



Εισαγωγή στην Πληροφορική και τον Προγραμματισμό Η/Υ

7^ο Μάθημα

Υλοποίηση πράξεων στο δυαδικό σύστημα μέσω Λογικών συναρτήσεων:
Λογικές πύλες, Λογικές Πράξεις. Λογικές συναρτήσεις.
Λογικά Κυκλώματα & Άλγεβρα Μπουλ

Λεωνίδας Αλεξόπουλος

Αν. Καθηγητής ΕΜΠ

E-mail: leo@mail.ntua.gr

Τηλ: 210 772-1666

Σύνοψη Μέχρι Σήμερα:

Στα προηγούμενα μαθήματα μάθαμε το **Δυαδικό Σύστημα**

→ Κωδικοποίηση – Επεξεργασία – Αποκωδικοποίηση

→ Bit/Byte/Word

→ Κωδικοποίηση αριθμών

→ Παράσταση Ακέραιων Θετικών / Αρνητικών, Fixed & Floating point, Πρόσθεση, Αφαίρεση, Πολλαπλασιασμός,

→ Κωδικοποίηση άλλων δεδομένων

→ Χαρακτήρες, Εικόνα, Ήχος, Αναλογικό Σήμα

→ Μετατροπή Αναλογικού Σήματος σε Ψηφιακό (ADC/DAC), Μέθοδοι Επαλήθευσης Δεδομένων

**Πως η θεωρία αυτή υλοποιείται σε επίπεδο Η/Υ?
Transistor**

Λογικά Κυκλώματα, Λογικών συναρτήσεις, Λογικές πύλες, Λογικές Πράξεις

Σήμερα:

Μέρος 1ο: (όχι στον Forouzan – εκτός ύλης)

Εισαγωγή σε

- Transistors
- Υλοποίηση με Transistor
- Κατασκευή Transistors (εκτός ύλης)

Μέρος 2ο:

Forouzan: μόνο 4.1 & Παράρτημα Ε (εκτός 4.2/4.3)

Εισαγωγή σε:

- Λογικές μεταβλητές, πύλες, συναρτήσεις
- Πίνακας αληθείας, Σύνθεση/ Ανάλυση
- Παραδείγματα / Ασκήσεις

Σήμερα:

Μέρος 1ο: (όχι στον Forouzan – εκτός ύλης)

Εισαγωγή σε

- Transistors
- Υλοποίηση με Transistor
- Κατασκευή Transistors (εκτός ύλης)

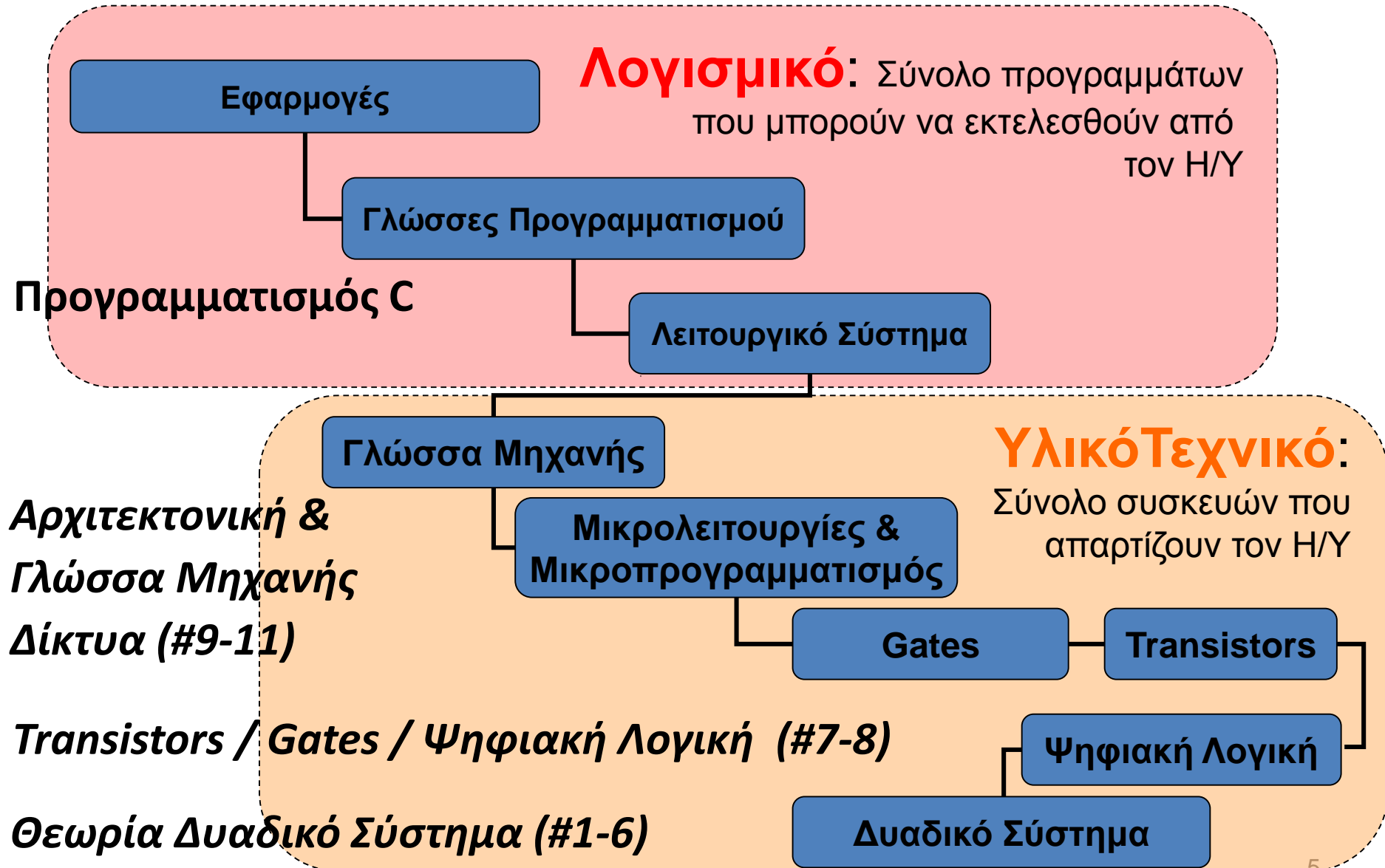
Μέρος 2ο:

Forouzan: μόνο 4.1 & Παράρτημα Ε (εκτός 4.2/4.3)

Εισαγωγή σε:

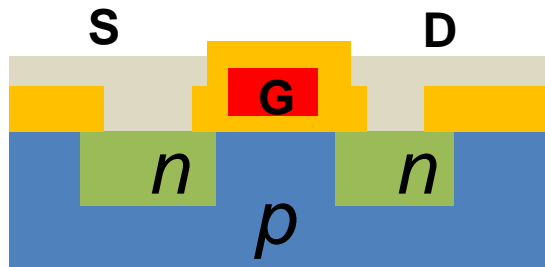
- Λογικές μεταβλητές, πύλες, συναρτήσεις
- Πίνακας αληθείας, Σύνθεση/ Ανάλυση
- Παραδείγματα / Ασκήσεις

Εισαγωγή στην Πληροφορική και τον Προγραμματισμό Η/Υ:



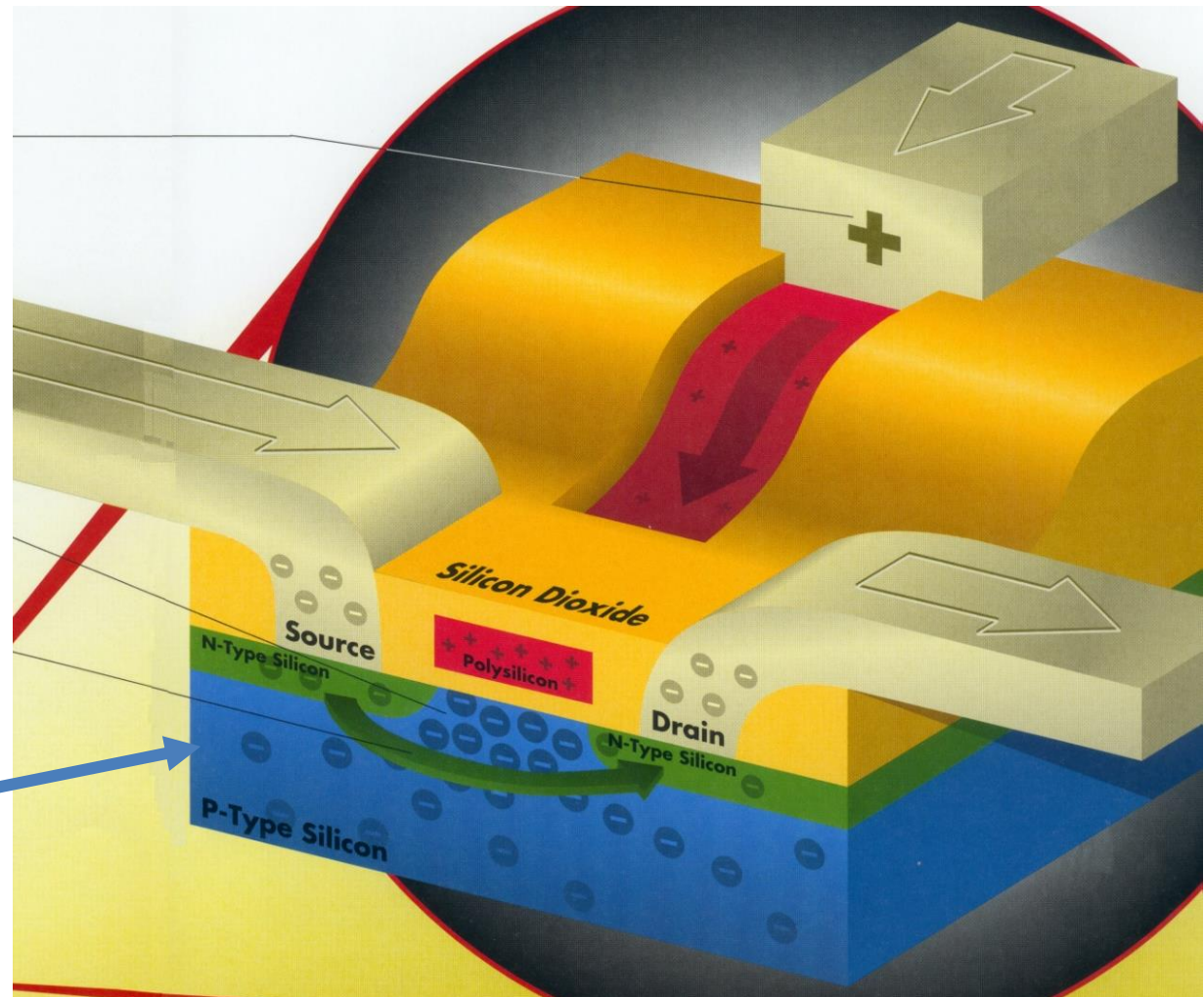
Transistor

Ελέγχουν την διέλευση ρεύματος από S → D



- Polysilicon
- Metal
- p substrate
- n material
- SiO₂

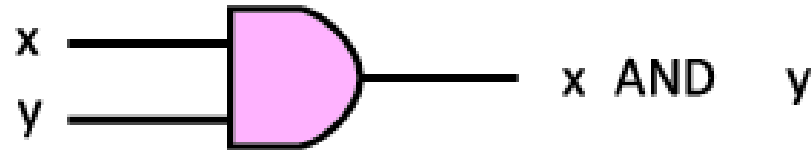
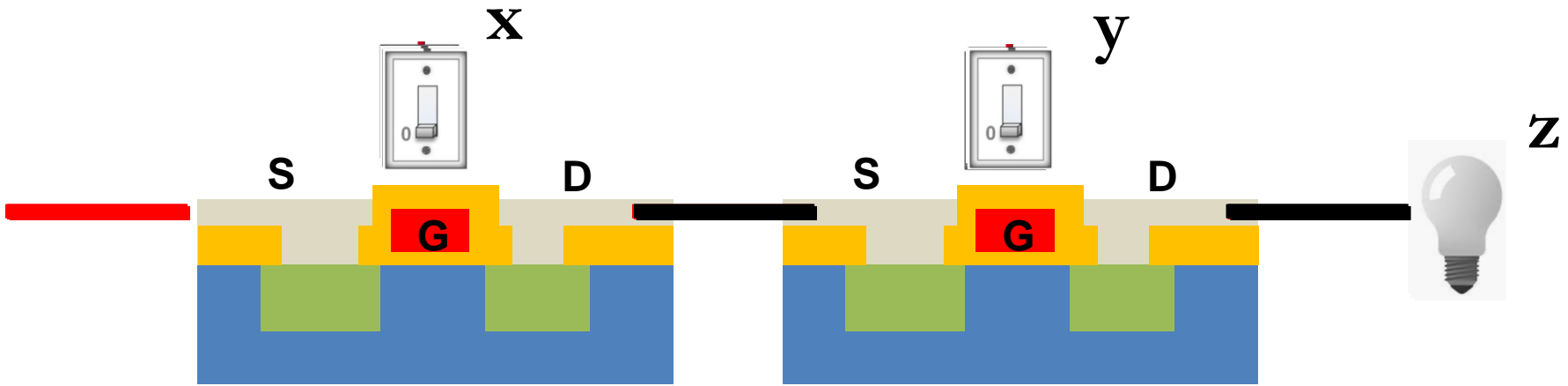
Silicon (πυρίτιο)



Transistors σε Η/Υς



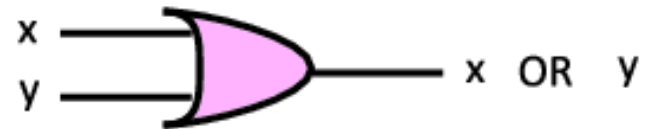
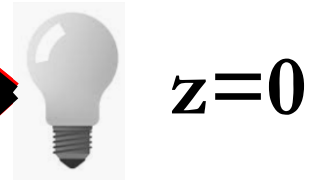
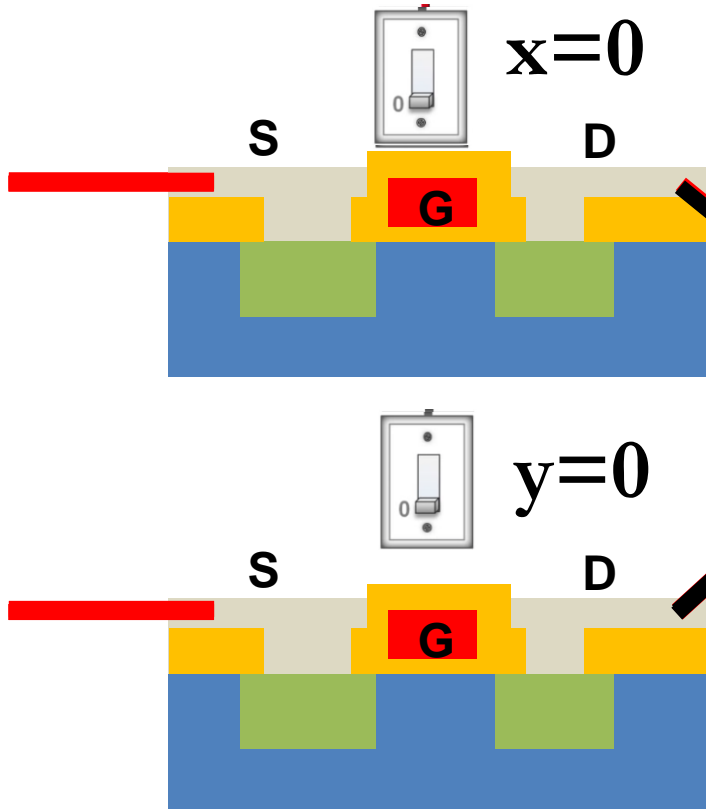
Πράξεις με transistors



AND

x	y	x AND y
0	0	0
0	1	0
1	0	0
1	1	1

Πράξεις με transistors

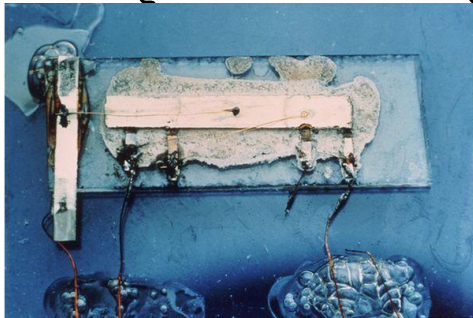


OR

x	y	x OR y
0	0	0
0	1	1
1	0	1
1	1	1

Ηλεκτρονική Υλοποίηση Πυλών

- Αν συνδυάσουμε, κατάλληλα ολοκληρωμένα, μεγάλο αριθμό τρανζίστορ μπορούμε να υλοποιήσουμε περισσότερες πύλες από αυτές που αντιστοιχούν στον ίδιο αριθμό τρανζίστορ μεμονωμένα. Ανάλογα με τον αριθμό των τρανζίστορ που χρησιμοποιούμε προκύπτουν οι παρακάτω περιπτώσεις:



1950

Αρ.
...	SSI	Small Scale Integration (1960)
~100	MSI	Medium Scale Integration
~1000	LSI	Large Scale Integration
~10 ⁶	VLSI	Very Large Scale Integration (1970)
~10 ⁹	ULSI	Ultra Large Scale Integration



2000 (VLSI)

2021 up to trillions....

