Πρόγραμμα Επικαιροποίησης Γνώσεων Αποφοίτων

ΕΝΟΤΗΤΑ Μ1
ΨΗΦΙΑΚΑ ΗΛΕΚΤΡΟΝΙΚΑ

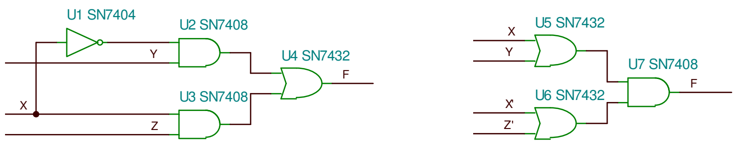
Εκπαιδευτής: Γ. Π. ΠΑΤΣΗΣ, Επικ. Καθηγητής, Τμήμα Ηλεκτρονικών Μηχανικών, ΤΕΙ Αθήνας

ΚΑΘΟΛΙΚΕΣ ΠΥΛΕΣ NAND – NOR – ΑΛΓΕΒΡΑ BOOLE – ΘΕΩΡΗΜΑ DE MORGAN

1. Διεπίπεδες υλοποιήσεις
2. Η καθολικότητα των πυλών NAND και NOR
3. Η πύλη XOR και η πύλη EQUIVALENCE
4. Αξιώματα – Θεωρήματα Άλγεβρας Boole
5. Ασκήσεις

Διεπίπεδες υλοποιήσεις

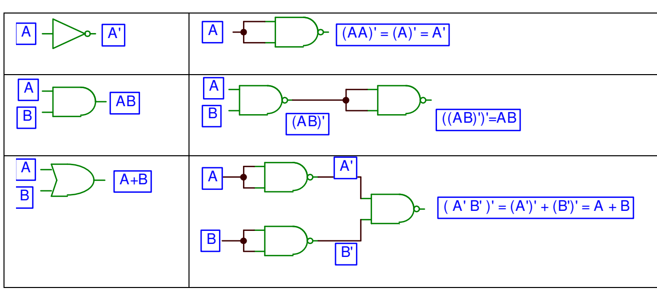
1. Με δεδομένη τη συνάρτηση λογικής ενός κυκλώματος, μπορούμε να το υλοποιήσουμε με AND-OR ή OR – AND τεχνική.
2. Για παράδειγμα, η συνάρτηση P = X'Y+XZ = (X+Y)(X'+Z). Η πρώτη μορφή είναι εκφρασμένη σε άθροισμα γινομένων και υλοποιείται εύκολα με AND-OR τεχνική, ενώ η δεύτερη μορφή είναι εκφρασμένη σε γινόμενο αθροισμάτων και υλοποιείται εύκολα με OR-AND τεχνική. Οι δύο αυτές ισοδύναμες υλοποιήσεις φαίνονται στη συνέχεια (**Εικόνα 1**).
3. Τέτοιες υλοποιήσεις απλών κυκλωμάτων, ονομάζονται δι-επίπεδες εξαιτίας του ενός επιπέδου με πύλες AND (OR) και του 2ου επιπέδου με πύλες OR (AND).



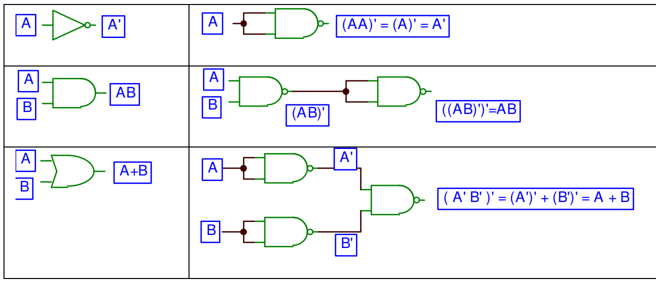
Εικόνα 1. Ισοδύναμες AND-OR και OR-AND υλοποιήσεις.

