

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής  
Εισαγωγή στην Επιστήμη των Υπολογιστών  
2016-17

# Πράξεις με δυαδικούς αριθμούς

(λογικές πράξεις)

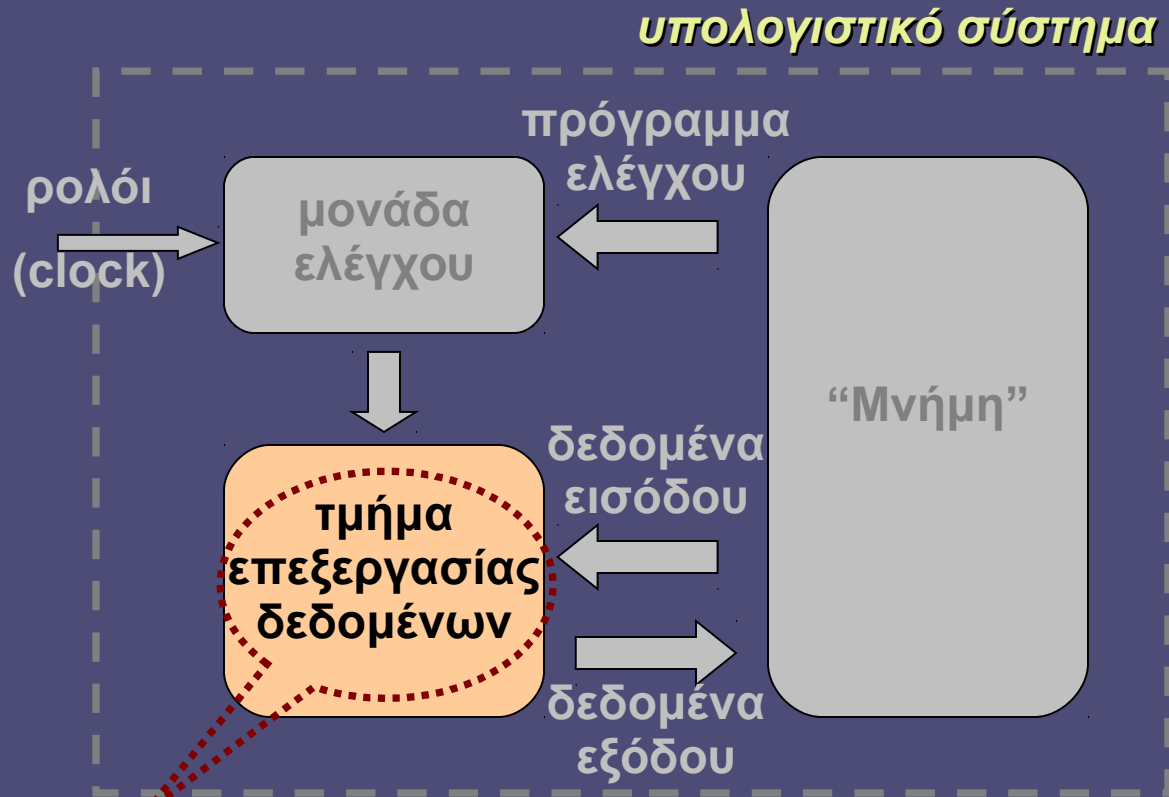
<http://mixstef.github.io/courses/csintro/>

Μ.Στεφανιδάκης



# Εκτέλεση πράξεων

- Εισαγωγή



- Επεξεργασία: ψηφιακά δυαδικά κυκλώματα
  - Εκτελούν πράξεις μεταξύ σειρών 0 και 1...
  - ...οι οποίες αναπαριστούν δυαδικούς αριθμούς

Ποιες κατηγορίες πράξεων;

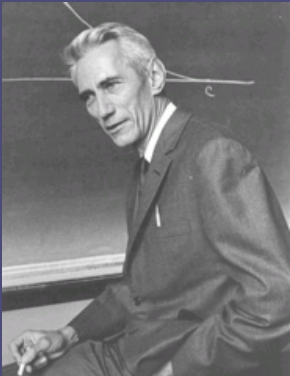
# Πράξεις με δυαδικούς αριθμούς

- Εισαγωγή

- Ο υπολογιστής μπορεί να εκτελέσει
  - Λογικές πράξεις (δυαδικής λογικής)
  - Αριθμητικές πράξεις
- Οι πράξεις εκτελούνται
  - Σε ομάδες bits: “δυαδικούς αριθμούς”

# Ψηφιακά Ηλεκτρονικά και Δυαδική Λογική

- Εισαγωγή
- Δυαδική λογική



C.E.Shannon

- 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)

# Ποσότητες Δυαδικής Λογικής

- Εισαγωγή
- Δυαδική λογική

- Στη δυαδική λογική άλγεβρα
  - Υπάρχουν 2 “ποσότητες” (σύμβολα):
    - Αληθές ή 1 ή ΝΑΙ
    - Ψευδές ή 0 ή ΟΧΙ
  - Ένα δυαδικό ψηφίο (**bit**) έχει τιμή 0 ή 1
- Στα ψηφιακά ηλεκτρονικά κυκλώματα:
  - 0 ή “χαμηλή τάση” ή “η μια φορά ρεύματος”
  - 1 ή “υψηλή τάση” ή “η άλλη φορά ρεύματος”
  - Ανάλογα με την τεχνολογία, ένα bit αναπαρίσταται με αντίστοιχη κατάσταση σε ένα ηλεκτρονικό κύκλωμα

# Bits & Bytes

- Εισαγωγή
- Δυαδική λογική

- **Bit**

- Η **μικρότερη** λογική ποσότητα - η μικρότερη μονάδα δεδομένων - 0 ή 1.

