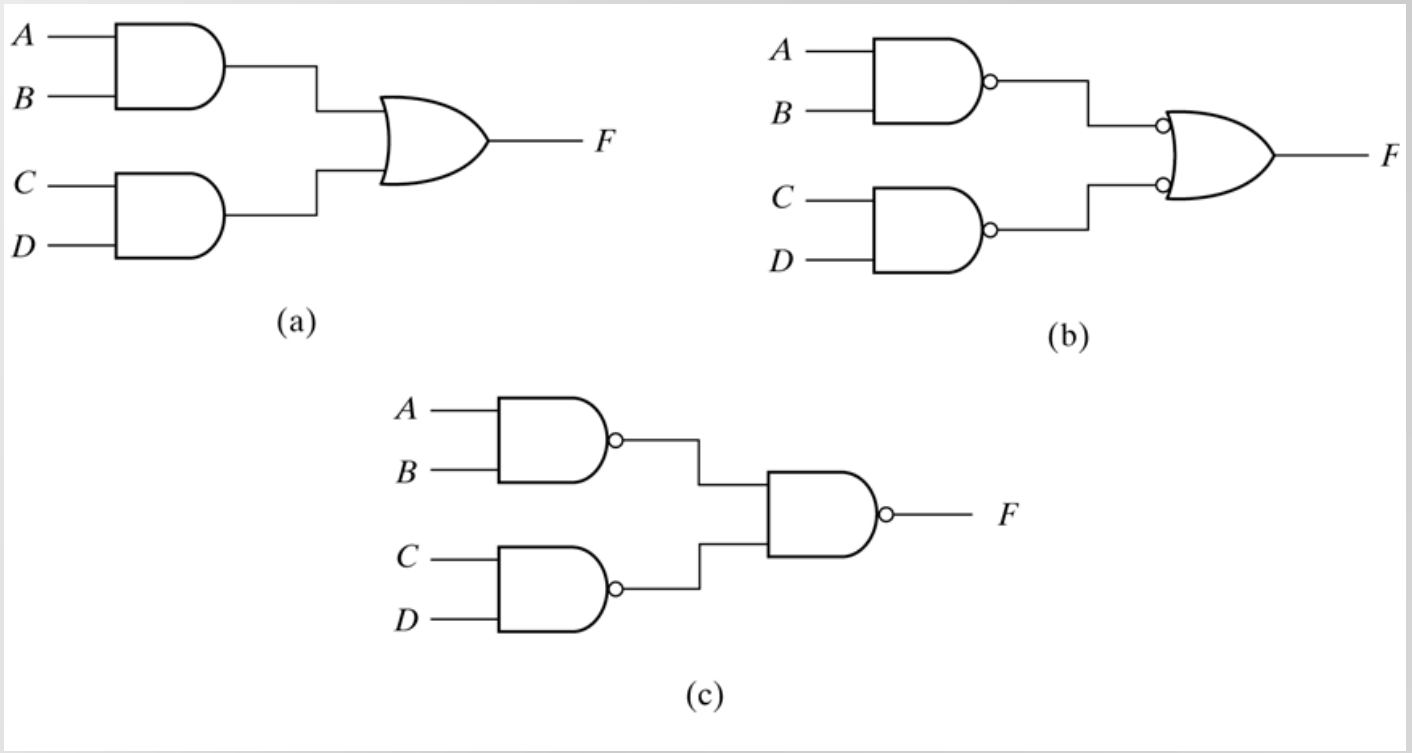


(b) Invert – OR

Έξοδος: x’ + y’ + z’ = (xyz)’

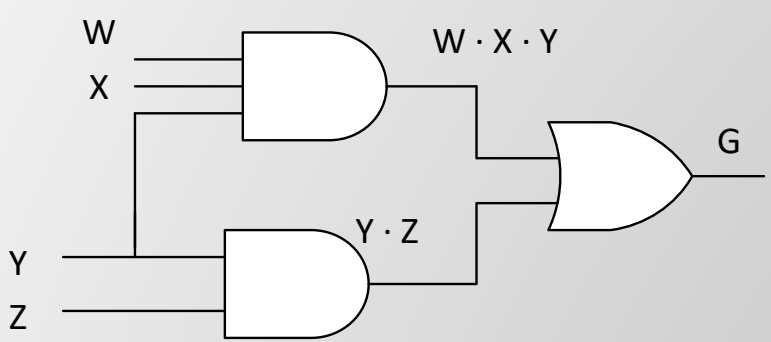
- Είναι ισοδύναμα λογικά σύμβολα.
- Είτε χρησιμοποιείται το ένα είτε το άλλο.
- Αν χρησιμοποιηθούν ταυτόχρονα τότε έχουμε «μικτή σημειογραφία».

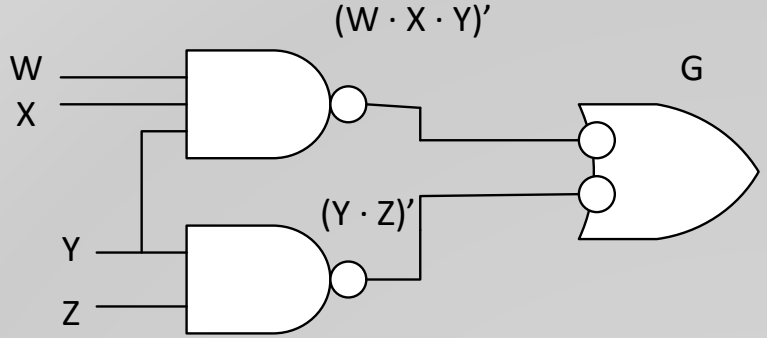
3 τρόποι υλοποίησης συνάρτησης



Τρεις τρόποι για να εφαρμοστεί η συνάρτηση F = AB + CD

Παράδειγμα SoP με NAND (1)



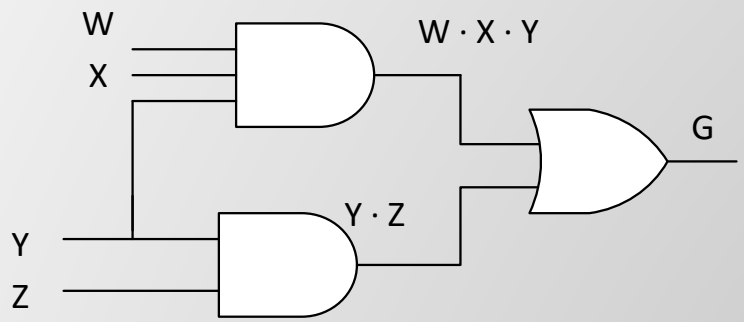


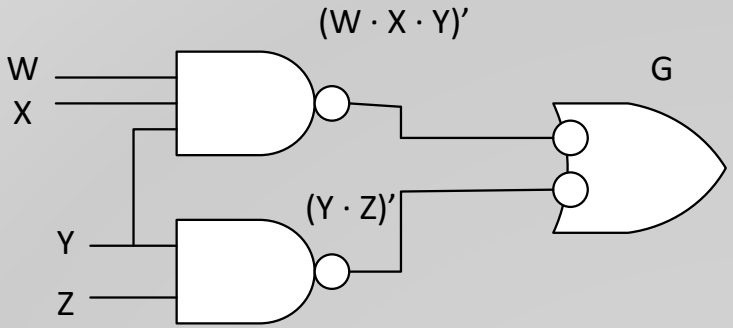
Υλοποίηση 2 επιπέδων

a) Αρχικό SOP (AND-OR κύκλωμα).

b) Υλοποίηση χρησιμοποιώντας πύλες NAND.

Παράδειγμα SoP με NAND





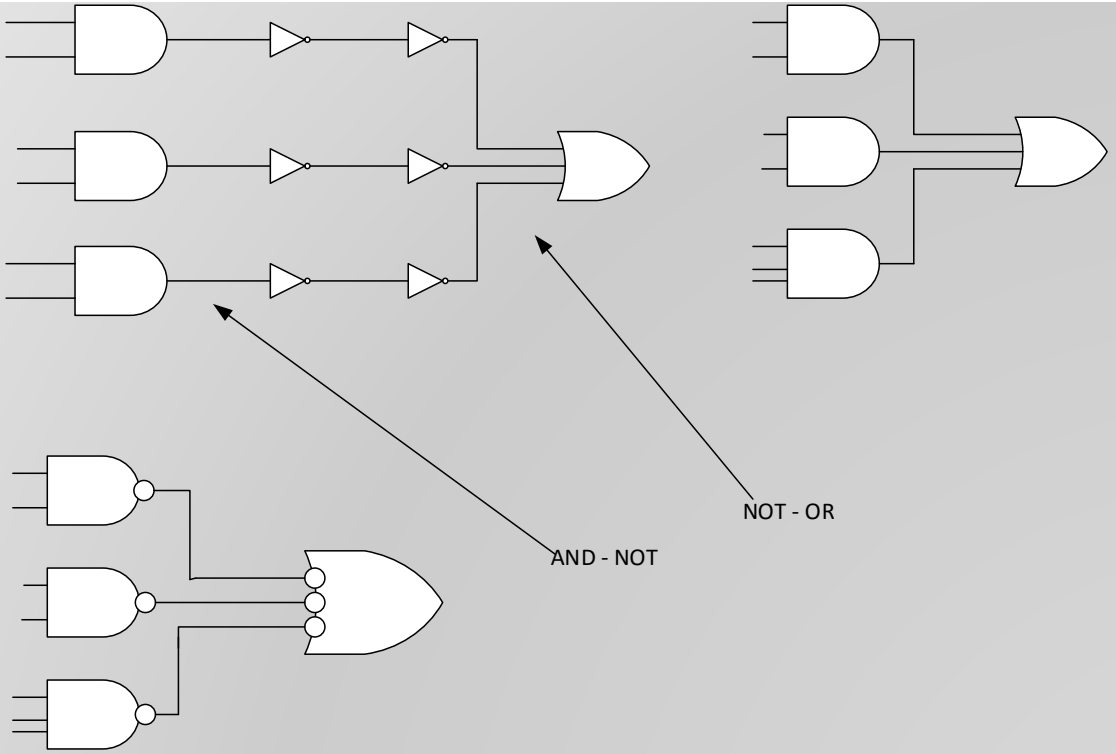
- Επαλήθευση:

(a) G = WXY + YZ

(b) G = ((WXY)' • (YZ)')'

= (WXY)'' + (YZ)'' = WXY + YZ

SoP με NAND



(α) Αρχικό SOP.

(b) Διπλή αντιστροφή (NOT) και ομαδοποίηση.

(c) Αντικατάσταση με πύλες NAND.

Υλοποίηση 2 επιπέδων με NAND (1)

- $F(X, Y, Z) = \sum m(0, 6)$

1. Εκφράστε την F σε SOP μορφή

   $F = X'Y'Z' + XYZ'$

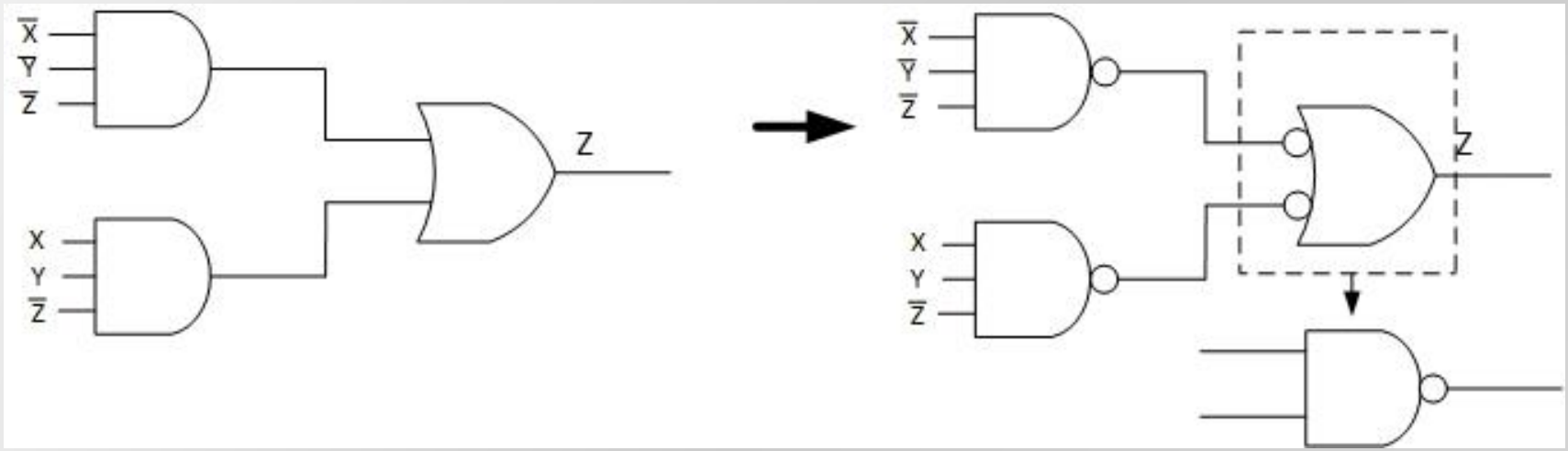
2. Βρείτε την SOP υλοποίηση της F.

3. Αντικατάσταση:

   AND → AND-NOT μορφή της NAND.

   OR→ NOT-AND μορφή της NAND.