Η καθολικότητα των πυλών NAND και NOR

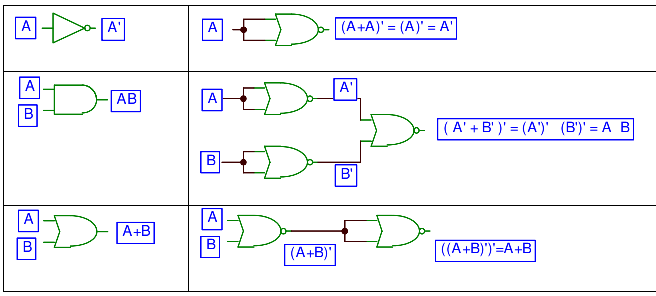
1. Κάθε ψηφιακό κύκλωμα που μπορεί να εκφραστεί με κανονική μορφή λογικής συνάρτησης μπορεί να υλοποιηθεί με πύλες NOT, AND, OR.
2. Καθεμία από αυτές τις πύλες μπορεί να υλοποιηθεί μόνο με πύλες NAND ή μόνο με πύλες NOR.
3. Συνεπώς κάθε ψηφιακό κύκλωμα μπορεί να υλοποιηθεί μόνο με πύλες NAND ή μόνο με πύλες NOR. Για το λόγο αυτό οι πύλες αυτές λέγονται καθολικές.
4. Στη συνέχεια δείχνουμε την ισοδυναμία των πυλών NOT, AND, OR μόνο πύλες NAND ή μόνο με πυλες NOR.



Εικόνα 1. Η καθολικότητα της NAND.



Εικόνα 1. Η καθολικότητα της NAND.

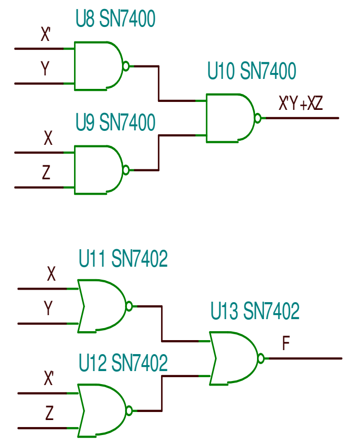


Εικόνα 1. Η καθολικότητα της NOR.

Παραδείγματα

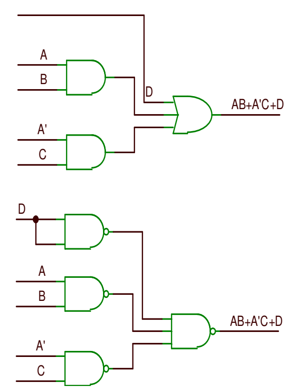
Εφαρμογή στην τάξη:
Επαλήθευση καθολικότητας πυλών NAND και NOR στο TINA…

1. Αποδεικνύεται ότι κάθε διεπίπεδη υλοποίηση στην οποία τα σήματα εισόδου συνδέονται όλα μόνο στο πρώτο επίπεδο πυλών, είναι ισοδύναμο με το ίδιο κύκλωμα όπου όλες οι πύλες ανεξαιρέτως έχουν αντικατασταθεί με πύλες NAND ή μόνο με πύλες NOR. Π.χ. στην **Εικόνα 4** φαίνονται δύο ισοδύναμα κυκλώματα, έναν με με NAND υλοποίηση και ένα με NOR, για το κύκλωμα της F= X’Y+XZ.
2. **Η διαδικασία μετατροπής των υλοποιήσεων AND-OR και OR-AND σε NAND – NAND και NOR – NOR αντίστοιχα, μπορεί να εφαρμοστεί σε όλες τις 2-επιπεδες υλοποιήσεις εφόσον δεν υπάρχει απευθείας είσοδος στο 2ο επίπεδο πυλών !!**



Εικόνα 4. Δύο ισοδύναμα κυκλώματα, a) ένα με με NAND υλοποίηση και b) ένα με NOR.

1. **Διαφορετικά, η είσοδος που εισάγεται απευθείας στο 2ο επίπεδο πυλών, θα πρέπει να αντιστραφεί πριν εισαχθεί στο 2ο επίπεδο της NAND – NAND ή NOR – NOR υλοποίησης. Μια τέτοια περίπτωση φαίνεται στην Εικόνα 5 για την είσοδο D.**
2. Αν το κύκλωμα που θέλουμε να μετασχηματίσουμε αποτελείται από περισσότερα από 2 επίπεδα πυλών, η προηγούμενη διαδικασία δεν ισχύει !!!
3. Κάθε πύλη στο κύκλωμα πρέπει να αντικατασταθεί από το αντίστοιχο ισοδύναμο NAND ή NOR κύκλωμα. Στη συνέχεια το κύκλωμα θα πρέπει να αναλυθεί παραπέρα για να απομακρύνουμε τις πλεονάζουσες πύλες.
4. Οι πύλες NAND και NOR είναι συμπληρωματικές. Η συνάρτηση NOR είναι το δυϊκό της συνάρτησης NAND. Δηλαδή **NAND‘ = NOR** και **NOR ‘ = NAND**. Για το λόγο αυτό, όλες οι διαδικασίες και οι κανόνες για τα κυκλώματα με πύλες NOR είναι τα δυϊκά των αντίστοιχων για τα κυκλώματα NAND.