- **Byte**

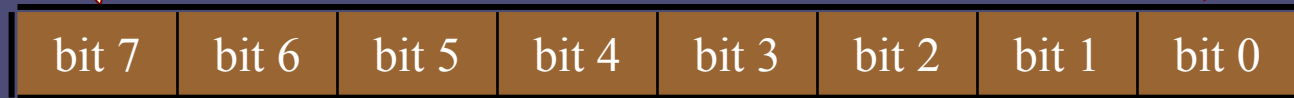
- Ομάδα 8 bits
- Η **ελάχιστη** ποσότητα που μπορεί να χειριστεί ο υπολογιστής κατά την εκτέλεση μιας πράξης
- Μια σειρά από bytes αναπαριστά έναν δυαδικό “αριθμό”
  - Αποθήκευση: σε καταχωρητές ή στη μνήμη

# Η ανατομία ενός byte

- Εισαγωγή
- Δυαδική λ

το περισσότερο  
σημαντικό bit

το λιγότερο  
σημαντικό  
bit



- Γιατί είναι αυτή η σειρά των bits;
  - Γιατί το λιγότερο σημαντικό bit είναι δεξιά και το περισσότερο σημαντικό αριστερά;
  - Θα φανεί όταν μιλήσουμε για αριθμητικές πράξεις
  - Προς το παρόν, το byte είναι απλώς μια οκτάδα bits

# Πράξεις Δυαδικής Λογικής

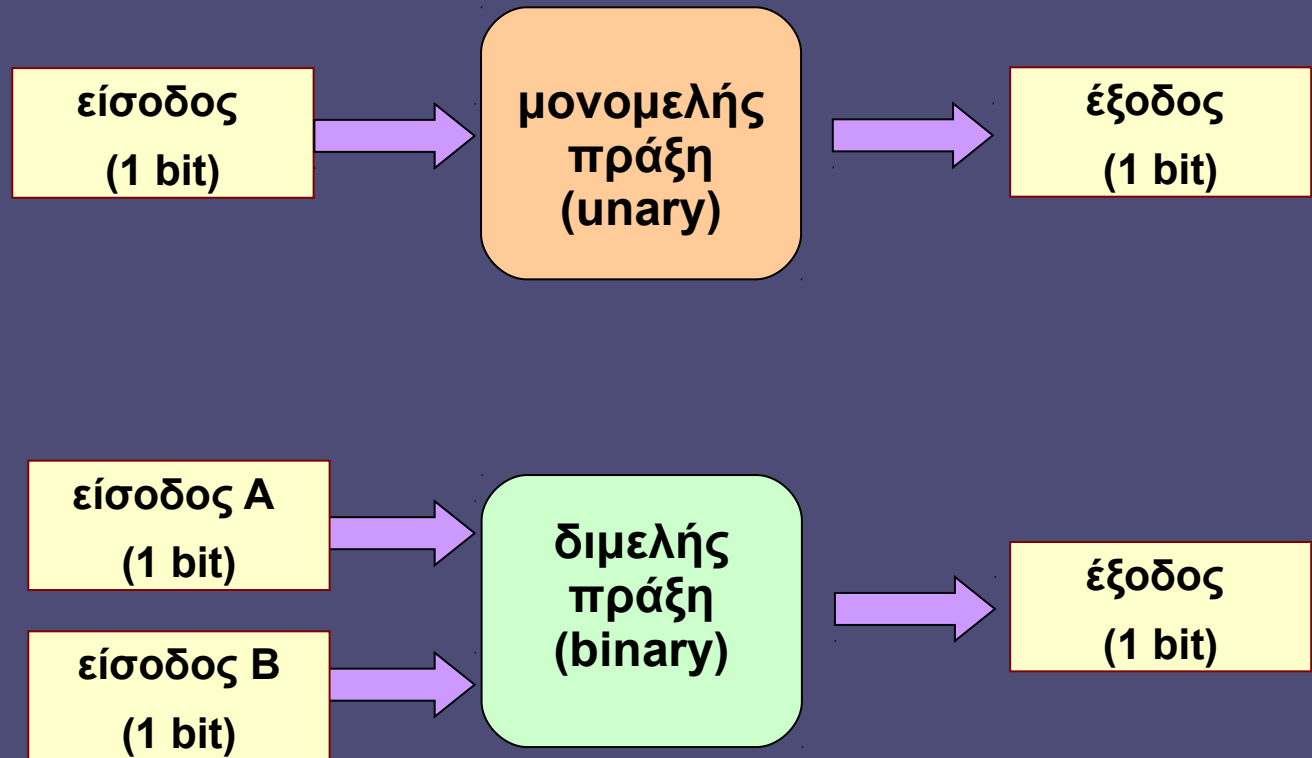
- Εισαγωγή
- Δυαδική λογική

- Στη δυαδική λογική άλγεβρα
  - Καθορίζονται λογικές πράξεις μεταξύ των λογικών ποσοτήτων 0 και 1 (bits)
- Στα ψηφιακά ηλεκτρονικά κυκλώματα:
  - Κύκλωμα δέχεται ως είσοδο την ηλεκτρική αναπαράσταση των 0 και 1
  - Και παράγει στην έξοδό του την ηλεκτρική αναπαράσταση του αποτελέσματος μιας λογικής πράξης
  - Το κύκλωμα υλοποίησης της λογικής πράξης ονομάζεται πύλη (gate).



