Λογική Σχεδίαση

CAL@DI

DSCAL

DIGITAL SYSTEMS & COMPUTER ARCHITECTURE LABORATORY

Λογική σχεδίαση

- Θεωρία: **Βασίλειος Καρακώστας, Επίκουρος Καθηγητής**
  Διονύσιος Βασιλόπουλος, ΕΔΙΠ
  - *E-mail: vkarakos@di.uoa.gr | denis@di.uoa.gr*
- Εργαστήριο: **Διονύσιος Βασιλόπουλος, ΕΔΙΠ**
- Πληροφορίες για το μάθημα θα βρείτε στο eclass: https://eclass.uoa.gr/courses/D13/
  - *Οι διαλέξεις του μαθήματος*
  - *Λυμένες ασκήσεις*
- Το εργαστήριο προσφέρει την κατανόηση των βασικών αρχών και πρακτικών της ψηφιακής λογικής, σύμφωνα με το αρχαίο κινέζικο ρητό:
  - ***"Ακούω και ξεχνώ,***
    βλέπω και θυμάμαι,
    εφαρμόζω και κατανοώ (μαθαίνω)"

Λογική σχεδίαση

- **Μαθησιακοί στόχοι**:
  - *Να μυηθούν βήμα-βήμα οι αμύητοι φοιτητές, χωρίς την αναγκαιότητα κάποιας πρότερης γνώσης, αρχικά στα αριθμητικά συστήματα και στις λογικές πύλες, κατόπιν στη συνδυαστική λογική και τέλος στην ακολουθιακή λογική*
  - *Η μύηση συμπεριλαμβάνει και θεμελιώδεις γνώσεις προγραμματισμού στη γλώσσα περιγραφής υλικού (VHDL)*
    - **Σήμερα, το hardware είναι κυρίως προγραμματισμός**
  - *Το ιδιαίτερο τελετουργικό μύησης στην ψηφιακή σχεδίαση, που ακολουθείται στο παρόν μάθημα, αφορά όλους τους φοιτητές που ενδιαφέρονται να εντρυφήσουν:*
    - στην ανάπτυξη του υλικού και του λογισμικού υπολογιστικών συστημάτων, τηλεπικοινωνιακών και δικτυακών συστημάτων, και συστημάτων ψηφιακής επεξεργασίας σήματος και πληροφοριών
    - στην επιστήμη και τεχνολογία των υπολογιστών

Λογική σχεδίαση

- Προσδοκώμενα μαθησιακά αποτελέσματα
- Με την επιτυχή ολοκλήρωση του μαθήματος ο φοιτητής/η φοιτήτρια θα είναι σε θέση να:
  - *περιγράφει τις αρχές και πρακτικές της συνδυαστικής και ακολουθιακής λογικής,*
  - *σχεδιάζει όλες τις βασικές μονάδες ενός ψηφιακού συστήματος στο επίπεδο της λογικής σχεδίασης,*
  - *αναλύει τον χρονισμό των ψηφιακών συστημάτων,*
  - *περιγράφει τις σύγχρονες διατάξεις μνήμης και λογικής,*
  - *κατέχει βασική γνώση προγραμματισμού σε γλώσσα περιγραφής υλικού (VHDL)*
    - κυρίως μέσω του εργαστηρίου του μαθήματος

Λογική σχεδίαση

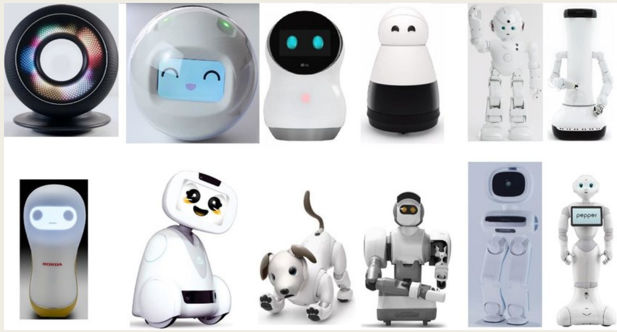
- Βιβλιογραφία
  - *ΨΗΦΙΑΚΗ ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ, ΕΚΔΟΣΗ ARM®*
  - *Κωδικός Βιβλίου στον Εύδοξο: 86055864*
  - *Έκδοση: 1η/2019*
  - *Συγγραφείς: SARAH L. HARRIS, DAVID MONEY HARRIS*
  - *ISBN: 978-960-461-961-0*
  - *Τύπος: Σύγγραμμα*
  - *Διαθέτης (Εκδότης): ΕΚΔΟΣΕΙΣ ΚΛΕΙΔΑΡΙΘΜΟΣ ΕΠΕ*

Τα ψηφιακά συστήματα βρίσκονται παντού!

- Ψηφιακές συσκευές καταναλωτών (consumer electronics):

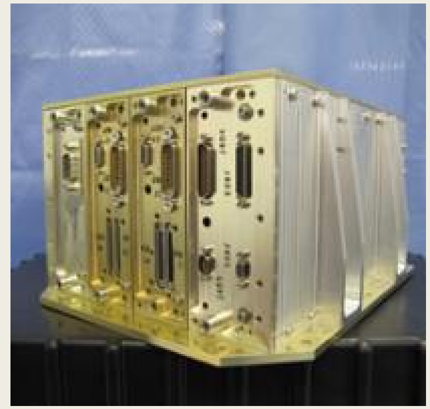
1960

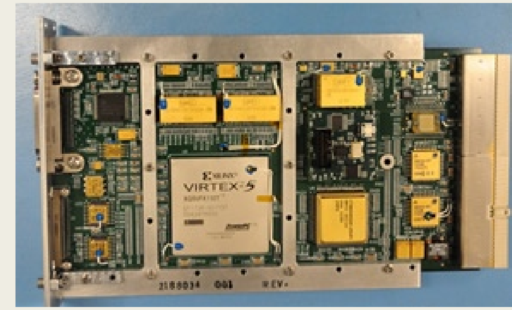
1980



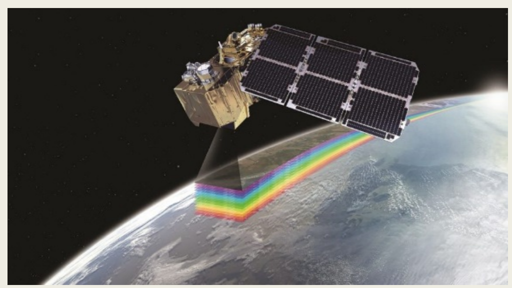
Τα ψηφιακά συστήματα βρίσκονται παντού!

- Μερικά παραδείγματα από την αεροδιαστημική:

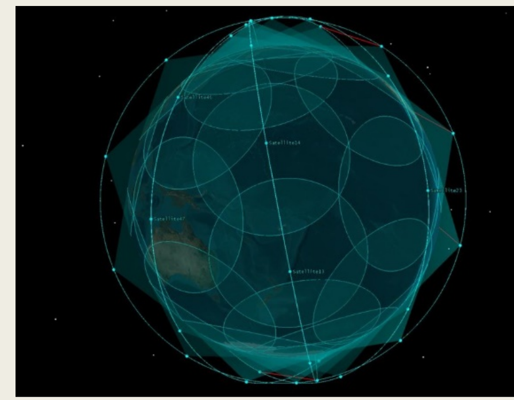




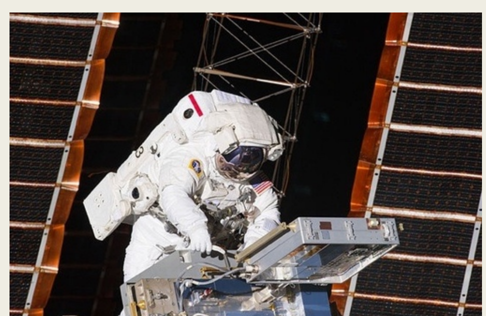
Adaptive image & video processing/compression



Earth observation and hyperspectral imaging

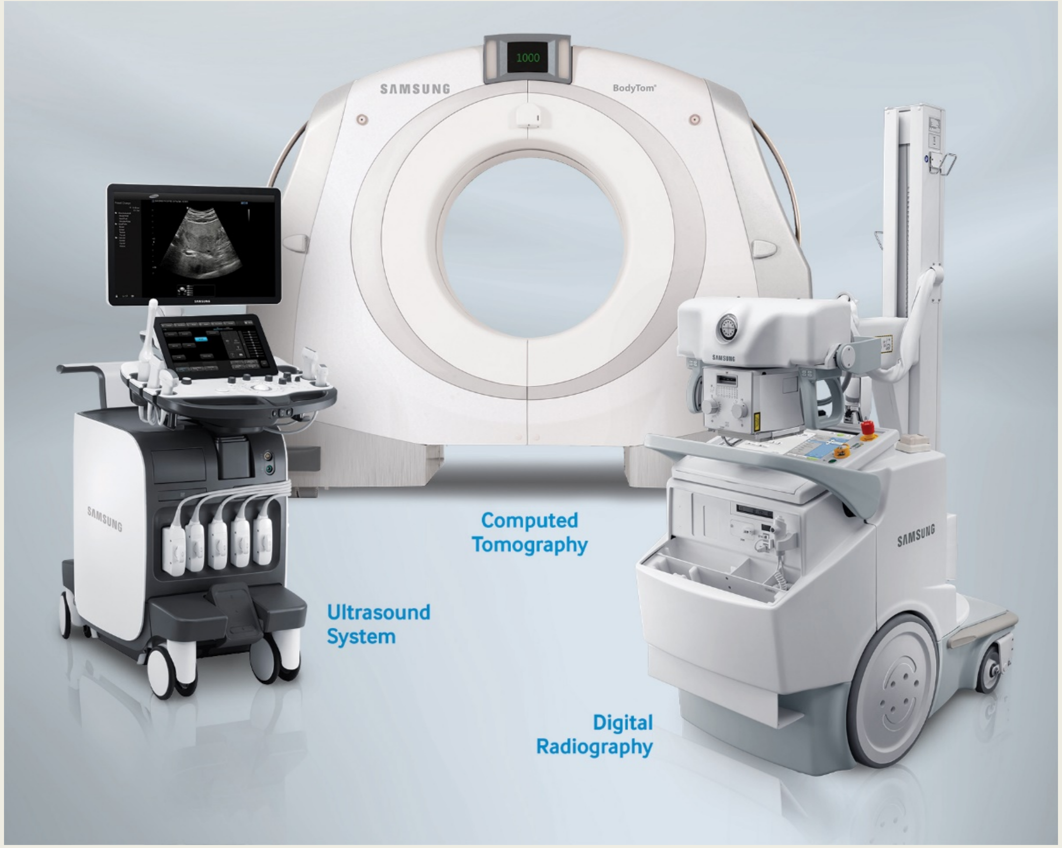


Internet of space and navigation



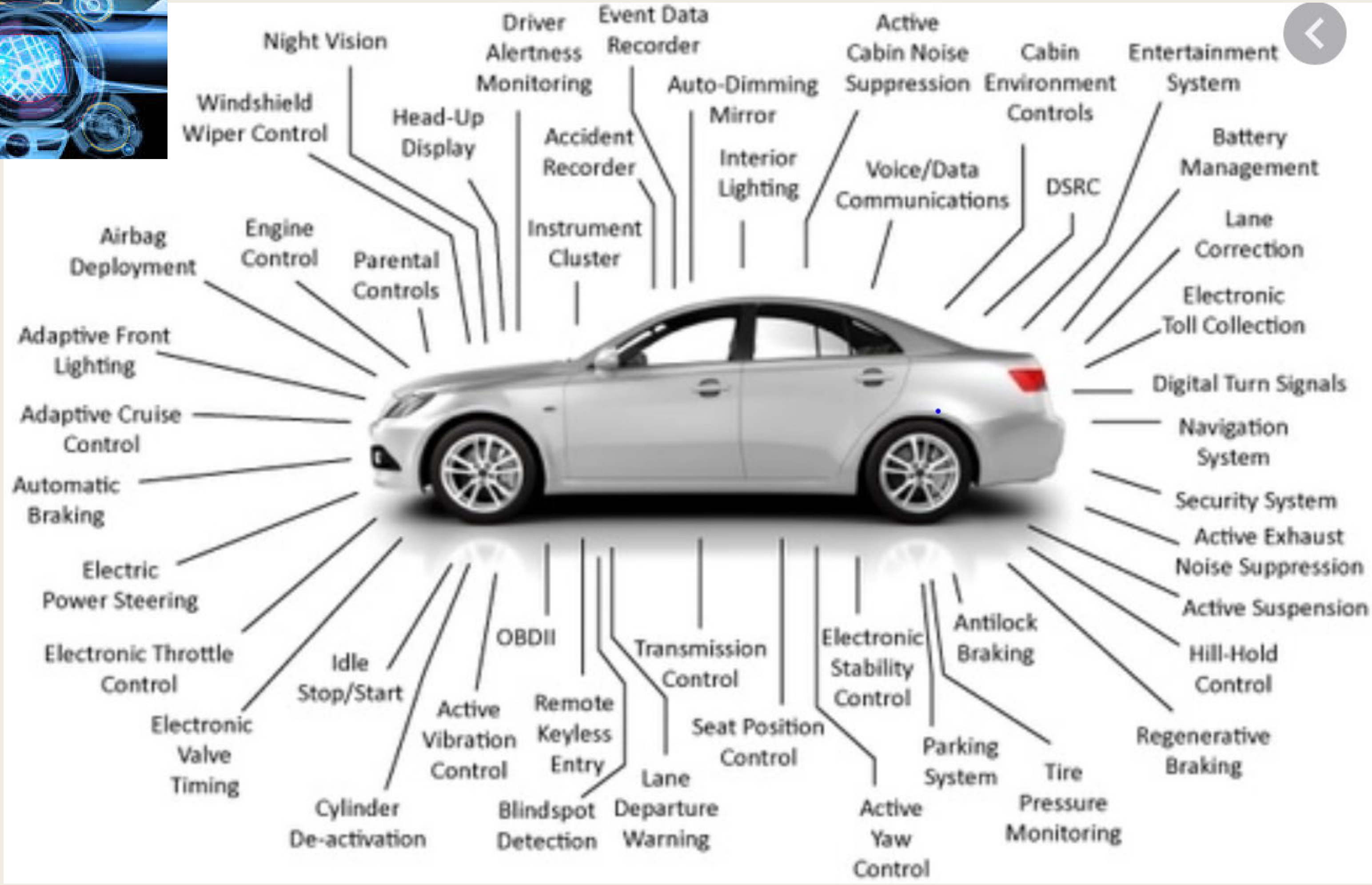
Τα ψηφιακά συστήματα βρίσκονται παντού!