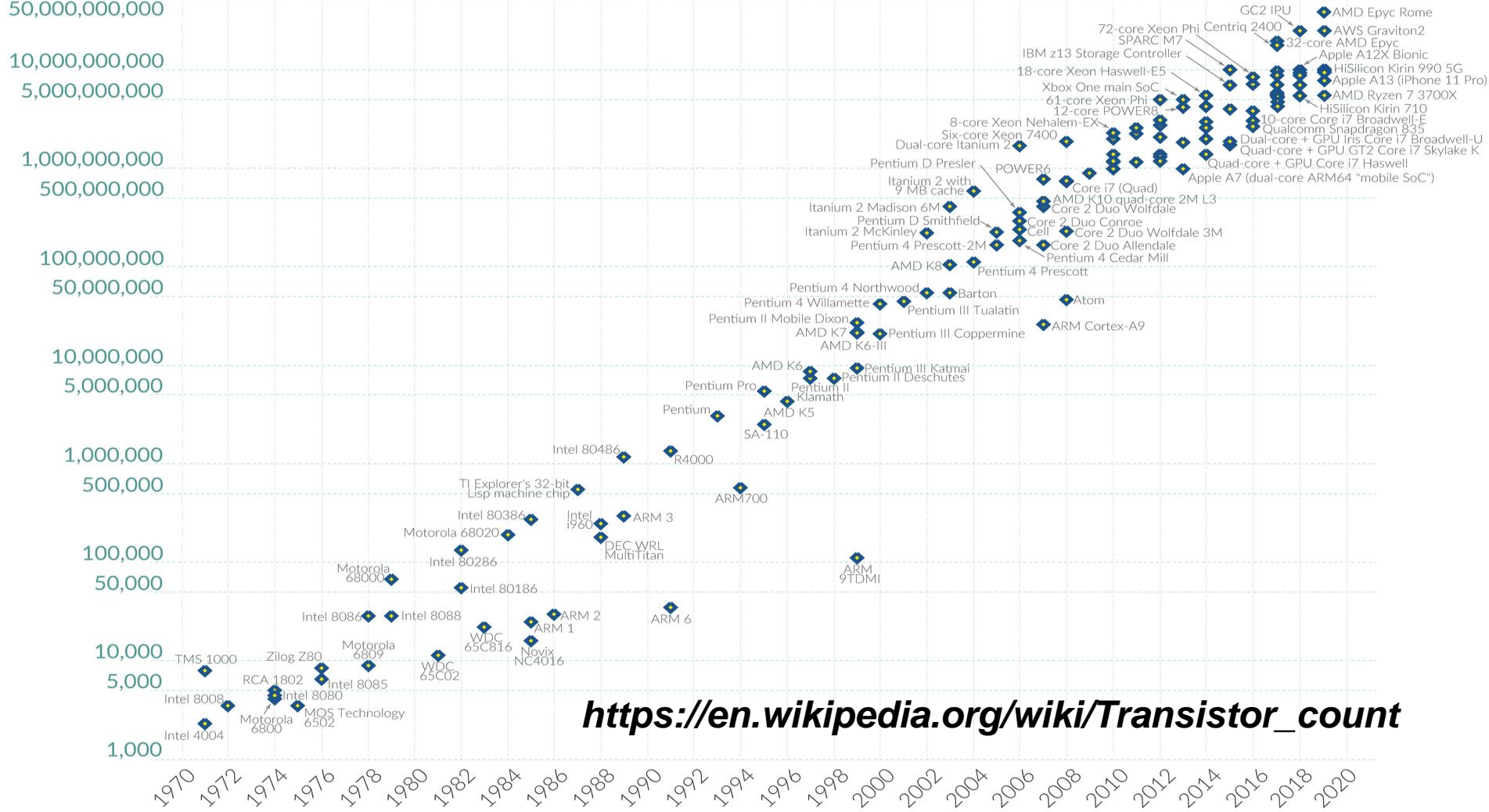
- Τα VLSI/ULSI είναι μικρά, παρουσιάζουν μικρή κατανάλωση ηλεκτρικής ενέργειας (άρα εκλύουν λιγότερη θερμότητα), είναι αξιόπιστα και το κόστος παραγωγής τους συνεχώς ελαττώνεται.

Αριθμός transistors per circuit chip

Moore's Law: The number of transistors on microchips doubles every two years

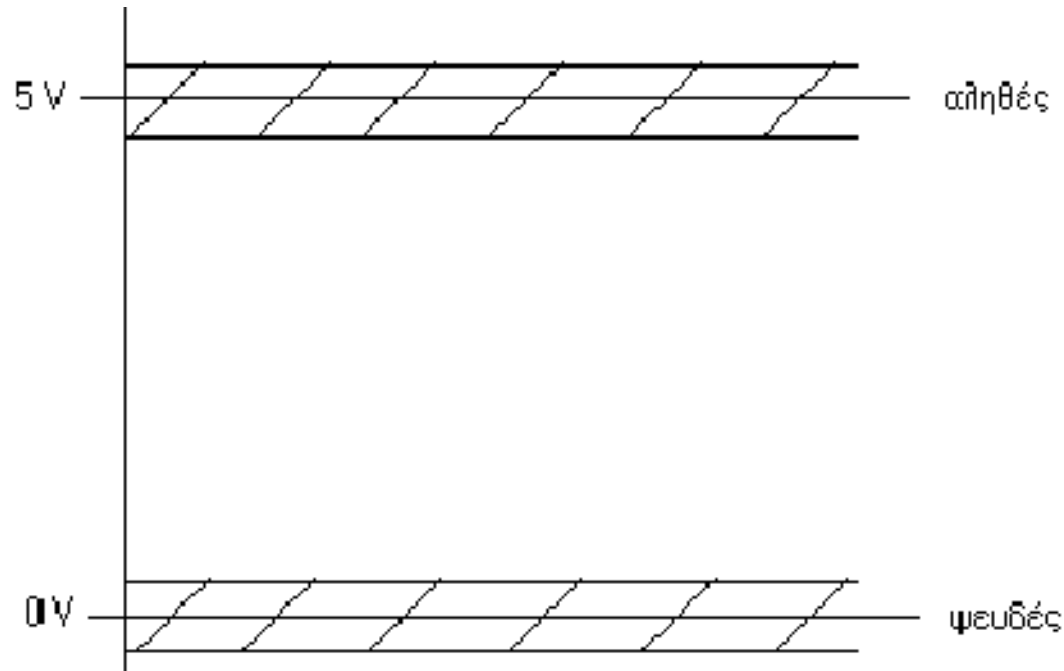
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Transistor count



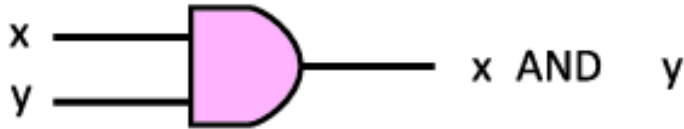
https://en.wikipedia.org/wiki/Transistor_count

Αληθές (1) / Ψευδές (0) \rightarrow 0 / +5 Volt

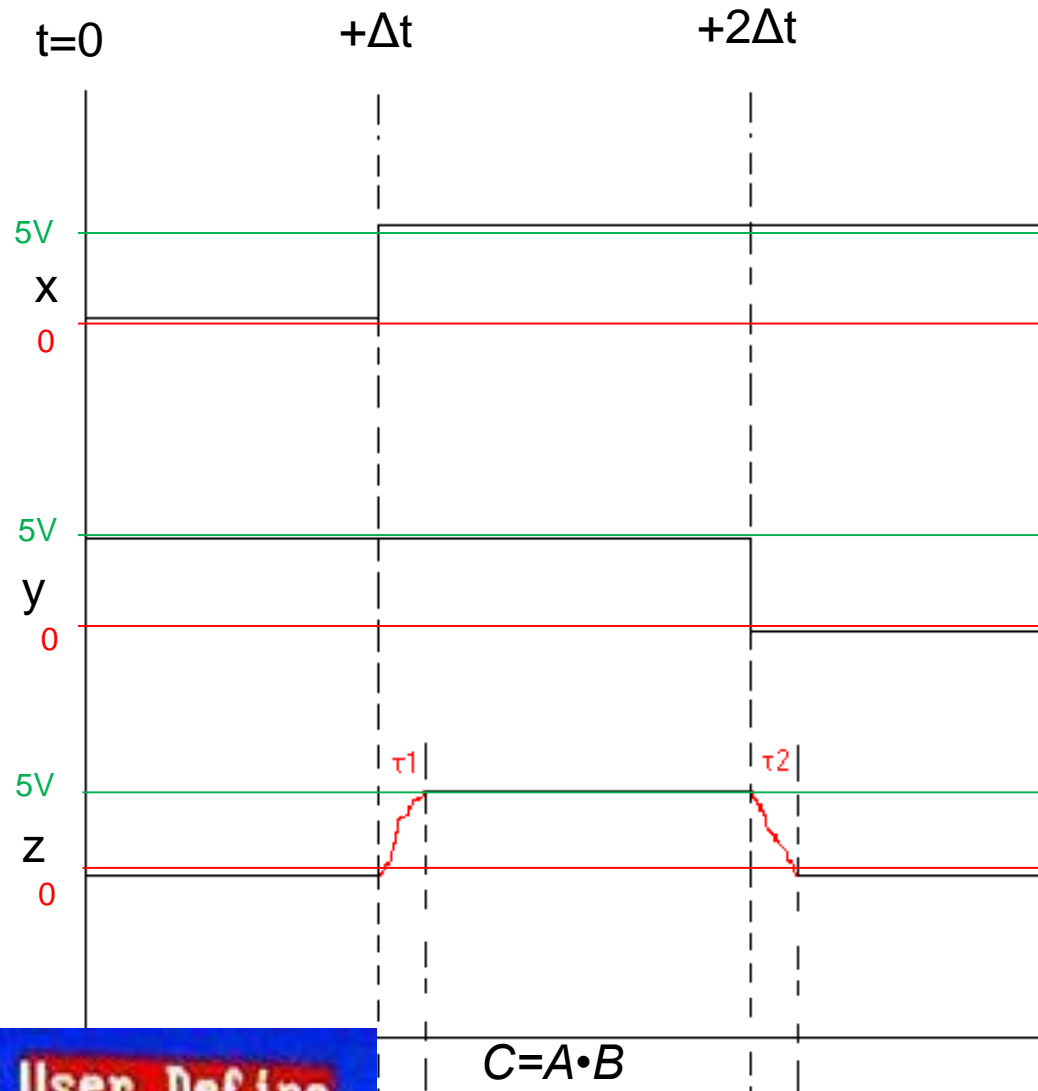


- Κατά την ηλεκτρονική υλοποίηση τύπου TTL (Transistor – to – Transistor Logic) αντιστοιχίζουμε το λογικό “0” (ψευδές) με τάση 0 Volt και το λογικό “1” (αληθές) με τάση +5 Volt. Όμως το αληθές και το ψευδές δεν αντιστοιχούν με απόλυτη ακρίβεια στα +5 και 0 Volt, αλλά σε μια περιοχή γύρω από αυτά.

Συχνότητα Υλοποίησης λογικής πράξης



- Έστω η λογική συνάρτηση
 $Z = f(X, Y) = X \cdot Y = X \text{ AND } Y$
- Παρατηρούμε ότι στις μεταβατικές καταστάσεις βρισκόμαστε εκτός των ορισθέντων ορίων για μικρό χρόνο τ , που λέγεται **χρονική καθυστέρηση** και δείχνει το χρόνο που πρέπει να περάσει από την αλλαγή μιας εισόδου για να σιγουρευτούμε για την τιμή της εξόδου.
- **Η περίοδος του συστήματος πρέπει να είναι μεγαλύτερη από την χρονική καθυστέρηση** για να αποφευχθεί το πρόβλημα του να βρεθεί το σύστημα σε «αόριστη» κατάσταση.
- Δηλαδή υπάρχει περιορισμός στη συχνότητα του συστήματος.
- Σήμερα έχει επιτευχθεί δραστική μείωση της χρονικής καθυστέρησης. Έτσι οι σημερινοί υπολογιστές έχουν φτάσει σε συχνότητες, της τάξης των GHz.