# Λογικές πράξεις με bits

- Εισαγωγή
- Δυαδική λογική



# Λογικές πράξεις με bits

- Εισαγωγή
- Δυαδική λογική

- Μονομελής λογική πράξη
  - NOT (αντιστροφή)
- Διμελείς λογικές πράξεις
  - AND (λογικό-ΚΑΙ)
  - OR (λογικό-Η)
  - XOR (αποκλειστικό-Η)

# Βασικές Λογικές Πράξεις

- Εισαγωγή
- Δυαδική λογική

- Αντιστροφή (NOT)
  - Αντιστροφή ενός bit

είσοδος A

έξοδος NOT (A)

ή A' ή  $\overline{A}$

πιθανές  
τιμές  
εισόδου

A	Y
0	1
1	0

αντίστοιχες  
τιμές εξόδου

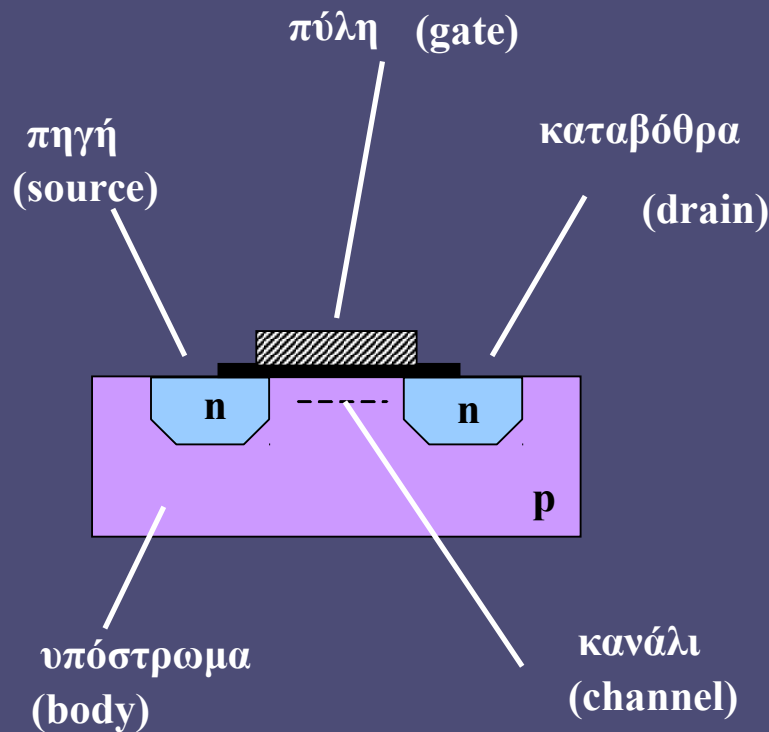
Πίνακας Αλήθειας

# Πώς υλοποιείται μια πύλη NOT;

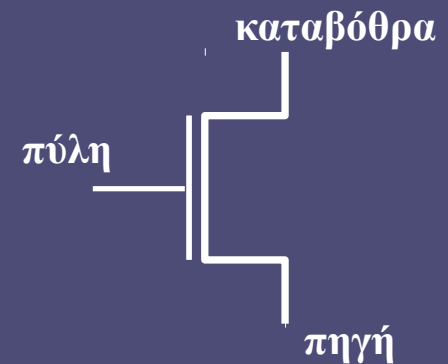
- Εισαγωγή
- Δυαδική λογική

- Από το προηγούμενο μάθημα:

**PMOS:** άγει όταν στην πύλη εφαρμόζεται '0'



Το τρανζίστορ NMOS

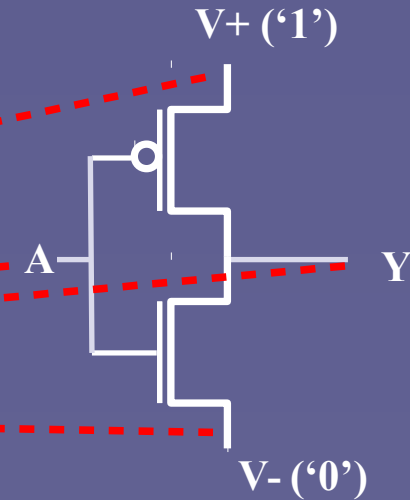
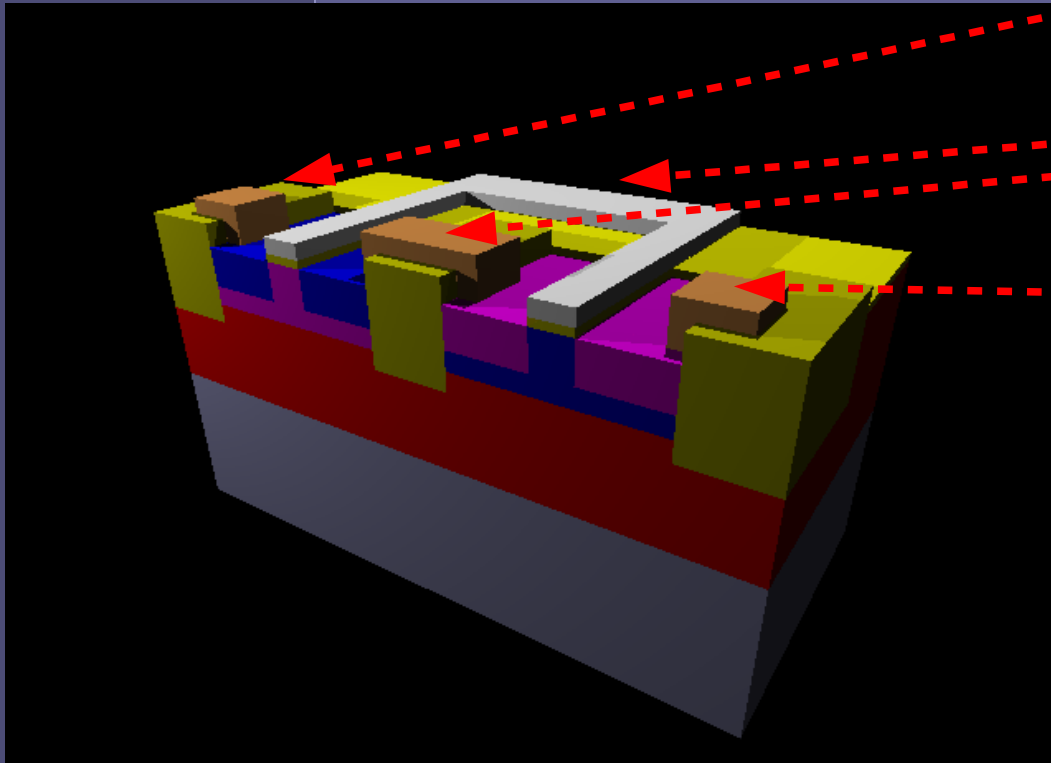


σύμβολο

**NMOS:** άγει όταν στην πύλη εφαρμόζεται '1'

# Από το προηγούμενο μάθημα: ποια η λειτουργία του;

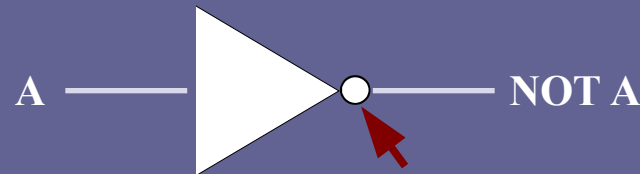
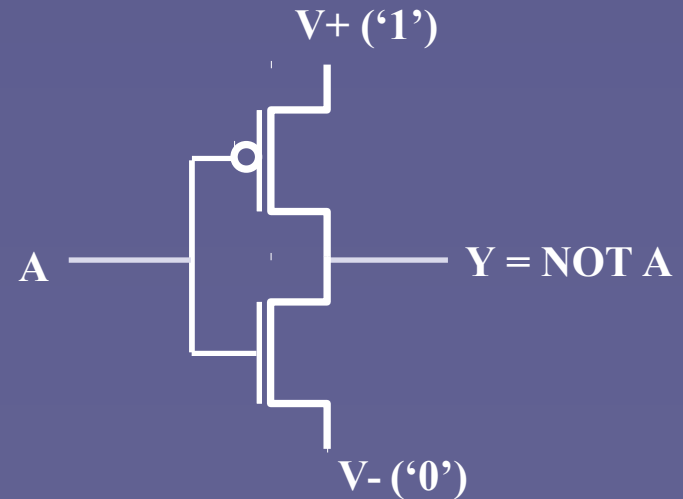
- Εισαγωγή
- Δυαδική Λογική



# Η πύλη NOT (αντιστροφέας)

- Εισαγωγή
- Δυαδική Λογική

A	Y
0	1
1	0



ο κύκλος συμβολίζει την αντιστροφή

**σύμβολο πύλης NOT**

# Βασικές Λογικές Πράξεις

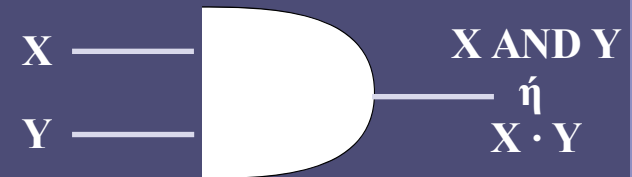
- Εισαγωγή
- Δυαδική Λογική

- Λογικό ΚΑΙ (AND)

- το αποτέλεσμα είναι 1, μόνο όταν και το X και το Y είναι 1
- $0 \text{ AND } X = X \text{ AND } 0 = 0$
- $1 \text{ AND } X = X \text{ AND } 1 = X$