- Μερικά παραδείγματα από την ιατρική:



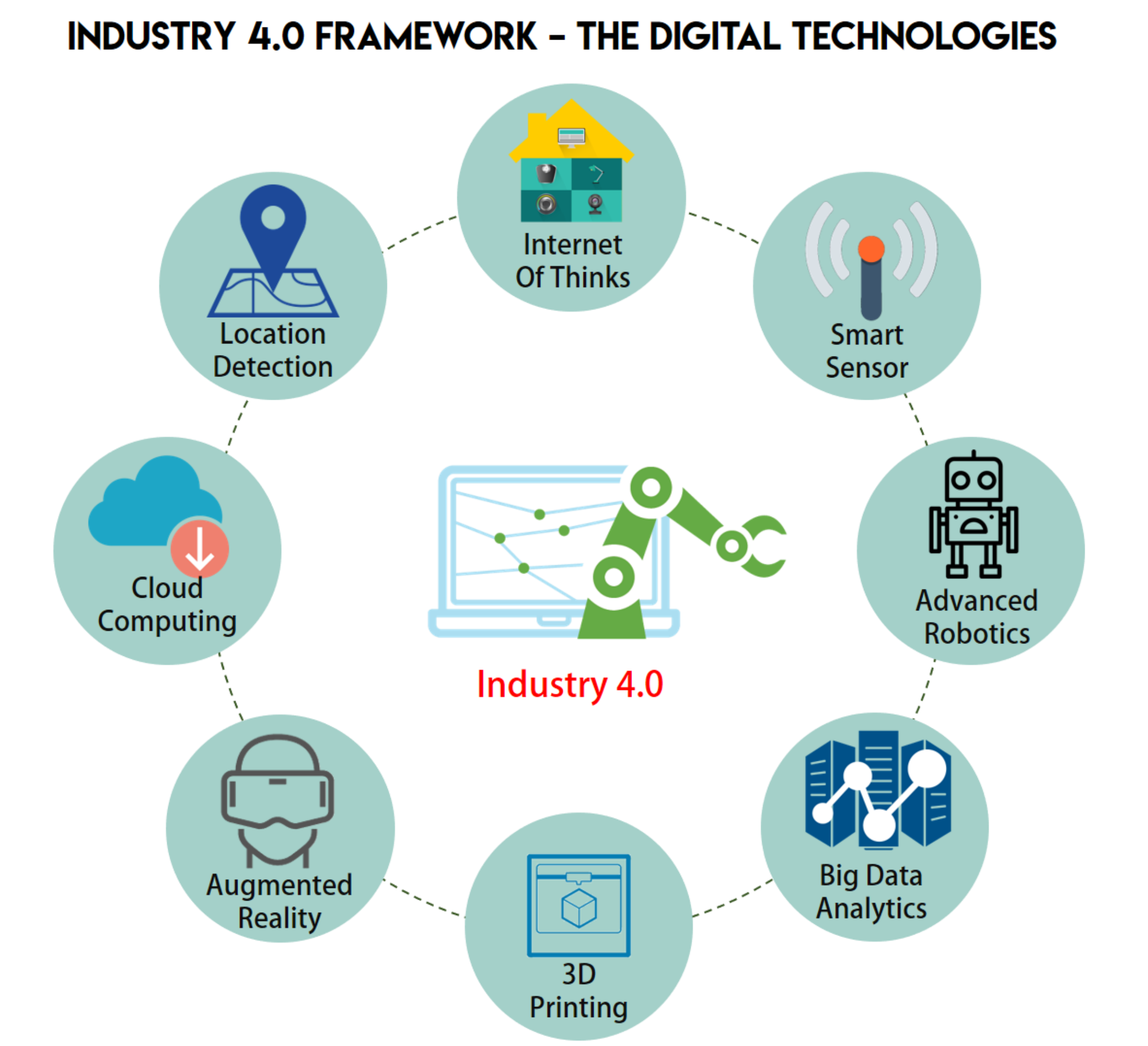
Τα ψηφιακά συστήματα βρίσκονται παντού!

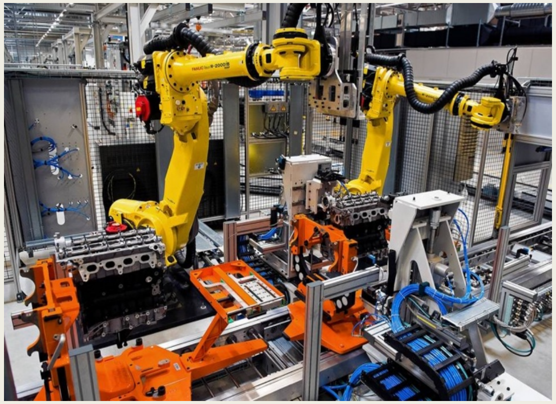
- Μερικά παραδείγματα από την αυτοκινητοβιομηχανία (AI, IoT):



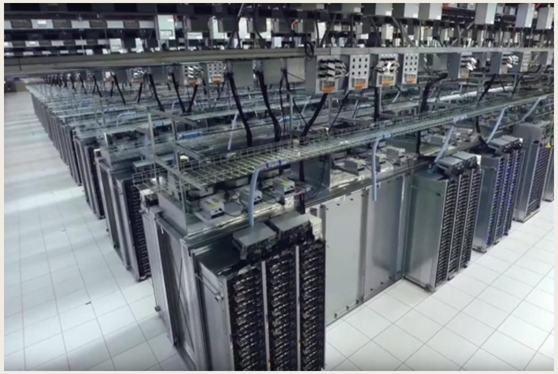
Τα ψηφιακά συστήματα βρίσκονται παντού!

- Μερικά παραδείγματα από την ψηφιακή βιομηχανία:



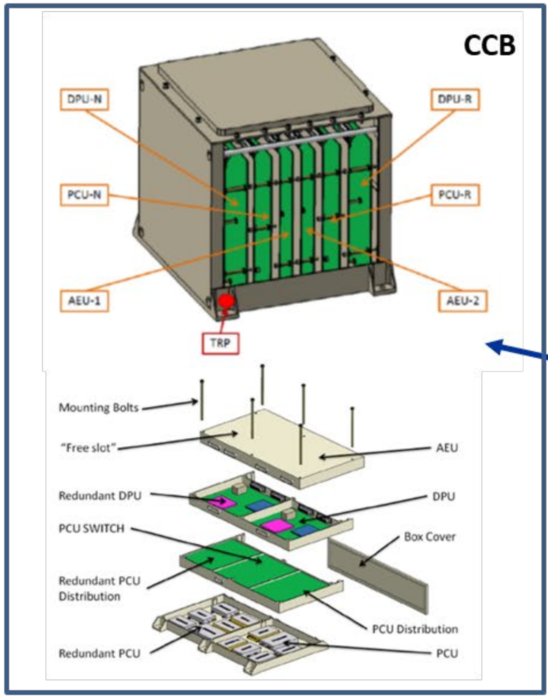


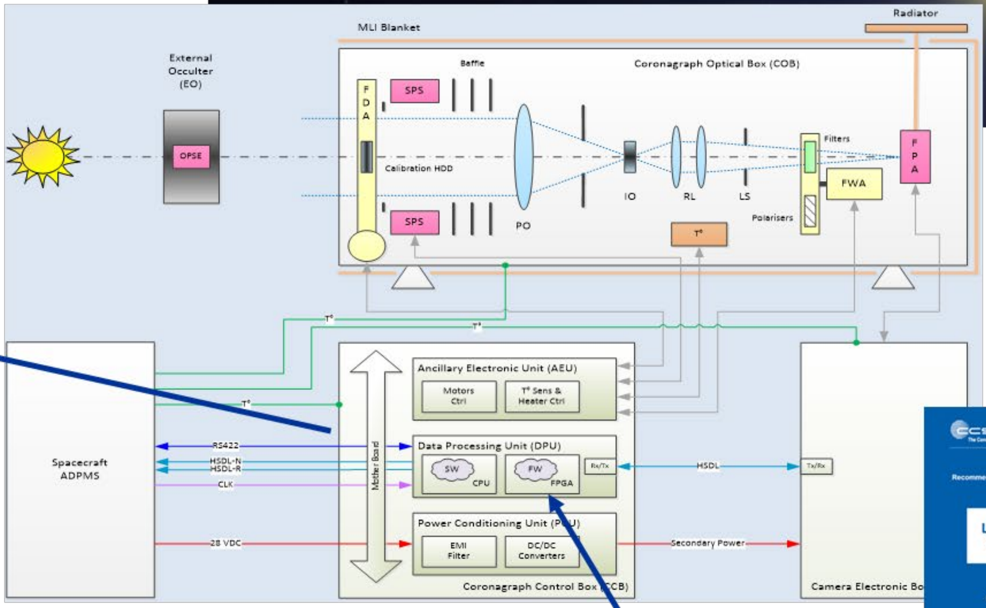
Robotics automation and machine learning @ industry



Big data analytics and machine learning @ data centers

ESA PROBA-3 Coronagraph System





CCSDS 121.0-B-2 Image Data Compression (IDC) IP core

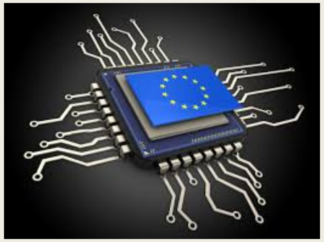
CAL@DI

arm

RISC-V®

RISC-V: The Free and Open RISC Instruction Set Architecture

epi European Processor Initiative



Κεφάλαιο 1

Εισαγωγή στη ψηφιακή σχεδίαση και τεχνολογία (από το μηδέν στο ένα)

Γιώργος Παπαδημητρίου, Αντώνης Πασχάλης,
Διονύσης Βασιλόπουλος

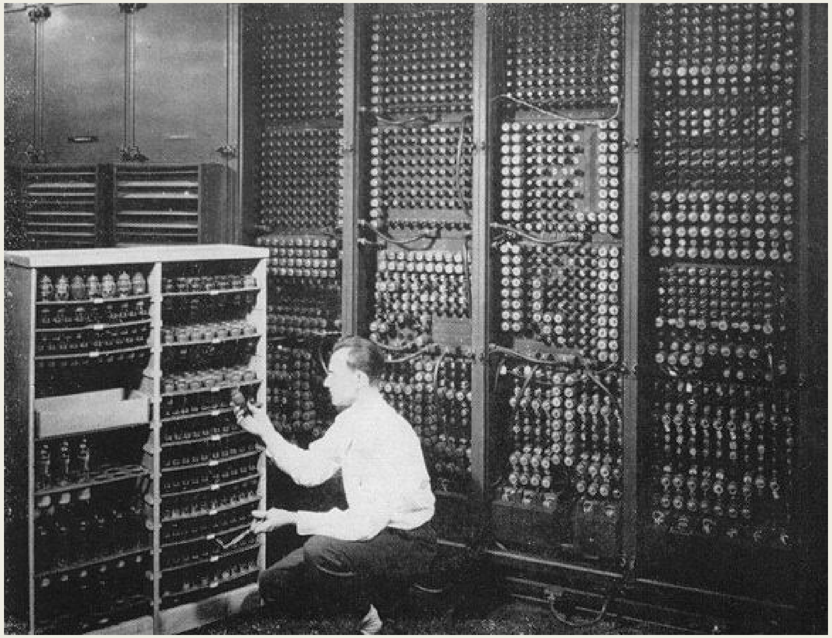
DIGITAL SYSTEMS & COMPUTER ARCHITECTURE LABORATORY