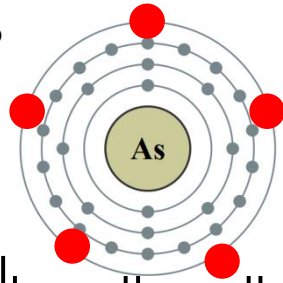


CPU Operating Speed
– External Clock

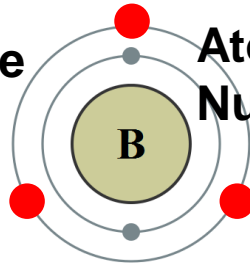
User Define
148 MHz

Κατασκευή Transistor

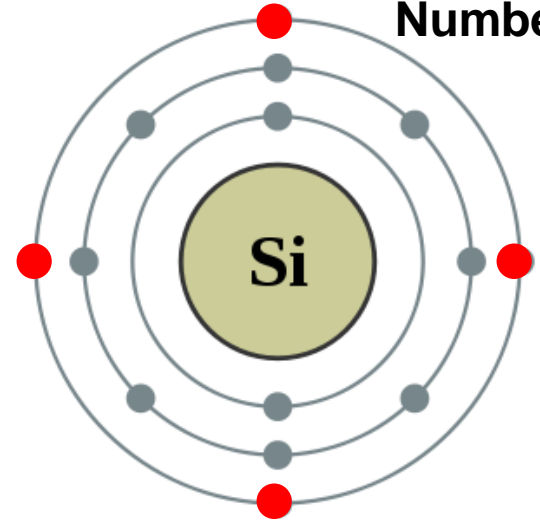
AN: 33



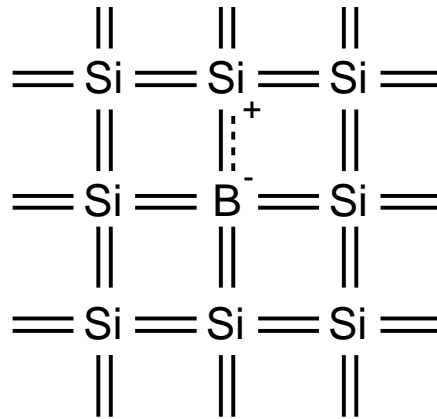
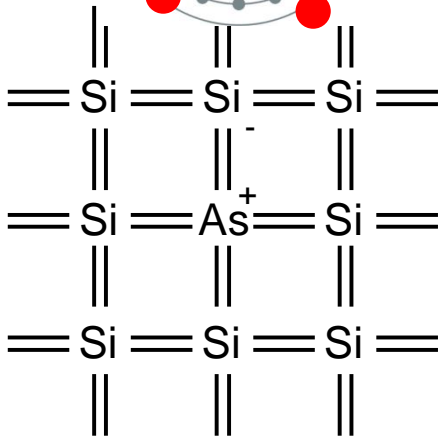
p-type Atomic Number: 5



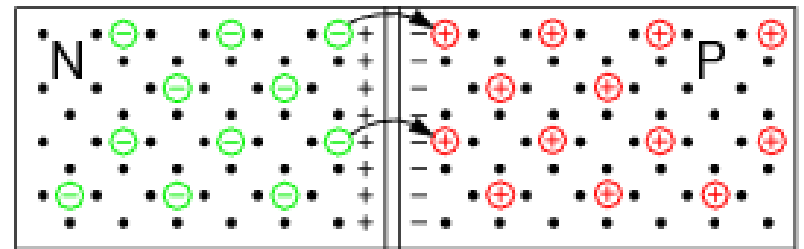
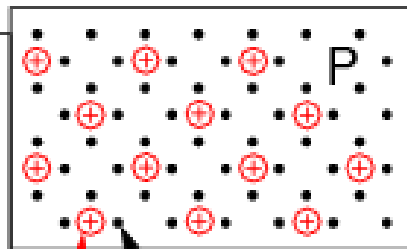
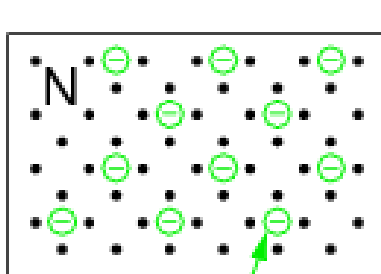
Atomic Number: 14



n-type



no charge separation



electron

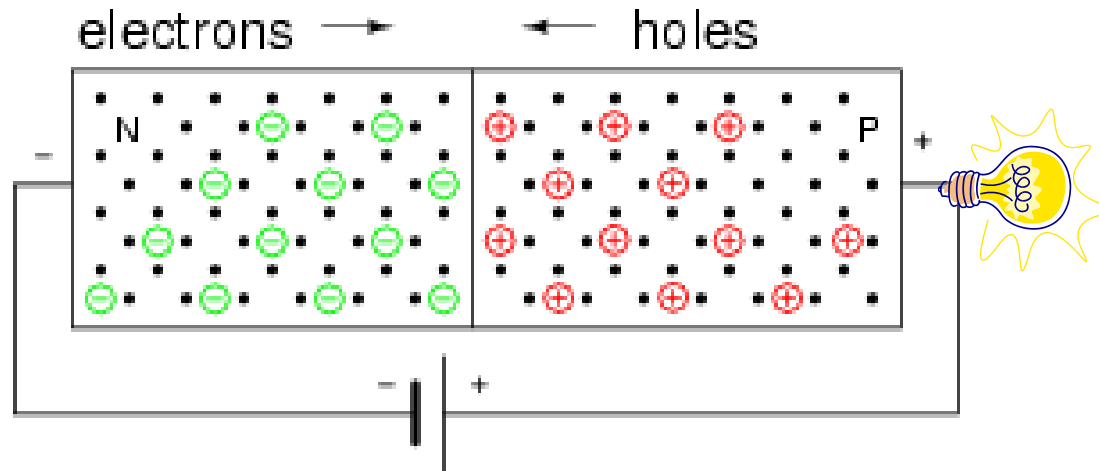
crystal lattice hole

intrinsic

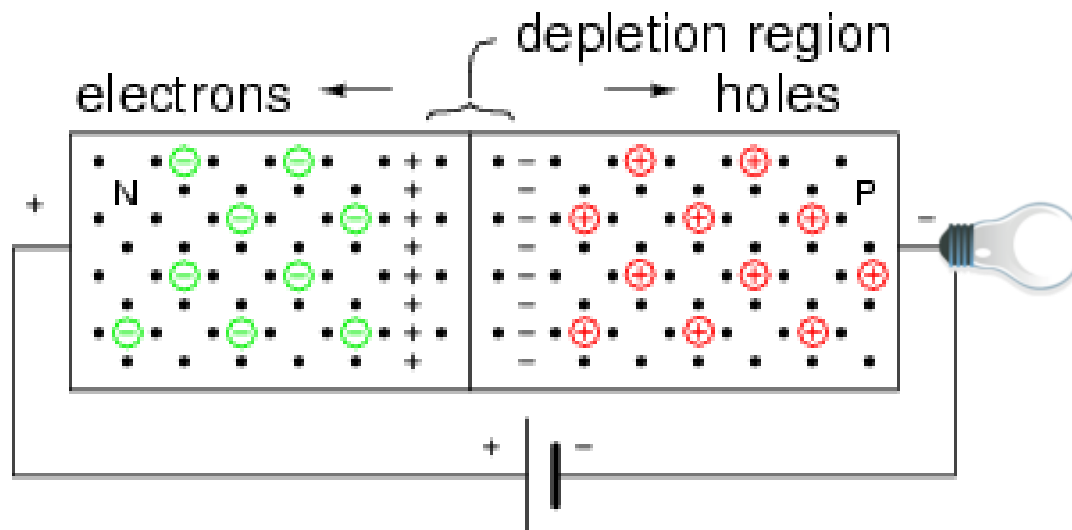


charge separation

Κατασκευή Transistor



(a) Forward



(b) Reverse

Κατασκευή Transistor



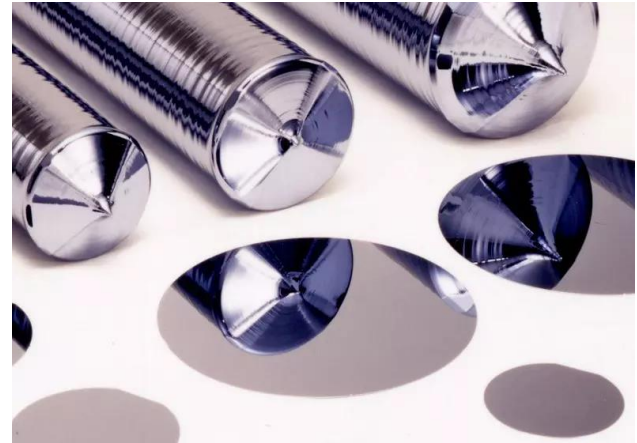
Clean Rooms

Κατασκευή Transistor

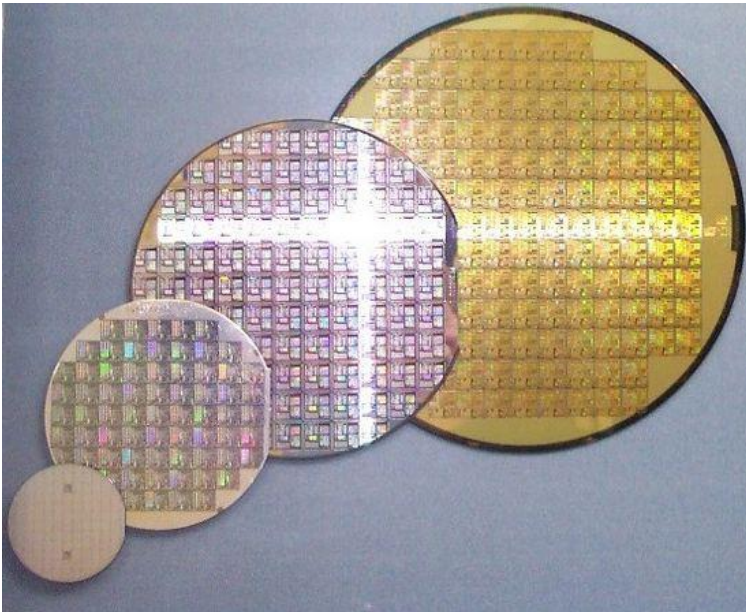
1. Sand (Si)



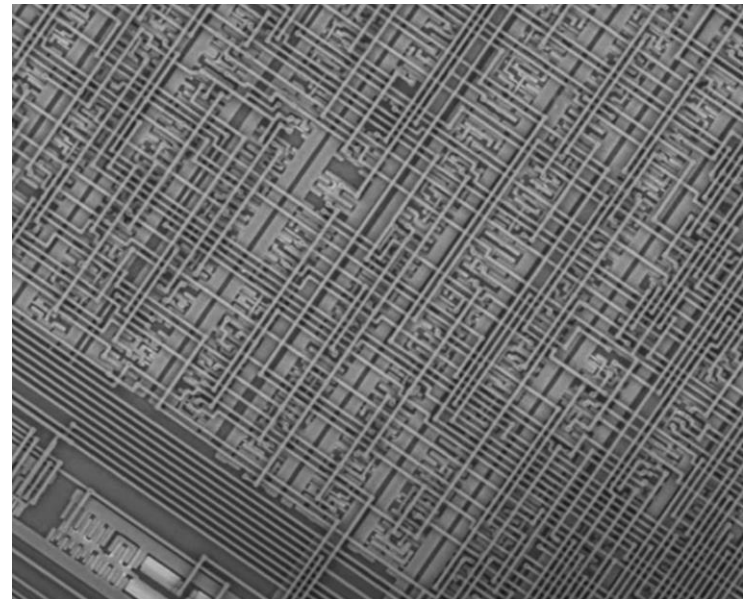
2. Wafers 2/4/6/8/12 inches



3. Photolithography on wafers



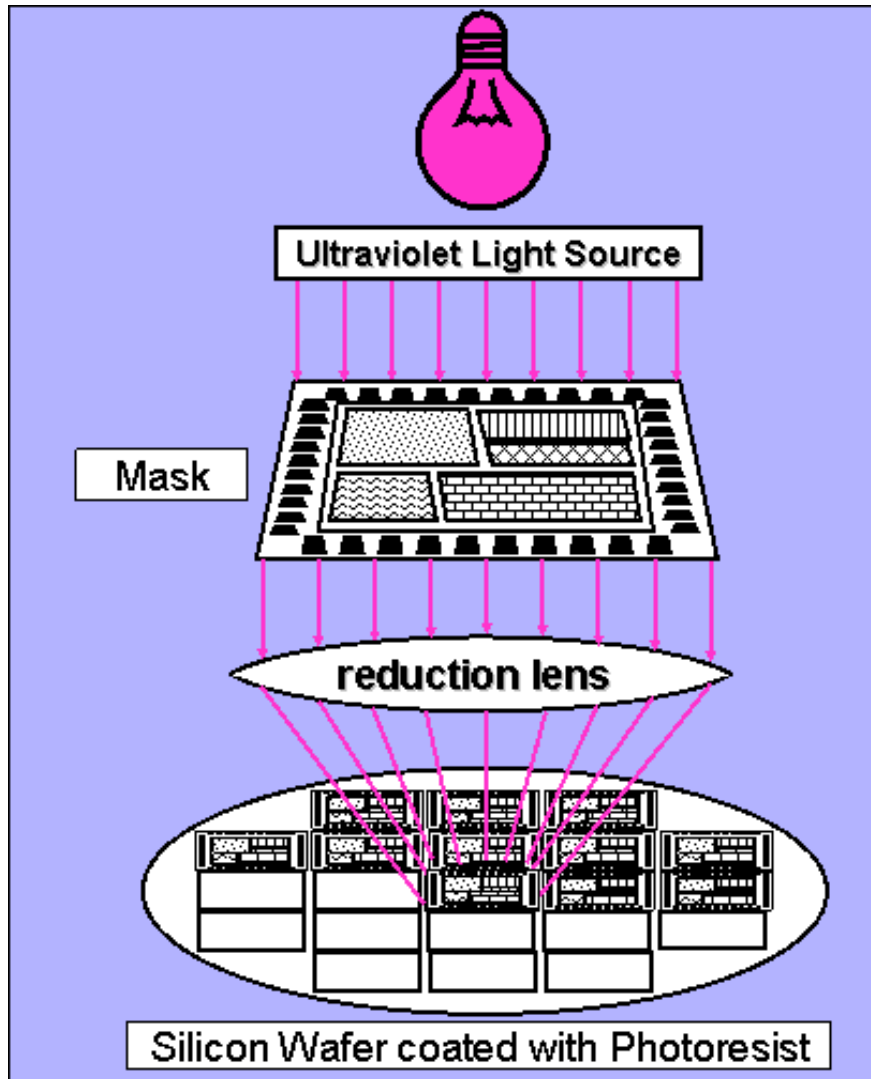
4. Wafers with several layers



Photolithography on wafers

<https://pt.coursera.org/lecture/nanotechnology/photolithography-sample-patterning-demonstration-TdtAR>