Πίνακας Αλήθειας

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



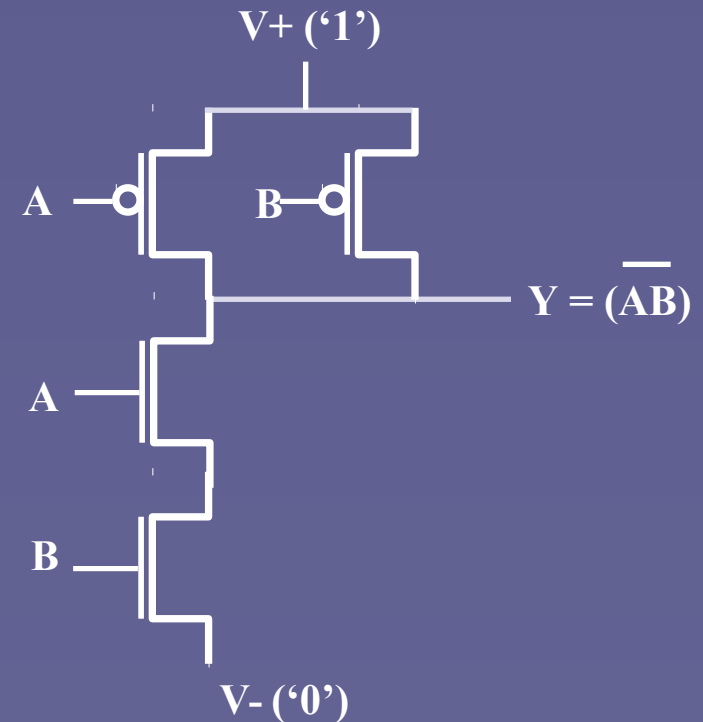
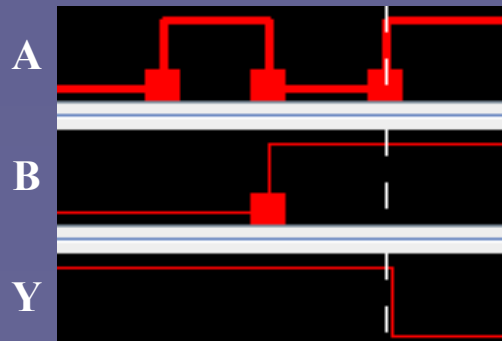
σύμβολο πύλης AND

# Παράδειγμα υλοποίησης: η πύλη NAND

- Εισαγωγή
- Δυαδική Λογική

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Πίνακας Αλήθειας NAND



σύμβολο πύλης NAND

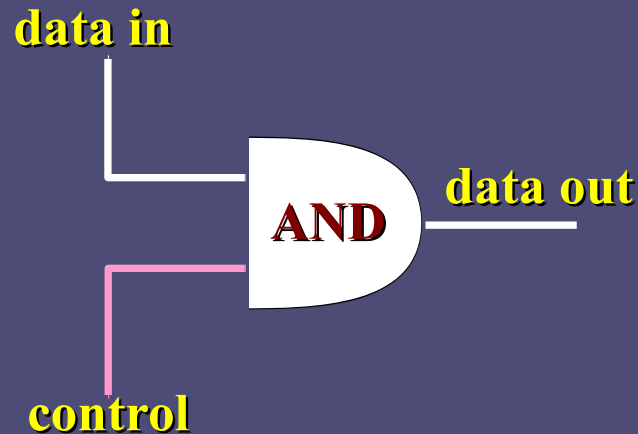
!  
NAND = NOT-AND

Υλοποίηση πύλης AND: χρησιμοποιώντας μια πύλη NAND και μια πύλη NOT



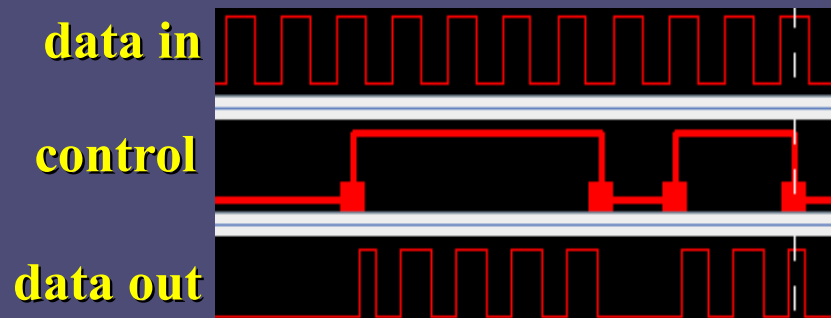
# Φραγή AND: για να θέσουμε σήμα στο 0

- Εισαγωγή
- Δυαδική Λογική



control = 0: η έξοδος data\_out είναι πάντα 0

control = 1: η έξοδος data\_out ισούται με το data\_in



$$0 \text{ AND } X = 0$$

$$1 \text{ AND } X = X$$

# Βασικές Λογικές Πράξεις

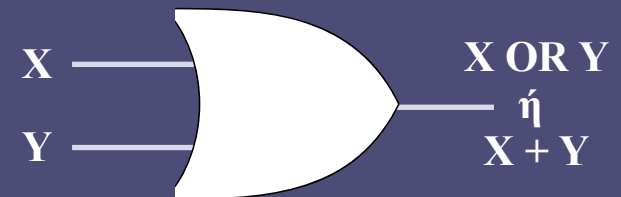
- Εισαγωγή
- Δυαδική Λογική

- Λογικό Ή (OR)

