

## Ψηφιακή Λογική και Σχεδίαση (σχεδίαση συνδυαστικών κυκλωμάτων)

<http://di.ionio.gr/~mistralt/tp/comparch/>

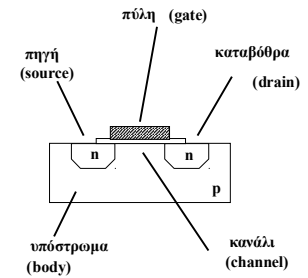


Μ.Στεφανιδάκης

## Το τρανζίστορ MOS(FET)

- Ηλεκτρονικά κυκλώματα

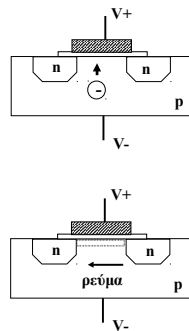
**i**  
Το τρανζίστορ αυτό είναι τύπου NMOS. Υπάρχει και το συμπληρωματικό PMOS.



## Λειτουργία του τρανζίστορ MOS(FET)

- Ηλεκτρονικά κυκλώματα

**;**  
Τι συμβαίνει στο τρανζίστορ PMOS;

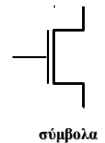


## Τρανζίστορ NMOS και PMOS

- Ηλεκτρονικά κυκλώματα

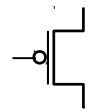
**i**  
Η πλειοψηφία των σημερινών κυκλωμάτων χρησιμοποιεί και τα δύο είδη τρανζίστορ (τεχνολογία CMOS)

**NMOS:** άγει όταν στην πύλη εφαρμόζεται ‘1’  
Περνά ισχυρό ‘0’



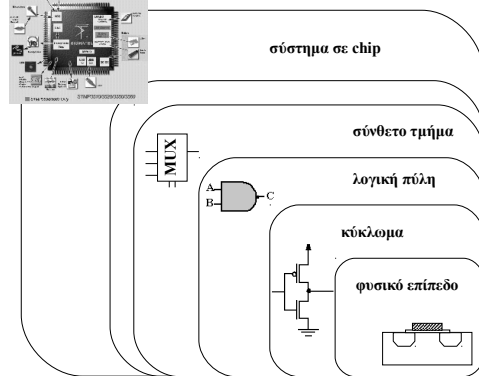
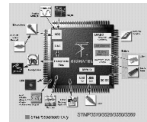
σύμβολα

**PMOS:** άγει όταν στην πύλη εφαρμόζεται ‘0’  
Περνά ισχυρό ‘1’



## Ψηφιακά Ηλεκτρονικά: Ιεραρχία σχεδίασης

- Ηλεκτρονικά κυκλώματα



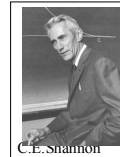
Αρχιτεκτονική Υπολογιστών – “Ψηφιακή Λογική και Σχεδίαση”

5

## Ψηφιακά Ηλεκτρονικά και Δυαδική λογική

- Ηλεκτρονικά κυκλώματα

- Η δυαδική λογική ταιριάζει με την τεχνολογία του τρανζίστορ
  - 2 καταστάσεις: ON-OFF, 1-0
  - Ψηφιακά ηλεκτρονικά (2 στάθμες)
- Δυαδική άλγεβρα Boole
  - Λογική άλγεβρα
  - Συσχέτιση με διακοπτικά κυκλώματα
    - Η εργασία του Shannon (1938)



C.E. Shannon

Αρχιτεκτονική Υπολογιστών – “Ψηφιακή Λογική και Σχεδίαση”

6

## Άλγεβρα Boole: επανάληψη

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

- $A + B$  (A OR B)
- $A \cdot B$  (ή απλά AB, A AND B)
- $\overline{A}$  (NOT A)
  - $A + 0 = A$  και  $A \cdot 1 = A$
  - $A + 1 = 1$  και  $A \cdot 0 = 0$
  - $A + \overline{A} = 1$  και  $A \cdot \overline{A} = 0$
  - $A + B = B + A$  και  $A \cdot B = B \cdot A$
  - $A + (B \cdot C) = (A + B) \cdot (A + C)$  και
  - $A(BC) = (AB)C$