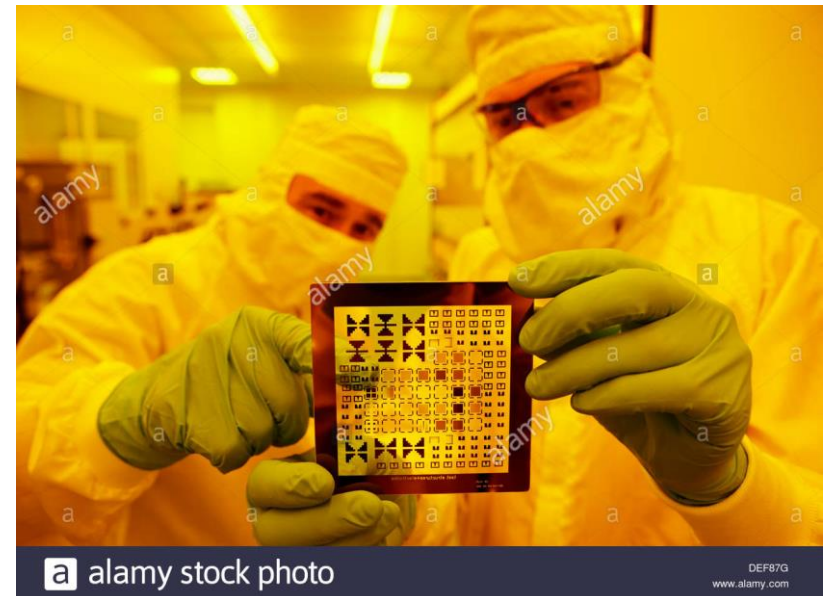
Photolithography



Wafer



Mask



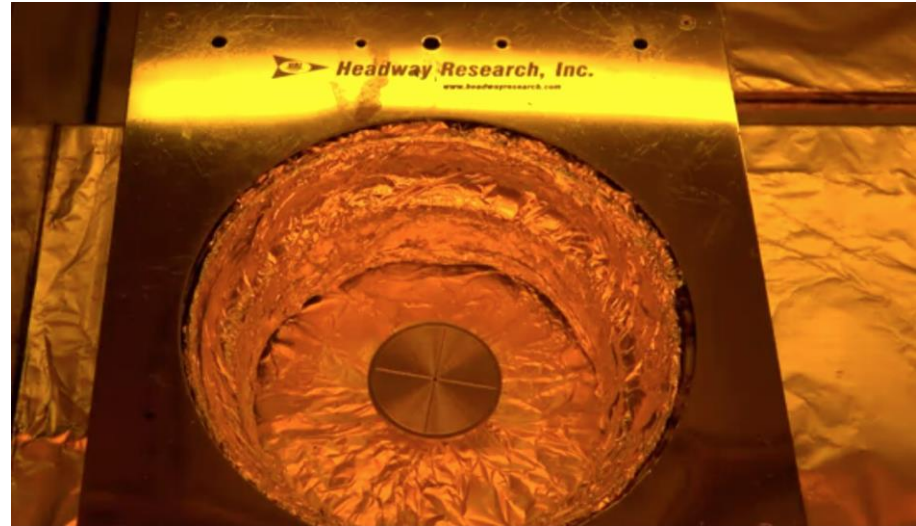
Photolithography on wafers

<https://pt.coursera.org/lecture/nanotechnology/photolithography-sample-patterning-demonstration-TdtAR>

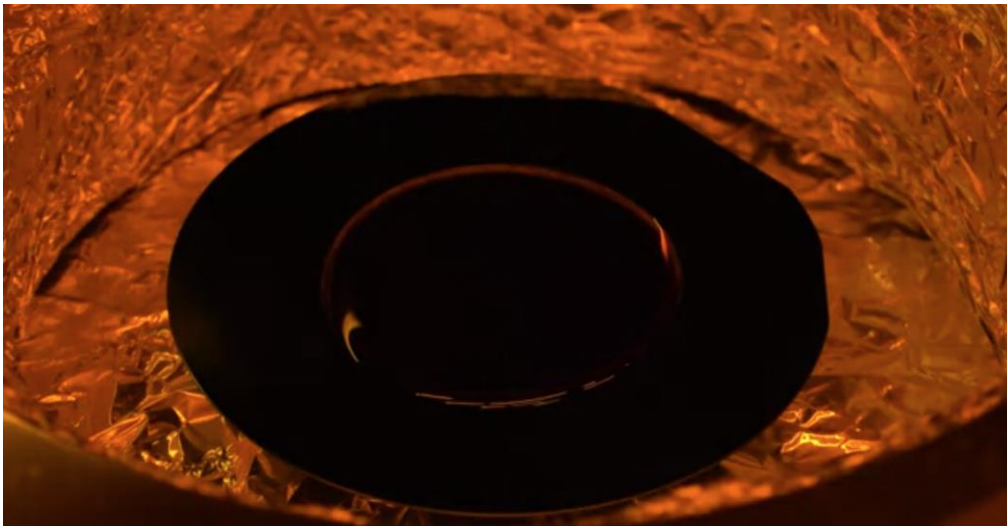
Wafer



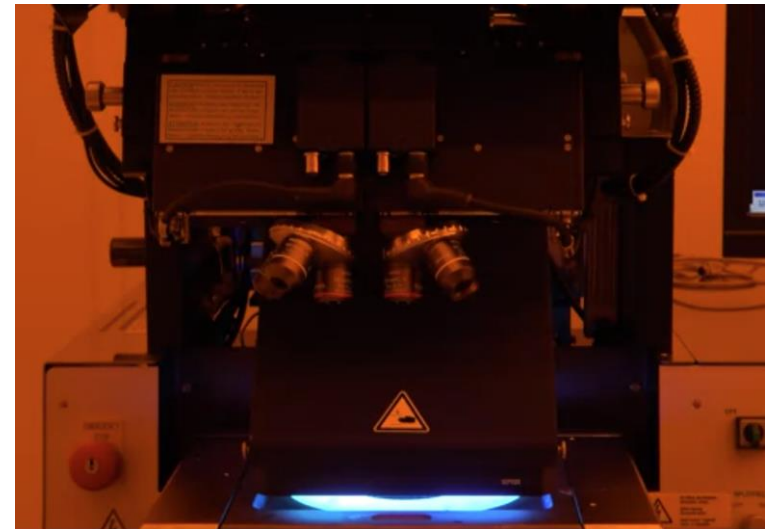
Spin Coater



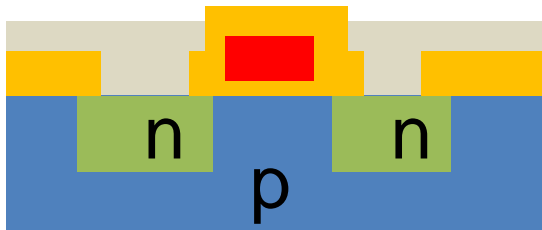
Photoresist








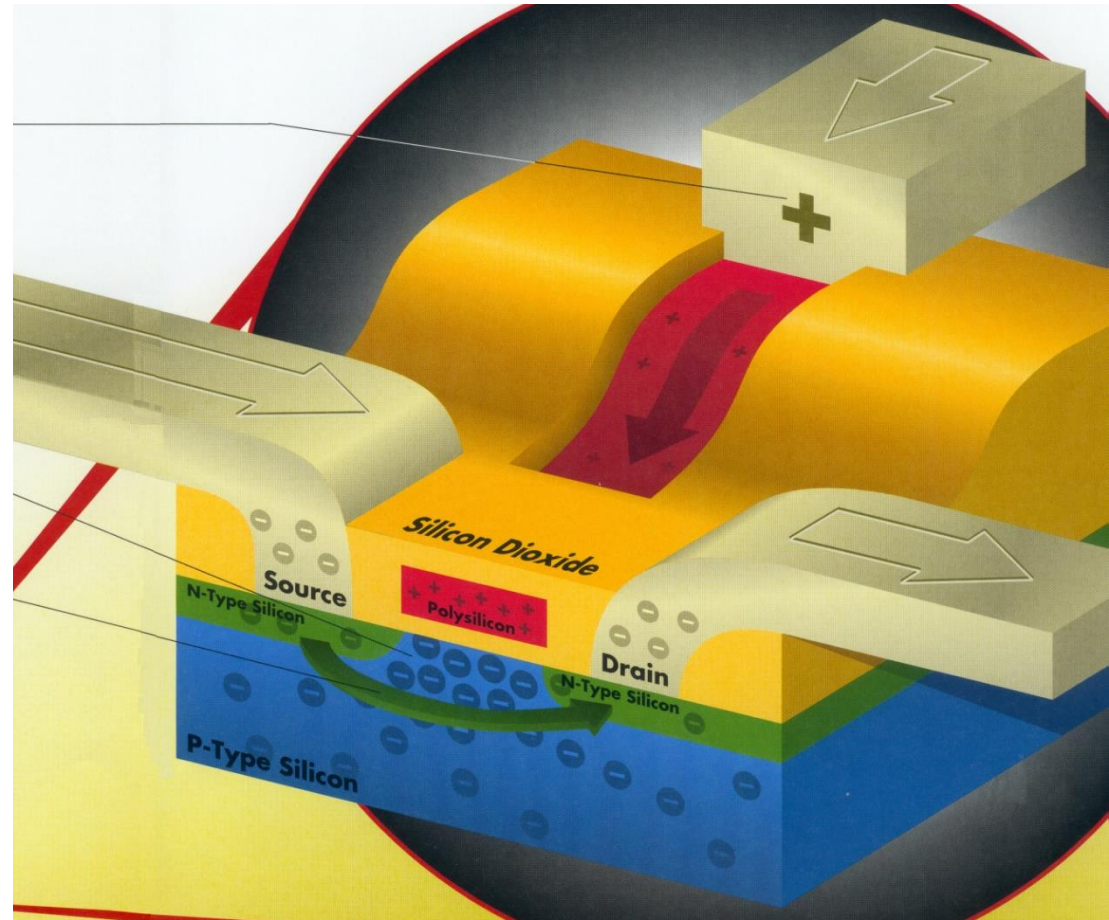
Exposure



Κατασκευή Transistor



-  Polysilicon
-  Metal
-  p substrate
-  n material
-  SiO₂

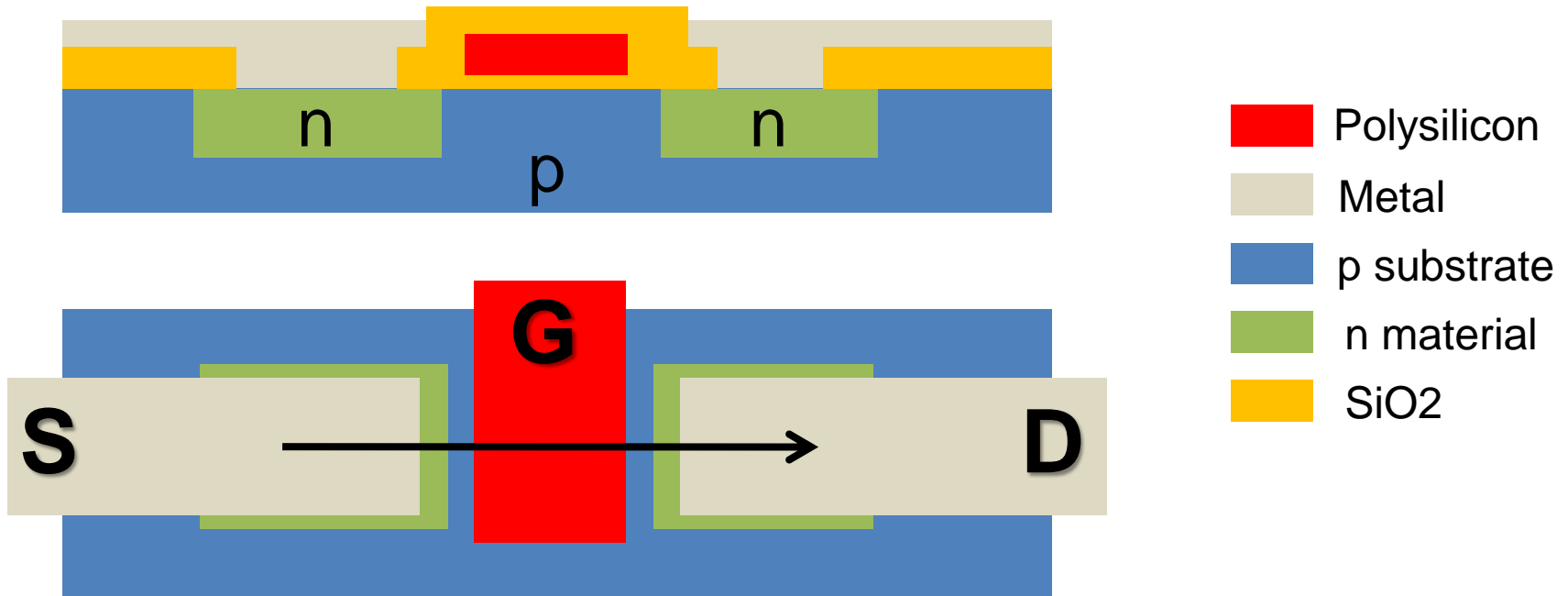


www.cmosvlsi.com/lect0.ppt

<http://lsmwww.epfl.ch/Education/former/2002-2003/VLSIDesign/ch02/ch02.html>

<http://www2.renesas.com/fab/en/line/line1.html>

Transistor Cross-section



- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- Polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



→ **Cover wafer with protective layer of SiO_2 (oxide)**

→ Spin photoresist

→ n-well mask

→ Etch oxide with hydrofluoric acid (HF!)

→ Strip off remaining photoresist (Piranah!)

→ Place wafer in furnace and heat until As atoms diffuse into exposed Si

→ Strip off the remaining oxide using HF

→ Deposit very thin layer of gate oxide

→ Place wafer in furnace with Silane gas (SiH_4) to form polysilicon

→ Polysilicon mask, etch oxide with HF

→ Cover wafer with SiO_2

→ Etch with HF to expose contacts

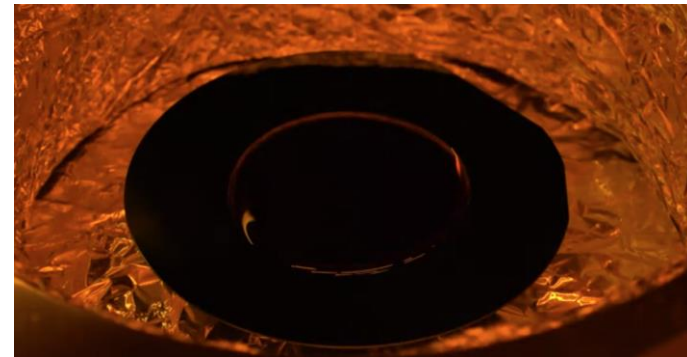
→ Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



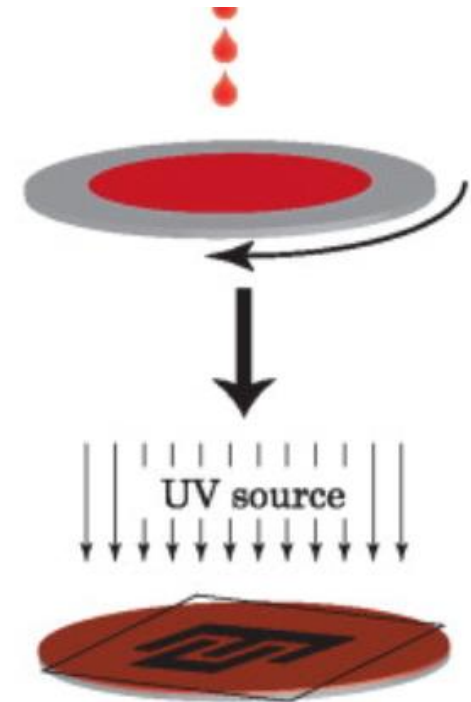
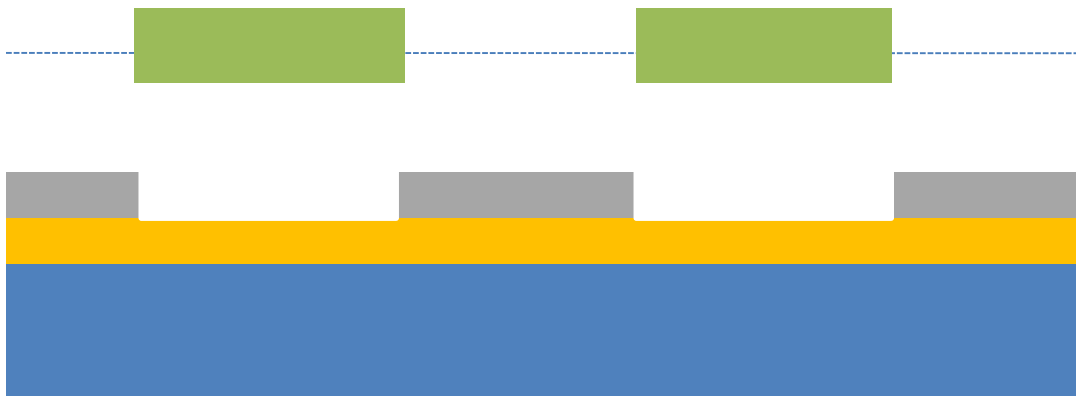
Grow SiO_2 on top of Si wafer

900 – 200 C with O_2 in oxidation furnace

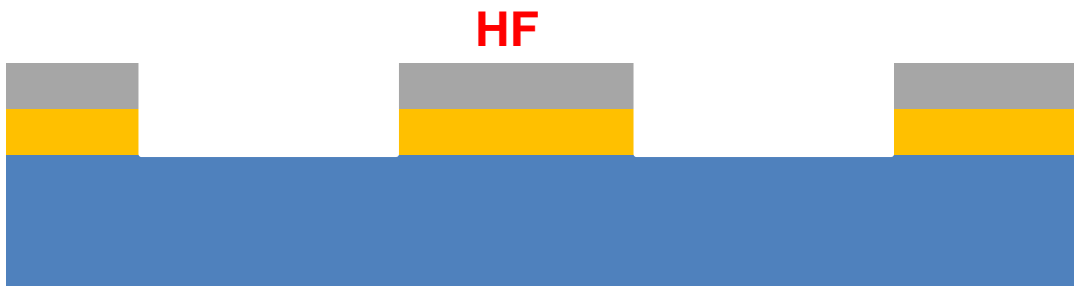
- Cover wafer with protective layer of SiO_2 (oxide)
- **Spin photoresist**
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- Polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- **n-well mask**
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- .
- .
- .



- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- **Etch oxide with hydrofluoric acid (HF!)**
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- Polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- **Strip off remaining photoresist (Piranah!)**
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- Polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires

Piranah



- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- **Place wafer in furnace and heat until As atoms diffuse into exposed Si**
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- Polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- **Strip off the remaining oxide using HF**
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- Polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires

HF



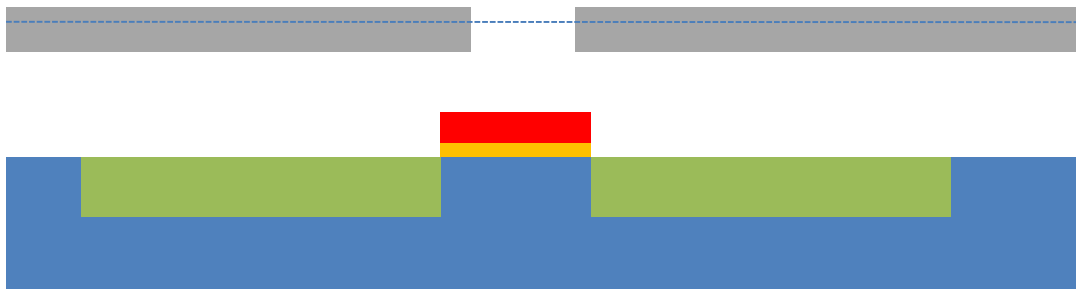
- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- **Deposit very thin layer of gate oxide**
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- Polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- **Place wafer in furnace with Silane gas (SiH_4) to form polysilicon**
- Polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- **Polysilicon mask, etch oxide with HF**
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



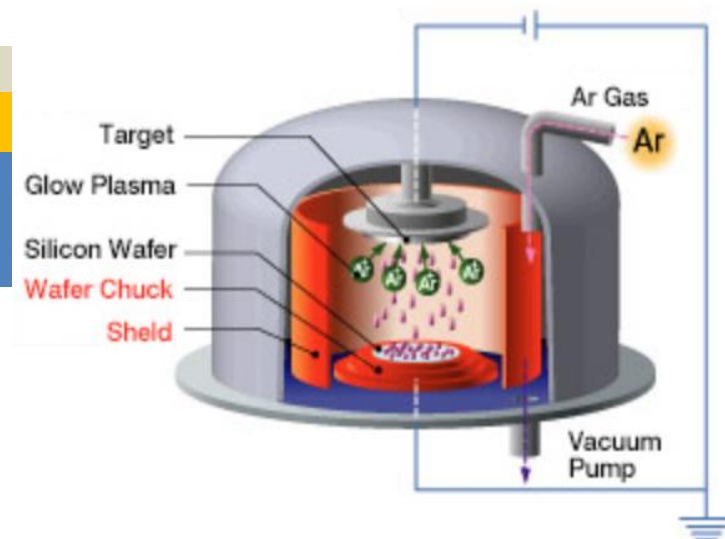
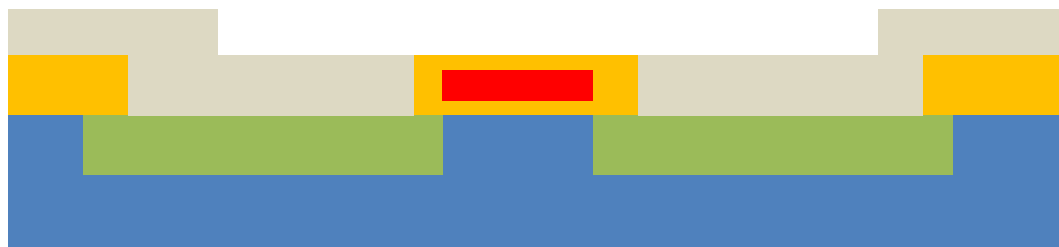
- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- polysilicon mask, etch oxide with HF
- **Cover wafer with SiO_2**
- Etch with HF to expose contacts
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



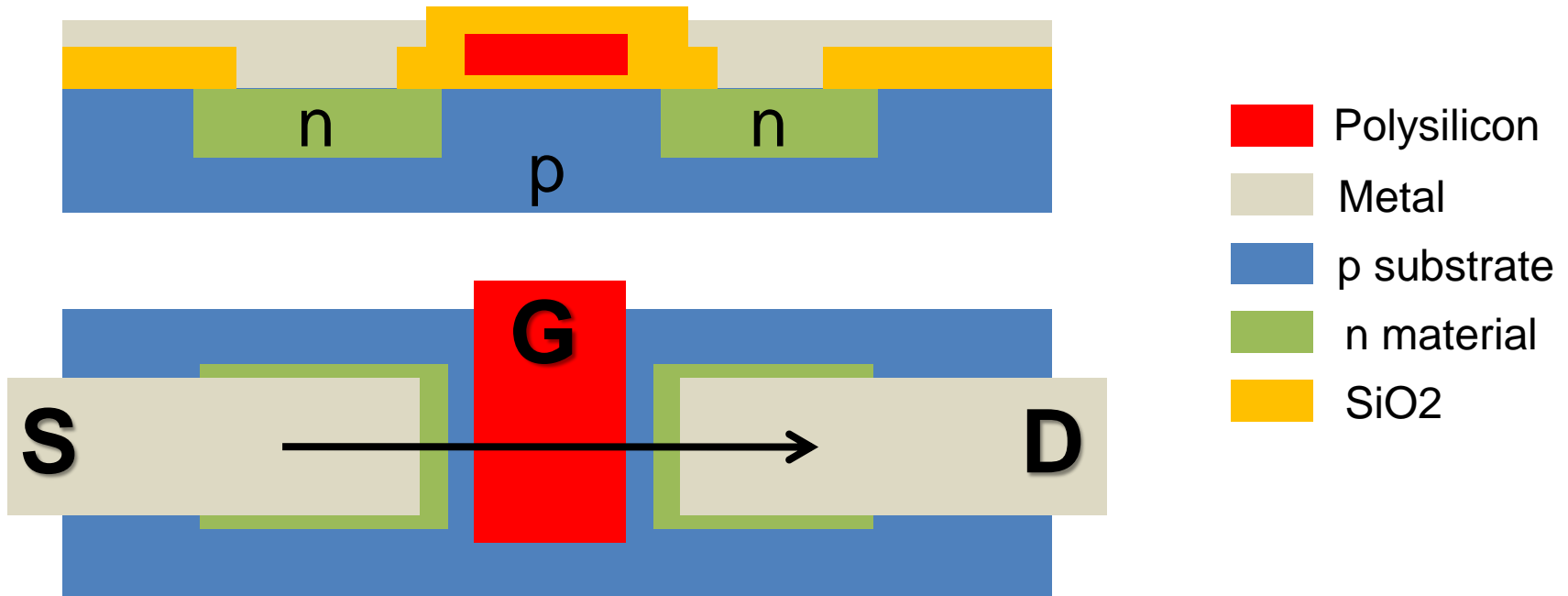
- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- **Etch with HF to expose contacts**
- Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires



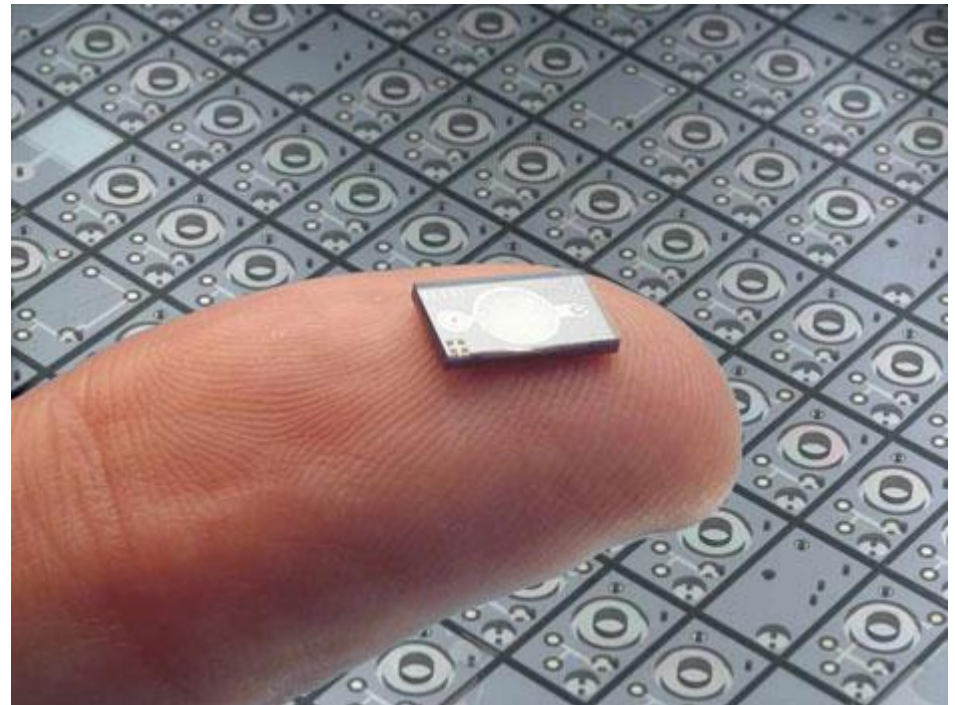
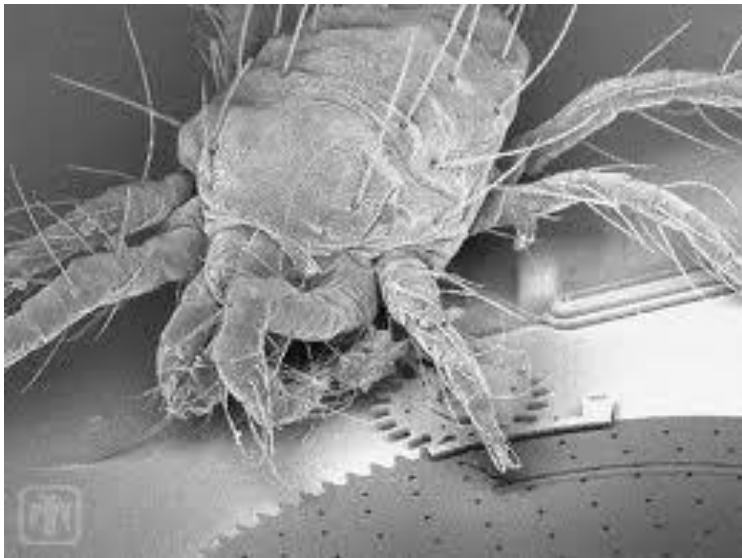
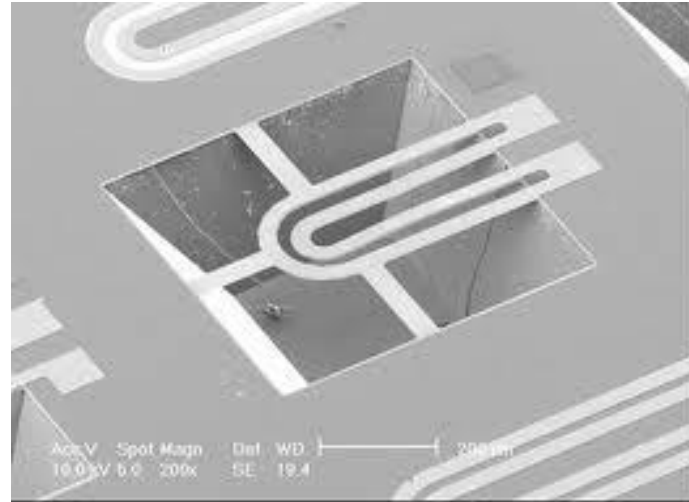
- Cover wafer with protective layer of SiO_2 (oxide)
- Spin photoresist
- n-well mask
- Etch oxide with hydrofluoric acid (HF!)
- Strip off remaining photoresist (Piranah!)
- Place wafer in furnace and heat until As atoms diffuse into exposed Si
- Strip off the remaining oxide using HF
- Deposit very thin layer of gate oxide
- Place wafer in furnace with Silane gas (SiH_4) to form polysilicon
- polysilicon mask, etch oxide with HF
- Cover wafer with SiO_2
- Etch with HF to expose contacts
- **Sputter on aluminum over whole wafer and pattern to remove excess metal, leaving wires**