- το αποτέλεσμα είναι 1, όταν το X ή το Y ή και τα δύο είναι 1
- $1 \text{ OR } X = X \text{ OR } 1 = 1$
- $0 \text{ OR } X = X \text{ OR } 0 = X$

Πίνακας Αλήθειας

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



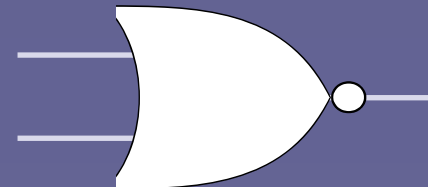
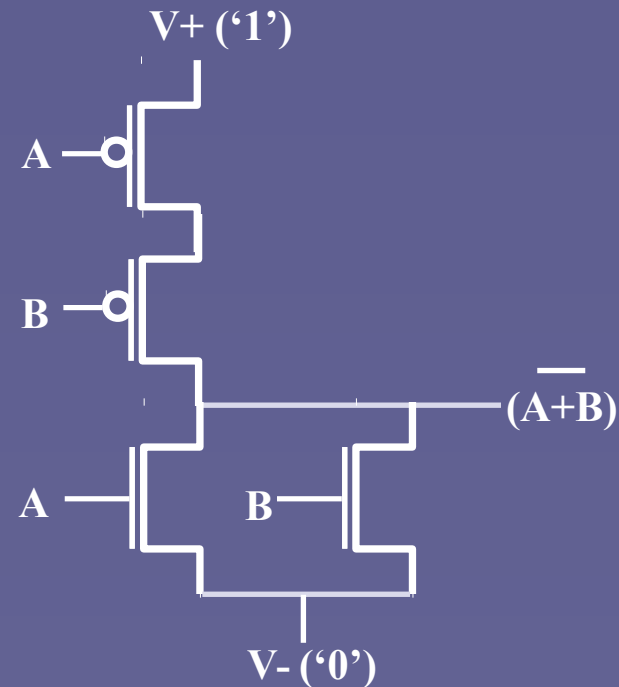
σύμβολο πύλης OR

# Παράδειγμα υλοποίησης: η πύλη NOR

- Εισαγωγή
- Δυαδική Λογική

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Πίνακας Αλήθειας NOR



σύμβολο πύλης NOR

NOR = NOT-OR

Υλοποίηση πύλης  
OR: χρησιμοποι-  
ώντας μια πύλη  
NOR και μια πύλη  
NOT

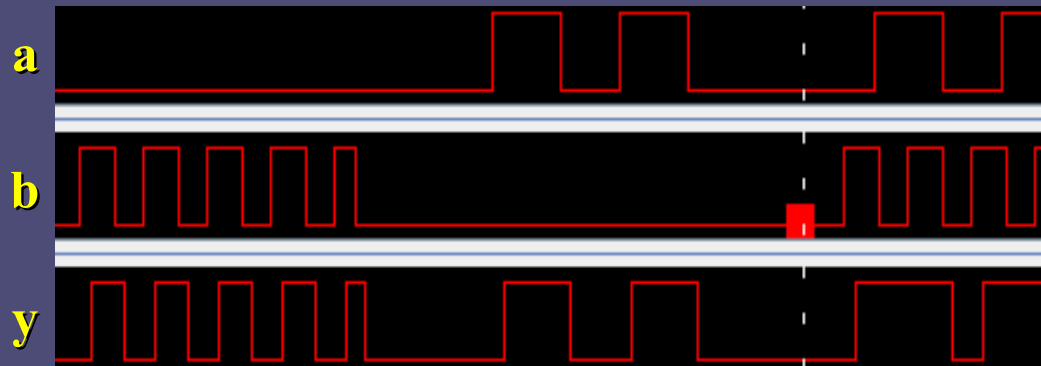
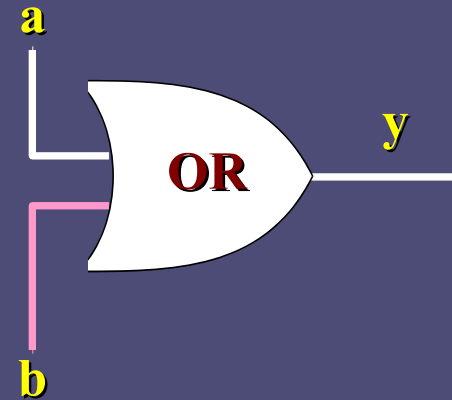
# Συγκέντρωση σημάτων με OR

- Εισαγωγή
- Δυαδική Λογική

$$0 \text{ OR } X = X$$

$$1 \text{ OR } X = 1$$

Προσοχή!  
Ποτέ δεν  
συνδέουμε  
εξόδους  
πυλών μαζί!



- Θα πρέπει ανά πάσα στιγμή όλα τα σήματα πλην ενός να είναι 0!