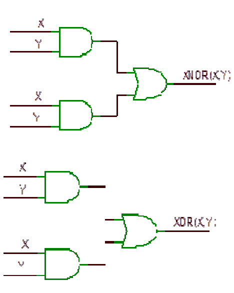


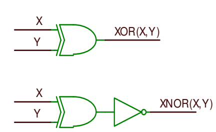
Εικόνα 5. a) Η είσοδος που εισάγεται απευθείας στο 2ο επίπεδο πυλών, θα πρέπει να b) αντιστραφεί πριν εισαχθεί στο 2ο επίπεδο της NAND – NAND ή NOR – NOR υλοποίησης.

1. Τα ψηφιακά κυκλώματα κατασκευάζονται πολύ συχνότερα με πύλες NAND, ή NOR παρά με πύλες NOT, AND, OR.
2. Οι πύλες NAND και NOR κατασκευάζονται ευκολότερα με ηλεκτρονικά κυκλώματα και είναι οι βασικές πύλες που χρησιμοποιούνται σε όλες τις οικογένειες ψηφιακών ολοκληρωμένων κυκλωμάτων.
3. Για την υλοποίηση μιας συνάρτησης Boole με πύλες NAND χρειαζόμαστε τη συνάρτηση απλοποιημενη σε μορφή αθροίσματος γινομένων. Ενώ για την υλοποιήσή της με πύλες NOR, χρειαζόμαστε τη συνάρτηση απλοποιημένη σε μορφή γινομένου αθροισμάτων.
4. Ο κανόνας για να βρίσκουμε το κύκλωμα μιας συνάρτησης Boole με πύλες NAND είναι ο εξής:
5. 1.Απλοποιούμε τη συνάρτηση και την εκφράζουμε ως άθροισμα γινομένων.
6. 2.Σχεδιάζουμε μια πύλη NAND για κάθε όρο γινομένου της συνάρτησης που περιέχει τουλάχιστον δύο παράγοντες. Οι είσοδοι κάθε τέτοιας πύλης είναι οι παράγοντες των όρων. Αυτές είναι οι πύλες του πρώτου επιπέδου.
7. 3.Σχεδιάζουμε μια πύλη NAND στο δεύτερο επίπεδο, με εισόδους που τροφοδοτούνται από τις εξόδους του πρώτου επιπέδου.
8. 4.Ένας όρος με έναν μόνο παράγοντα χρειάζεται έναν αντιστροφέα στο πρώτο επίπεδο ή αν έχουμε ήδη το συμπλήρωμα του μπορούμε να το τροφοδοτήσουμε κατευθείαν σε μια είσοδο πύλης NAND δεύτερου επιπέδου.
9. Η υλοποίηση συναρτήσεων Boole με πύλες NOR απαιτεί αυτές να έχουν απλοποιηθεί σε γινόμενο αθροισμάτων. Μια τέτοια έκφραση μεταφράζεται σε μια ομάδα πυλών OR για τους όρους αθροίσματος, ακολουθούμενη από μια πύλη AND για το γινόμενο.
10. Ο κανόνας για την υλοποίηση συναρτήσεων Boole με πύλες NOR είναι παρόμοιος με τον κανόνα για την υλοποίηση με πύλες NAND (τρία-βήματα), μόνο που η απλοποιημένη έκφραση πρέπει να είναι σε μορφή γινομένου αθροισμάτων και οι όροι για τις πύλες NOR του πρώτου επιπέδου είναι τα αθροίσματα.
11. Οι όροι αθροίσματος που έχουν μόνον έναν όρο χρειάζονται πύλες NOR μιας εισόδου ή αντιστροφείς.
12. Ένας δεύτερος τρόπος για να υλοποιήσουμε μια συνάρτηση με πύλες NOR, είναι να χρησιμοποιήσουμε το συμπλήρωμα της συνάρτησης εκφρασμένο σαν γινόμενο αθροισμάτων. Αυτό δίνει μια υλοποίηση της F' σε δύο επίπεδα λογικής ή μιας υλοποίηση της F σε τρία επίπεδα.
13. Όπως θα δούμε στο επόμενο κεφάλαιο, για να πάρουμε το απλοποιημένο γινόμενο αθροισμάτων από το χάρτη Karnaugh, πρέπει να συνδυάσουμε τα μηδενικά του χάρτη και μετά να πάρουμε το συμπλήρωμα της συνάρτησης. Για να πάρουμε το απλοποιημένο γινόμενο αθροισμάτων για το συμπλήρωμα της συνάρτησης πρέπει να συνδυάσουμε τις μονάδες του χάρτη και να πάρουμε το συμπλήρωμα της συνάρτησης.