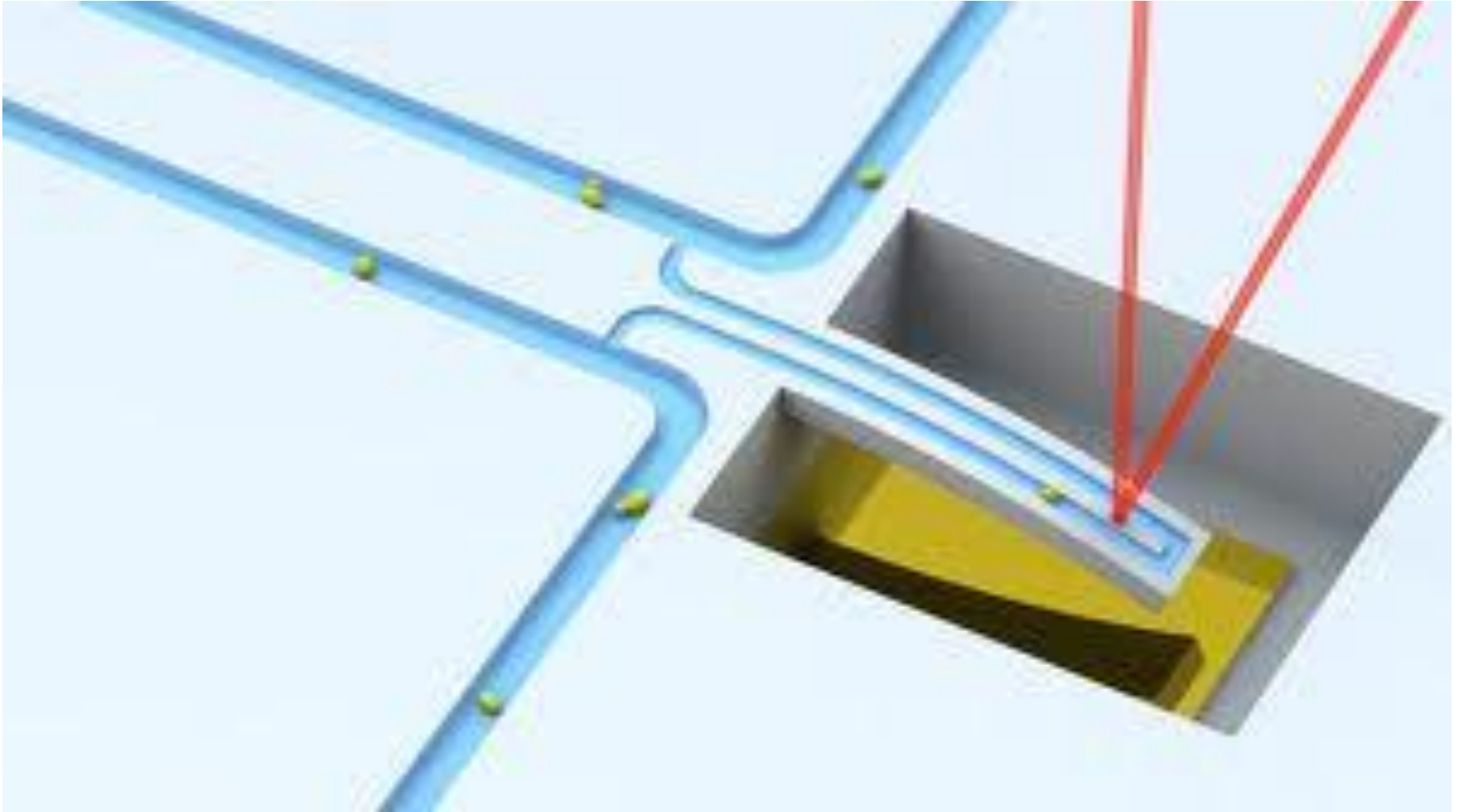
Transistor Cross-section



Fab for MEMS

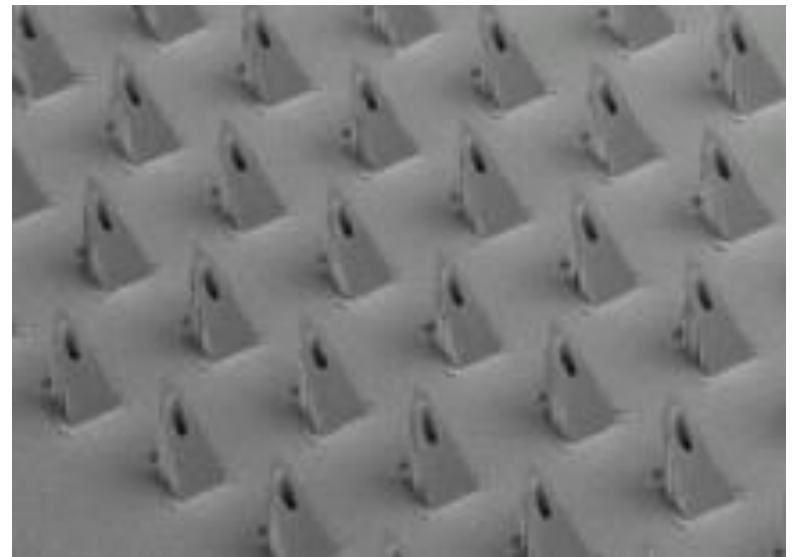
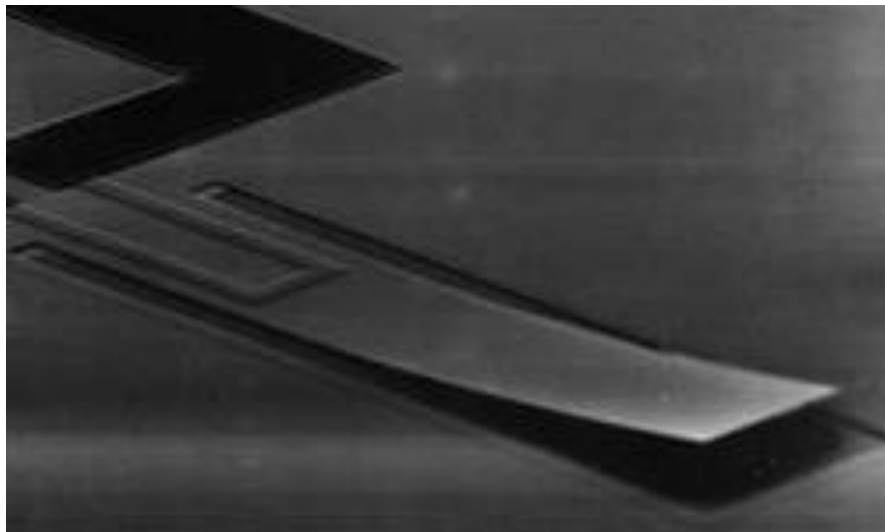
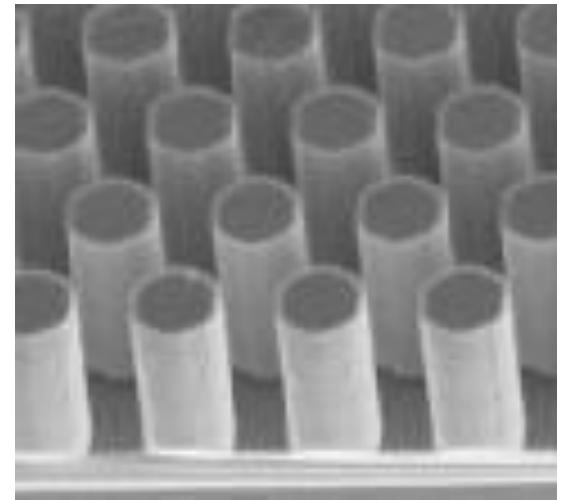
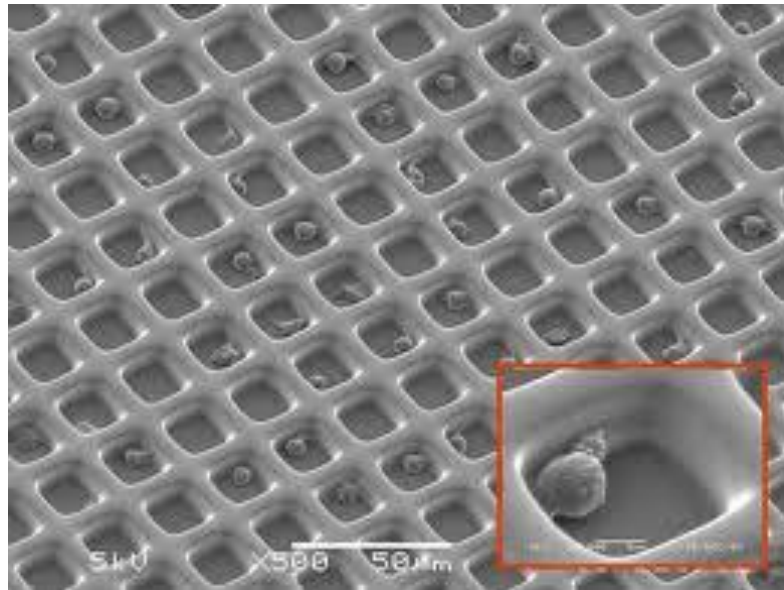


Fab for BIO-MEMS



Manalis Lab, MIT

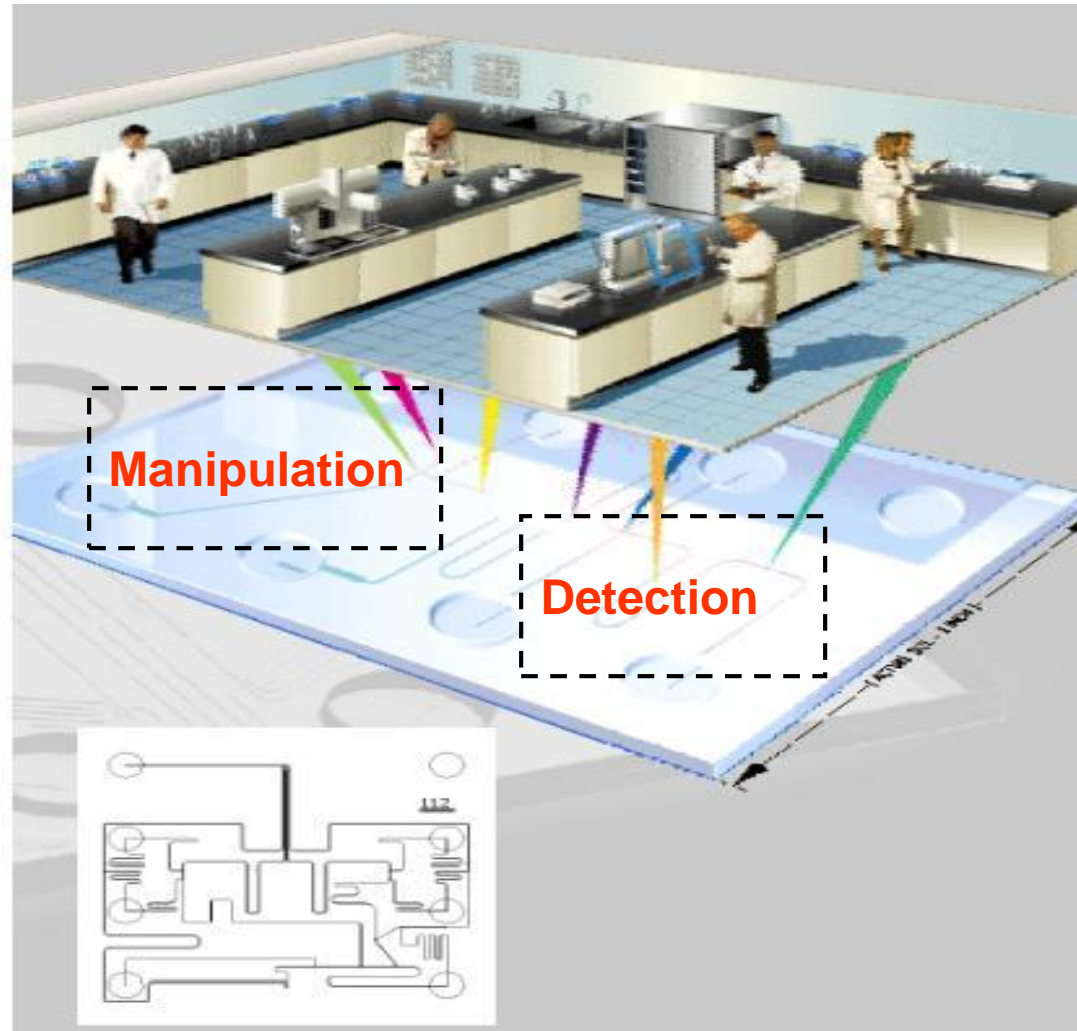
Fab for BIO-MEMS



Lab-on-Chip Technology

Features/Benefits

- Miniaturization
- Automation
- Integration
- Data Quality
- Reagent Savings
- Portability
- Precision



Σήμερα:

Μέρος 1ο: (όχι στον Forouzan – εκτός ύλης)

Εισαγωγή σε

- Transistors
- Υλοποίηση με Transistor
- Κατασκευή Transistors (εκτός ύλης)

Μέρος 2ο:

Forouzan: μόνο 4.1 & Παράρτημα Ε (εκτός 4.2/4.3)

Εισαγωγή σε:

- Λογικές μεταβλητές, πύλες, συναρτήσεις
- Πίνακας αληθείας, Σύνθεση/ Ανάλυση
- Παραδείγματα / Ασκήσεις

Άλγεβρα Boole & Λογικά Κυκλώματα

Σταθερές Άλγεβρας Boole

“0”: ψευδές
θέση OFF
0V
Ανοιχτός διακόπτης
Σβηστή λάμπα

“1”: Αληθές
θέση ON
5V
κλειστός διακόπτης
αναμένη λάμπα

Μεταβλητές Άλγεβρας Boole

x, y, z, \dots Όπου παίρνουν τιμές 0 / 1

Τελεστές (Boolean operators)

Λογική Πράξη	Μαθηματικός Συμβολισμός
NOT	\bar{A}, \bar{B}
AND	$A \cdot B$
OR	$A + B$

(συμπλήρωμα)

(γινόμενο)

(άθροισμα)

Το "και" δεν είναι AND!

AND \rightarrow \cdot
OR \rightarrow $+$
NOT \rightarrow $'$

Παραδείγματα Πράξεων

$X+Y \rightarrow X \text{ OR } Y$
 $X \cdot Y \rightarrow X \text{ AND } Y$

(AND)

$$x \cdot 1$$

(OR)

$$x + 0$$

$$x \cdot y$$

$$x \cdot y$$

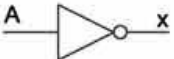






$$x \cdot y \cdot z \cdot 1$$

$$x + y + z$$

$$x \cdot (y + z)$$

$$(ABC'D) + (AB'C'D) + (A'BCD) + (AB'CD) + (A'BCD')$$

Λογικές Πύλες

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR																																																																																																
Alg. Expr.	\bar{A}	AB	\overline{AB}	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$																																																																																																
Symbol																																																																																																							
Truth Table	<table border="1"> <thead> <tr> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	X	0	1	1	0	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	B	A	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	B	A	X	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"> <thead> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	1
A	X																																																																																																						
0	1																																																																																																						
1	0																																																																																																						
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	1																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					

Οι πύλες NAND και NOR ονομάζονται «ΟΙΚΟΥΜΕΝΙΚΕΣ ΠΥΛΕΣ» γιατί οι πύλες αυτές μπορούν να παράξουν όλες τις AND, OR, NOT.