# Βασικές Λογικές Πράξεις

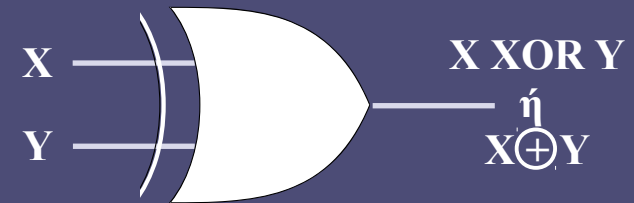
- Εισαγωγή
- Δυαδική Λογική

- Αποκλειστικό Ή (XOR)

- το αποτέλεσμα είναι 1, όταν **μόνο το X** ή **μόνο το Y** είναι 1
- $1 \text{ XOR } X = X \text{ XOR } 1 = \text{NOT } X$
- $0 \text{ XOR } X = X \text{ XOR } 0 = X$
- $X \text{ XOR } Y = A \cdot B' + A' \cdot B$

Πίνακας Αλήθειας

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0



**σύμβολο πύλης XOR**

# Βασικές Λογικές Πράξεις

- Εισαγωγή
- Δυαδική Λογική



Υλοποίηση πύλης  
XNOR:  
χρησιμοποιώντας  
συνδυασμούς  
άλλων πυλών  
 $X \text{ xnor } Y =$   
 $XY + X'Y'$

- XNOR: Η συμπληρωματική συνάρτηση της XOR
  - το αποτέλεσμα είναι 1, όταν τα  $X$  και  $Y$  είναι όμοια
  - συνάρτηση “ισοδυναμίας”

Πίνακας Αλήθειας

X	Y	XNOR
0	0	1
0	1	0
1	0	0
1	1	1

# Λογικές πράξεις σε ομάδες bits

- Λογικές Πράξεις

- Ο υπολογιστής μπορεί να εφαρμόσει λογικές πράξεις στα δεδομένα μας

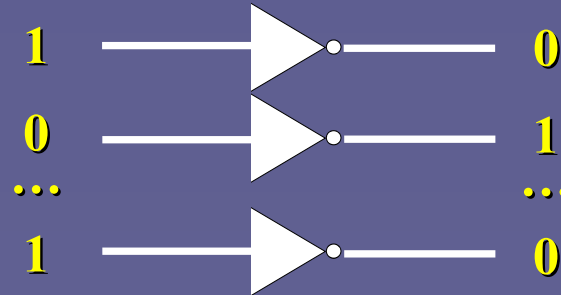
- Δεδομένα = σειρές από 0 και 1
- Όχι όμως σε μεμονωμένα bits!!
- Αλλά: σε ομάδες των 8, 16, 32 ή 64 bits ταυτόχρονα

$$\begin{array}{ccccccc} A_n & \dots & A_i & \dots & A_2 & A_1 & A_0 \\ B_n & \dots & B_i & \dots & B_2 & B_1 & B_0 \\ \hline Y_n & \dots & Y_i & \dots & Y_2 & Y_1 & Y_0 \end{array} \quad \text{op} \quad (=AND, OR, XOR)$$

$Y_i = A_i \text{ op } B_i$

# Ο τελεστής NOT σε δυαδικούς αριθμούς

- Λογικές πράξεις



!

Η “μέθοδος” του υλικού (hardware): πολλαπλές ίδιες μονάδες εκτελούν την ίδια λειτουργία παράλληλα

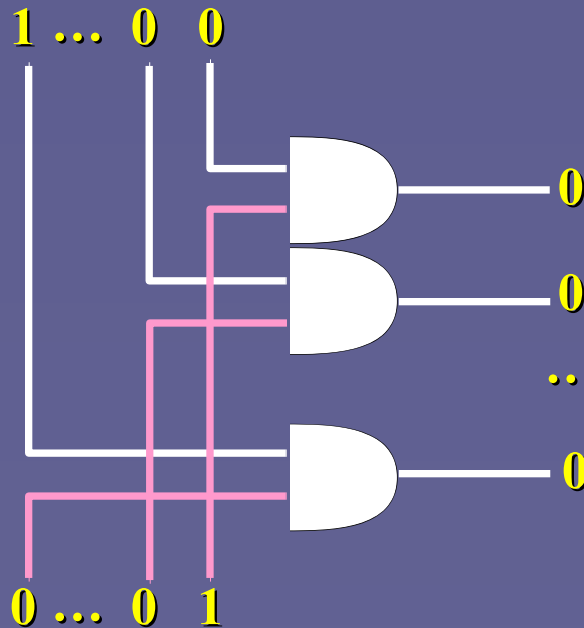
$$\begin{array}{r} 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0 \\ \hline 0\ 1\ 1\ 0\ 0\ 1\ 1\ 1 \end{array} \quad \text{NOT}$$