Περιεχόμενα κεφαλαίου 1

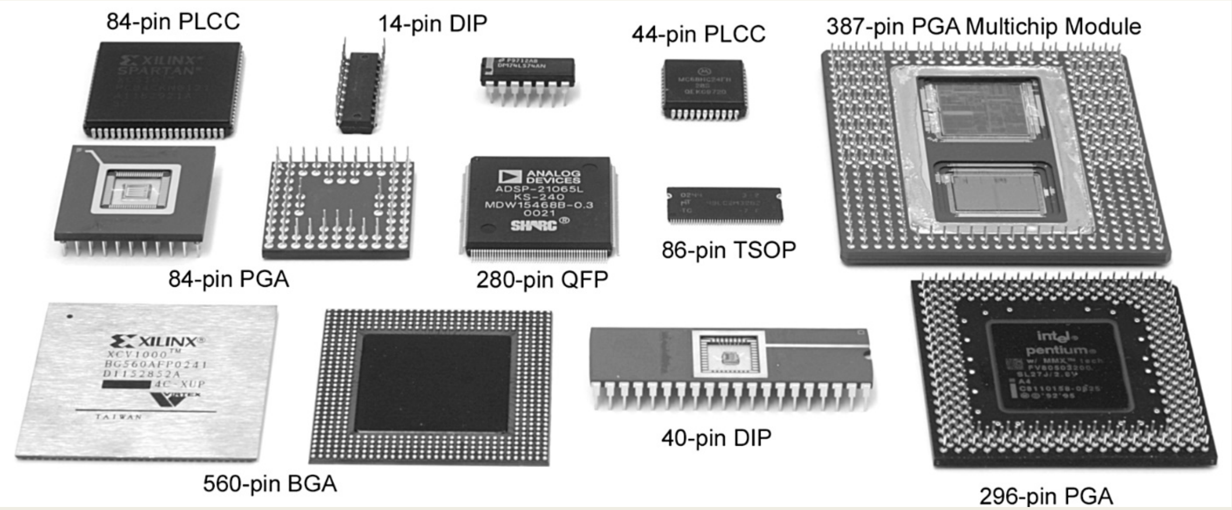
- Εξέλιξη ψηφιακής τεχνολογίας (Εισαγωγή στην Πληροφορική)
- Περί μικροεπεξεργαστών (Εισαγωγή στην Πληροφορική)
- Διαχείριση πολυπλοκότητας υπολογιστικών συστημάτων (Εισαγωγή στην Πληροφορική)
- Ψηφιακή «αφαίρεση» (Εισαγωγή στην Πληροφορική)
- Αναπαράσταση αριθμών σε δυαδική μορφή
- Πράξεις (+, -) μεταξύ δυαδικών αριθμών
- Αθροιστές - Αφαιρέτες
- Λογικές πύλες
- Λογικά επίπεδα – Περιθώρια θορύβου
- Τρανζίστορ CMOS – Μοντέλο διακοπτών – CMOS πύλες
- Κατανάλωση ισχύος

Οι παρουσιάσεις βασίζονται στο βιβλίο με τίτλο **«Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών – Έκδοση ARM»** των Sarah L. Harris & David Money Harris. Τίτλοι με κόκκινο υποδηλώνουν εκπαιδευτικό υλικό εκτός αυτού του βιβλίου.

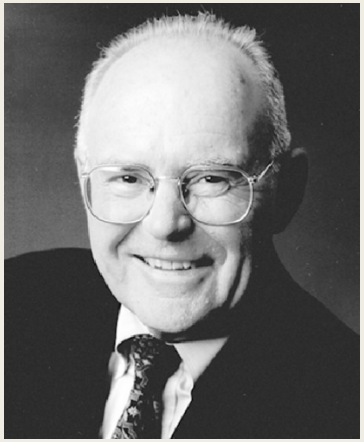
- ENIAC Αντικατάσταση λυχνίας (19.000 πιθανές βλάβες)



- Ολοκληρωμένα κυκλώματα



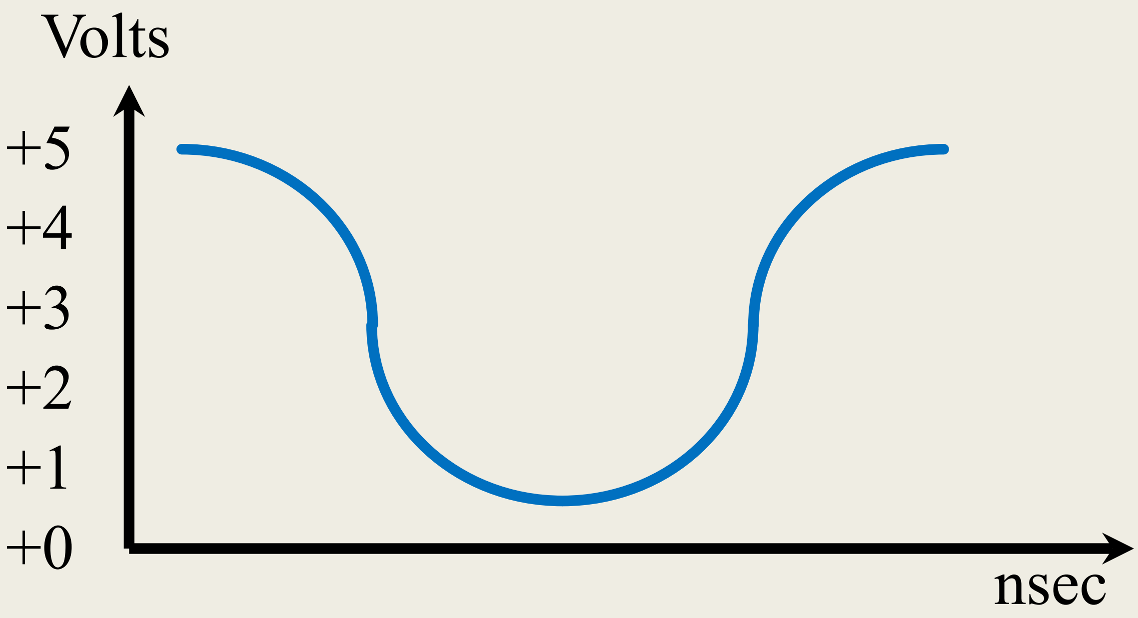
Νόμος του Moore



- Gordon Moore (Σαν Φρανσίσκο, 1929)
- Κάτοχος προπτυχιακού διπλώματος στη Χημεία από το Πανεπιστήμιο Berkeley και διδακτορικού διπλώματος στη Χημεία και τη Φυσική από το Πανεπιστήμιο Caltech
- Το 1968 ίδρυσε την εταιρεία Intel μαζί με τον Robert Noyce
- Το 1965 παρατήρησε ότι **το πλήθος των τρανζίστορ σε ένα τσιπ υπολογιστών διπλασιάζεται κάθε χρόνο**. Αυτή η τάση έχει γίνει ευρύτερα γνωστή ως **Νόμος του Moore**
- Από το 1975 και έπειτα, **το πλήθος των τρανζίστορ διπλασιάζεται κάθε δύο χρόνια,** ενώ **η απόδοση των μικροεπεξεργαστών διπλασιάζεται κάθε 18 με 24 μήνες**
- Το νόμο του Moore ακολουθούν και **οι μνήμες (τεχνολογίας SRAM, DRAM, HDD, SSD)** του ιεραρχικού συστήματος μνήμης, αλλά με μικρότερους ρυθμούς
- Οι **πωλήσεις των ημιαγωγών** εμφανίζουν και αυτές εκθετική αύξηση
- Ο Νόμος του Moore έχει αποτελέσει την κινητήρια δύναμη πίσω από την απίστευτη πρόοδο που έχει σημειωθεί στη **βιομηχανία των ημιαγωγών** τα 50 τελευταία χρόνια, που οδηγεί στη ψηφιακή εποχή

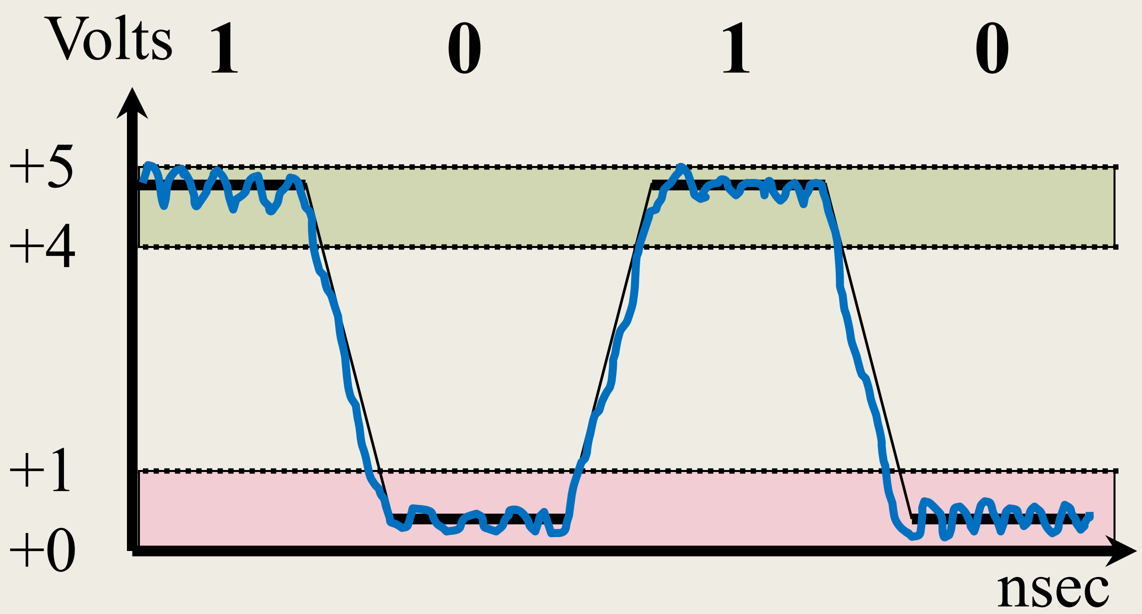
Αναλογικά συστήματα

- Επεξεργάζονται συνεχή ηλεκτρικά σήματα, που μεταβάλλονται σαν συναρτήσεις του χρόνου



Ψηφιακά συστήματα

- Επεξεργάζονται δυαδικά ηλεκτρικά σήματα
- Τα δυαδικά σήματα λαμβάνουν δύο μόνο τιμές:
  - *Το μηδέν (0 ή LOW)*
  - *Το ένα (1 ή HIGH)*

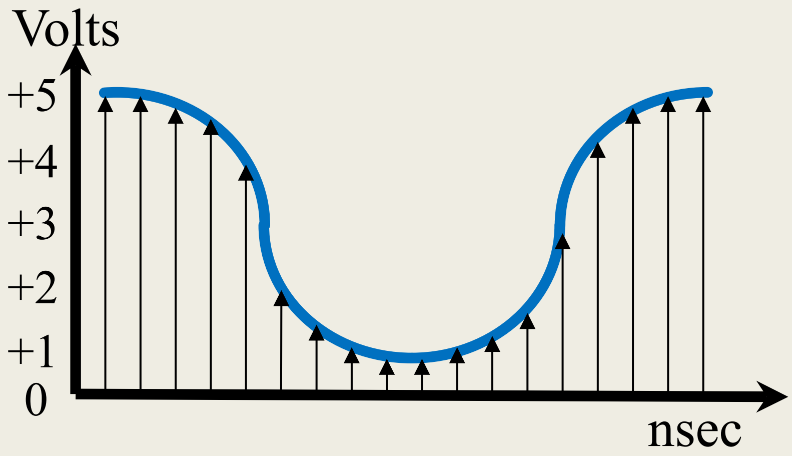


Ψηφιακά συστήματα

- Συστήματα Υπολογιστών
- Συστήματα Επικοινωνιών
- Συστήματα Επεξεργασίας Σήματος
- Συστήματα Αυτομάτου Ελέγχου
  - *ψηφιακή επεξεργασία δεδομένων*
  - *ψηφιακή λήψη δεδομένων*
  - *ψηφιακή απόκριση*
  - *αναλογική λήψη δεδομένων*

    μετατροπή της πληροφορίας από αναλογική σε ψηφιακή
    με Analog to Digital Converter – ADC
    (δειγματοληψία + κβαντοποίηση σε ακέραιους αριθμούς 10-16 bit)
  - *αναλογική απόκριση*

    μετατροπή της πληροφορίας από ψηφιακή σε αναλογική
    με Digital to Analog Converter – DAC



George Boole, 1815–1864

- Όντας παιδί εργατών και μην έχοντας την οικονομική δυνατότητα να μορφωθεί μέσω του εκπαιδευτικού συστήματος, ο Boole έμαθε μόνος του μαθηματικά και στη συνέχεια έγινε μέλος του διδακτικού προσωπικού στο Κολέγιο Κουίνς (Ιρλανδία).
- Έγραψε τη μονογραφία An Investigation of the Laws of Thought (1854), όπου παρουσίασε για πρώτη φορά τις δυαδικές μεταβλητές και τις τρεις θεμελιώδεις λογικές πράξεις (logic operations): AND, OR και NOT.
  - *Οι δυαδικές μεταβλητές του Boole μπορούσαν να λάβουν τις τιμές TRUE ή FALSE*
  - *Θεωρούμε ότι είναι συνώνυμοι οι όροι:*
    - 1, TRUE, HIGH
    - 0, FALSE, LOW

Μεταφορά πληροφορίας

- Η ποσότητα πληροφορίας D σε μία μεταβλητή διακριτών τιμών με N διαφορετικές καταστάσεις χαρακτηρίζεται απο την σχέση
  - $D=\log_2 N$
  - *Ο λογάριθμος έχει βάση το 2*
  - *Η μονάδα μέτρησης του D ειναι το bit*
- Κάθε δυαδική μεταβλητή μεταφέρει $\log_2 2=1$ bit πληροφοριών
- Θεωρητικά, ένα συνεχές σήμα μεταφέρει άπειρη ποσότητα πληροφοριών, αφού μπορεί να πάρει άπειρο πλήθος τιμών
- Στην πράξη, ο θόρυβος και το σφάλμα μέτρησης περιορίζουν τις πληροφορίες που μεταφέρονται στα 10-16 bit, για τα περισσότερα συνεχή σήματα