Κάθε ψηφιακό κύκλωμα μπορεί να υλοποιηθεί μόνο με πύλες NAND ή μόνο με πύλες NOR

AND → •
OR → +
NOT → '

Κάντε τις λογικές πράξεις

$$0 \text{ AND } 1 = ? \quad 0$$

$$0 \bullet 1 = ? \quad 0$$

$$x \bullet 0 = ? \quad 0$$

$$x \text{ OR } 0 = ? \quad x$$

$$x + 0 = ? \quad x$$

$$(1 + 1) + 1 = ? \quad 1$$

$$(1 + 1) + 0 = ? \quad 1$$

$$(1 + 0) \bullet 1' = ? \quad 0$$

$$x + x = ? \quad x$$

$$x + x' = ? \quad 1$$

$$x + 1 = ? \quad 1$$

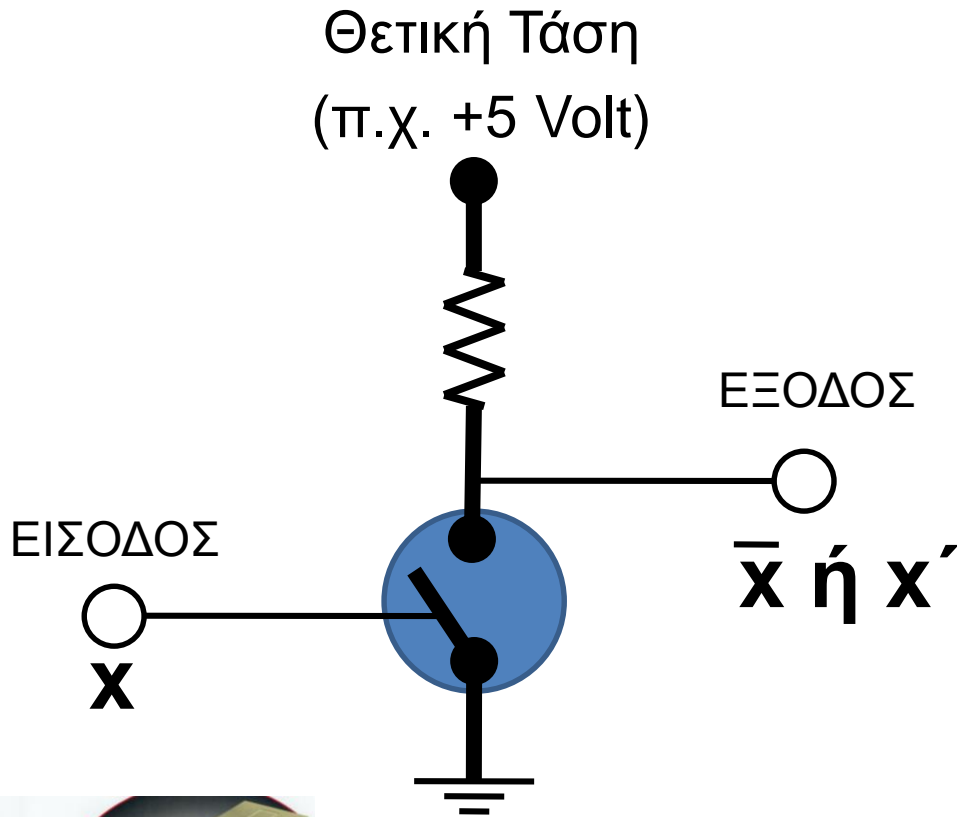
$$x \bullet x = ? \quad x$$

$$x \bullet x' = ? \quad 0$$

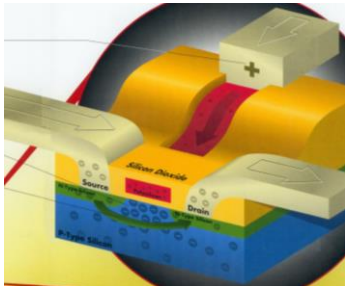
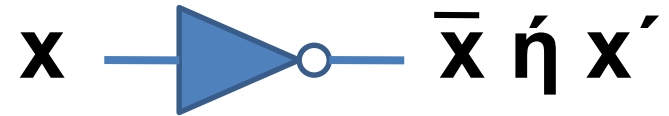
$$(1 + 0) \bullet 1 = ? \quad 1$$

η λογική πράξη AND (•) προηγείται της λογικής πράξης OR (+). Σε κάθε περίπτωση όμως χρησιμοποιούμε παρενθέσεις ανάμεσα στις λογικές πράξεις AND OR που κάνουν εμφανή την προτεραιότητα.

Υλοποίηση Πύλης NOT



x	x'
0	1
1	0

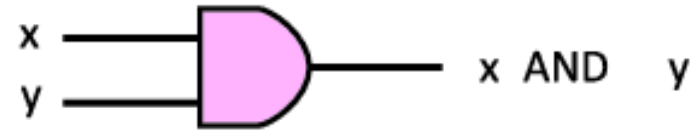


Πύλη AND (.)

ΕΞΟΔΟΣ



$(x \cdot y)$

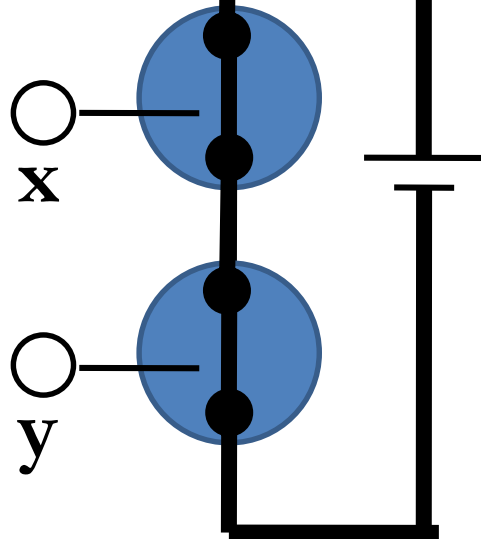


ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

x	y	$(x \cdot y)$
0	0	0
0	1	0
1	0	0
1	1	1

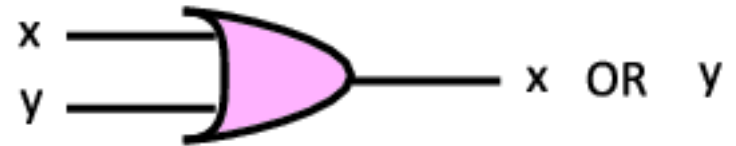


ΕΙΣΟΔΟΣ
(0: Ανοιχτή)

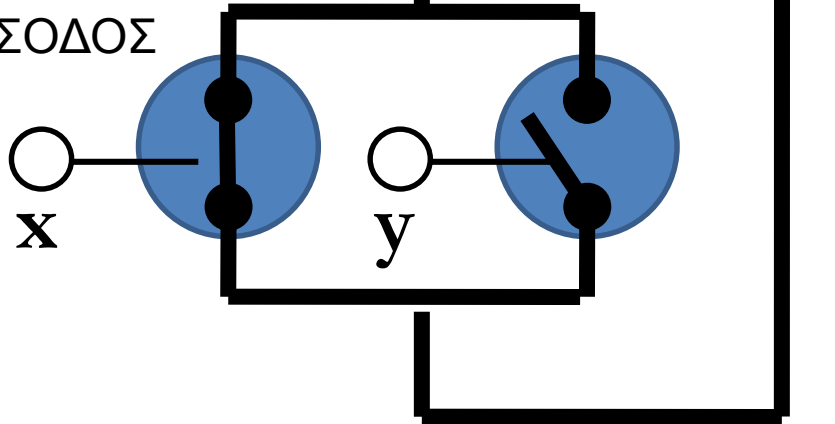


Πύλη OR (+)

ΕΞΟΔΟΣ
 $(x + y)$



ΕΙΣΟΔΟΣ

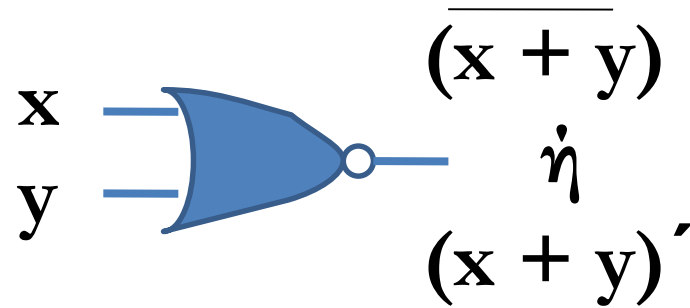


ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ

x	y	$(x + y)$
0	0	0
0	1	1
1	0	1
1	1	1

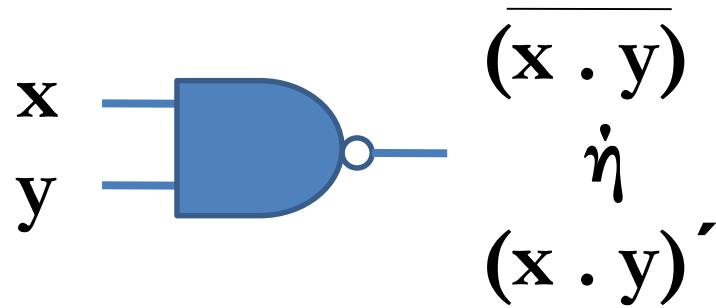


Πύλη NOR



x	y	$(x + y)'$
0	0	1
0	1	0
1	0	0
1	1	0

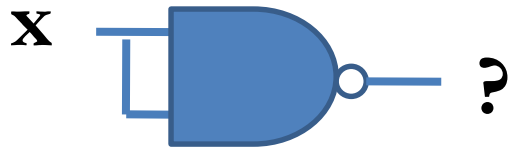
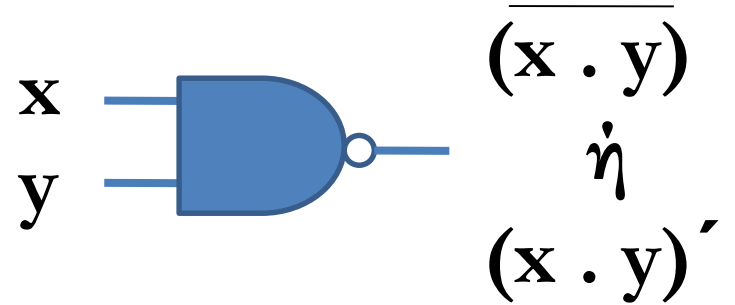
Πύλη NAND



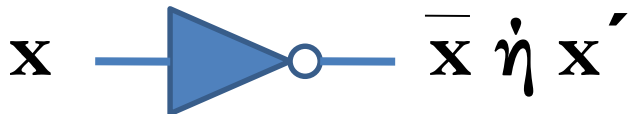
x	y	$(x \cdot y)'$
0	0	1
0	1	1
1	0	1
1	1	0

Υλοποίηση άλλων πυλών από NAND

Τι συμβαίνει
όταν $x=y$;



1 NAND

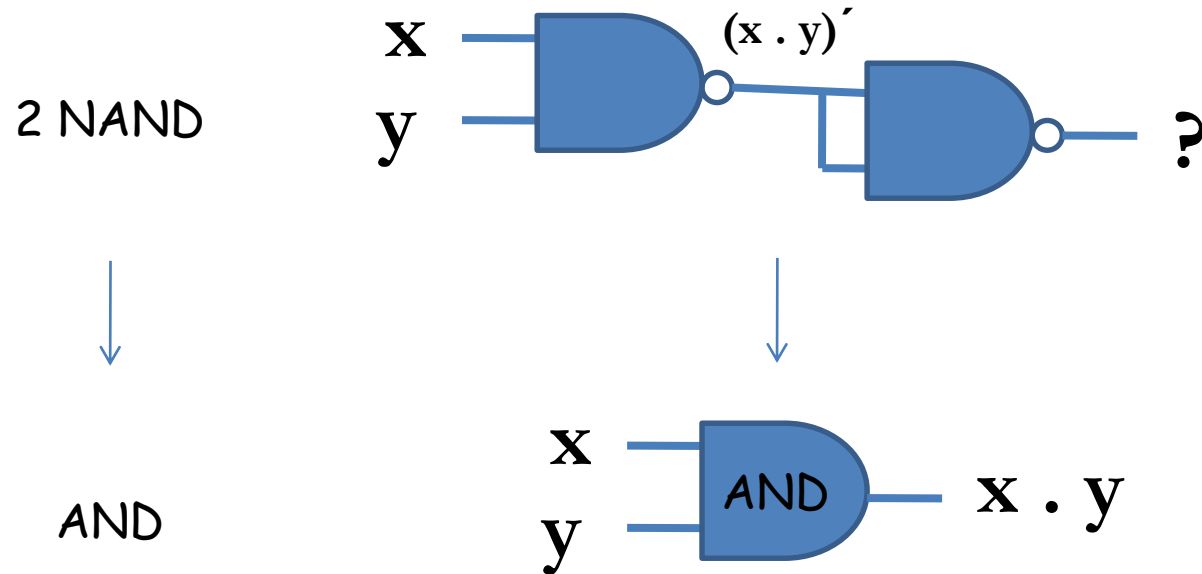


NOT

x	y	$(x \cdot y)'$
0	0	1
0	1	1
1	0	1
1	1	0

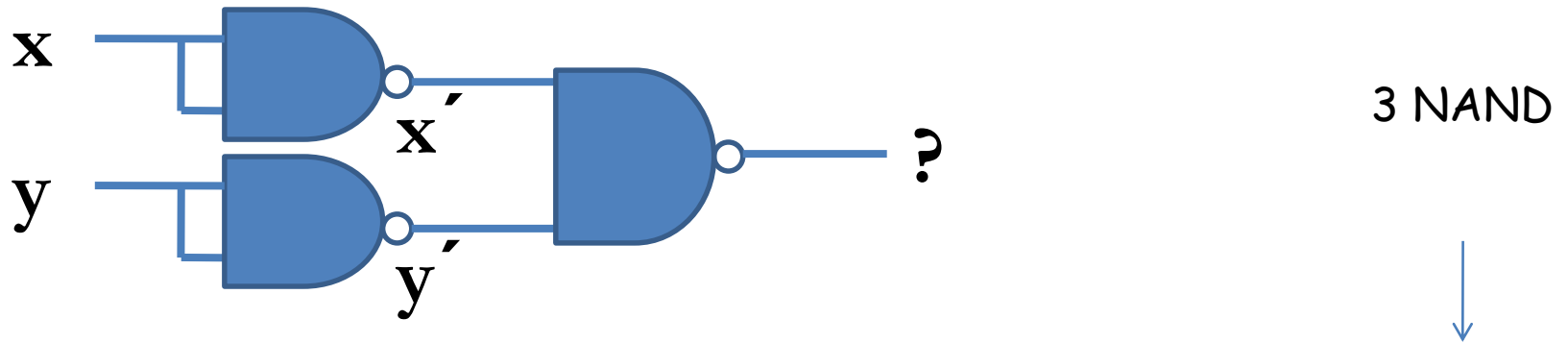
Υλοποίηση άλλων πυλών από NAND

Τι συμβαίνει όταν συνδέσουμε μία NOT με μία NAND;



Υλοποίηση άλλων πυλών από NAND

Τι συμβαίνει με την παρακάτω συνδεσμολογία;



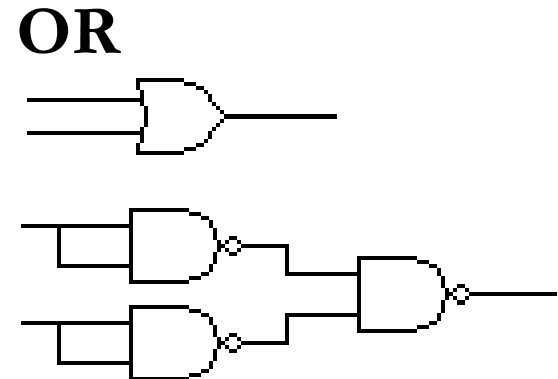
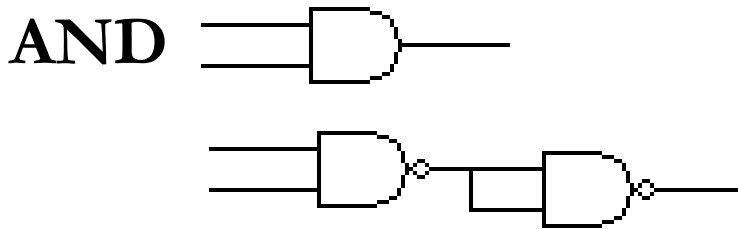
x	y	x'	y'	x' NAND y'
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1



Μία πύλη --> όλες οι πράξεις

- Κάθε ψηφιακό κύκλωμα μπορεί να υλοποιηθεί μόνο με πύλες NAND ή μόνο με πύλες NOR (ΟΙΚΟΥΜΕΝΙΚΕΣ ΠΥΛΕΣ) γιατί οι πύλες αυτές μπορούν να παράξουν όλα τα AND, OR, NOT

-- NAND --



AND → ·
OR → +
NOT → '

Λογικές συναρτήσεις

Συναρτήσεις: $z = f(x_1, x_2, \dots, x_n) \in \{0,1\}$ $x_i \in \{0,1\}, i = 1, 2, \dots, n$

Η λογική συνάρτηση z των n μεταβλητών ορίζεται σε 2^n σημεία που αντιστοιχούν στους n συνδυασμούς τους.