Η πύλη XOR και η πύλη EQUIVALENCE

1. Μια πύλη που χρησιμοποιείται συχνά στην κατασκευή κυκλωμάτων υπολογισμών είναι η XOR, ενώ στην κατασκευή κυκλωμάτων αναγνώρισης/διόρθωσης χρησιμοποιείται συχνά η EQUIVALENCE.
2. Αυτές οι δύο πύλες είναι συμπληρωματικές.
3. Ο πίνακας αληθείας τους, τα σύμβολά τους και η ισοδύναμες AND-OR υλοποιήσεις τους φαίνονται στην **Εικόνα 6**.

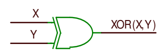


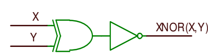


XY	$X \oplus Y$	$X \odot Y$
00	0	1
01	1	0
10	1	0
11	0	1

Εικόνα 6. Ο πίνακας αληθείας, τα σύμβολά και η ισοδύναμες AND-OR υλοποιήσεις των XOR και EQUIVALENCE.

1. Μια XOR n-εισόδων είναι ισοδύναμη προς μια Boolean συνάρτηση με 2^{n-1} ελαχιστόρους, των οποίων το δυαδικό ισοδύναμο έχει περιττό πλήθος 1.
2. Μια EQUIVALENCE n-εισόδων είναι ισοδύναμη προς μια Boolean συνάρτηση με 2^{n-1} ελαχιστόρους των οποίων το δυαδικό ισοδύναμο έχει άρτιο πλήθος 1.
3. Όταν το πλήθος των μεταβλητών της συνάρτησης είναι περιττός, οι ελαχιστόροι με άρτιο αριθμό 0 είναι όσοι με περιττό αριθμό 1. Τότε οι XOR και EXCLUSIVE είναι ίσες.
4. Όταν το πλήθος των μεταβλητών είναι άρτιο, αυτές οι δύο πύλες είναι συμπληρωματικές.





Εφαρμογή στην τάξη:
Επαλήθευση καθολικότητας πυλών NAND και NOR στο TINA...

Αξιώματα – Θεωρήματα Άλγεβρας Boole

1. Η άλγεβρα Boole είναι το μαθηματικό εργαλείο με το οποίο χειριζόμαστε τα ψηφιακά κυκλώματα.
2. Πρόκειται για μια αλγεβρική δομή που ορίζεται στο σύνολο B={0,1} μαζί με δύο δυαδικούς τελεστές + και *.

Τα **αξιώματα της άλγεβρας Boole** αναφέρονται στη συνέχεια:	Τα **θεωρήματα της άλγεβρας Boole** αναφέρονται στη συνέχεια:
1. Είναι κλειστή ως προς τον τελεστή + και τον τελεστή *. 2. Το στοιχείο 0 είναι το ουδέτερο στοιχείο ως προς την πρόσθεση: x+0=x. Το στοιχείο 1 είναι το ουδέτερο στοιχείο ως προς τον πολλαπλασιασμό: x*1=x. 3. Ισχύει η αντιμεταθετικότητα ως προς τον τελεστή +: x+y=y+x. Ισχύει η αντιμεταθετικότητα ως προς τον τελεστή *: x*y=y*x. 4. Ισχύει η επιμεριστικότητα του * ως προς τον + και το αντίστροφο: x*(y+z)=(x*y)+(x*z) και x+(y*z)=(x+y)*(x+z). 5. Για κάθε στοιχείο x στο B, υπάρχει το στοιχείο x’ του B (που ονομάζεται συμπλήρωμα του x), τέτοιο ώστε: x’ + x = 1 και x’ * x = 0. 6. Υπάρχουν τουλάχιστον 2 στοιχεία x,y στο B τέτοια ώστε x ≠ y.	1. x*x=x, x+x=x. 2. 1+x=1, 0*x=0. 3. x’’=x. 4. x*(y*z) = (x*y)*z, x+(y+z)=(x+y)+z. 5. x*(x+y)=x. 6. x+x*y=x. 7. x+x’*y=x+y. 8. Θεώρημα de Morgan: (x+y)’=x’*y’ , (x*y)’=x’+y’.