Επιλεγμένες ασκήσεις

- **Άσκηση 1.4**
  Μια αναλογική τάση παίρνει τιμές από 0 έως 5 V. Αν είναι εφικτό να μετρηθεί με ακρίβεια ±50 mV, πόσα bit πληροφοριών μπορεί να μεταφέρει το μέγιστο;

- **Άσκηση 1.6**
  Οι Βαβυλώνιοι ανέπτυξαν το εξηνταδικό (sexagesimal, με βάση το 60) αριθμητικό σύστημα πριν από 4000 χρόνια περίπου. Πόσα bit πληροφοριών μεταφέρονται με ένα εξηνταδικό ψηφίο;

Επιλεγμένες ασκήσεις

- Άσκηση 1.4
  Μια αναλογική τάση παίρνει τιμές από 0 έως 5 V. Αν είναι εφικτό να μετρηθεί με ακρίβεια ±50 mV, πόσα bit πληροφοριών μπορεί να μεταφέρει το μέγιστο;
  - *Ακρίβεια ±50 mV σημαίνει ότι το αναλογικό σήμα διαμερίζεται ανά 100 mV , δηλαδή σε 50 διακριτές τιμές από από 0 έως 5 V*
  - $\log_2 50 = 5.64$ bits
- Άσκηση 1.6
  Οι Βαβυλώνιοι ανέπτυξαν το εξηνταδικό (sexagesimal, με βάση το 60) αριθμητικό σύστημα πριν από 4000 χρόνια περίπου. Πόσα bit πληροφοριών μεταφέρονται με ένα εξηνταδικό ψηφίο;
  - $\log_2 60 = 5.91$ bits

Ερωτήσεις συνεντεύξεων

- Ερώτηση 1.2
  Ένας βασιλιάς παραλαμβάνει 64 χρυσά νομίσματα σε φόρους, αλλά έχει βάσιμες υποψίες ότι ένα από αυτά είναι κάλπικο. Σας καλεί στο παλάτι για να εντοπίσετε το κάλπικο νόμισμα.
  Έχετε στη διάθεσή σας μια ζυγαριά παλαιού τύπου με δύο δίσκους στους οποίους μπορείτε να τοποθετείτε νομίσματα.
  Πόσες φορές πρέπει να χρησιμοποιήσετε τη ζυγαριά για να βρείτε το ελαφρύτερο, κάλπικο νόμισμα;
  - *Επιλέξτε μία από τις απαντήσεις:*
    (α) 6
    (β) 5
    (γ) 4
    (δ) 3

Ερωτήσεις συνεντεύξεων

- Ερώτηση 1.2
  Ένας βασιλιάς παραλαμβάνει 64 χρυσά νομίσματα σε φόρους, αλλά έχει βάσιμες υποψίες ότι ένα από αυτά είναι κάλπικο. Σας καλεί στο παλάτι για να εντοπίσετε το κάλπικο νόμισμα.
  Έχετε στη διάθεσή σας μια ζυγαριά παλαιού τύπου με δύο δίσκους στους οποίους μπορείτε να τοποθετείτε νομίσματα.
  Πόσες φορές πρέπει να χρησιμοποιήσετε τη ζυγαριά για να βρείτε το ελαφρύτερο, κάλπικο νόμισμα;
  - *Για διαίρεση στα 2, με δύο ζυγαριές Α και Β*
    - 1η Ζύγιση, Από 32 σε Α και Β : Έστω ελαφρύτερη η Α
    - 2η Ζύγιση. Από 16 σε Α και Β : Έστω ελαφρύτερη η Α
    - 3η Ζύγιση. Από 8 σε Α και Β : Έστω ελαφρύτερη η Α
    - 4η Ζύγιση. Από 4 σε Α και Β : Έστω ελαφρύτερη η Α
    - 5η Ζύγιση. Από 2 σε Α και Β : Έστω ελαφρύτερη η Α
    - 6η Ζύγιση. Από 1 σε Α και Β : Έστω ελαφρύτερη η Α (κάλπικο νόμισμα)

Ερωτήσεις συνεντεύξεων

- Ερώτηση 1.2
  Ένας βασιλιάς παραλαμβάνει 64 χρυσά νομίσματα σε φόρους, αλλά έχει βάσιμες υποψίες ότι ένα από αυτά είναι κάλπικο. Σας καλεί στο παλάτι για να εντοπίσετε το κάλπικο νόμισμα.
  Έχετε στη διάθεσή σας μια ζυγαριά παλαιού τύπου με δύο δίσκους στους οποίους μπορείτε να τοποθετείτε νομίσματα.
  Πόσες φορές πρέπει να χρησιμοποιήσετε τη ζυγαριά για να βρείτε το ελαφρύτερο, κάλπικο νόμισμα;
  - *Για διαίρεση στα 3, με δύο ζυγαριές A και B*
    - 1η Ζύγιση, Από 22 σε A και B και 20 στην άκρη: Έστω ελαφρύτερη η A
    - 2η Ζύγιση. Από 8 σε A και B και 6 στην άκρη: Έστω ελαφρύτερη η A
    - 3η Ζύγιση. Από 3 σε A και B και 2 στην άκρη : Έστω ελαφρύτερη η A
    - 4η Ζύγιση. Από 1 σε A και B και 1 στην άκρη : Αν A=B τότε κάλπικο αυτό που είναι στην άκρη. Αλλιώς κάλπικο αυτό που είναι το ελαφρύτερο από A και B

    (υπάρχουν και άλλες διαιρέσεις π.χ. στην 1η ζύγιση 21/21/22, κ.λ.π. που δίνουν το ίδιο αποτέλεσμα)

Ερωτήσεις συνεντεύξεων

- Ερώτηση 1.2
  Ένας βασιλιάς παραλαμβάνει 64 χρυσά νομίσματα σε φόρους, αλλά έχει βάσιμες υποψίες ότι ένα από αυτά είναι κάλπικο. Σας καλεί στο παλάτι για να εντοπίσετε το κάλπικο νόμισμα.
  Έχετε στη διάθεσή σας μια ζυγαριά παλαιού τύπου με δύο δίσκους στους οποίους μπορείτε να τοποθετείτε νομίσματα.
  Πόσες φορές πρέπει να χρησιμοποιήσετε τη ζυγαριά για να βρείτε το ελαφρύτερο, κάλπικο νόμισμα;
  – *Για διαίρεση στα 4, με δύο ζυγαριές Α και Β και στην άκρη τα υπόλοιπα (Α περίπτωση worst case scenario)*
    - 1η Ζύγιση, Από 16 σε Α και Β και 32 στην άκρη: Έστω Α=Β, ασχολούμαστε με τα 32 που είναι στην άκρη.
    - 2η Ζύγιση. Από 8 σε Α και Β και 16 στην άκρη: Έστω Α=Β, ασχολούμαστε με τα 16 που είναι στην άκρη.
    - 3η Ζύγιση. Από 4 σε Α και Β και 8 στην άκρη : Έστω Α=Β, ασχολούμαστε με τα 8 που είναι στην άκρη.
    - 4η Ζύγιση. Από 2 σε Α και Β και 4 στην άκρη : Έστω Α=Β, ασχολούμαστε με τα 4 που είναι στην άκρη.
    - 5η Ζύγιση. Από 1 σε Α και Β και 2 στην άκρη : Έστω Α=Β, ασχολούμαστε με τα 2 που είναι στην άκρη.
    - 6η Ζύγιση. Από 1 σε Α και Β. Έστω ελαφρύτερη η Α (κάλπικο νόμισμα)

Ερωτήσεις συνεντεύξεων

- Ερώτηση 1.2
  Ένας βασιλιάς παραλαμβάνει 64 χρυσά νομίσματα σε φόρους, αλλά έχει βάσιμες υποψίες ότι ένα από αυτά είναι κάλπικο. Σας καλεί στο παλάτι για να εντοπίσετε το κάλπικο νόμισμα.
  Έχετε στη διάθεσή σας μια ζυγαριά παλαιού τύπου με δύο δίσκους στους οποίους μπορείτε να τοποθετείτε νομίσματα.
  Πόσες φορές πρέπει να χρησιμοποιήσετε τη ζυγαριά για να βρείτε το ελαφρύτερο, κάλπικο νόμισμα;
  - *Για διαίρεση στα 4, με δύο ζυγαριές Α και Β και στην άκρη τα υπόλοιπα (Β περίπτωση best case scenario)*
    - 1η Ζύγιση, Από 16 σε Α και Β και 32 στην άκρη: Έστω ελαφρύτερη η Α.
    - 2η Ζύγιση. Από 4 σε Α και Β και 8 στην άκρη: Έστω ελαφρύτερη η Α.
    - 3η Ζύγιση. Από 1 σε Α και Β και 2 στην άκρη : Έστω ελαφρύτερη η Α (κάλπικο νόμισμα)

    Φαίνεται ως <u>πιθανή καλύτερη λύση</u>, αλλά δεν είναι, γιατί δεν εξασφαλίζει ότι ΠΑΝΤΑ με 3 ζυγίσεις βρίσκουμε το σωστό αποτέλεσμα. Μας ενδιαφέρει τι γίνεται στη χειρότερη των περιπτώσεων.

Αριθμητικά Συστήματα

- Δεκαδικό σύστημα αναπαράστασης αριθμών (0-9)

Στήλη των 1000
Στήλη των 100
Στήλη των 10
Στήλη των 1

6598_{10}

Κάθε στήλη ενός δεκαδικού αριθμού έχει δεκαπλάσιο βάρος από την προηγούμενη στήλη. Ξεκινώντας από τα δεξιά προς τα αριστερά τα βάρη είναι: **10^0, 10^1, 10^2, 10^3, ...**

- Δυαδικό σύστημα αναπαράστασης αριθμών (0-1)

Στήλη των 8
Στήλη των 4
Στήλη των 2
Στήλη των 1

1101_2

Κάθε στήλη ενός δυαδικού αριθμού έχει διπλάσιο βάρος από την προηγούμενη στήλη. Ξεκινώντας από τα δεξιά προς τα αριστερά τα βάρη είναι: **2^0, 2^1, 2^2, 2^3, ...**

Αριθμητικά Συστήματα

- Δεκαδικό σύστημα αναπαράστασης αριθμών (0-9)

Στήλη των 1000
Στήλη των 100
Στήλη των 10
Στήλη των 1

$$6598_{10} = 6 \times 10^3 + 5 \times 10^2 + 9 \times 10^1 + 8 \times 10^0$$

Έξι Χιλιάδες · Πέντε Εκατοντάδες · Εννέα Δεκάδες · Οκτώ Μονάδες

- Δυαδικό σύστημα αναπαράστασης αριθμών (0-1)

Στήλη των 8
Στήλη των 4
Στήλη των 2
Στήλη των 1

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = \mathbf{13_{10}}$$

Μια Οκτάδα · Μια Τετράδα · Μηδέν Δυάδες · Μια Μονάδα

Δυνάμεις του 2

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1024$
- $2^{11} = 2048$
- $2^{12} = 4096$
- $2^{13} = 8192$
- $2^{14} = 16384$
- $2^{15} = 32768$

Προσπαθήστε να θυμάστε μέχρι το 2^{10}

Παράδειγμα 1.1

- Μετατροπή από το **Δυαδικό** στο Δεκαδικό σύστημα
  - *Μετατρέψτε τον αριθμό* 10110_2 *στο δεκαδικό*

Παράδειγμα 1.1

- Μετατροπή από το **Δυαδικό** στο Δεκαδικό σύστημα
  - *Μετατρέψτε τον αριθμό* 10110_2 *στο δεκαδικό*

$$16\times1 + 8\times0 + 4\times1 + 2\times1 + 1\times0 = 22_{10}$$

Παράδειγμα 1.2

- Μετατροπή από το **Δεκαδικό** στο **Δυαδικό** σύστημα
  – *Μετατρέψτε τον αριθμό 84_{10} στο δυαδικό*

Μετατροπή από το δεκαδικό στο δυαδικό σύστημα

- Μέθοδος 1: Από αριστερά προς τα δεξιά
  1. *Βρίσκω την μεγαλύτερη δύναμη του 2 που χωράει (≤) στον αριθμό*
  2. *Εάν χωράει, στη στήλη αυτής της δύναμης το αντίστοιχο ψηφίο του δυαδικού αριθμού είναι 1, αλλιώς είναι 0*
  3. *Αφαιρώ τη δύναμη του 2 από τον αριθμό*
  4. *Επαναλαμβάνω*

Μετατροπή από το δεκαδικό στο δυαδικό σύστημα

Μέθοδος 1: Από αριστερά προς τα δεξιά

84_{10}	$2^6=64<84$	64×1
$84-64=20_{10}$	$2^5=32>20$	32×0
20_{10}	$2^4=16<20$	16×1
$20-16=4_{10}$	$2^3=8>4$	8×0
4_{10}	$2^2=4=4$	4×1
$4-4=0_{10}$	$2^1=2>0$	2×0
0_{10}	$2^0=1>0$	1×0

$= 1010100_2$

Μετατροπή από το δεκαδικό στο δυαδικό σύστημα

- Μέθοδος 2: Από δεξιά προς τα αριστερά
  1. *Επαναληπτικά διαιρώ με το 2*
  2. *Βάζω το υπόλοιπο (0 ή 1) ως ψηφίο του δυαδικού αριθμού από δεξιά προς τα αριστερά*
  3. *Επαναλαμβάνω*