Υλοποίηση 2 επιπέδων με NAND (2)



Διεπίπεδη υλοποίηση με πύλες NAND

F = X'Y'Z' + XYZ'

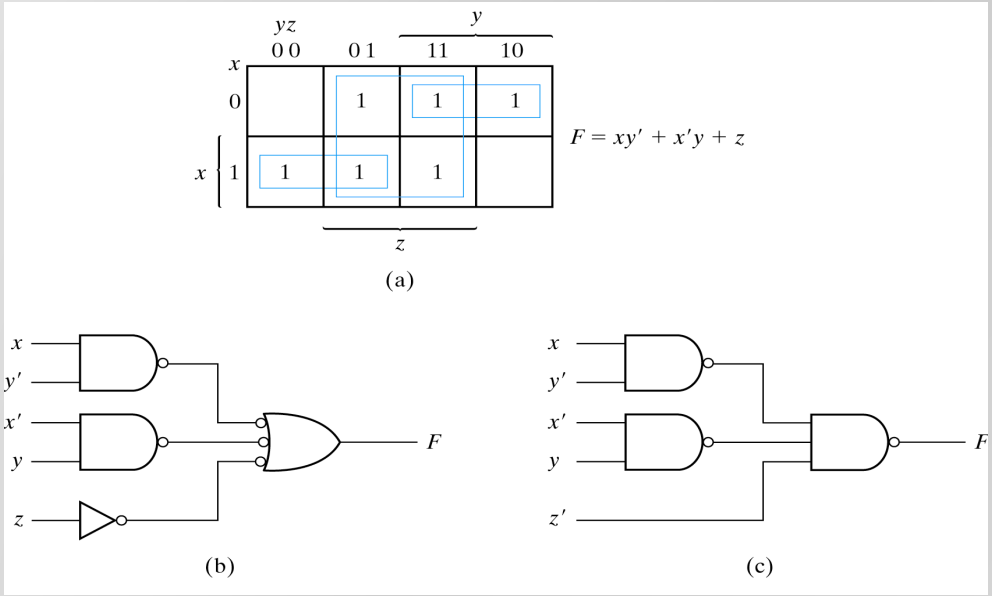
Διαδικασία για σχεδιασμό με NAND

- Απλοποιούμε τη συνάρτηση σε SoP.
- Τοποθετούμε NAND για κάθε όρο γινομένου.
- Τοποθετούμε μια AND-NOT ή NOT-OR στο δεύτερο επίπεδο.
- Ένας όρος με μια μεταβλητή χρειάζεται έναν αντιστροφέα στο πρώτο επίπεδο.

Παράδειγμα υλοποίησης με NAND

- Υλοποιήστε την παρακάτω συνάρτηση Boole με πύλες ΟΧΙ-ΚΑΙ.

F = (1, 2, 3, 4, 5, 7):



– F = xy’ + x’y + z

NAND πολλαπλών επιπέδων

- Ξεκινά από ένα κύκλωμα πολλαπλών επιπέδων:
  1. Μετατροπη όλων των πυλών AND σε NAND με σύμβολα AND-NOT.
  2. Μετατροπή όλων των πυλών OR σε NAND με σύμβολα NOT-OR.
  3. Έλεγχος όλων των αντιστροφέων (bubbles) στο διάγραμμα. Για κάθε bubble που δεν εξουδετερώνεται με άλλο bubble στην ίδια γραμμή, προσθέτουμε μια πύλη NOT ή συμπληρώνουμε την είσοδο.

Πολυ-επίπεδη σχεδίαση με NAND

- Παράδειγμα:
  - Χρησιμοποιήστε πύλες NAND και NOT για την υλοποίηση της:

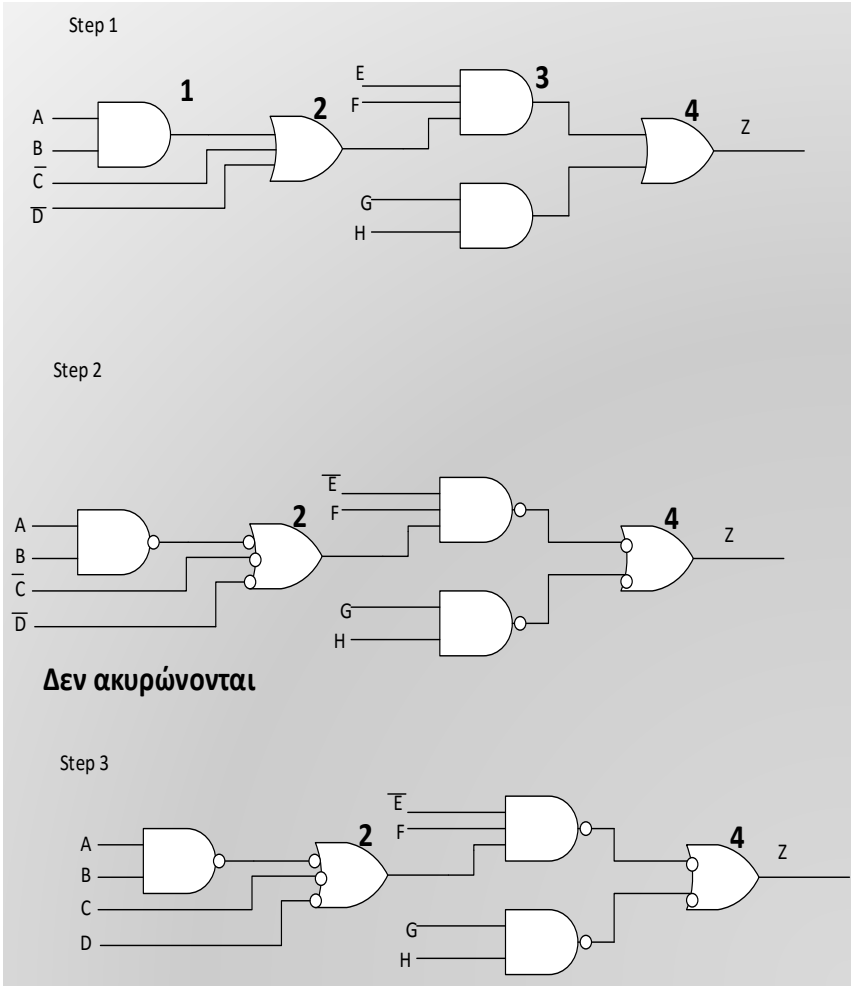
$Z = E'F\,(\,AB + C' + D'\,) = GH$

AB (1)

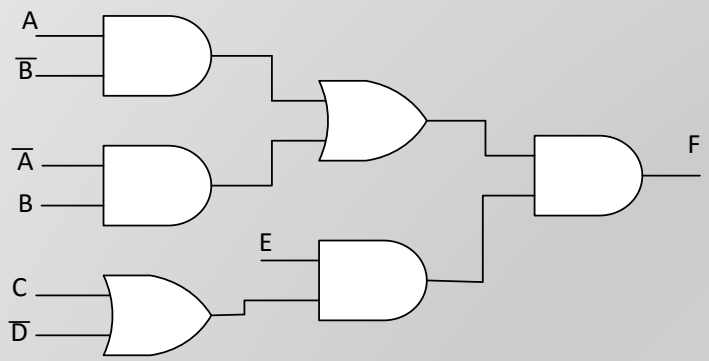
$AB + C' + D'$ (2)

$E'F(\,AB + C' + D'\,)$ (3)

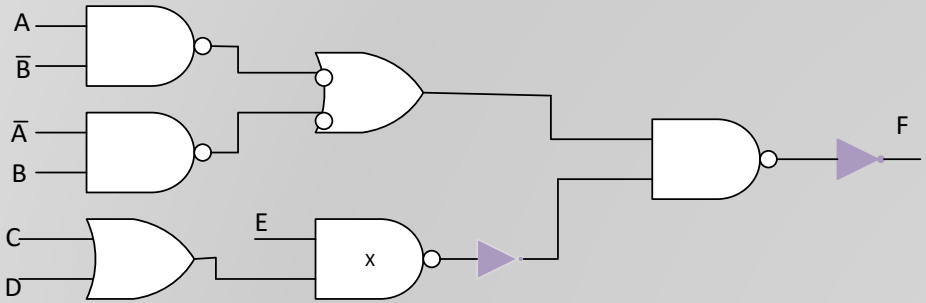
$E'F(\,AB + C' + D'\,) + GH$ (4)



Μετατροπές πυλών σε NAND (1)



(a) AND – OR gates

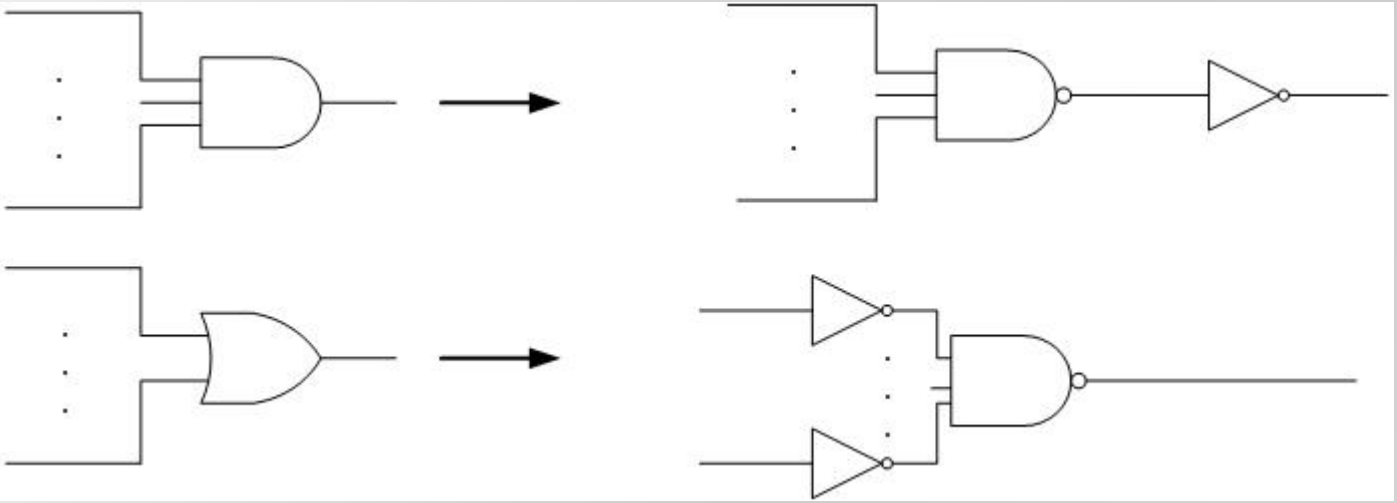


(b) NAND gates

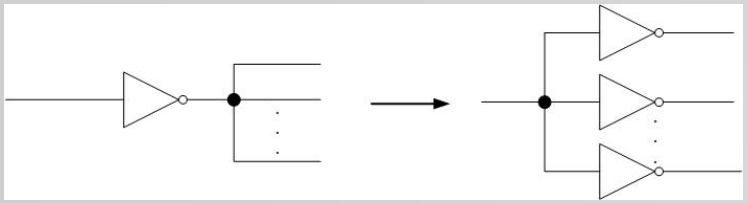
- $F = (A\overline{B} + \overline{A}\, B)\, E\, (C+\overline{D})$

Μετατροπές πυλών σε NAND (2)

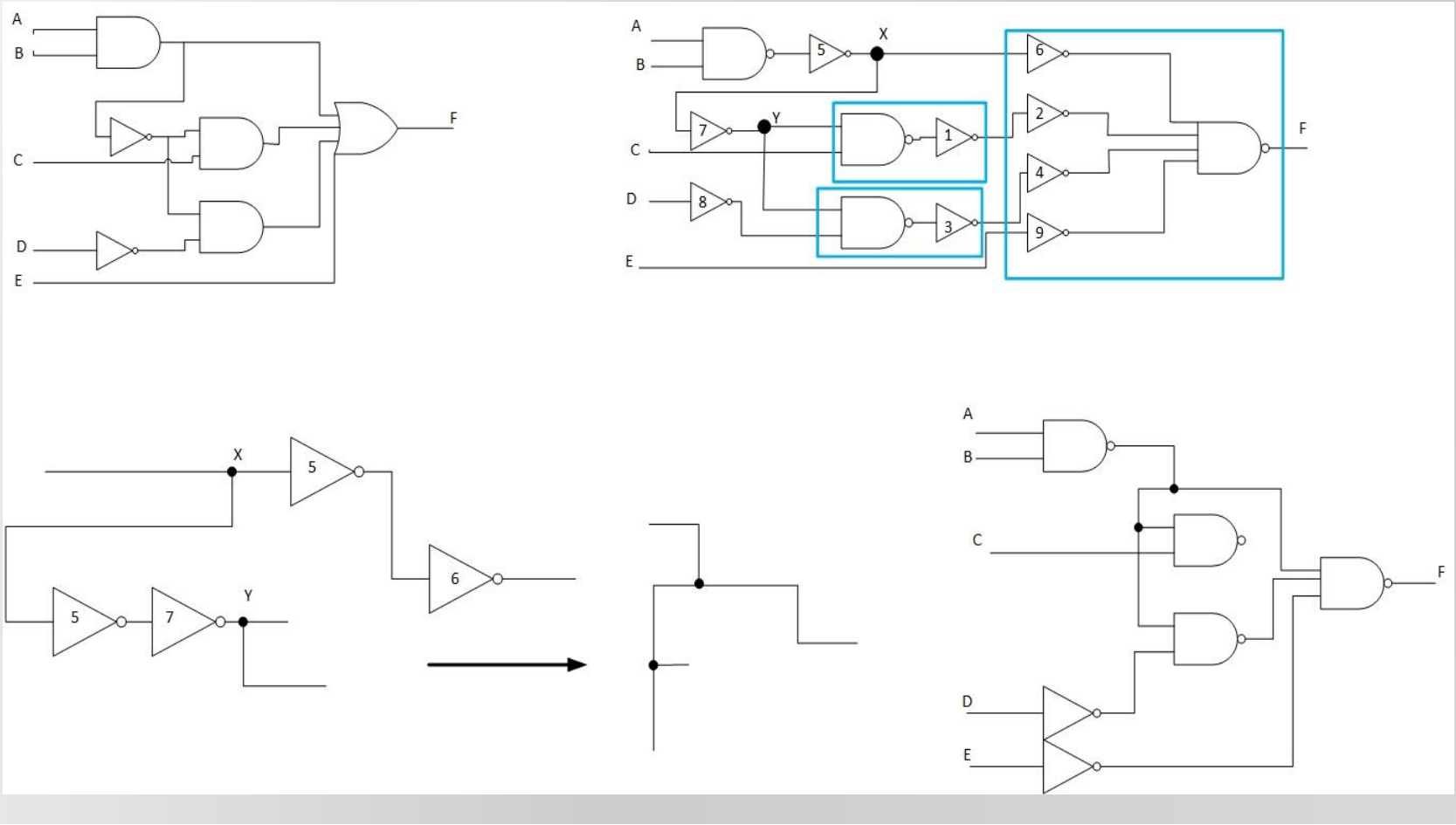
- Αντικατάσταση πυλών τύπου AND και OR:



- Πύλες NOT:

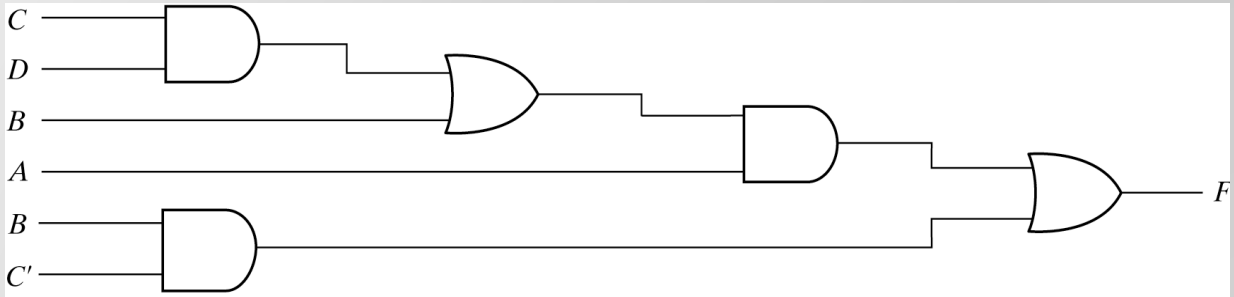


Παραδείγματα (1)

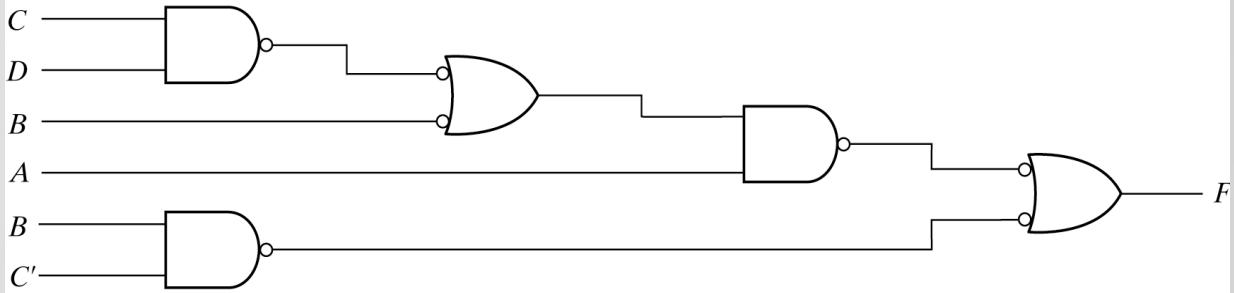


Παραδείγματα (2)

- Υλοποίηση: F = A (CD + B) + BC



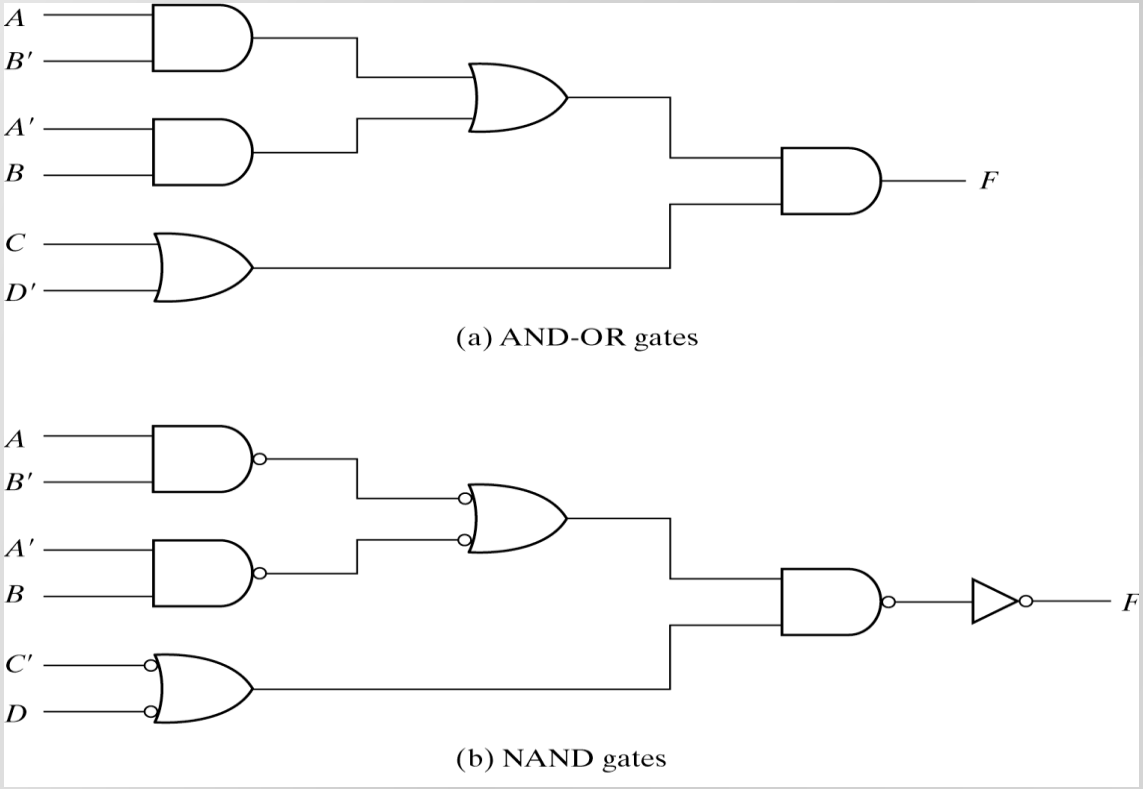
(a) AND-OR gates



(a) NAND gates

Παραδείγματα (3)

- F = (AB′ + A′B)(C + D′)



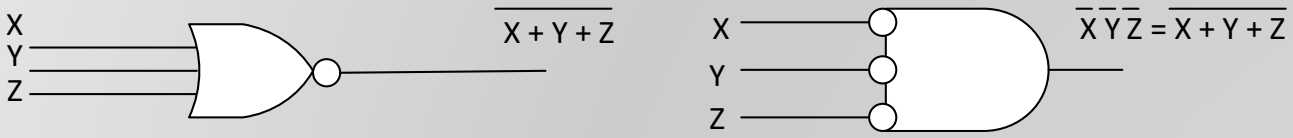
Σχεδιασμός με πύλες NOR

- Επίσης «οικουμενική» πύλη αφού οποιοδήποτε ψηφιακό κύκλωμα μπορεί να υλοποιηθεί μόνο με πύλες NOR.

- Μπορούμε να το αποδείξουμε με τον ίδιο τρόπο που έχουμε αποδείξει την πύλη NAND.

Κυκλώματα NOR

- Για την υλοποίηση μιας συνάρτησης με πύλες NOR:
  - Βρείτε ένα απλοποιημένο POS.
  - Το POS είναι ένα κύκλωμα OR-AND.
  - Αλλάξτε το OR-AN κύκλωμα σε NOR κύκλωμα.
  - Χρησιμοποίστε τα πιο κάτω σύμβολα.

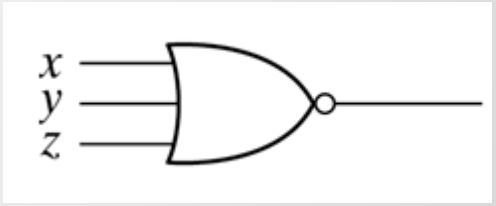


(a) OR – NOT

(b) NOT - AND

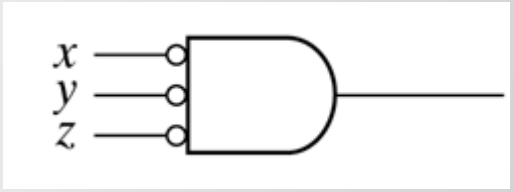
2 ισοδύναμα σύμβολα

(a) OR-invert



(x + y + z)’

(a) Invert-AND



x’y’z’ = (x + y + z)’

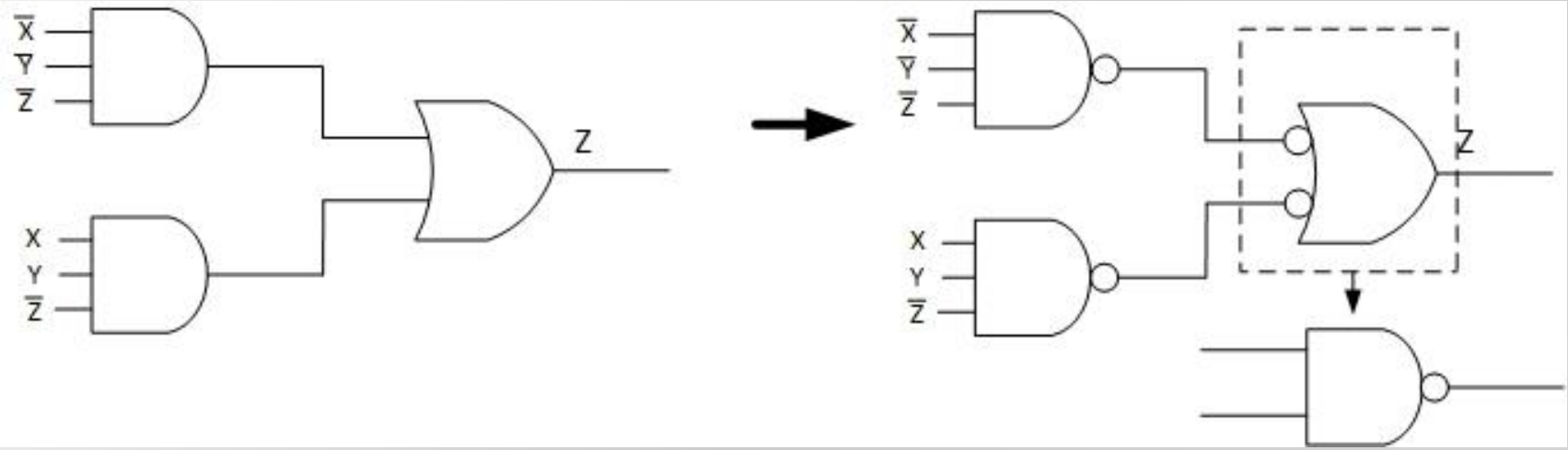
Υλοποίηση 2 επιπέδων με NOR

- $F(X, Y, Z) = \sum m(0, 6)$

1. Εκφράστε την $F'( )$ se SOP μορφή:
   1. $F' = \sum m(1, 2, 3, 4, 5, 7)$
      $X'Y'Z + X'YZ' + X'YZ + XY'Z' + XY'Z + XYZ$
   2. $F = XY' + X'Y + Z$
2. Πάρτε το συμπλήρωμα της F' να υπολογίσετε την F σε μορφή POS:

   $F = (F')' = (X' + Y)(X + Y')(Z')$
3. Βρείτε την OR-AND υλοποίηση της F.
4. Προσθέστε bubbles και αντιστροφείς για την μετατροπή μιας OR-AND υλοποίησης σε μια NOR-NOR υλοποίηση.