- Η έξοδος  $Y_i$  εξαρτάται μόνο από την είσοδο  $A_i$



# Ο τελεστής AND σε δυαδικούς αριθμούς

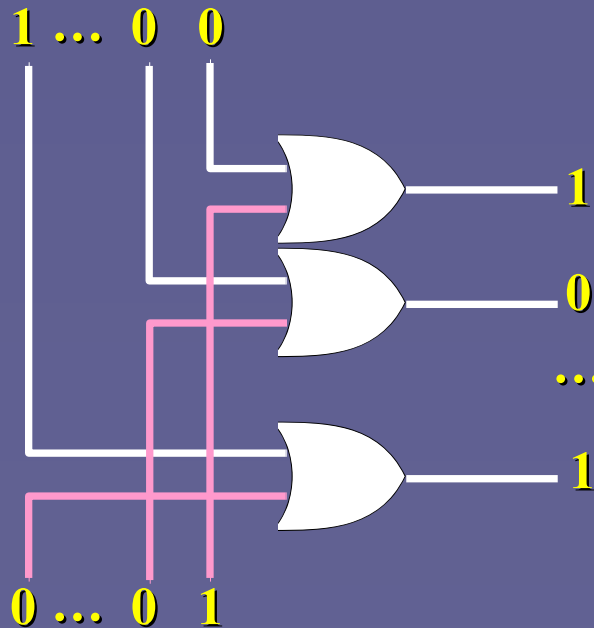
- Λογικές πράξεις



1	0	0	1	1	0	0	0	AND
0	0	1	1	0	1	0	1	
<hr/>								
0	0	0	1	0	0	0	0	

# Ο τελεστής OR σε δυαδικούς αριθμούς

- Λογικές πράξεις



1	0	0	1	1	0	0	0	OR
0	0	1	1	0	1	0	1	
<hr/>								
1	0	1	1	1	1	0	1	

# Μάσκες

- Λογικές Πράξεις

- Για να αλλάξουμε την τιμή μεμονωμένων bits μέσα σε μια ομάδα
  - Για να θέσουμε επιλεγμένα bits σε 1
  - Για να θέσουμε επιλεγμένα bits σε 0
  - Για να αντιστρέψουμε επιλεγμένα bits
  - Χωρίς να επηρεάζουμε τα υπόλοιπα!
    - αυτά διατηρούν την τιμή τους, είτε 0 είτε 1
- Μάσκα: σειρά bits, επιλεγμένη ώστε:  
*Bits Εισόδου op Μάσκα  $\rightarrow$  Νέα ομάδα bits*
  - op = AND, OR ή XOR
  - Νέα ομάδα περιέχει το επιθυμητό αποτέλεσμα

# Μάσκα AND: για να θέσουμε bits στο 0

- Λογικές πράξεις

- Ζητούμενο: σε λέξη των 8 bits να τεθούν σε 0 τα 3 λιγότερο σημαντικά bits.

Λέξη:	1 0 0 1 1 0 1 0	AND
Μάσκα:	1 1 1 1 1 0 0 0	
Νέα:	1 0 0 1 1 0 0 0	

- Η AND μάσκα περιέχει:
  - 0 στα bits που θα γίνουν 0
  - 1 στα bits που θα παραμείνουν ως έχουν

$$0 \text{ AND } X = 0$$

$$1 \text{ AND } X = X$$

# Μάσκα OR: για να θέσουμε bits στο 1

- Λογικές πράξεις

- Ζητούμενο: σε λέξη των 8 bits να τεθούν σε 1 τα bits 0,4 και 5.

Λέξη:	1 0 0 1 1 0 0 0	OR
Μάσκα:	0 0 1 1 0 0 0 1	
Νέα:	1 0 1 1 1 0 0 1	

- Η OR μάσκα περιέχει:
  - 1 στα bits που θα γίνουν 1
  - 0 στα bits που θα παραμείνουν ως έχουν

$$0 \text{ OR } X = X$$

$$1 \text{ OR } X = 1$$

# Μάσκα XOR: για να αντιστρέψουμε bits

- Λογικές πράξεις

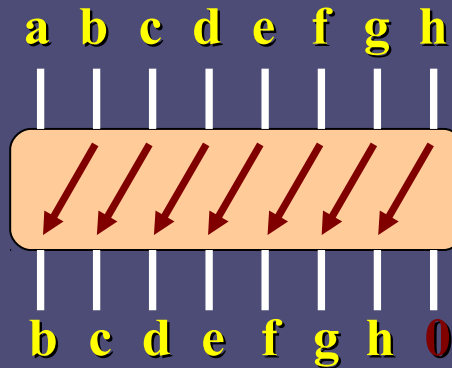
- Ζητούμενο: σε λέξη των 8 bits να αντιστραφούν τα bits 3,6 και 7.

Λέξη:	1 0 0 1 1 0 0 0	XOR
Μάσκα:	1 1 0 0 1 0 0 0	
Νέα:	0 1 0 1 0 0 0 0	

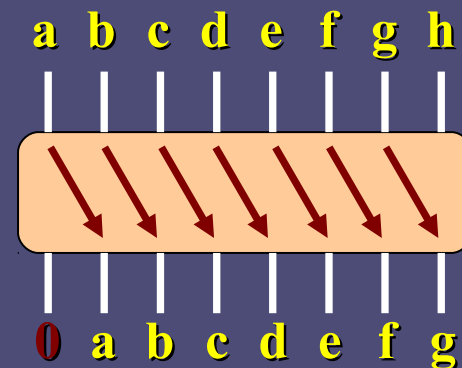
- Η XOR μάσκα περιέχει:
  - 1 στα bits που θα αντιστραφούν
  - 0 στα bits που θα παραμείνουν ως έχουν

# Ολίσθηση (Shift)

- Λογικές πράξεις
- Ολίσθηση



αριστερή ολίσθηση



δεξιά ολίσθηση