Μετατροπή από το δεκαδικό στο δυαδικό σύστημα

Μέθοδος 2: Από δεξιά προς τα αριστερά (διαίρεση δια 2)

84_{10} =	84/2 = 42	Υπ. 0
	42/2 = 21	Υπ. 0
	21/2 = 10	Υπ. 1
	10/2 = 5	Υπ. 0
	5/2 = 2	Υπ. 1
	2/2 = 1	Υπ. 0
	1/2 = 0	Υπ. 1

= 1010100_2

Εύρος τιμών ενός αριθμού με Ν ψηφία

- **Δεκαδικός Αριθμός Ν ψηφίων**
  - *Πόσες τιμές μπορεί να πάρει?* 10^N
  - *Εύρος?* $[0, 10^N - 1]$
  - *Παράδειγμα: Δεκαδικός αριθμός 3 ψηφίων:*
    - 10^3 = 1000 πιθανές τιμές
    - Εύρος: [0, 999]
- **Δυαδικός Αριθμός Ν ψηφίων**
  - *Πόσες τιμές μπορεί να πάρει?* 2^N
  - *Εύρος?* $[0, 2^N - 1]$
  - *Παράδειγμα: Δυαδικός αριθμός 3 ψηφίων:*
    - 2^3 = 8 πιθανές τιμές
    - Εύρος: [0, 7] = [000_2 to 111_2]

Δεκαεξαδικοί Αριθμοί (hexadecimal)

- Η βάση είναι το 16
- Ευκολότερη αναπαράσταση των Δυαδικών αριθμών

Δεκαεξαδικό	Δεκαδικό	Δυαδικό
0	0	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	

Δεκαεξαδικό	Δεκαδικό	Δυαδικό
8	8	
9	9	
A	10	
B	11	
C	12	
D	13	
E	14	
F	15	

Δεκαεξαδικοί Αριθμοί (hexadecimal)

- Η βάση είναι το 16
- Ευκολότερη αναπαράσταση των Δυαδικών αριθμών

Δεκαεξαδικό	Δεκαδικό	Δυαδικό
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111

Δεκαεξαδικό	Δεκαδικό	Δυαδικό
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Αριθμητικά Συστήματα

- Δεκαεξαδικό σύστημα αναπαράστασης αριθμών (0-9,A-F)

Στήλη των 1
Στήλη των 16
Στήλη των 256

Κάθε στήλη ενός δεκαεξαδικού αριθμού έχει δεκαεξαπλάσιο βάρος από την προηγούμενη στήλη. Ξεκινώντας από τα δεξιά προς τα αριστερά τα βάρη είναι: **16^0, 16^1, 16^2 =256, 16^3 =4096, …**

$$2ED_{16} = 2 \times 16^2 + E \times 16^1 + D \times 16^0$$

Δύο διακοσιο-πενηνταεξάδες

Δεκατέσσερις δεκαεξάδες

Δεκατρείς μονάδες

Παράδειγμα 1.3

- Μετατροπή από το **Δεκεξαδικό** στο Δεκαδικό σύστημα
  - *Μετατρέψτε τον αριθμό $2ED_{16}$ στο δεκαδικό*

$$2ED_{16} = 2 \times 16^2 + 14 \times 16^1 + 13 \times 16^0 = 749_{10}$$

Δεκαεξαδικό	Δεκαδικό
A	10
B	11
C	12
D	13
E	14
F	15

Παράδειγμα 1.3

- Μετατροπή από το **Δεκεξαδικό** στο Δυαδικό σύστημα
  - *Μετατρέψτε τον αριθμό $2ED_{16}$ στο δυαδικό*
- Κάθε δεκαεξαδικό ψηφίο μετατρέπεται σε τέσσερα δυαδικά ψηφία σύμφωνα με τον πίνακα

$2ED_{16} = 0010\ 1110\ 1101 = 001011101101_2$

Δεκαεξαδικό	Δυαδικό
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Δεκαεξαδικό	Δυαδικό
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Παράδειγμα 1.4

- Μετατροπή από το **Δυαδικό** στο **Δεκεξαδικό** σύστημα
  - *Μετατρέψτε τον αριθμό 1111010_2 στο δεκαεξαδικό*
- Ξεκινώντας από δεξιά προς τα αριστερά χωρίζουμε σε τετράδες δυαδικών ψηφίων
- Η περισσότερο σημαντική τετράδα (στα αριστερά) μπορεί να έχει λιγότερα από τέσσερα ψηφία, οπότε συμπληρώνεται με **μηδενικά στα αριστερά**
- Κάθε τετράδα δυαδικών ψηφίων μετατρέπεται σε ένα δεκαεξαδικό ψηφίο σύμφωνα με τον προηγούμενο πίνακα

$$1111010_2 = 0111_2\ 1010_2 = 7_{16}\ A_{16} = 7A_{16}$$

Παράδειγμα 1.5

- Μετατροπή από το **Δεκαδικό** στο **Δεκεξαδικό** σύστημα
  - *Μετατρέψτε τον αριθμό 333_{10} στο δεκαεξαδικό*

Αρχικά μετατρέπουμε τον δεκαδικό αριθμό στο δυαδικό και στη συνέχεια μετατρέπουμε το δυαδικό αριθμό στο δεκαεξαδικό (απλούστερη μέθοδος με πιο απλές πράξεις)

333_{10} = 333/2 = 166 Υπ. 1
166/2 = 83 Υπ. 0
83/2 = 41 Υπ. 1
41/2 = 20 Υπ. 1
20/2 = 10 Υπ. 0
10/2 = 5 Υπ. 0
5/2 = 2 Υπ. 1
2/2 = 1 Υπ. 0
1/2 = 0 Υπ. 1 = 101001101_2

101001101_2 = 0001 0100 1101 = $14D_{16}$

Επιλεγμένες ασκήσεις

- Άσκηση 1.13 / 1.15
  Μετατρέψτε τους ακόλουθους μη προσημασμένους δυαδικούς αριθμούς σε δεκαδικούς και σε δεκαεξαδικούς. Δείξτε αναλυτικά τη μετατροπή.

  (β) 110110_2 (γ) 11110000_2

- Άσκηση 1.17
  Μετατρέψτε τους ακόλουθους δεκαεξαδικούς αριθμούς σε δεκαδικούς. Δείξτε αναλυτικά τη μετατροπή.

  (α) $A5_{16}$ (γ) $FFFF_{16}$

- Άσκηση 1.25 / 1.27
  Μετατρέψτε τους ακόλουθους δεκαδικούς αριθμούς σε μη προσημασμένους δυαδικούς και στη συνέχεια σε δεκαεξαδικούς.

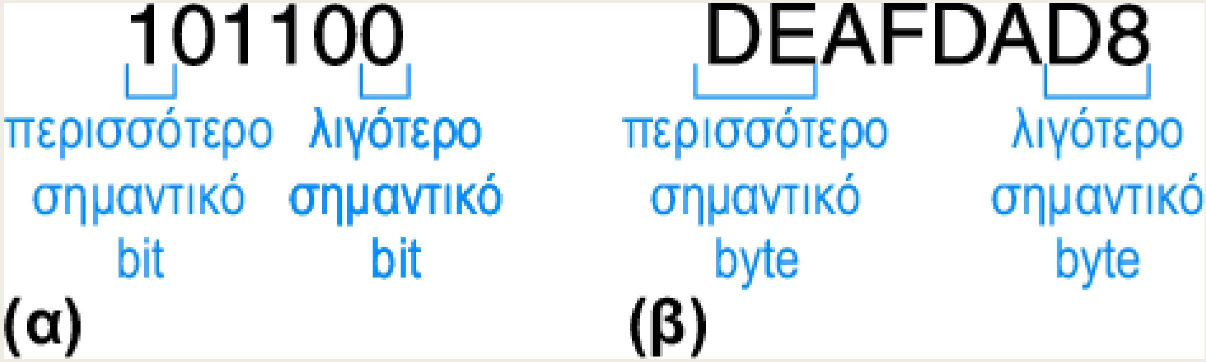
  (β) 63_{10} (δ) 845_{10}

Byte, nibble και λέξεις

- *Byte* ονομάζεται μία ομάδα των οκτώ bits
  - *Αναπαριστά μία από $2^8 = 256$ πιθανές τιμές*
  - *Χρησιμοποιείται ως μέγεθος αποθήκευσης στη μνήμη*
- *Nibble* ονομάζεται μία ομάδα των τεσσάρων bits ή μισού byte
  - *Αναπαριστά μία από $2^4 = 16$ πιθανές τιμές*
  - *Σε ένα δεκαεξαδικό ψηφίο μπορεί να αποθηκευτεί ένα nibble, ενώ σε δύο δεκαεξαδικά ψηφία μπορεί να αποθηκευτεί ένα πλήρες byte*
- Οι μικροεπεξεργαστές χειρίζονται δεδομένα σε τεμάχια τα οποία ονομάζονται *λέξεις* (words).
  - *Το μέγεθος μιας λέξης εξαρτάται από την αρχιτεκτονική του μικροεπεξεργαστή*
  - *Στα σύγχρονα υπολογιστικά συστήματα χρησιμοποιούνται μικροεπεξεργαστές των 64 ή των 32 bit.*
  - *Ενσωματωμένοι επεξεργαστές σε απλές οικιακές συσκευές χρησιμοποιούν λέξεις των 8 ή των 16 bit*

Περισσότερο & Λιγότερο Σημαντικά Bit/Byte

- Σε μία ομάδα από bits
  - *το πιο αριστερό bit ονομάζεται **περισσότερο σημαντικό bit** (**most significant bit - MSB**)*
  - *το bit που βρίσκεται στο άλλο άκρο δεξιά λέγεται **λιγότερο σημαντικό bit** (**least significant bit - LSB**)*
- Ομοίως, στο εσωτερικό μιας λέξης
  - *το πιο αριστερό byte ονομάζεται **περισσότερο σημαντικό byte***
  - *το byte που βρίσκεται στο δεξί άκρο λέγεται **λιγότερο σημαντικό byte***



Bytes, bits/sec και διαβαθμίσεις

- Ο όρος **kilo** (χιλιάδα) αντιστοιχεί στο $2^{10} \approx 10^{3}$
- Ο όρος **mega** (εκατομμύριο) αντιστοιχεί στο $2^{20} \approx 10^{6}$
- Ο όρος **giga** (δισεκατομμύριο) αντιστοιχεί στο $2^{30} \approx 10^{9}$
- Η χωρητικότητα της μνήμης συνήθως μετριέται σε **bytes**
  - *τα 2^{10} bytes είναι ένα kilobyte (1KB) δηλαδή 1024 bytes*
  - *τα 2^{20} bytes είναι ένα megabyte (1MB) δηλαδή 1024 x 1024 = 1.048.576 bytes*
  - *τα 2^{30} bytes είναι ένα gigabyte (1GB) δηλαδή 1024 x 1024 x 1024 = 1.073.741.824 bytes*
- Η ταχύτητα των καναλιών επικοινωνίας μετριέται σε **bits/sec (bps)**
  - *τα 2^{10} bps είναι ένα kilobit/sec (1Kbps) δηλαδή 1024 bits/sec*
  - *τα 2^{20} bps είναι ένα megabit/sec (1Mbps) δηλαδή 1024 x 1024 = 1.048.576 bits/sec*
  - *τα 2^{30} bps είναι ένα gigabit/sec (1Gbps) δηλαδή 1024 x 1024 x 1024 = 1.073.741.824 bits/sec*

Επιλεγμένες ασκήσεις

- Άσκηση 1.45
  Ένα συγκεκριμένο μόντεμ DSL λειτουργεί στα 768 Kbit/sec. Πόσα Kbyte μπορεί να λάβει σε 1 λεπτό;

Επιλεγμένες ασκήσεις

- Άσκηση 1.45
  Ένα συγκεκριμένο μόντεμ DSL λειτουργεί στα 768 Kbit/sec. Πόσα Kbyte μπορεί να λάβει σε 1 λεπτό;

– Σε ένα λεπτό μπορεί να λάβει 60*768Kbit=>46080Kbit/min

– 1 Byte=8 bit=>46080Kbit/min=5760Kbyte/min

Επιλεγμένες ασκήσεις

- Άσκηση 1.47
  Όταν οι κατασκευαστές σκληρών δίσκων χρησιμοποιούν τους όρους «megabyte» και «gigabyte», εννοούν 10^6 byte και 10^9 byte, αντίστοιχα. Πόσα πραγματικά GB μουσικής μπορείτε να αποθηκεύσετε σε έναν σκληρό δίσκο με χωρητικότητα 50 GB;

Επιλεγμένες ασκήσεις

- Άσκηση 1.47
  Όταν οι κατασκευαστές σκληρών δίσκων χρησιμοποιούν τους όρους «megabyte» και «gigabyte», εννοούν 10^6 byte και 10^9 byte, αντίστοιχα. Πόσα πραγματικά GB μουσικής μπορείτε να αποθηκεύσετε σε έναν σκληρό δίσκο με χωρητικότητα 50 GB;
  - *50GB=50.000.000.000 bytes*
  - *1 (real) kbyte=1024bytes=> 50.000.000.000 bytes = (50.000.000.000/1024) kbytes=48.828.125 Kbytes*
  - *1 (real) Mbyte=1024Kbyte=> 48.828.125 Kbytes = (48.828.125 /1024) Mbytes=47.683,72 Mbytes*
  - *1 (real) Gbyte=1024Mbyte=> 47.683,72 Mbytes = (47.683,72 /1024) Gbytes=46,566 Gbytes*

Παράδειγμα 1.6

- Υπολογισμός κατά προσέγγιση δυνάμεων του 2 στο δεκαδικό σύστημα
  - *Βρείτε κατά προσέγγιση την τιμή του 2^{24}*
  - *Χωρίζουμε τον εκθέτη σε δύο μέρη*
    - ένα πολλαπλάσιο του δέκα
    - και το υπόλοιπο

    $2^{24} = 2^{20} \times 2^{4}$

    $2^{20} = 2^{10} * 2^{10} \approx 10^{3} * 10^{3} = 1000*1000 = 1$ εκατομμύριο.