Παραδείγματα (4)

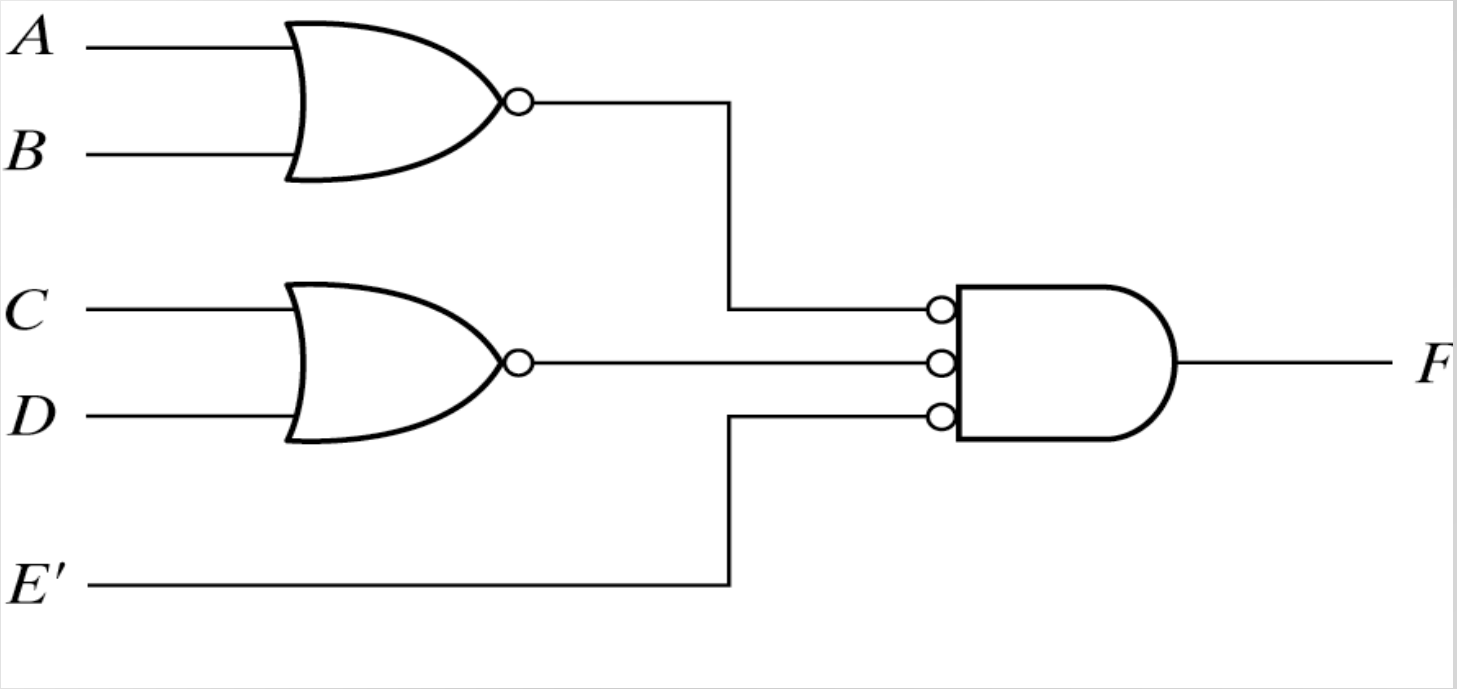


Υλοποίηση 2 επιπέδων με πύλες NOR.

$$F = (F')' = (X' + Y)(X + Y')Z'$$

Παραδείγματα (5)

- F = (A + B) (C + D) E



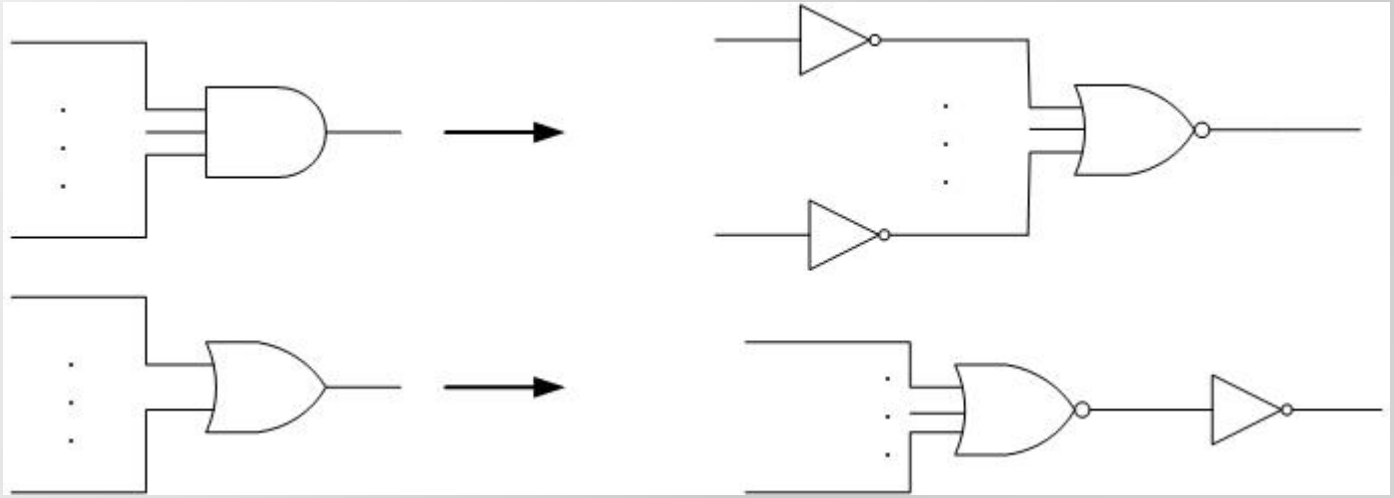
Πολλαπλών επιπέδων NOR

- Ξεκινά από ένα κύκλωμα πολλάπλών επιπέδων:

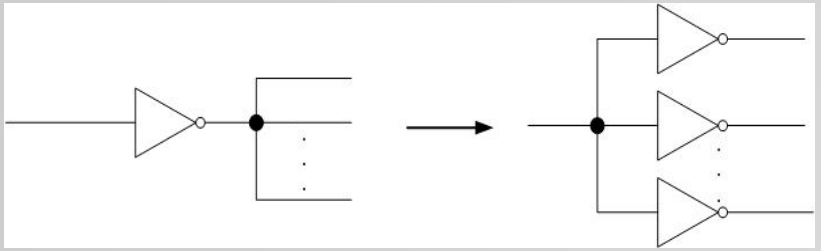
1. Μετατροπή όλων των πυλών OR σε NOR με σύμβολα OR-NOT.
2. Μετατροπή όλων των πυλών AND σε NOR με σύμβολα NOT-AND.
3. Έλεγχος όλων των αντιστροφέων (bubbles) στο διάγραμμα. Για κάθε bubble που δεν εξουδετερώνεται με άλλο bubble πάνω στην ιδια γραμμή, προσθέτουμε μια πύλη NOT ή συμπληρώνουμε την είσοδο.

Ένας απλός τρόπος

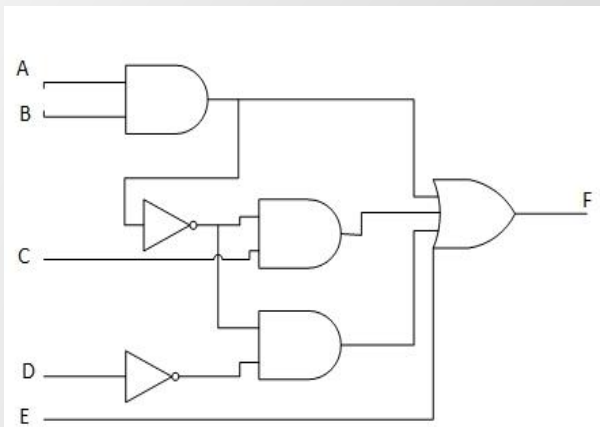
- Αντικατάσταση πυλών τύπου AND και OR:

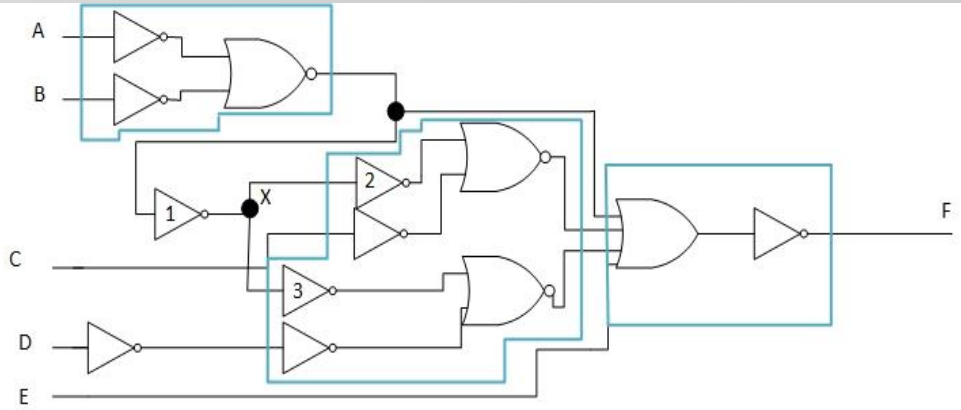


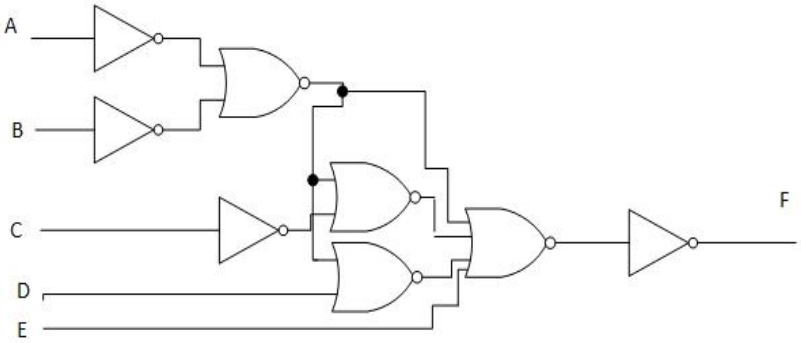
- Πύλες NOT:



Παράδειγμα

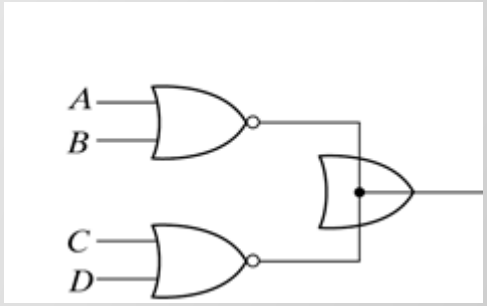




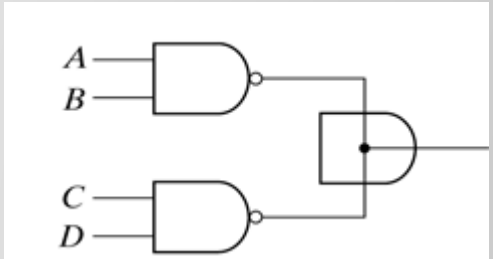


Καλωδιωμένη Λογική (1)

- Μερικές πύλες NAND, NOR (αλλά όχι όλες) επιτρέπουν να συνδεθούν οι έξοδοι των πυλών με καλώδιο.
- Αυτό ονομάζεται « καλωδιωμένη λογική ».



F = (AB + CD)’



F = [(A + B) (C + D)]’

Καλωδιωμένη Λογική (2)

- Μια πύλη καλωδιωμένης λογικής δεν αντιστοιχεί σε πραγματική πύλη δεύτερου επιπέδου.

- Μπορεί όμως για πρακτικούς λόγους να θεωρήσουμε ότι είναι πύλη.

Συνάρτηση eXclusive OR (XOR)

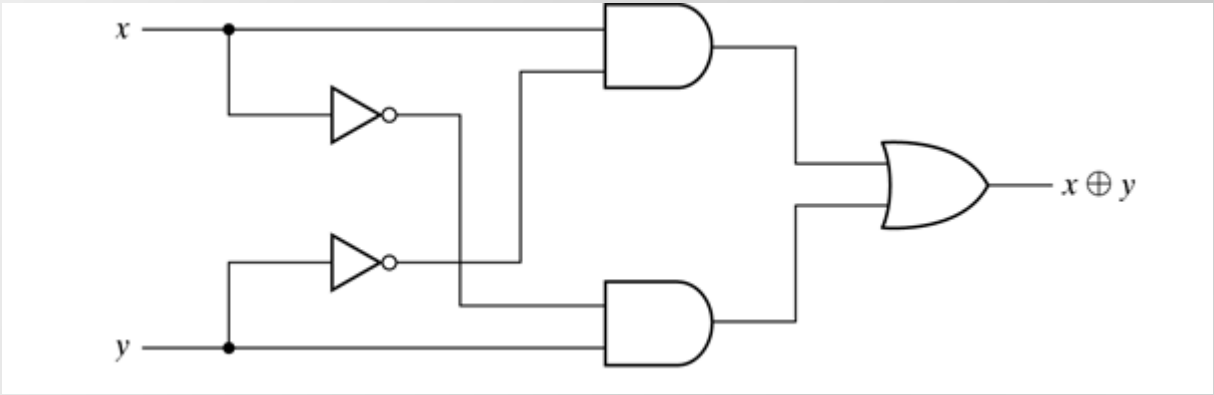
- XOR (συμβολίζεται με $\oplus$): η συνάρτηση μη-ισότητας
- XOR(X, Y) = $X \oplus Y = X'Y + XY'$
- Ταυτότητες:
  - $X \oplus 0 = X$
  - $X \oplus 1 = X'$
  - $X \oplus X = 0$
  - $X \oplus X' = 1$
- Ιδιότητες:
  - $X \oplus Y = Y \oplus X$ --Αντιμεταθετική
  - $(X \oplus Y) \oplus W = X \oplus (Y \oplus W)$ --Προσεταιριστική

Υλοποίηση XOR συνάρτησης

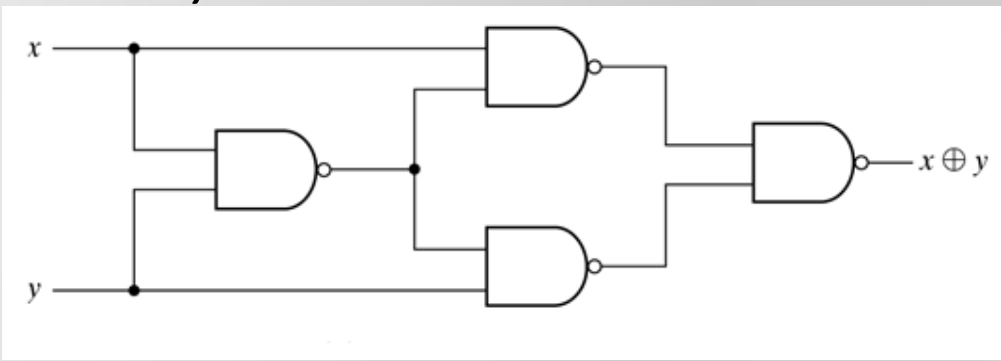
- XOR(a, b) = ab′ + a′b
- Άμεσος τρόπος: 5 πύλες
  - 2 αντιστροφείς, δύο AND 2-εισοδων, μια OR 2-εισόδων ή
  - 2 αντιστροφείς & 3 NAND 2-εισόδων.
- Έμμεσος τρόπος:
  - 4 πύλες NAND.