Αρχιτεκτονική Υπολογιστών – “Ψηφιακή Λογική και Σχεδίαση”

7

## Άλγεβρα Boole: επανάληψη

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole

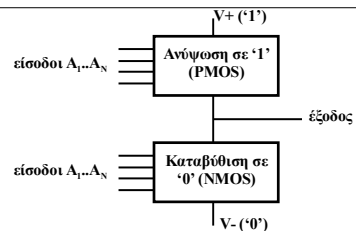
- $A(B+C) = (AB)+(AC)$  και
- $A+(BC) = (A+B)(A+C)$
- $\overline{(A+B)} = \overline{A} \cdot \overline{B}$  και
- $\overline{(A \cdot B)} = \overline{A} + \overline{B}$  (DeMorgan)

Αρχιτεκτονική Υπολογιστών – “Ψηφιακή Λογική και Σχεδίαση”

8

## Απλές λογικές πύλες

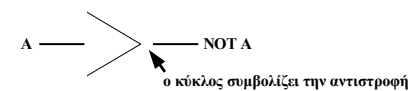
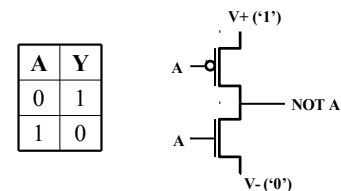
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες



- Στατική τεχνολογία CMOS
  - Αναστρέφουσες συναρτήσεις
    - NOT, NAND, NOR ...
  - Αξιόπιστη λειτουργία, εύκολη σχεδίαση
  - Όχι πάντα η αποδοτικότερη λύση

## Η πύλη NOT

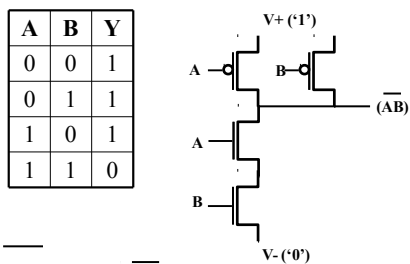
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες



σύμβολο πύλης NOT

## Η πύλη NAND

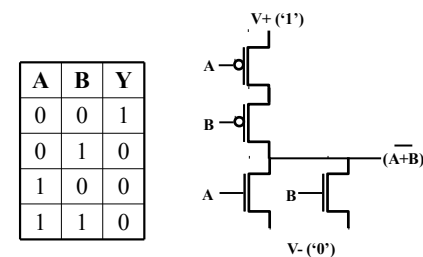
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες



σύμβολο πύλης NAND

## Η πύλη NOR

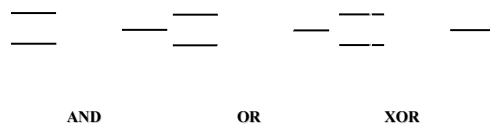
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες



σύμβολο πύλης NOR

## Άλλες λογικές πύλες

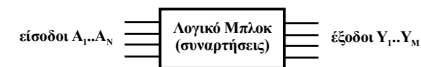
- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες



- Ως συνδυασμός των βασικών πυλών
  - NOT, NAND, NOR
- Υπάρχουν και εναλλακτικές μέθοδοι σχεδίασης!

## Συνδυαστική Λογική

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική



- Μπλοκ λογικών συναρτήσεων
  - Οι έξοδοι εξαρτώνται αποκλειστικά από την τρέχουσα τιμή των εισόδων
  - Δεν υπάρχει μνήμη προηγούμενων καταστάσεων
  - Αλλαγή των εισόδων θα επηρεάσει τις εξόδους μετά από χρονικό διάστημα (καθυστερήσει διάδοσης)
  - Η συνάρτηση που υλοποιεί το μπλοκ μπορεί να εκφραστεί με έναν πίνακα αλήθειας

## Υλοποίηση συναρτήσεων

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική

!