    $2^{4} = 16.$

Παράδειγμα 1.6

- Υπολογισμός κατά προσέγγιση δυνάμεων του 2 στο δεκαδικό σύστημα
  - *Βρείτε κατά προσέγγιση την τιμή του 2^{24}*
  - *Χωρίζουμε τον εκθέτη σε δύο μέρη*
    - ένα πολλαπλάσιο του δέκα
    - και το υπόλοιπο

    $2^{24} = 2^{20} \times 2^{4}$

    $2^{20} = 2^{10} * 2^{10} \approx 10^{3} * 10^{3}$ =1000*1000=1 εκατομμύριο.

    $2^{4} = 16$.

    Άρα, $2^{24} \approx 16$ εκατομμύρια (καλή προσέγγιση)

    Με ακρίβεια είναι $2^{24} = 16.777.216$

Επιλεγμένες ασκήσεις

- Άσκηση 1.48
  Εκτιμήστε την τιμή του 2^{31} χωρίς τη χρήση αριθμομηχανής

- Άσκηση 1.49
  Μια μνήμη στον μικροεπεξεργαστή Pentium II είναι οργανωμένη ως ορθογώνια διάταξη bit με 2^8 γραμμές και 2^9 στήλες. Εκτιμήστε πόσα bit (σε χιλιάδες) διαθέτει η μνήμη χωρίς να χρησιμοποιήσετε αριθμομηχανή

Επιλεγμένες ασκήσεις

- Άσκηση 1.48
  Εκτιμήστε την τιμή του 2^{31} χωρίς τη χρήση αριθμομηχανής
  - *2 Δις*
- Άσκηση 1.49
  Μια μνήμη στον μικροεπεξεργαστή Pentium II είναι οργανωμένη ως ορθογώνια διάταξη bit με 2^8 γραμμές και 2^9 στήλες. Εκτιμήστε πόσα bit (σε χιλιάδες =>Kbyte) διαθέτει η μνήμη χωρίς να χρησιμοποιήσετε αριθμομηχανή
  - $2^9 * 2^8 = 2^{17} = 2^{10} * 2^7 \approx 1000 * 2^7$
  - $2^9 * 2^8 \text{bit} \approx 1000 * 2^7 \text{ bit} = 2^7 \text{ Kbit} = 128 \text{ Kbit} = 16 \text{ Kbyte}$

Ερωτήσεις συνεντεύξεων

- Ερώτηση 1.3
  Ένας καθηγητής, ένας βοηθός διδασκαλίας, ένας φοιτητής σχεδίασης ψηφιακών συστημάτων και ένας πρωτοετής που είναι το αστέρι της ομάδας στίβου του πανεπιστημίου πρέπει να διασχίσουν μια ετοιμόρροπη γέφυρα κατά τη διάρκεια μιας νύχτας χωρίς φεγγάρι.
  Η γέφυρα είναι τόσο ασταθής που **μόνο δύο άτομα** μπορούν να τη διασχίζουν **ταυτόχρονα και θα πρέπει να έχουν μαζί τους φακό για να βλέπουν**.
  Οι τέσσερις πρωταγωνιστές έχουν μόνο έναν φακό στη διάθεσή τους, και η απόσταση της γέφυρας από τη μία άκρη έως την άλλη είναι υπερβολικά μεγάλη για να πετούν τον φακό, οπότε κάποιος πρέπει να τον μεταφέρει πίσω στους άλλους.
  Το αστέρι του στίβου μπορεί να διασχίσει τη γέφυρα σε 1 λεπτό. Ο φοιτητής σχεδίασης ψηφιακών συστημάτων μπορεί να τη διασχίσει σε 2 λεπτά. Ο βοηθός διδασκαλίας μπορεί να τη διασχίσει σε 5 λεπτά. Ο καθηγητής πάντα αφαιρείται, με αποτέλεσμα να χρειάζεται 10 λεπτά για να διασχίσει τη γέφυρα.
  Ποιος είναι ο ταχύτερος χρόνος που απαιτείται για να μεταβούν όλοι οι πρωταγωνιστές μας από τη μία άκρη της γέφυρας στην άλλη;

Ερωτήσεις συνεντεύξεων

- Ερώτηση 1.3 (Απάντηση)

1. Αρχικά διασχίζουν τη γέφυρα οι 2 πιο γρήγοροι (δρομέας, φοιτητής) σε 2 λεπτά.
2. Ο πιο γρήγορος (δρομέας) επιστρέφει το φακό σε 1 λεπτό
3. Οι δύο πιο αργοί (βοηθός, καθηγητής) διασχίζουν τη γέφυρα σε 10 λεπτά.
4. Το φακό τον επιστρέφει ο φοιτητής (που είναι ο πιο γρήγορος από όσους έχουν διασχίσει τη γέφυρα) σε 2 λεπτά
5. Ο δρομέας και ο φοιτητής διασχίζουν τη γέφυρα σε 2 λεπτά.

Σύνολο: 17 λεπτά.

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

Πρόσθεση δυαδικών αριθμών

- Ακριβώς όπως και στο δεκαδικό σύστημα πρόσθέτουμε τα ψηφία της ίδιας στήλης (ή αλλιώς ίδιου βάρους) μαζί με το bit κρατούμενου (αν υπάρχεί)

- Στο δεκαδικό

```
    11    ← κρατούμενο
   4277
+  5499
-------
   9776
```

- Στο δυαδικό

```
    11    ← κρατούμενο
   1011
+  0011
-------
   1110
```

Παράγεται bit κρατούμενου:
$1+1 = 2_{10} = 10_2$
$1+1+1 = 3_{10} = 11_2$

Υπερχείλιση μη προσημασμένων

- Ενας μη προσημασμένος (unsigned) δυαδικός αριθμός με N bit μπορεί να πάρει τιμές από το διάστημα $[0, 2^{N}-1]$
  - *Υπάρχει η περίπτωση το αποτέλεσμα μιας πρόσθεσης να χρειάζεται N+1 bit για αναπαρασταθεί*
  - *Σε αυτή την περίπτωση το bit που βρίσκεται στην N+1 θέση* ***αγνοείται*** *και έτσι τα υπόλοιπα N bits αναπαριστούν ένα λάθος αποτέλεσμα*
  - *Αυτό το φαινόμενο λέγεται* ***υπερχείλιση***
    - μπορεί να ανιχθευθεί μέσω του ελέγχου για κάποιο **κρατούμενο εξόδου (carry)**, το οποίο παράγεται στη στήλη του πιο σημαντικού bit
- Όταν κάνουμε πράξεις με δυαδικούς αριθμούς των N bit, συνήθως είναι ζητούμενο το αποτέλεσμα να αποθηκεύεται επίσης σε αριθμό των N bit.

Δυαδικές κωδικοποιήσεις με 4 bit

- Μη προσημασμένοι αριθμοί
- Εύρος τιμών στα 4 bit: $[0, 2^4-1]$

Δεκαδικός	Μη προσημασμένος
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Δεκαδικός	Μη προσημασμένος
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Παράδειγμα 1.7

- Προσθέστε τους ακόλουθους μη προσημασμένους δυαδικούς αριθμούς μεγέθους 4 bit (Παράδειγμα 1.7)

$$
\begin{array}{r}
0111 \\
+\quad 0101 \\
\hline
\end{array}
$$

Παράδειγμα 1.7

- Προσθέστε τους ακόλουθους μη προσημασμένους δυαδικούς αριθμούς μεγέθους 4 bit (Παράδειγμα 1.7)

$$
\begin{array}{rl}
 & 111 \quad \leftarrow \text{κρατούμενο} \\
 & 0111 \\
+ & 0101 \\
\hline
 & 1100
\end{array}
$$

  – *Επαλήθευση* $0111_2 = 7_{10}$, $0101_2 = 5_{10}$, $1100_2 = 12_{10}$

Παράδειγμα 1.8

- Προσθέστε τους ακόλουθους μη προσημασμένους δυαδικούς αριθμούς μεγέθους 4 bit (Παράδειγμα 1.8)

$$\begin{array}{r} 1101 \\ +\quad 0101 \\ \hline \end{array}$$

Παράδειγμα 1.8

- Προσθέστε τους ακόλουθους μη προσημασμένους δυαδικούς αριθμούς μεγέθους 4 bit (Παράδειγμα 1.8)

  – *Επαλήθευση* $1101_2 = 13_{10}$, $0101_2 = 5_{10}$, $10010_2 = 18_{10}$

$$
\begin{array}{rr}
 & 11\ 1 \\
 & 1101 \\
+ & 0101 \\
\hline
 & 0010
\end{array}
$$

← κρατούμενο

Παράδειγμα 1.8

- Προσθέστε τους ακόλουθους μη προσημασμένους δυαδικούς αριθμούς μεγέθους 4 bit (Παράδειγμα 1.8)

  – *Επαλήθευση* $1101_2 = 13_{10}$, $0101_2 = 5_{10}$, $10010_2 = 18_{10}$

$$
\begin{array}{rr}
 & 11\,\,1 \\
 & 1101 \\
+ & 0101 \\
\hline
 & \cancel{1}0010
\end{array}
$$

← κρατούμενο (Carry)

Υπερχείλιση: 2 αντί 18!
(Overflow)

Συνήθως όμως θέλουμε οι αριθμοί που προσθέτουμε και το αποτέλεσμα να έχουν ίδιο αριθμό από bit. Σε αυτή την περίπτωση δεν μπορούμε να αποθηκεύσουμε το 5^o bit στο αποτέλεσμα και έχουμε Υπερχείλιση : Αποτέλεσμα 2 αντί 18!. Αν μπορούσαμε αν έχουμε 5 bit στο αποτέλεσμα, τότε αυτό θα ήταν σωστό!

Προσημασμένοι δυαδικοί αριθμοί

- Μέχρι στιγμής μελετήσαμε **μη προσημασμένους (unsigned)** ακέραιους αριθμούς στο δυαδικό σύστημα αρίθμησης στους οποίους απεικονίζονται μόνο οι **φυσικοί ακέραιοι αριθμοί**.
- Για να απεικονίστουν **αρνητικοί ακέραιοι αριθμοί** στο δυαδικό σύστημα αρίθμησης πρέπει να γίνει χρήση ενός **προσημασμένου (signed) συστήματος αρίθμησης**
- Υπάρχουν αρκετές μέθοδοι για την αναπαράσταση προσημασμένων ακεραίων αριθμών (θετικών και αρνητικών) στο δυαδικό σύστημα. Οι πιο διαδεδομένες σήμερα είναι:
  - *Αριθμοί προσήμου-μεγέθους (sign/magnitude)*
  - *Αριθμοί συμπληρώματος ως προς δύο (two’s complement)*

Αριθμοί προσήμου-μεγέθους

- Σε έναν αριθμό προσήμου-μεγέθους των N bit **το πιο σημαντικο bit** του αριθμού υποδηλώνει το πρόσημο του
  - *Το ‘0’ στο MSB δηλώνει οτι ο αριθμός είναι* ***θετικός***
  - *Το ‘1’ στο MSB δηλώνει οτι ο αριθμός είναι* ***αρνητικός***
- Τα υπόλοιπα N-1 bit του αριθμού δηλώνουν το μέγεθος του αριθμού, δηλαδή την **απόλυτη τιμή** του
- Άρα, για έναν αριθμό πρόσημου-μεγέθους N-bit ισχύει
  - *1 bit πρόσημου, N-1 bits μέγεθος (απόλυτη τιμή)*
- Ένας αριθμός προσήμου-μεγέθους των N bit παίρνει τιμές από το κλειστό διάστημα: $[-(2^{N-1} - 1), 2^{N-1} - 1]$

Δυαδικές κωδικοποιήσεις με 4 bit

- Προσημασμένοι αριθμοί προσήμου-μεγέθους
- Εύρος τιμών στα 4 bit: $[-(2^3-1), 2^3-1]$

Δεκαδικός	Προσημασμένος προσήμου μεγέθους
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Δεκαδικός	Προσημασμένος προσήμου μεγέθους
–0	1000
–1	1001
–2	1010
–3	1011
–4	1100
–5	1101
–6	1110
–7	1111

Παράδειγμα 1.9

- Μετατροπή προσημασμένων δεκαδικών αριθμών σε δυαδικούς αριθμούς προσήμου-μεγέθους
- Οι δεκαδικοί ακέραιοι αριθμοί ± 5 σε αναπαράσταση δυαδικού αριθμού προσήμου-μεγέθους με 4 bit είναι:

Παράδειγμα 1.9

- Μετατροπή προσημασμένων δεκαδικών αριθμών σε δυαδικούς αριθμούς προσήμου-μεγέθους
- Οι δεκαδικοί ακέραιοι αριθμοί ± 5 σε αναπαράσταση δυαδικού αριθμού προσήμου-μεγέθους με 4 bit είναι:
  - *Αρχικά βρίσκουμε το μέγεθος τους αριθμού μετατρέποντας τον δεκαδικό αριθμό σε μη προσημασμένο δυαδικό αριθμό*
  - $5_{10} = 101_2$