XOR με πύλες AND-OR-NOT (άμεσος)

Με πύλες AND – OR – NOT :

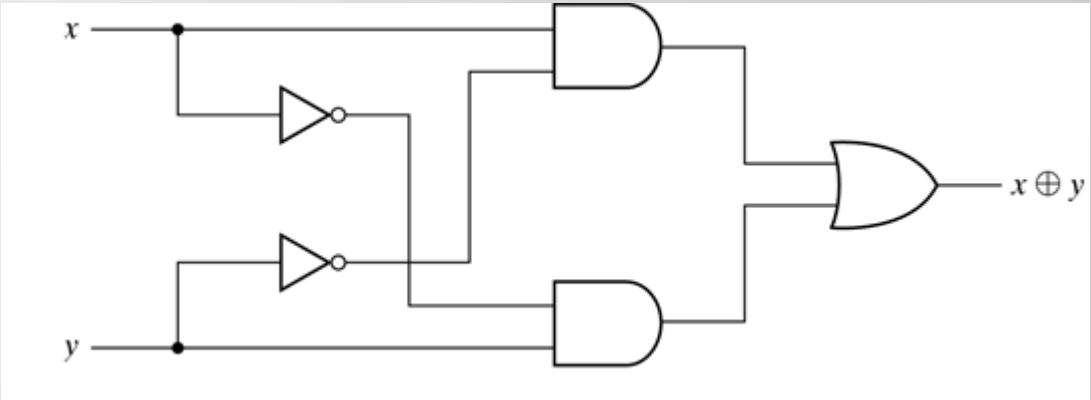


Με πύλες NAND :

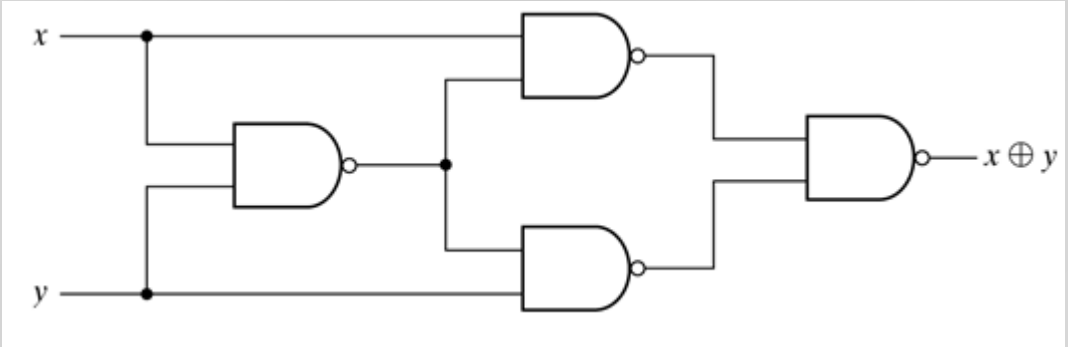


Κύκώμα XOR με 4 NAND (έμμεσος)

Με πύλες AND – OR – NOT :



Με πύλες NAND :



Η XOR

- Η πύλη XOR υλοποιεί μια «περιττή συνάρτηση».
- Η XOR τριών ή περισσότερων μεταβλητών είναι ίση με 1 αν μόνο μια μεταβλητή είναι ίση με 1 ή αν και οι τρεις μεταβλητές είναι ίσες με 1.
- Διαφορετικά: Για να έχει τιμή 1 περιττός αριθμός από μεταβλητές πρέπει να είναι ίσες με 1 → περιττή συνάρτηση.

Συνάρτηση X-NOR

- XNOR: η συνάρτηση ισότητας
- XNOR(a, b) = ab + a′b′
- Παρατηρήστε ότι

  XNOR(a, b) = (XOR(a, b))′

  – (a ⊕ b) = (a′b + ab′) ′

  = (a′b)′(ab′)′

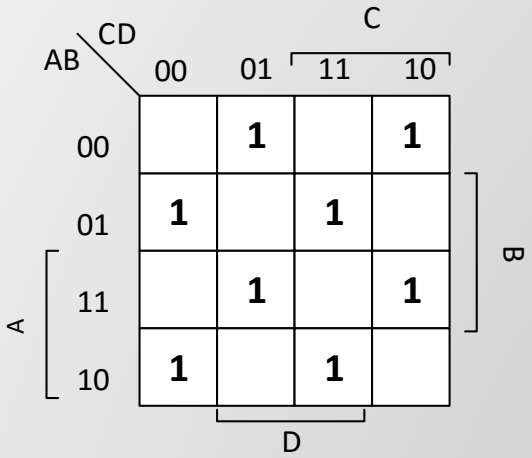
  = (a + b′)(a′ + b)

  = ab + a′b′
- a ⊕ b′ = (a ⊕ b)′ = a′ ⊕ b

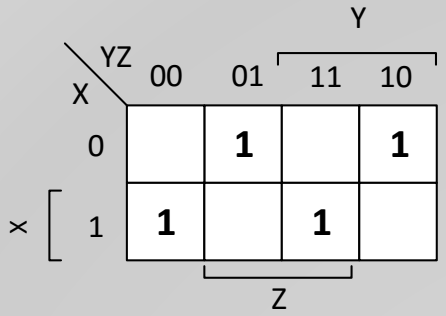
Περιττή συνάρτηση (odd function)(1)

- $x \oplus y = x'y + xy'$
- $x \oplus y \oplus z = xy'z' + x'yz' + x'y'z + xyz$
- $x \oplus y \oplus z \oplus w = x'yzw + xy'zw + xyz'w + xyzw' + x'y'z'w + x'yz'w + x'y'zw + xy'z'w'$
- Παρατηρείτε κάτι που επαναλαμβάνεται εδώ:
- Μια συνάρτηση XOR n-εισόδων είναι αληθής (=1) για όλους τους ελαχιστόρους που έχουν περιττό αριθμό από 1.
- Γι αυτό το XOR είναι γνωστό ως «η περιττή συνάρτηση».

Περιττή συνάρτηση (odd function)(2)



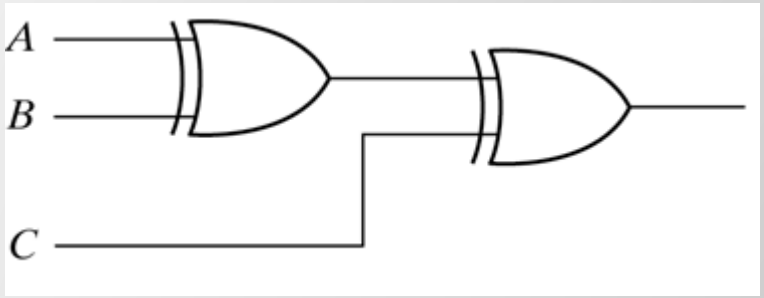
(a) X ⊕ Y ⊕ Z



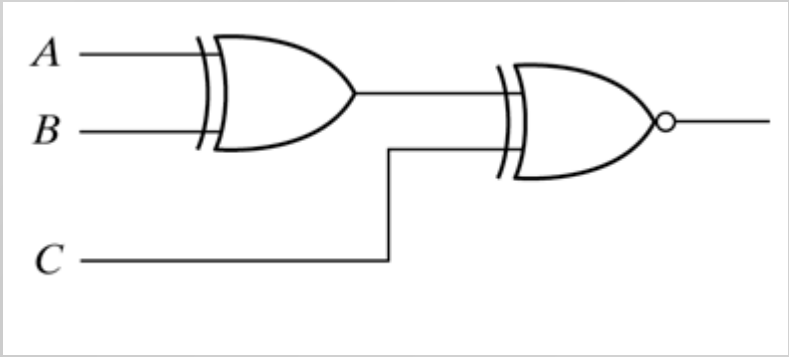
(b) A ⊕ B ⊕ C ⊕ D

Οι ελαχιστόροι απέχουν 2 τετράγωνα ο ένας από τον άλλον.

Περιττή συνάρτηση 3 εισόδων



(a) 3 είσοδοι με περιττή συνάρτηση



(b) 3 είσοδοι με άρτια συνάρτηση

Άρτια συνάρτηση

- Πώς θα υλοποιούσατε μια άρτια συνάρτηση;

Από το συμπλήρωμα του XOR → XNOR.

Δημιουργία και έλεγχος ισοτιμίας

- Οι περιττές και άρτιες συναρτήσεις μπορούν να χρησιμοποιηθούν για την υλοποίηση κυκλωμάτων ελέγχου ισοτιμίας (parity check) που χρησιμοποιούνται για εξεύρεση λαθών και τη διόρθωση τους.
- Γεννήτρια Ισοτιμίας (Parity Generator):

  Το κύκλωμα που παράγει το bit ισοτιμίας, πριν τη μετάδοση από τον αποστολέα.
- Έλεγχος Ισοτιμίας (Parity Check):

  Το κύκλωμα που ελέγχει την ισοτιμία στον παραλήπτη, για εξεύρεση λαθών.

Παραγωγή Άρτιας Ισοτιμίας

- Η P(X, Y, Z) πρέπει να παράγει 1 για κάθε συνδυασμό εισόδων που περιέχει περιττό αριθμό από 1.
- Είναι μια περιττή συνάρτηση $3^{ων}$-εισόδων $P = X \oplus Y \oplus Z$.

X	Y	Z	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

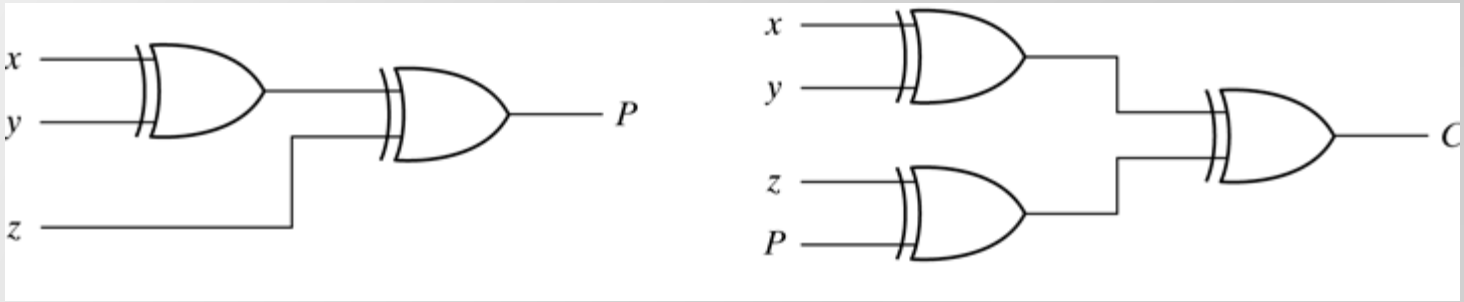
Έλεγχος Άρτιας Ισοτιμίας

- Πως θα υλοποιούσατε τον έλεγχο ισοτιμίας για το προηγούμενο παράδειγμα;

Α) Χρησιμοποιήστε ένα κύκλωμα XOR 4^{ων}-εισόδων (περιττή συνάρτηση) C = X $\oplus$ Y $\oplus$ Z $\oplus$ P $\rightarrow$ 1 υποδεικνύει ένα λάθος ή

Β) Χρησιμοποιείστε ένα XNOR κύκλωμα 4^{ων}-εισόδων (άρτια συνάρτηση) C = (X $\oplus$ Y $\oplus$ Z $\oplus$ P)΄ $\rightarrow$ 1 υποδεικνύει ορθή ισοτιμία.

Κυκλώματα γεννήτριας και ελεγκτής άρτιας ισοτιμίας 3 bit



(a) 3-bit άρτιας γεννήτριας ισοτιμίας (b) 4-bit άρτιος ελεγκτής ισοτιμίας

- Αν C = 1 τότε υπάρχει λάθος.

  Η γεννήτρια ισοτιμίας μπορεί να γίνει από το (β) αν συνδεθεί το P σταθερά στο 0.

Ισχύει: z XOR 0 = z

Ολοκληρωμένα κυκλώματα (IC)

- Ένα chip που περιέχει όλα τα ηλεκτρονικά στοιχεία για την κατασκευή ψηφιακών πυλών. Οι πύλες είναι συνδεδεμένες έτσι ώστε να σχηματίσουν το επιθυμητό κύκλωμα.
- Υπάρχουν ακροδέκτες εισόδου/εξόδου δεδομένων και τροφοδοσίας.
- Κάθε ολοκληρωμένο κύκλωμα (Integrated Circuit) έχει μια αριθμητική ονομασία.

Επίπεδα ολοκλήρωσης

- Κατηγοριοποίηση ανάλογα με την πολυπλοκότητα:
  - SSI (Small Scale Integration, μικρής κλίμακας ολοκλήρωση) μόνο μερικές πύλες (μικρότερες από 10).
  - MSI (Medium S.I. μεσαίας κλίμακας ολοκλήρωση) 10 εώς 10.000 πύλες (π.χ. αθροιστές αποκωδικοποιητές).
  - LSI (Large S.I. μεγάλης κλίμακας ολοκλήρωση) χιλιάδες πύλες (π.χ. επεξεργαστές, chip μνήμης).
  - VLSI (Very Large S.I. πολύ μεγάλης κλίμακας ολοκλήρωση) Εκατοντάδες χιλιάδες πύλες (π.χ. σύγχρονοι επεξεργαστές).