Παραδείγματα

$$Z = x \cdot 1$$

$$Z = x \cdot y'$$

$$Z = x + 0$$

$Z = x \cdot y \cdot 1 \leftarrow$ Σε πόσα σημεία ορίζεται η Z ?

$Z = x \cdot (y+c) \leftarrow$ Σε πόσα σημεία ορίζεται η Z ?

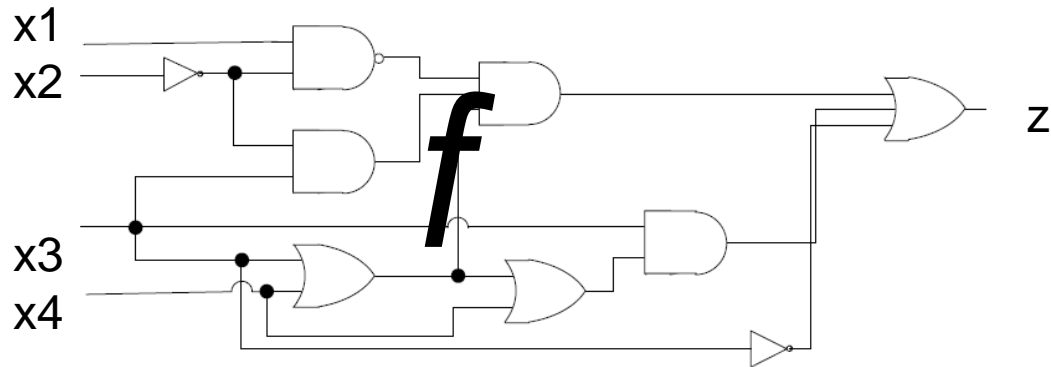
Λογικές Μεταβλητές, Πύλες, Πράξεις, Συναρτήσεις

- **Λογικές μεταβλητές:** x_i όπου x_i 0 ή 1
- **Λογικές πύλες:** απλές και παράγωγες.
- **Λογικές πράξεις:** απλές λογικές συναρτήσεις της μαθηματικής λογικής στις οποίες ανάγονται όλες οι λογικές συναρτήσεις.
- **Λογικές συναρτήσεις:** συναρτήσεις όπου τόσο οι ανεξάρτητες μεταβλητές όσο και η εξαρτημένη μεταβλητή είναι όλες λογικές μεταβλητές, δηλαδή
$$z = f(x_1, x_2, \dots, x_n) \hat{\in} \{0,1\} \quad x_i \hat{\in} \{0,1\}, i = 1, 2, \dots, n$$
- Επειδή κάθε μεταβλητή x μπορεί να λάβει 2 τιμές, η λογική συνάρτηση z των n μεταβλητών ορίζεται σε 2^n σημεία.
- **Πίνακας αληθείας:** εργαλείο συστηματικής παράστασης όλων των τιμών μιας λογικής συνάρτησης που αντιστοιχούν στους συνδυασμούς των x_i μεταβλητών της.

Λογικές Συναρτήσεις σε λογικό διάγραμμα

Συναρτήσεις όπου τόσο οι ανεξάρτητες μεταβλητές όσο και η εξαρτημένη μεταβλητή είναι όλες λογικές μεταβλητές, δηλαδή

$$z = f(x_1, x_2, \dots, x_n) \quad z \in \{0,1\} \quad x_i \in \{0,1\}, i = 1, 2, \dots, n$$

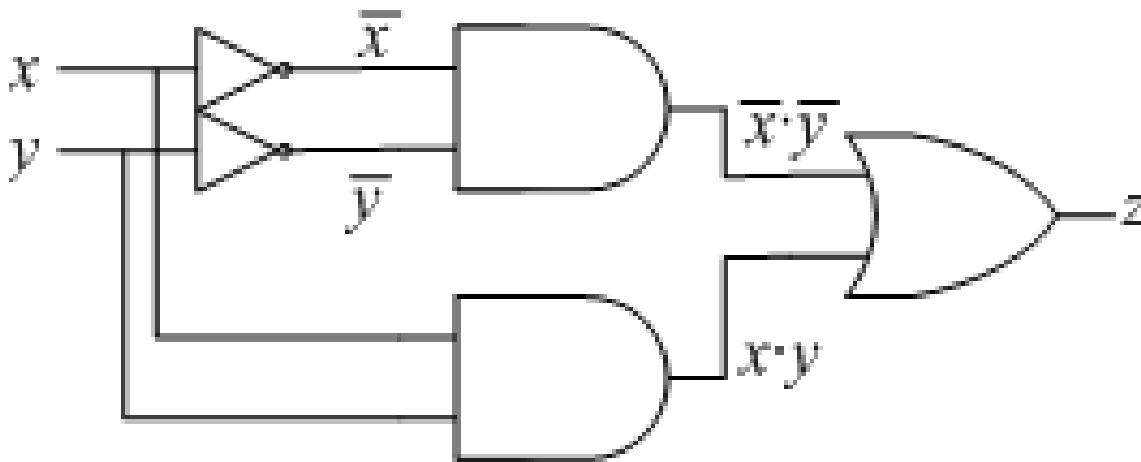


Η λογική συνάρτηση z των n μεταβλητών ορίζεται σε 2^n σημεία που αντιστοιχούν στους n συνδυασμούς τους.

Πίνακες Αληθείας

Πίνακας **όλων** των τιμών μιας λογικής συνάρτησης που αντιστοιχούν στους συνδυασμούς των μεταβλητών της.

Λογικό Διάγραμμα



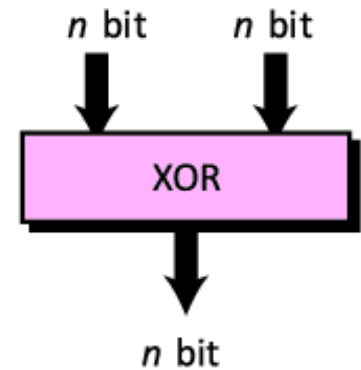
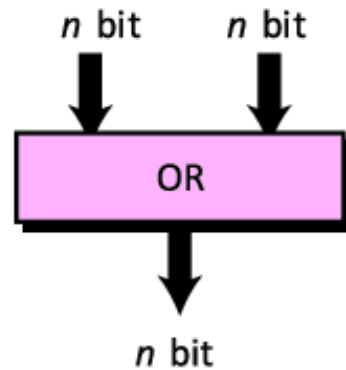
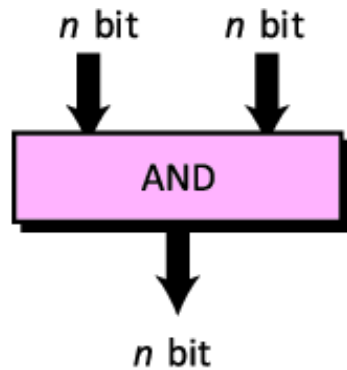
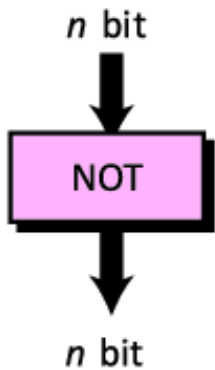
Πίνακας Αληθείας

x	y	z
0	0	1
0	1	0
1	0	0
1	1	1

Λογική Συνάρτηση $z = \bar{x} \cdot \bar{y} + x \cdot y$

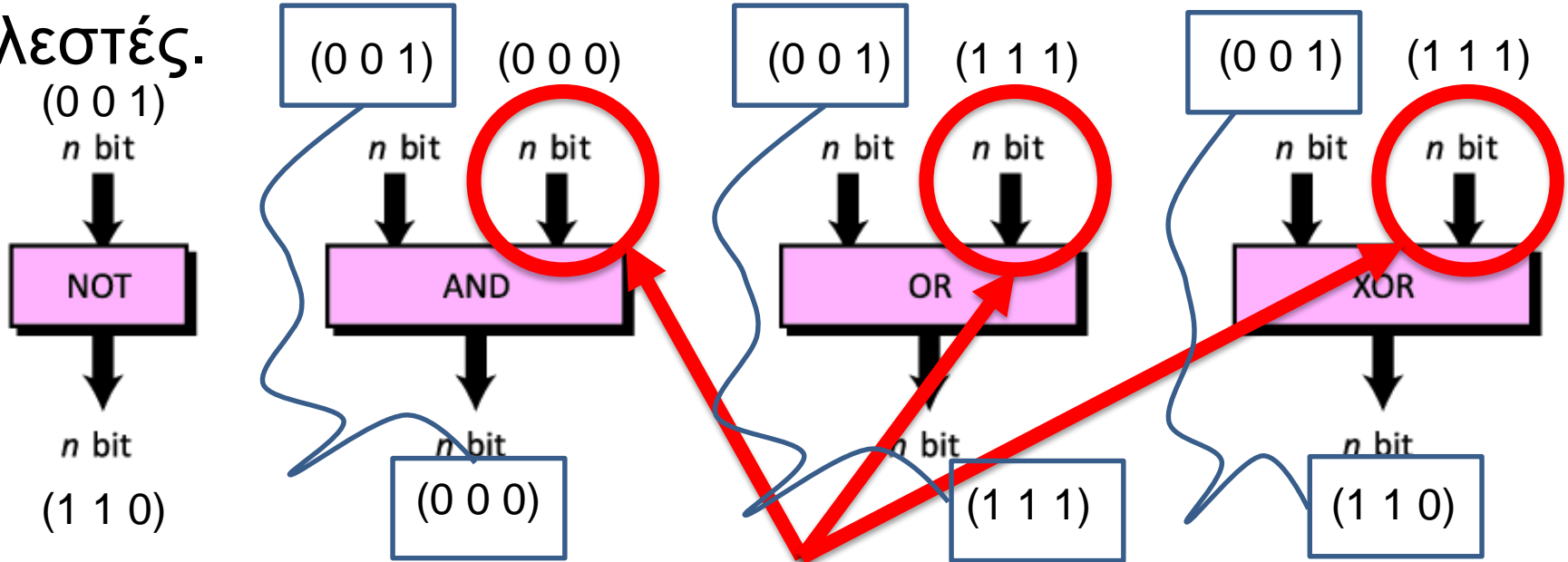
Λογικές πράξεις σε επίπεδο σχήματος

Και οι τέσσερις τελεστές (NOT, AND, OR, και XOR) μπορούν να εφαρμοστούν σε ένα σχήμα με n bit. Το αποτέλεσμα είναι το ίδιο όπως αν εφαρμόζαμε κάθε τελεστή σε κάθε μεμονωμένο bit για τον τελεστή NOT, και σε κάθε αντίστοιχο ζεύγος bit για τους άλλους τρεις τελεστές.



Λογικές πράξεις σε επίπεδο σχήματος

Και οι τέσσερις τελεστές (NOT, AND, OR, και XOR) μπορούν να εφαρμοστούν σε ένα σχήμα με n bit. Το αποτέλεσμα είναι το ίδιο όπως αν εφαρμόζαμε κάθε τελεστή σε κάθε μεμονωμένο bit για τον τελεστή NOT, και σε κάθε αντίστοιχο ζεύγος bit για τους άλλους τρεις τελεστές.



ΜΑΣΚΕΣ

Εφαρμογές

Για την τροποποίηση ενός σχήματος bit μπορούν να εφαρμοστούν οι τέσσερις λογικές πράξεις.

- ❑ Συμπλήρωμα (NOT)
- ❑ Απενεργοποίηση (AND) (όταν η μάσκα είναι 0)
- ❑ Ενεργοποίηση (OR) (όταν η μάσκα είναι 1)
- ❑ Αντιστροφή (XOR) (όταν η μάσκα είναι 1)

Παράδειγμα AND – ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ (ΟΤΑΝ ΜΑΣΚΑ ΕΊΝΑΙ 0)

Χρησιμοποιήστε μια μάσκα για να απενεργοποιήσετε τα πέντε αριστερότερα bit ενός σχήματος. Ελέγξτε τη μάσκα με το σχήμα 10100110.

Λύση

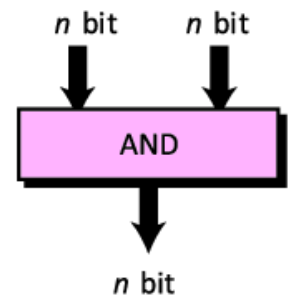
Η μάσκα είναι 00000111. Το αποτέλεσμα της εφαρμογής της μάσκας είναι το εξής:

	1	0	1	0	0	1	1	0
AND	0	0	0	0	0	1	1	1
	0	0	0	0	0	1	1	0

Είσοδος

Μάσκα

Έξοδος



(όταν η μάσκα είναι 0)

Παράδειγμα OR – ΕΝΕΡΓΟΠΟΙΗΣΗ (ΟΤΑΝ ΜΑΣΚΑ ΕΊΝΑΙ 1)

Χρησιμοποιήστε μια μάσκα για να ενεργοποιήσετε τα πέντε αριστερότερα bit ενός σχήματος. Ελέγξτε τη μάσκα με το σχήμα 10100110.

Λύση

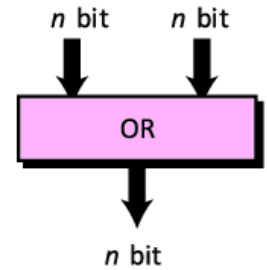
Η μάσκα είναι 11111000. Το αποτέλεσμα της εφαρμογής της μάσκας είναι το εξής:

	1	0	1	0	0	1	1	0
OR	1	1	1	1	1	0	0	0
	1	1	1	1	1	1	1	0

Είσοδος

Μάσκα

Έξοδος



(όταν η μάσκα είναι 1)

Παράδειγμα XOR – ΑΝΤΙΣΤΡΟΦΗΣ (ΟΤΑΝ ΜΑΣΚΑ ΕΊΝΑΙ 1)

Χρησιμοποιήστε μια μάσκα για να αντιστρέψετε τα πέντε αριστερότερα bit ενός σχήματος. Ελέγξτε τη μάσκα με το σχήμα 10100110.

Λύση

Η μάσκα είναι 11111000. Το αποτέλεσμα της εφαρμογής της μάσκας είναι το εξής:

	1	0	1	0	0	1	1	0	Είσοδος 1
XOR	1	1	1	1	1	0	0	0	Μάσκα
	0	1	0	1	1	1	1	0	Έξοδος

(όταν η μάσκα είναι 1)

END