Πάντοτε προσπαθούμε να απλοποιήσουμε τις συναρτήσεις (πίνακες Karnaugh ή άλλες υπολογιστικές μέθοδοι)

A	B	C	Y	Ελαχιστόροι	Μεγιστόροι
0	0	0	1	$a'b'c'$	$a+b+c$
0	0	1	0	$a'b'c$	$a+b+c'$
0	1	0	0	$a'bc'$	$a+b'+c$
0	1	1	1	$a'bc$	$a+b'+c'$
1	0	0	0	$ab'c'$	$a'+b+c$
1	0	1	0	$ab'c$	$a'+b+c'$
1	1	0	1	$abc'$	$a'+b'+c$
1	1	1	1	$abc$	$a'+b'+c'$

- $Y = a'b'c' + a'bc + abc' + abc$
- $Y = (a+b+c')(a+b'+c)(a'+b+c)(a'+b'+c')$

## Βασικά συνδυαστικά τμήματα

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

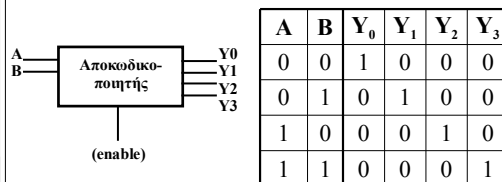
- Αποκωδικοποιητής (decoder)
  - N εισοδοί ενεργοποιούν 1 από  $2^N$  εξόδους
- Πολυπλέκτης (multiplexer)
  - N εισοδοί επιλέγουν 1 από  $2^N$  εισόδους
- Αθροιστής (adder)
  - Αριθμητική πρόσθεση δυαδικών ψηφίων

## Αποκωδικοποιητής (decoder)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Πώς υλοποιείται ένας αποκωδικοποιητής;



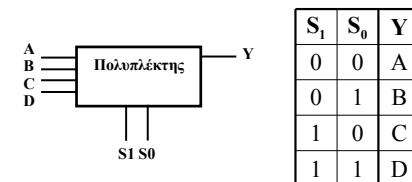
- Αποκωδικοποιητής N-σε-2<sup>N</sup>
  - N εισοδοι ενεργοποιούν μία από 2<sup>N</sup> εξόδους

## Πολυπλέκτης (multiplexer)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Πώς υλοποιείται ένας πολυπλέκτης;



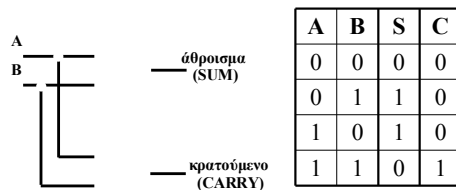
- Πολυπλέκτης 2<sup>N</sup> γραμμών σε 1
  - Επιλογή μίας από 2<sup>N</sup> εισόδους με τη βοήθεια N σημάτων ελέγχου

## Ημιαθροιστής (half-adder)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Αν απαιτείται πρόσθεση αριθμών με περισσότερα bits;

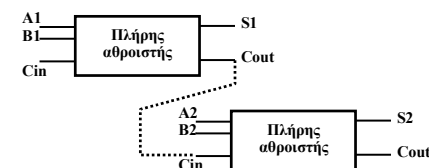


## Πλήρης αθροιστής (full-adder)

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

;

Πώς υλοποιείται ένας πλήρης αθροιστής;



- Πολλαπλά τμήματα πλήρη αθροιστή
  - Όμως: πόσο γρήγορα διαδίδεται το κρατούμενο; (ripple carry)
  - Τεχνικές πρόβλεψης κρατουμένου (carry look-ahead)

## Πέρα από τη συνδυαστική λογική...

- Ηλεκτρονικά κυκλώματα
- Άλγεβρα Boole
- Λογικές Πύλες
- Συνδυαστική Λογική
- Βασικά Συνδυαστικά Τμήματα

### • Στο επόμενο μάθημα...

- Πώς εισάγω την έννοια της κατάστασης ενός λογικού τμήματος;
- Που φυλάσσεται η κατάσταση;
- Πότε ενημερώνεται;
- Πώς συγχρονίζονται τα διάφορα τμήματα λογικής;