Οικογένειες Ψηφιακής Λογικής

- Κατηγοριοποίηση ανάλογα με την οικογένεια με την οποία είναι κατασκευασμένα:
  - TTL (Transistor – Transistor Logic)
  - ECL (Emitter – Coupled Logic)
  - MOS (Metal – Oxide – Semiconductor)
  - CMOS (Complementary MOS)

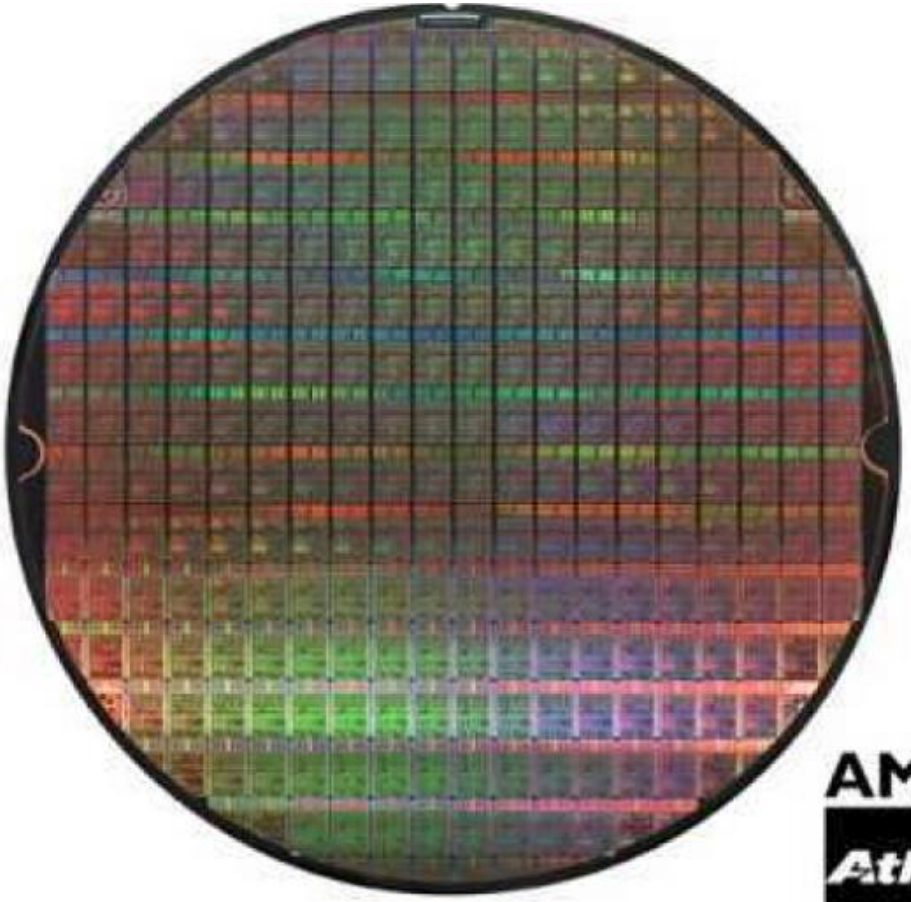
CMOS, η πιο γνωστή οικογένεια

- Η επικρατέστερη οικογένεια λογικής είναι η CMOS, γιατί μας δημιουργεί κυκλώματα με χαμηλή κατανάλωση ισχύος.
- TTL, ECL δε χρησιμοποιουνται πια όσο στο παρελθόν.

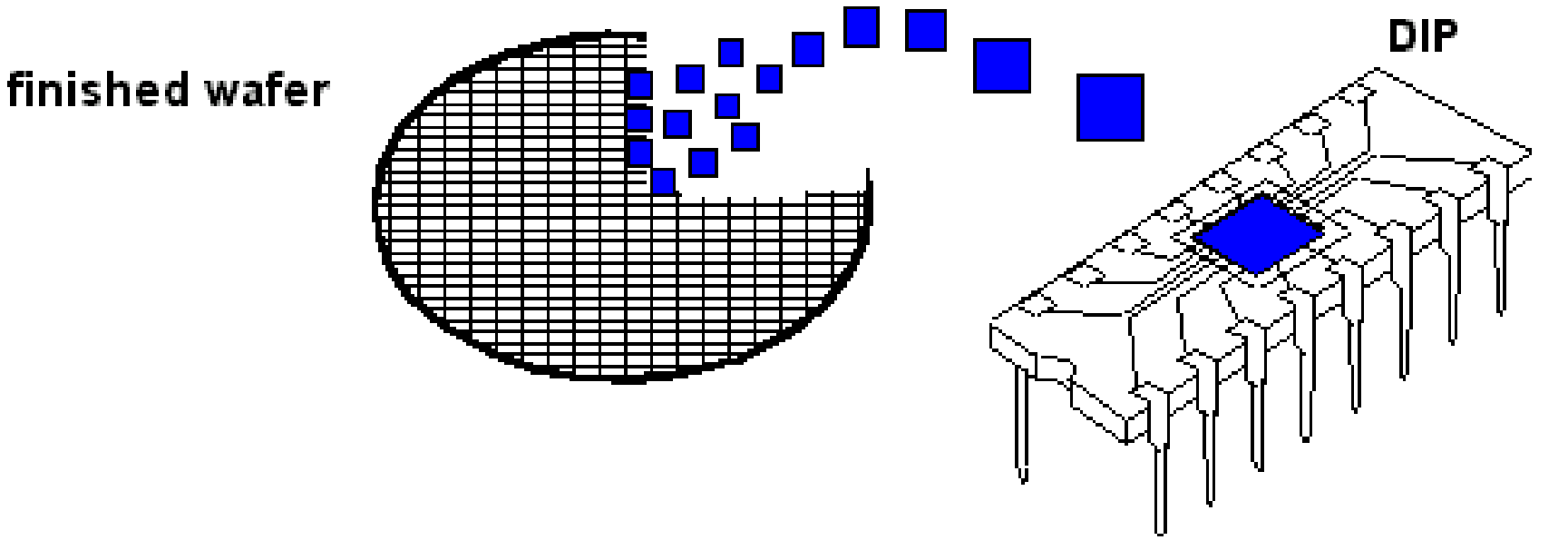
Παράμετροι Σύγκρισης Οικογενειων Ψηφιακών IC

- Fan-out (βαθμός οδήγησης εξόδου): πόσα φορτία μπορεί να αντέξει στην έξοδο μια πύλη.
- Fan-In (βαθμός οδήγησης εισόδου): αριθμός εισόδων μιας πύλης.
- Power dissipation (ισχύς κατανάλωσης).
- Propagation delay (καθυστέρηση διάδοσης): μέσος χρόνος που απαιτείται για να μεταδοθεί το σήμα από την είσοδο στην έξοδο.
- Noise margin (περιθώριο θορύβου): μέγιστη τάση εξωτερικού θορύβου που μπορεί να προστεθεί σε ένα σήμα χωρίς να προκληθεί ανεπιθύμητη αλλαγή στην έξοδο.

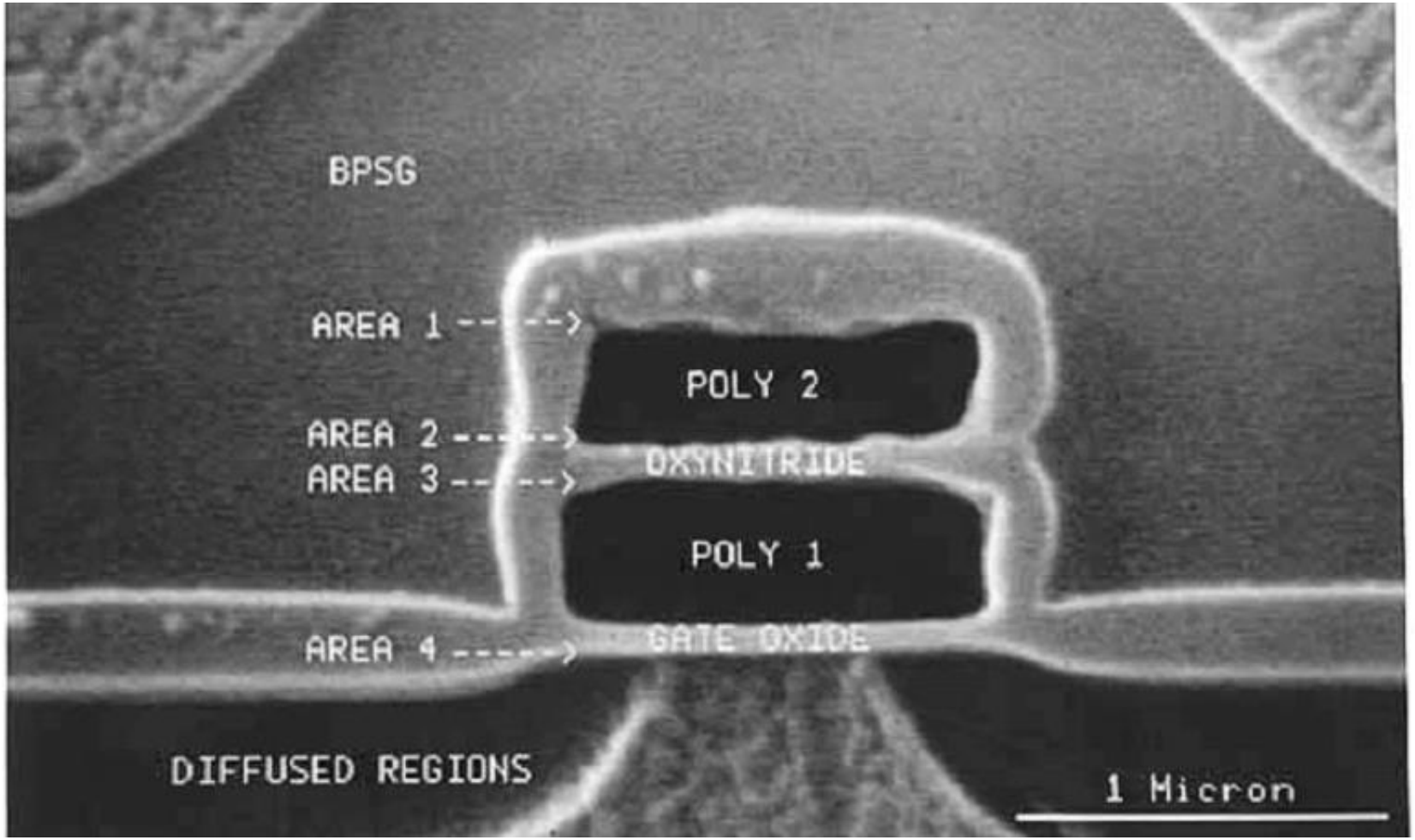
Κατασκευή IC από WAFERS(1)



Κατασκευή IC από WAFERS(2)



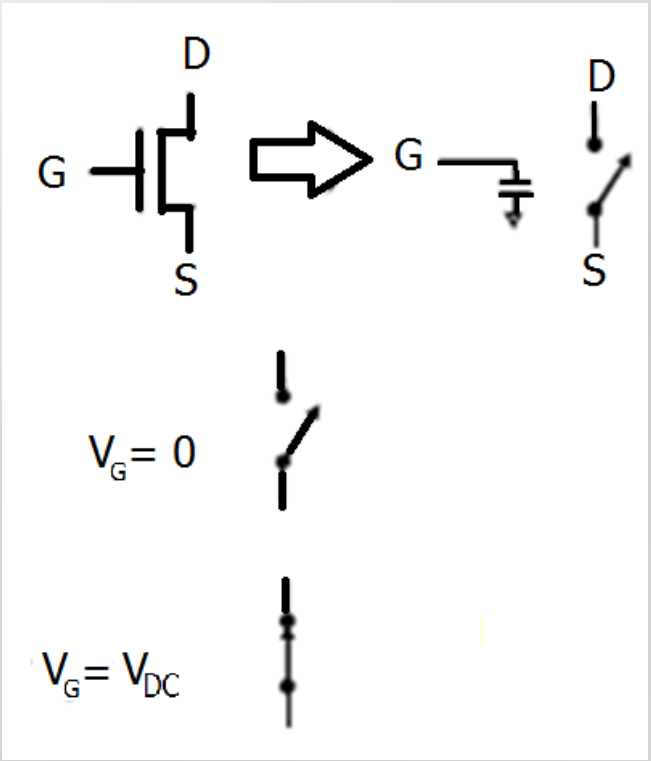
Όλα δημιουργούνται από το transistor



Το transistor είναι ένας διακόπτης

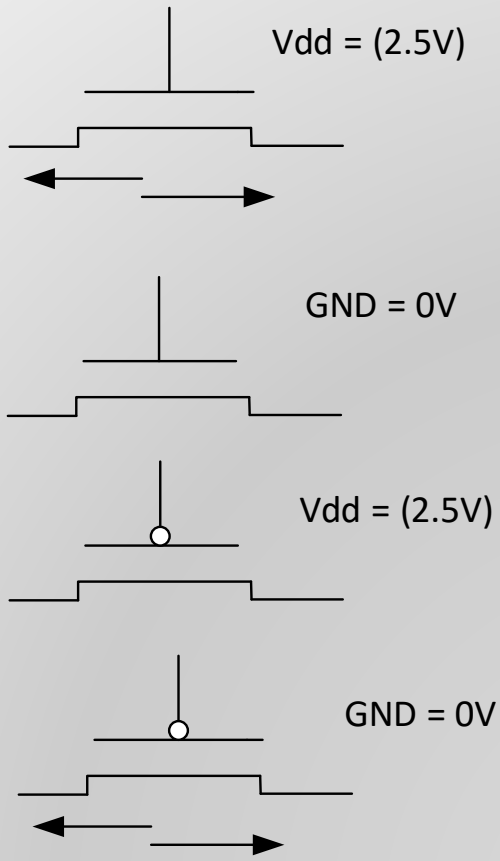
- Ιδανικός διακόπτης ελεγχόμενης τάσης.