Παράδειγμα 1.9

- Μετατροπή προσημασμένων δεκαδικών αριθμών σε δυαδικούς αριθμούς προσήμου-μεγέθους
- Οι δεκαδικοί ακέραιοι αριθμοί ± 5 σε αναπαράσταση δυαδικού αριθμού προσήμου-μεγέθους με 4 bit είναι:
  - *Αρχικά βρίσκουμε το μέγεθος τους αριθμού μετατρέποντας τον δεκαδικό αριθμό σε μη προσημασμένο δυαδικό αριθμό*
  - $5_{10} = 101_2$
  - *Στη συνέχεια βάζουμε και το bit προσήμου*

    +5 = 0101

    –5 = 1101

Αριθμοί προσήμου-μεγέθους

- **Προβλήματα** στην αναπαράσταση προσήμου-μεγέθους
  - *Η **πρόσθεση** δεκαδικών αριθμών στην αναπαράσταση προσήμου-μεγέθους **δεν δίνει σωστά αποτελέσματα***
  - *Για παράδειγμα, –5 + 5 = 0:*

Αριθμοί προσήμου-μεγέθους

- **Προβλήματα** στην αναπαράσταση προσήμου-μεγέθους
  - *Η **πρόσθεση** δεκαδικών αριθμών στην αναπαράσταση προσήμου-μεγέθους **δεν δίνει σωστά αποτελέσματα***
  - *Για παράδειγμα, –5 + 5 = 0:*

$$
\begin{array}{r}
1101 \\
+\quad 0101 \\
\hline
\cancel{1}0010
\end{array}
\quad (2_{10} \text{ αντί για 0 – Λάθος!})
$$

- Επίσης, υπάρχουν δύο αναπαραστάσεις του 0
  - ***0000*** *(+0)*
  - ***1000*** *(–0)*

Αριθμοί συμπληρώματος ως προς 2

- Δεν έχουν τα προβλήματα των αριθμών προσήμου-μεγέθους
  - *Η **πρόσθεση** δίνει **σωστά αποτελέσματα** είτε οι αριθμοί είναι μη προσημασμένοι (unsigned) είτε είναι προσημασμένοι (signed) σε αναπαράσταση συμπληρώματος ως προς δύο*
  - *Υπάρχει **μόνο μία αναπαράσταση για το 0***
  - *Το **κρατούμενο εξόδου** που παράγεται κατά την πρόσθεση **αγνοείται** χωρίς να απαιτείται άλλη ενέργεια*
  - *Δεν <u>αλλάζει το πλήθος των ψηφίων του αριθμού μετά την πρόσθεση</u>, αλλά θέλει προσοχή στην **υπερχείλιση***

**** Η υπερχείλιση παρότι υπάρχει ως έννοια και στους μη προσημασμένους αλλά και στους προσημασμένους, αντιμετωπίζεται (εντοπίζεται) διαφορετικά.***

Αριθμοί συμπληρώματος ως προς δύο

- Είναι ίδιοι με τους μη προσημασμένους δυαδικούς αριθμούς των N bit με τη μόνη διαφορά ότι η θέση του περισσότερου σημαντικού bit (MSB) έχει βάρος ίσο με -2^{N-1} αντί για 2^{N-1}
- Το **πιο σημαντικό bit** συνεχίζει να δηλώνει το **πρόσημο** (**0 = θετικός**, **1 = αρνητικός**)
- Ο μεγαλύτερος θετικός αριθμός στα 4 bit είναι ο αριθμός **+7**: **0111**
- Το **0** αναπαρίσταται πάντα με τα ψηφία **όλα-0** (0000)
- Το **–1** αναπαρίσταται πάντα με τα ψηφία **όλα-1** (1111)
- Ο μικρότερος αρνητικός αριθμός στα 4 bit είναι ο αριθμός **–8**: 1000
- Ένας αριθμός συμπληρώματος ως προς δύο των N bit παίρνει τιμές από το κλειστό διάστημα: $[-2^{N-1}, 2^{N-1} - 1]$

Δυαδικές κωδικοποιήσεις με 4 bit

- Προσημασμένοι αριθμοί συμπληρώματος ως προς δύο
- Εύρος τιμών στα N bit: $[-2^3, 2^3-1]$

Δεκαδικός	Προσημασμένος συμπλ. ως προς δύο
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Δεκαδικός	Προσημασμένος συμπλ. ως προς δύο
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111

Αριθμοί συμπληρώματος ως προς δύο

- **Μετατροπή** προσημασμένου αριθμού συμπληρώματος ως προς δύο σε δεκαδικό προσημασμένο ακέραιο αριθμό:
  - *Η διαδικασία είναι ίδια με τους μη προσημασμένους δυαδικούς αριθμούς με τη μόνη διαφορά ότι το πιο σημαντικό bit (το bit προσήμου) και έχει βάρος -2^{N-1} αντί για 2^{N-1}*
  - **$1011 = 1 \times -2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = -5$**
- **Υπολογισμός του συμπληρώματος ως προς δύο** ενός αριθμού με αναπαράσταση συμπληρώματος ως προς δύο (δηλαδή από τον αριθμό Χ οδηγούμαστε στον –Χ)
  - *Αντιστροφή όλων των bit του δυαδικού αριθμού*
  - *Πρόσθεση του ‘1’*
- Για παράδειγμα ο αριθμός $3_{10} = 0011_2$ έχει συμπλήρωμα ως προς δύο το -3_{10} που βρίσκεται ως εξής:
  1. Αντιστροφή bit: 1100
  2. Πρόσθεση +1: +0001

$1101 = -3_{10}$

Παράδειγμα 1.10

- Αναπαράσταση ενός αρνητικού αριθμού με χρήση συμπληρώματος ως προς δυο
- Βρείτε πώς μπορεί να αναπαρασταθεί το -2_{10} ως αριθμός συμπληρώματος ως προς δύο με 4 bit.

Παράδειγμα 1.10

- Αναπαράσταση ενός αρνητικού αριθμού με χρήση συμπληρώματος ως προς δυο
- Βρείτε πώς μπορεί να αναπαρασταθεί το -2_{10} ως αριθμός συμπληρώματος ως προς δύο με 4 bit.
  - *Αρχικά βρίσκουμε τον δυαδικό αριθμό με το ίδιο μέγεθος*
    $+2_{10} = 0010_2$

Παράδειγμα 1.10

- Αναπαράσταση ενός αρνητικού αριθμού με χρήση συμπληρώματος ως προς δυο
- Βρείτε πώς μπορεί να αναπαρασταθεί το -2_{10} ως αριθμός συμπληρώματος ως προς δύο με 4 bit.
  - *Αρχικά βρίσκουμε τον δυαδικό αριθμό με το ίδιο μέγεθος* $+2_{10} = 0010_2$
  - *Στη συνέχεια αντιστρέφουμε τα bit του* 0010_2 *οπότε παράγεται το* 1101_2

Παράδειγμα 1.10

- Αναπαράσταση ενός αρνητικού αριθμού με χρήση συμπληρώματος ως προς δυο
- Βρείτε πώς μπορεί να αναπαρασταθεί το -2_{10} ως αριθμός συμπληρώματος ως προς δύο με 4 bit.
  - *Αρχικά βρίσκουμε τον δυαδικό αριθμό με το ίδιο μέγεθος*
    $+2_{10} = 0010_2$
  - *Στη συνέχεια αντιστρέφουμε τα bit του* 0010_2
    οπότε παράγεται το 1101_2
  - *Τέλος, προσθέτουμε το 1 στο προηγούμενο αποτέλεσμα*
    $1101_2 + 0001_2 = 1110_2 = -2_{10}$

Παράδειγμα 1.11

- <u>Εύρεση της τιμής των αρνητικών αριθμών με αναπαράσταση συμπληρώματος ως προς δύο</u>
- Βρείτε τη δεκαδική τιμή του αριθμού 1001_2, ο οποίος αναπαρίσταται με τη χρήση συμπληρώματος ως προς δύο.
- Μέθοδος 1: Εύρεση συμπληρώματος ως προς 2
  - *Αντιστρέφουμε τα bit και προσθέτουμε 1*
  - $1001_2 = 0110_2$
  - $0110_2 + 0001_2 = 0111_2 = 7_{10}$
  - *Άρα,* $1001_2 = -7_{10}$
- Μέθοδος 2: Μετατροπή στον αντίστοιχο δεκαδικό αριθμό με βάση ότι το βάρος του MSB είναι -2^3 αντί για 2^3
  - $1001 = 1 \times -2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = -7$

    (Προσοχή ότι μόνο το MSB έχει αρνητικό βάρος)

Συμπλήρωμα ως προς δύο του 0

- Βρείτε πώς μπορεί να αναπαρασταθεί το -0_{10} ως αριθμός συμπληρώματος ως προς δύο με 4 bit.
  - *Αρχικά βρίσκουμε τον δυαδικό αριθμό με το ίδιο μέγεθος* $+0_{10} = 0000_2$
  - *Στη συνέχεια αντιστρέφουμε τα bit του* 0000_2 *οπότε παράγεται το* 1111_2
  - *Τέλος, προσθέτουμε το 1 στο προηγούμενο αποτέλεσμα* $1111_2 + 0001_2 = \cancel{1}0000_2 = -0_{10}$
- Δεν υπάρχει ξεχωριστή αναπαράσταση για το -0_{10} με αναπαράσταση συμπληρώματος ως προς δύο
- Το <u>μηδέν θεωρείται θετικό</u> επειδή στο bit του προσήμου του υπάρχει το ψηφίο 0
- Συνεπώς οι θετικοί αριθμοί που μπορούν να αναπαρασταθούν με συμπλήρωμα ως προς δύο είναι κατά ένας λιγότεροι σε σχέση με τους αντίστοιχους αρνητικούς αριθμούς

Συμπλήρωμα ως προς δύο του -2^{N-1}

- Βρείτε εάν μπορεί να αναπαρασταθεί το $\mathbf{+2^{N-1}}_{10}$ ως αριθμός συμπληρώματος ως προς δύο με 4 bit (N=4, ισχύει για κάθε N).

Συμπλήρωμα ως προς δύο του -2^{N-1}

- Βρείτε εάν μπορεί να αναπαρασταθεί το $+2^{N-1}{}_{10}$ ως αριθμός συμπληρώματος ως προς δύο με 4 bit (N=4, ισχύει για κάθε N).
  - *Αρχικά βρίσκουμε τον αριθμό συμπληρώματος ως προς δύο του* $-2^3{}_{10} = 1000_2$
  - *Στη συνέχεια αντιστρέφουμε τα bit του* 1000_2 *οπότε παράγεται το* 0111_2
  - *Τέλος, προσθέτουμε το 1 στο προηγούμενο αποτέλεσμα* $0111_2 + 0001_2 = 1000_2 = -2^3{}_{10}$
- Δεν υπάρχει δυνατότητα αναπαράστασης για το $+2^3{}_{10}$ λόγω του φαινομένου της υπερχείλισης των προσημασμένων αριθμών συμπληρώματος ως προ δύο
  - *Προσθέσαμε δύο θετικούς αριθμούς και το αποτέλεσμα ήταν ένας αρνητικός αριθμός*

Πρόσθεση προσημασμένων αριθμών

- Η πρόσθεση προσημασμένων αριθμών με αναπαράσταση συμπληρώματος ως προς δύο υλοποιείται ακριβώς όπως και στους μη προσημασμένους αριθμούς με τις ακόλουθες διαφοροποιήσεις ως προς την ερμηνεία του αποτελέσματος:
  - *Σε αντίθεση με τους μη προσημασμένους αριθμούς η παραγωγη ενός κρατούμενου εξόδου από τη στήλη του πιο σημαντικού bit* ***δεν σημαίνει πάντοτε υπερχείλιση***
  - *Η πρόσθεση δύο* ***ετερόσημων αριθμών*** *δεν παράγει ποτέ υπερχείλιση*
  - *Η πρόσθεση δύο* ***ομόσημων αριθμών*** *των N bits μπορεί να προκαλέσει υπερχείλιση αν το αποτέλεσμα είναι μεγαλύτερο του* $2^{N-1} - 1$ *ή μικρότερο του* -2^{N-1}
- Αν δύο ομόσημοι αριθμοί προστεθούν και το αποτέλεσμα έχει αντίθετο πρόσημο, τότε παρουσιάζεται **υπερχείλιση**
- **Προσοχή: Δεν ταυτίζονται οι υπερχειλίσεις που παράγονται κατά την πρόσθεση μεταξύ προσημασμένων και μη προσημασμένων αριθμών**

Παράδειγμα 1.12

- Πρόσθεση αριθμών με αναπαράσταση συμπληρώματος ως προς δύο
- Υπολογίστε τα αθροίσματα (α) $-2_{10} + 1_{10}$ και (β) $-7_{10} + 7_{10}$ χρησιμοποιώντας αριθμούς συμπληρώματος ως προς δύο, των 4 bit.