- Τρία τερματικά
  - Πύλη (gate)
  - Σωλήνας (drain)
  - Πηγή (source)



Τεχνολογία CMOS

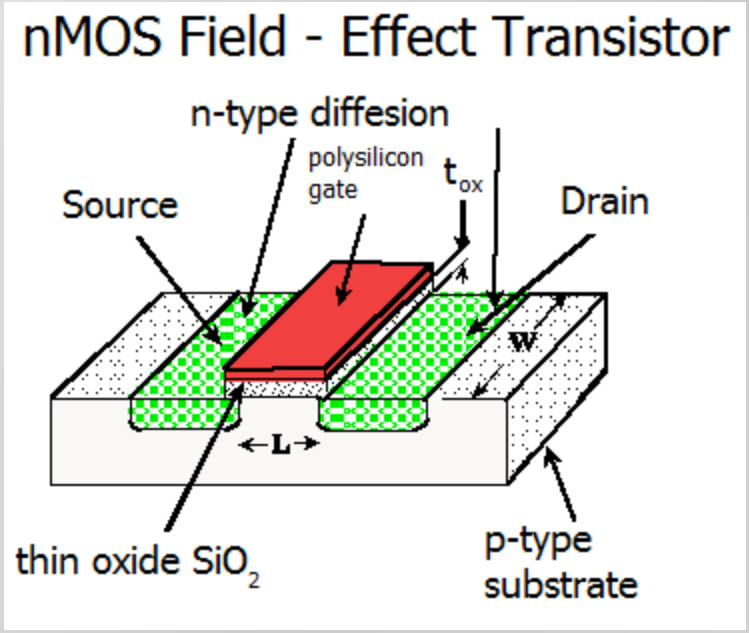
- CMOS: Συμπληρωματικός ημιαγωγός μεταλλικού οξειδίου
  - NMOS (N-τύπου ημιαγωγός μεταλλικού οξειδίου) transistors.
  - PMOS (P-τύπου ημιαγωγός μεταλλικού οξειδίου) transistors.
- NMOS Transistor
  - Η εφαρμογή ΥΨΗΛΟΥ (Vdd) στην πύλη μετατρέπει το τρανζίστορ σε «αγωγό»
  - Η εφαρμογή ΧΑΜΗΛΟΥ (GND) στην πύλη διακόπτει την διαδρομή αγωγής.
- PMOS Transistor
  - Η εφαρμογή ΥΨΗΛΟΥ (Vdd) στην πύλη διακόπτει την διαδρομή αγωγής
  - Η εφαρμογή ΧΑΜΗΛΟΥ (GND) στην πύλη μετατρέπει το τρανζίστορ σε «αγωγό».



Κυκλώματα CMOS

- Υλοποίηση λογικών πυλών και άλλων δομών χρησιμοποιώντας τεχνολογία CMOS.
- Βασικό στοιχείο: transistor:
  - n-κανάλι (n-channel): transistor nMOS
  - p-κανάλι (p-channel): transistor pMOS
  - Ο τύπος εξαρτάται από τα υλικά του ημιαγωγού που χρησιμοποιήθηκαν για την υλοποίηση του transistor.

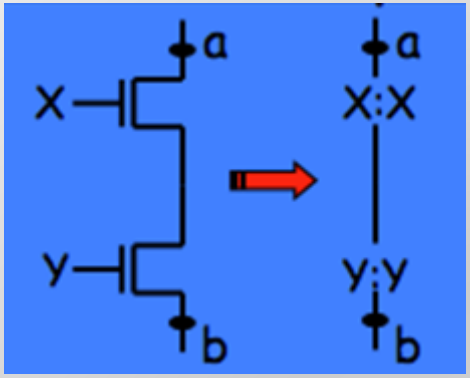
nMOS


nMOS Field - Effect Transistor
n-type diffesion
polysilicon gate
t_{ox}
Source
Drain
W
L
thin oxide SiO_2
p-type substrate

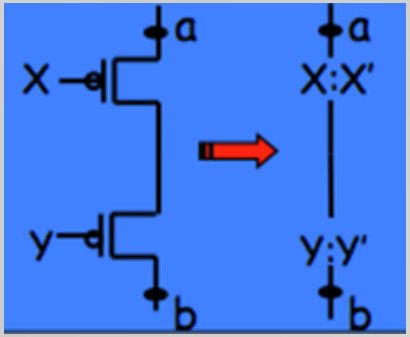
Τρανζίστορ σε Σειρά

- nMOS σε σειρά:
  - Υπάρχει μονοπάτι μεταξύ των σημείων a και b εάν X και Y είναι 1→ X • Y
- pMOS σε σειρά:
  - Υπάρχει μονοπάτι μεταξύ των σημείων a και b εάν X και Y είναι 0→ X′ • Y′

nMOS σε σειρά:



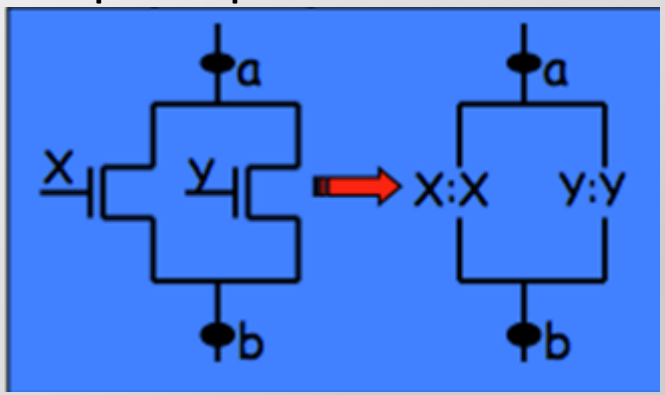
pMOS σε σειρά:



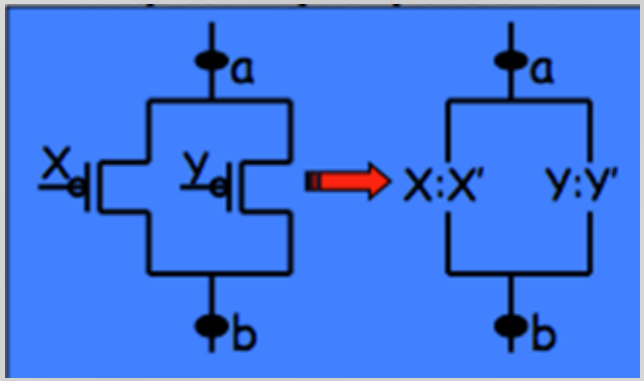
Τρανζίστορ Παράλληλα

- Παράλληλο nMOS:
  - Υπάρχει μονοπάτι μεταξύ των σημείων a και b εάν X και Y είναι 1→ X + Y
- Παράλληλο pMOS:
  - Υπάρχει μονοπάτι μεταξύ των σημείων a και b εάν X και Y είναι 0→ X′ + Y′

- Παράλληλο nMOS:

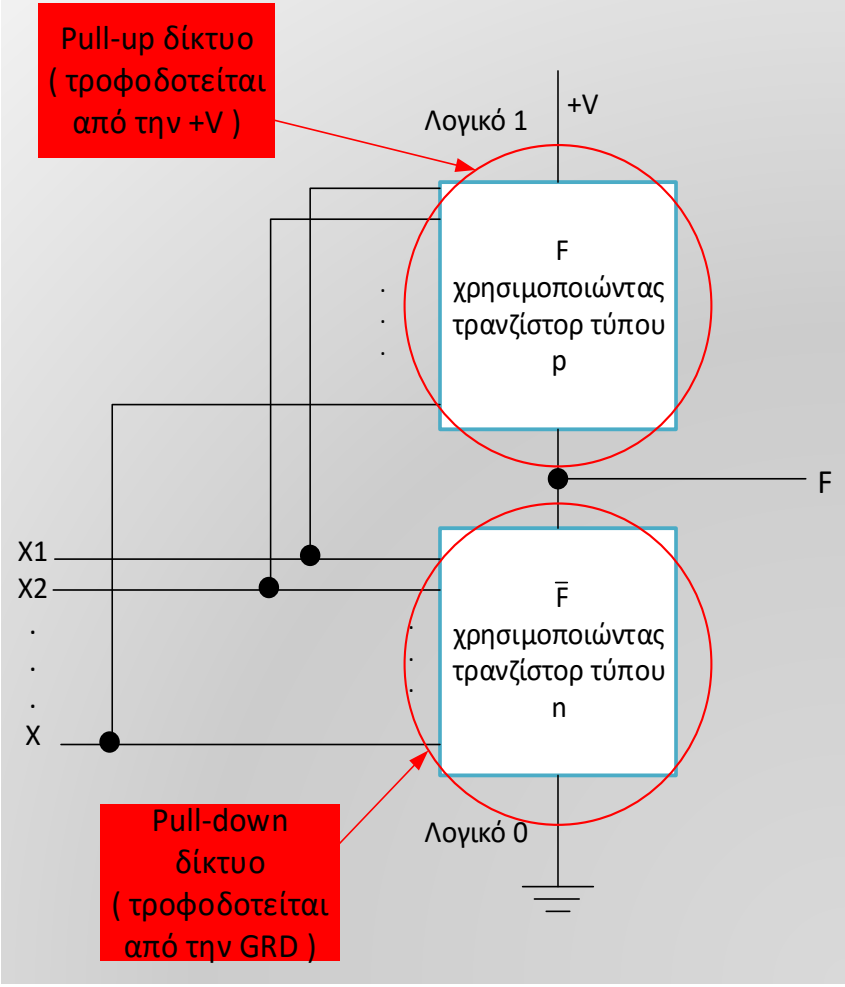


- Παράλληλο pMOS:

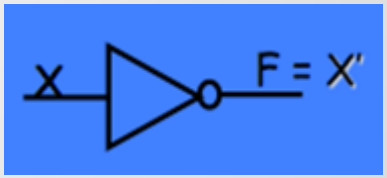


Η CMOS χρησιμοποιεί pMOS & nMOS

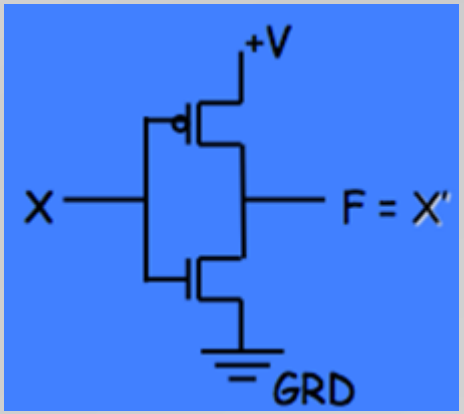
- Κάθε πλήρως συμπληρωματικό δίκτυο CMOS ακολουθεί τη δομή στα δεξιά.
- Το κάθε ένα απο τα δύο υπο-δίκτυα υλοποιεί τη συνάρτηση δυϊσμού του άλλου.
- Στατική CMOS (Static CMOS): υλοποιεί την F( ) (όλους τους συνδυασμούς που δίνουν 1) και το συμπλήρωμα της F΄( ) (όλους τους συνδυασμούς που δίνουν 0).
- Υπάρχει πάντα ένα μονοπάτι που οδηγεί στην έξοδο (F), είτε από την πηγή +V (λογικό 1) είτε από την γείωση (λογικό 0).
- Γιατι;



Θεμελιώδες στοιχείο CMOS: Ο αντιστροφέας



Λογικό σύμβολο

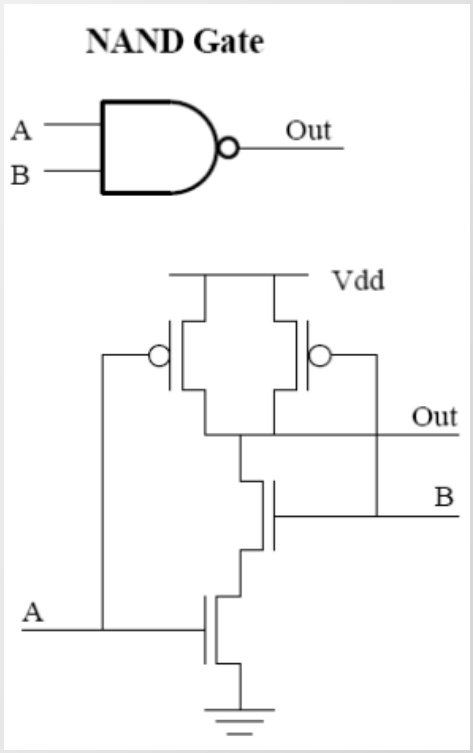


Σχηματικό σε επίπεδο τρανζίστορ

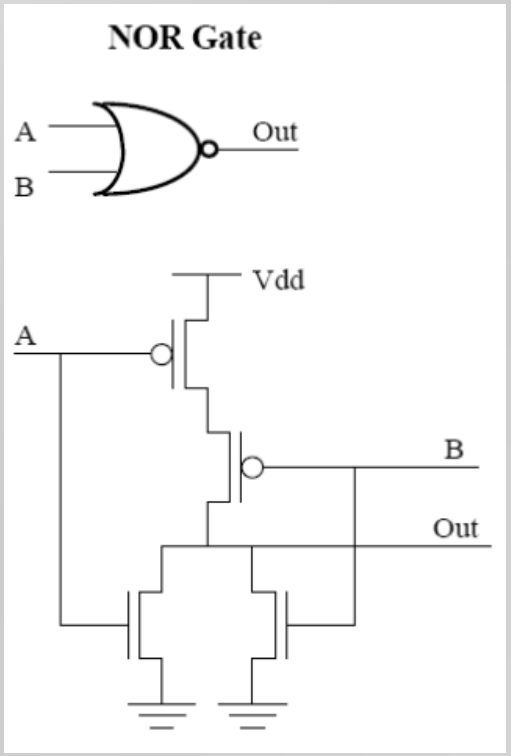
Λειτουργία:

- X = 1 → ο διακόπτης nMOS κλείνει (pMOS παραμένει ανοικτός) και η έξοδος άγει από το GRD → F = 0
- X = 0 → ο διακόπτης pMOS κλείνει (nMOS παραμένει ανοικτός) και η έξοδος άγει από το +V → F = 1

Θεμελιώδη στοιχεία CMOS: NAND, NOR



A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

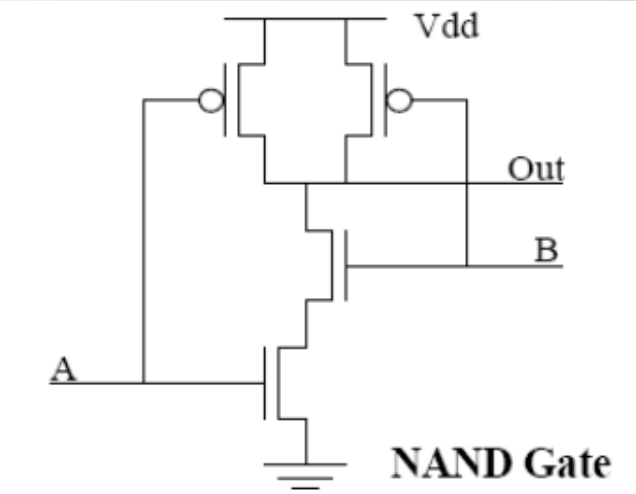


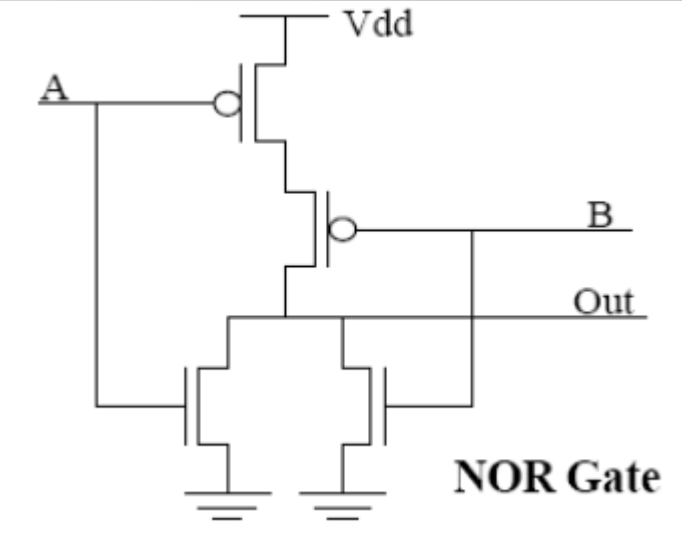
A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Ταυτόχρονη χρήση pMOS και nMOS

- Γιατί τα δίκτυα pMOS είναι συνδεδεμένα στο +V και τα nMOS στο GRD?
  - Τα στοιχεία pMOS είναι *σχεδόν ιδανικά* όταν τα διαπερνά υψηλή τάση (H) και *αδύνατα* όταν τα διαπερνά χαμηλή τάση (L).
  - Τα στοιχεία nMOS είναι *σχεδόν ιδανικά* όταν τα διαπερνά χαμηλή τάση και *αδύνατα* όταν τα διαπερνά υψηλή τάση (H).
  - Η δομή του CMOS εξασφαλίζει την παραμονή των τιμών των διαφόρων σημάτων στα κατάλληλα υψηλά και χαμηλά λογικά επίπεδα, όταν μεταδίδονται δια μέσω του δικτύου και φθάνουν στην έξοδο.

Σύγκριση Πυλών: NAND vs NOR

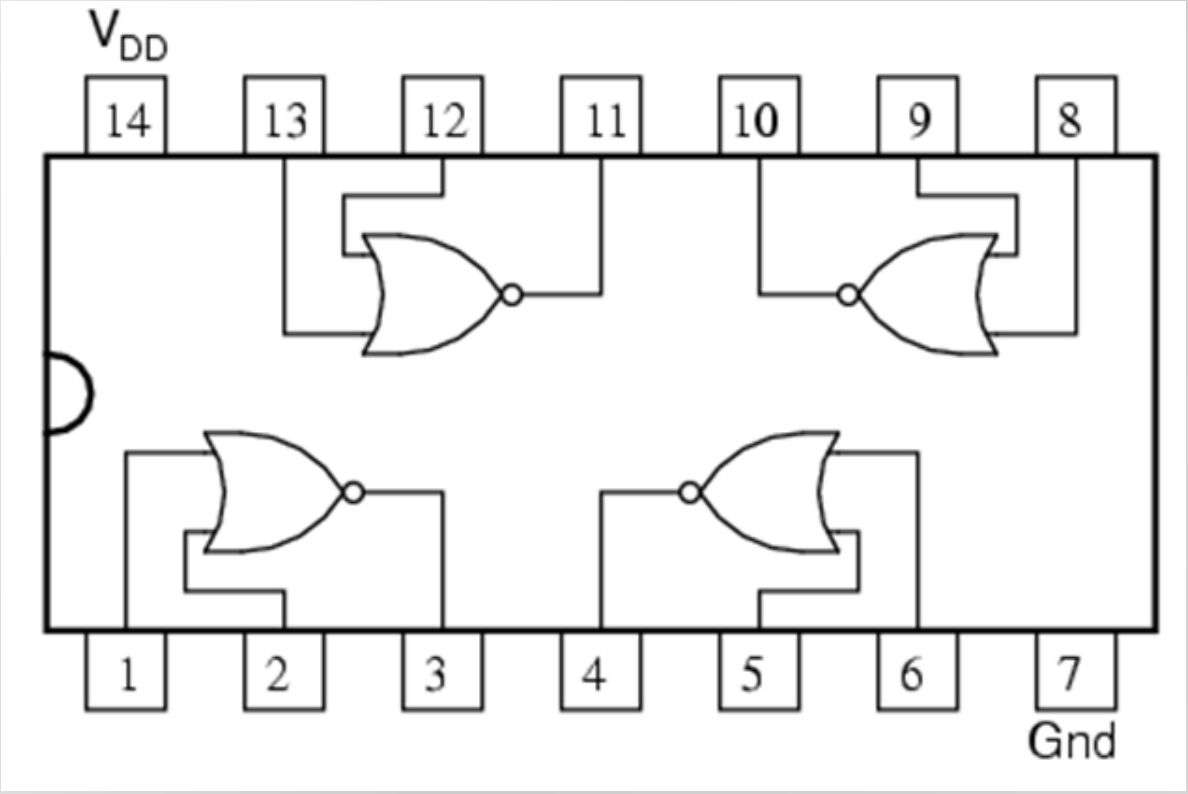




- Αν τα PMOS τρανζίστορ είναι ταχύτερα:
  - Είναι εντάξει να έχουμε PMOS τρανζίστορ σε σειρά.
  - Προτιμάται η πύλη NOR.
  - Η πύλη NOR προτιμάται επίσης αν H -> L είναι πιο κρίσιμη από L -> H.
- Αν τα NMOS τρανζίστορ είναι ταχύτερα:
  - Είναι εντάξει να έχουμε NMOS τρανζίστορ σε σειρά.
  - Προτιμάται η πύλη NAND.
  - Η πύλη NAND προτιμάται επίσης αν L -> H είναι πιο κρίσιμη από H -> L.

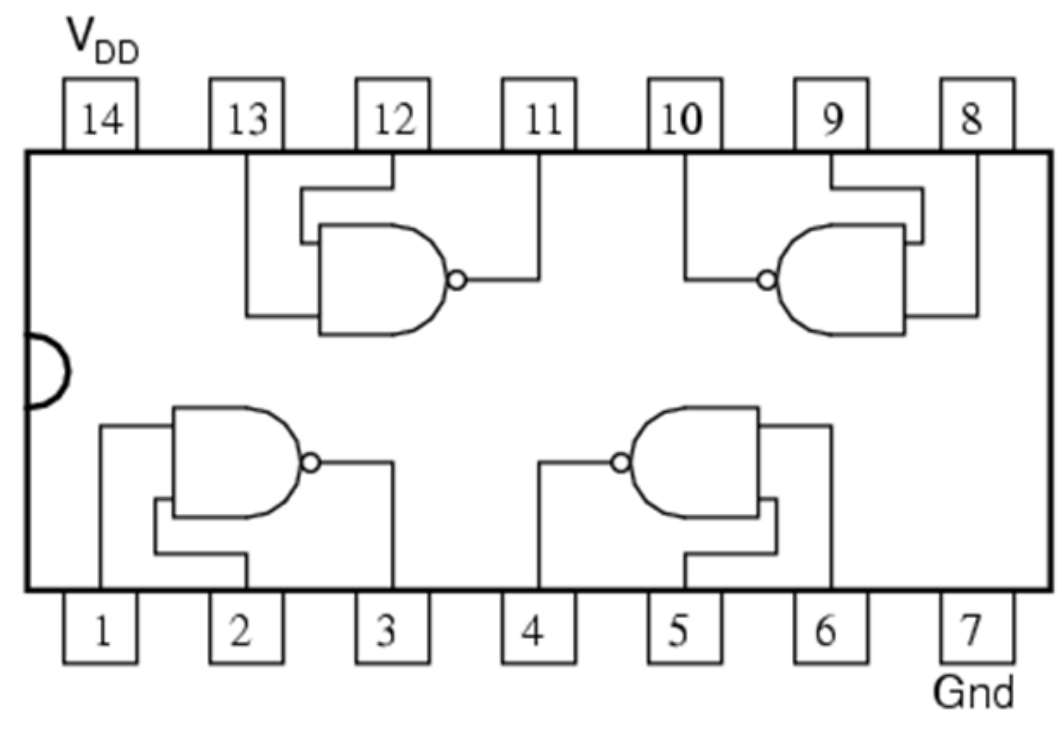
Συσκευασία NOR

Διάγραμμα σύνδεσμολογίας ή " Pinout " για την 4011 quad NOR πύλη.



Συσκευασία NAND

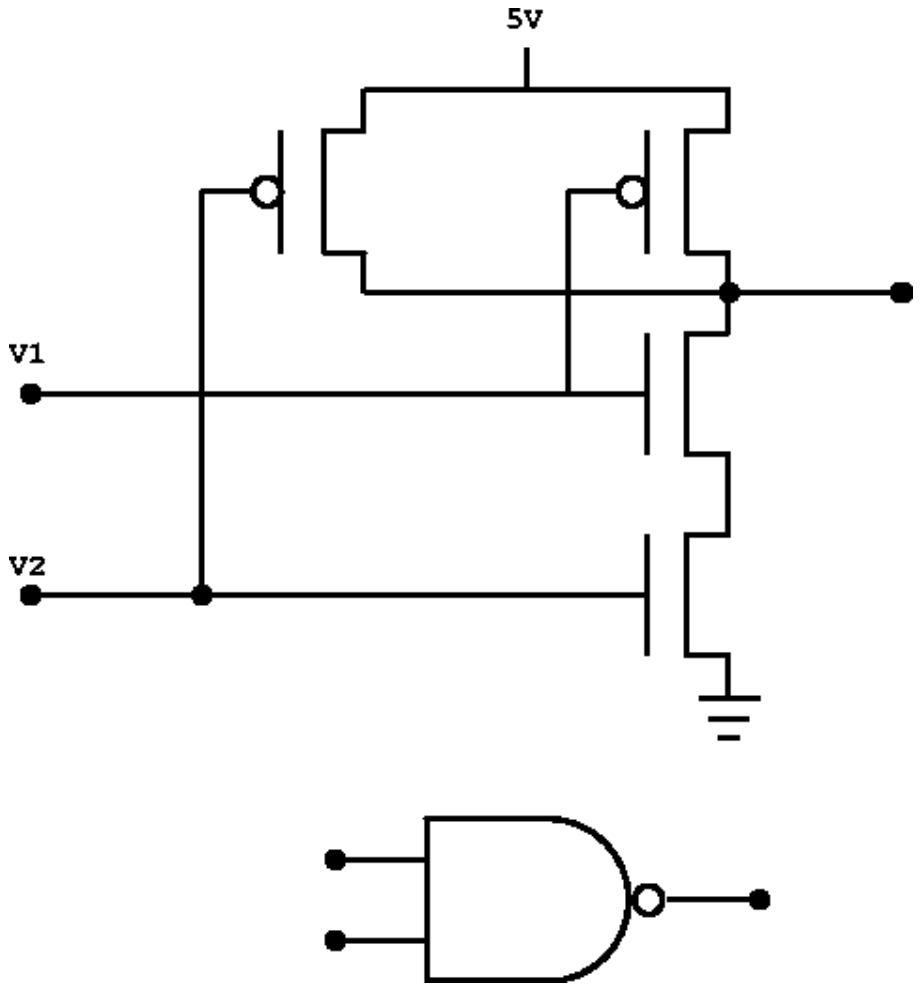
Διάγραμμα σύνδεσμολογίας ή " Pinout " για την 4011 quad NAND πύλη.



Γρηγοροτέρο το pMOS ή το nMOS;

- nMOS πιο γρήγορο transistor
- στο nMOS μετακινούνται ηλεκτρόνια με μεγαλύτερη ταχύτητα από ότι οι θετικές οπές στο pMOS.
- Η διαρροή φορτίου στη NAND είναι πιο μικρή από ότι στη NOR.
- Οι πύλες NAND μπορούν να οδηγήσουν (δηλ. να συνδεθούν σε) περισσότερες πύλες από ότι οι πύλες NOR.

Και ο νικητής είναι η… NAND



Τέλος Ενότητας