Παράδειγμα 1.12

- Πρόσθεση αριθμών με αναπαράσταση συμπληρώματος ως προς δύο
- Υπολογίστε τα αθροίσματα (α) $-2_{10} + 1_{10}$ και (β) $-7_{10} + 7_{10}$ χρησιμοποιώντας αριθμούς συμπληρώματος ως προς δύο.
  - *α)* $-2_{10} + 1_{10} = 1110_2 + 0001_2 = 1111_2 = -1_{10}$
  - *β)* $-7_{10} + 7_{10} = 1001_2 + 0111_2 = \cancel{1}0000_2 = 0000_2 = 0_{10}$

    Το κρατούμενο εξόδου (πέμπτο bit) αγνοείται, οπότε μένει το σωστό αποτέλεσμα (με 4 bit), δηλαδή το 0000_2.

Παράδειγμα 1.14

- Πρόσθεση αριθμών συμπληρώματος ως προς δύο με υπερχείλιση
- Υπολογίστε το άθροισμα $4_{10} + 5_{10}$ χρησιμοποιώντας αριθμούς συμπληρώματος ως προς δύο με 4 bit

Παράδειγμα 1.14

- Πρόσθεση αριθμών συμπληρώματος ως προς δύο με υπερχείλιση
- Υπολογίστε το άθροισμα $4_{10} + 5_{10}$ χρησιμοποιώντας αριθμούς συμπληρώματος ως προς δύο με 4 bit
  - $4_{10} + 5_{10} = 0100_2 + 0101_2 = 1001_2 = -7_{10}$
  - *Το αποτέλεσμα υπερχειλίζει το εύρος τιμών των θετικών αριθμών συμπληρώματος ως προς δύο με 4 bit (που είναι το $+7_{10}$,) παράγοντας ένα εσφαλμένο αρνητικό αποτέλεσμα, το -7_{10}*
  - *Εάν χρησιμοποιούσαμε αριθμούς τουλάχιστον των 5 bits τότε το αποτέλεσμα θα ήταν σωστό*
  - $4_{10} + 5_{10} = 00100_2 + 00101_2 = 01001_2 = 9_{10}$
  - *Προσοχή: Εάν θεωρήσουμε ότι οι αριθμοί των 4 bit είναι μη προσημασμένοι, το αποτέλεσμα είναι σωστό χωρίς υπερχείλιση!*
  - $4_{10} + 5_{10} = 0100_2 + 0101_2 = 1001_2 = 9_{10}$

Το ατύχημα του πύραυλου Ariane 5

- Ο πύραυλος Ariane 5, που κόστισε 7 δισεκατομμύρια δολάρια και εκτοξεύτηκε στις 4 Ιουνίου 1996, απέκλινε από την πορεία του 40 δευτερόλεπτα μετά από την εκτόξευση, κόπηκε στα δύο και εξερράγη.
- Η αποτυχία προκλήθηκε όταν ο υπολογιστής που έλεγχε τον πύραυλο **υπερχείλισε** το εύρος (16 bit) των προσημασμένων τιμών του και κατέρρευσε.
- Ο εν λόγω κώδικας είχε ελεγχθεί διεξοδικά στον πύραυλο Ariane 4. Όμως, ο Ariane 5 διέθετε πιο γρήγορη μηχανή η οποία παρήγαγε μεγαλύτερες τιμές για τον υπολογιστή ελέγχου, προκαλώντας έτσι την υπερχείλιση

Αφαίρεση αριθμών συμπληρώματος ως προς δύο

- Μέθοδος 1:

  Για να πραγματοποιήσουμε την αφαίρεση υπολογίζουμε το συμπλήρωμα ως προς δύο του αφαιρετέου και το προσθέτουμε στον μειωτέο

- Μέθοδος 2: Εναλλακτική μέθοδος

  Για να πραγματοποιήσουμε την αφαίρεση αντιστρέφουμε τα bit του αφαιρετέου και το προσθέτουμε στο μειωτέο ξεκινώντας με κρατούμενο εισόδου 1, αντί για 0

Παράδειγμα 1.13

- Αφαίρεση αριθμών με αναπαράσταση συμπληρώματος ως προς δύο
- Υπολογίστε τα (α) $5_{10} - 3_{10}$ και (β) $3_{10} - 5_{10}$ χρησιμοποιώντας αριθμούς συμπληρώματος ως προς δύο με 4 bit.
- Μέθοδος 1:

Παράδειγμα 1.13

- Αφαίρεση αριθμών με αναπαράσταση συμπληρώματος ως προς δύο
- Υπολογίστε τα (α) $5_{10} - 3_{10}$ και (β) $3_{10} - 5_{10}$ χρησιμοποιώντας αριθμούς συμπληρώματος ως προς δύο με 4 bit.
- **Μέθοδος 1:**

  (α) Ο μειωτέος είναι $5_{10} = 0101_2$

  Ο αφαιρετέος είναι $3_{10} = 0011_2$

  – *Υπολογίζουμε το συμπλήρωμα του αφαιρετέου ως προς δύο και παίρνουμε* $-3_{10} = 1101_2$

  – *Εκτελούμε πρόσθεση αντί για αφαίρεση*

  – $5_{10} + (-3_{10}) = 0101_2 + 1101_2 = \cancel{1}0010_2 = 2_{10}$

  – *Το κρατούμενο στην πιο σημαντική θέση αγνοείται*

Παράδειγμα 1.13

- Αφαίρεση αριθμών με αναπαράσταση συμπληρώματος ως προς δύο
- Υπολογίστε τα (α) $5_{10} - 3_{10}$ και (β) $3_{10} - 5_{10}$ χρησιμοποιώντας αριθμούς συμπληρώματος ως προς δύο με 4 bit.
- **Μέθοδος 1:**

  (β) *Ο μειωτέος είναι* $3_{10} = 0011_2$

  Ο αφαιρετέος είναι $5_{10} = 0101_2$

  – *Υπολογίζουμε το συμπλήρωμα του αφαιρετέου ως προς δύο και παίρνουμε* $-5_{10} = 1011_2$

  – *Εκτελούμε πρόσθεση αντί για αφαίρεση*

  – $3_{10} + (-5_{10}) = 0011_2 + 1011_2 = 1110_2 = -2_{10}$

Παράδειγμα 1.13

- Αφαίρεση αριθμών με αναπαράσταση συμπληρώματος ως προς δύο
- Υπολογίστε τα (α) $5_{10} - 3_{10}$ και (β) $3_{10} - 5_{10}$ χρησιμοποιώντας αριθμούς συμπληρώματος ως προς δύο με 4 bit.
- **Μέθοδος 2:**

  (α) Ο μειωτέος είναι $5_{10} = 0101_2$

  Ο αφαιρετέος είναι $3_{10} = 0011_2$

  – *Αντιστρέφουμε τα bit του αφαιρετέου και παίρνουμε* 1100_2
  – *Εκτελούμε πρόσθεση αντί για αφαίρεση με κρατούμενο εισόδου 1*
  – $5_{10} + (-3_{10}) = 0101_2 + 1100_2 + 1 = 0101_2 + 1100_2 + 0001 = 2_{10}$
  – *Το κρατούμενο στην πιο σημαντική θέση αγνοείται*

Παράδειγμα 1.13

- Αφαίρεση αριθμών με αναπαράσταση συμπληρώματος ως προς δύο
- Υπολογίστε τα (α) $5_{10} - 3_{10}$ και (β) $3_{10} - 5_{10}$ χρησιμοποιώντας αριθμούς συμπληρώματος ως προς δύο με 4 bit.
- **Μέθοδος 2:**

  (β) Ο μειωτέος είναι $3_{10} = 0011_2$

  Ο αφαιρετέος είναι $5_{10} = 0101_2$

  – *Αντιστρέφουμε τα bit του αφαιρετέου και παίρνουμε* 1010_2

  – *Εκτελούμε πρόσθεση αντί για αφαίρεση με κρατούμενο εισόδου 1*

  – $3_{10} + (-5_{10}) = 0011_2 + 1010_2 + 1 = 0011_2 + 1010_2 + 0001 = 1110_2 = -2_{10}$

Επέκταση προσήμου/μηδενός

- Ένας προσημασμένος αριθμός συμπληρώματος ως προς δύο μπορεί να επεκταθεί ώστε να έχει περισσότερα bits
  - *Αυτή η διαδικασία λέγεται **επέκταση προσήμου** (sign extension) Τα επιπλέον bits που θα προστεθούν θα αντιγράφονται από το bit προσήμου*
  - *Έστω ότι θέλουμε να προσθέσουμε 4 παραπάνω bit αριστερά:*

    3 = 0011 = 00000011

    –5 = 1011 = 11111011

- Ένας μη προσημασμένος αριθμός μπορεί να επεκταθεί ώστε να έχει περισσότερα bits
  - *Αυτή η διαδικασία λέγεται **επέκταση μηδενός** (zero extension)*
  - *Τα επιπλέον bits που θα προστεθούν θα είναι 0*
  - *Έστω ότι θέλουμε να προσθέσουμε 4 παραπάνω bit αριστερά*

    3 = 0011 = 00000011

    11 = 1011 = 00001011

Επιλεγμένες ασκήσεις

- Άσκηση 1.58 – Λύση
  Εκτελέστε τις ακόλουθες προσθέσεις **μη προσημασμένων δεκαεξαδικών** αριθμών. Αναφέρετε αν το άθροισμα θα προκαλέσει υπερχείλιση ή όχι στην περίπτωση που το αποτέλεσμα έχει 8 bit (δύο δεκαεξαδικά ψηφία).

  (γ) $AB_{16} + 3E_{16}$ (δ) $8F_{16} + AD_{16}$

Επιλεγμένες ασκήσεις

- Άσκηση 1.58 – Λύση
  Εκτελέστε τις ακόλουθες προσθέσεις **μη προσημασμένων δεκαεξαδικών** αριθμών. Αναφέρετε αν το άθροισμα θα προκαλέσει υπερχείλιση ή όχι στην περίπτωση που το αποτέλεσμα έχει 8 bit (δύο δεκαεξαδικά ψηφία).

  (γ) $AB_{16} + 3E_{16}$ (δ) $8F_{16} + AD_{16}$

Κρατούμενο 11111

(γ) 10101011

\+ 00111110

\-------------

11101001

(171+62=233)

ΔΕΝ υπάρχει υπερχείλιση

Κρατούμενο 1111

(δ) 10001111

\+ 10101101

\-------------

100111100(=60)

Carry

(143+173=316)

Υπάρχει υπερχείλιση

Επιλεγμένες ασκήσεις

- Άσκηση 1.61 – Λύση
  Μετατρέψτε τους παρακάτω δεκαδικούς αριθμούς σε δυαδικούς αριθμούς **συμπληρώματος ως προς δύο** των 6 bit και αφαιρέστε τους. Αναφέρετε αν η διαφορά θα προκαλέσει υπερχείλιση ή όχι στην περίπτωση που το αποτέλεσμα έχει 6 bit.

  (γ) $-28_{10} - 3_{10}$

Επιλεγμένες ασκήσεις

- Άσκηση 1.61 – Λύση
  Μετατρέψτε τους παρακάτω δεκαδικούς αριθμούς σε δυαδικούς αριθμούς **συμπληρώματος ως προς δύο** των 6 bit και αφαιρέστε τους. Αναφέρετε αν η διαφορά θα προκαλέσει υπερχείλιση ή όχι στην περίπτωση που το αποτέλεσμα έχει 6 bit.

(γ) $-28_{10} - 3_{10}$

$28_{10}=011100_2$
$-28_{10}=100011_2 + 1$
$=100100_2$

$3_{10}=000011_2$
$-3_{10}=111100_2 + 1$
$=111101_2$

$-28_{10} - 3_{10}=(-28_{10})+(-3_{10})$

```
  111
  100100₂
 +111101₂
 ---------------------
 1100001
```

Overflow?
ΟΧΙ

ΔΕΝ υπάρχει υπερχείλιση

Επιλεγμένες ασκήσεις

- Άσκηση 1.61 – Λύση
  Μετατρέψτε τους παρακάτω δεκαδικούς αριθμούς σε δυαδικούς αριθμούς **συμπληρώματος ως προς δύο** των 6 bit και αφαιρέστε τους. Αναφέρετε αν η διαφορά θα προκαλέσει υπερχείλιση ή όχι στην περίπτωση που το αποτέλεσμα έχει 6 bit.

  (δ) $\mathbf{-16_{10} - 21_{10}}$

Επιλεγμένες ασκήσεις

- Άσκηση 1.61 – Λύση
  Μετατρέψτε τους παρακάτω δεκαδικούς αριθμούς σε δυαδικούς αριθμούς συμπληρώματος ως προς δύο των 6 bit και αφαιρέστε τους. Αναφέρετε αν η διαφορά θα προκαλέσει υπερχείλιση ή όχι στην περίπτωση που το αποτέλεσμα έχει 6 bit.

(δ) $-16_{10} - 21_{10}$

$16_{10}=010000_2$
$-16_{10}=101111_2 + 1$
110000_2

$21_{10}=010101_2$
$-21_{10}=101010_2 + 1$
$=101011_2$

$-16_{10} – 21_{10}=(-16_{10})+(– 21_{10})$

110000_2
$+101011_2$

1011011

Overflow?
ΝΑΙ

Υπάρχει υπερχείλιση

Πράξεις αριθμών συμπληρώματος ως προς δύο

- Υπερχείλιση μπορεί να έχουμε ΜΟΝΟ όταν προσθέτουμε ομόσημους αριθμούς.
  – *Αν το αποτέλεσμα έχει διαφορετικό πρόσημο από τους αριθμούς που προσθέτουμε τότε έχουμε υπερχείλιση*

- Παράδειγμα, υπολογίστε την ακόλουθη πράξη (στα 3 bit)
  – $-4_{10} + -4_{10}$

Πράξεις αριθμών συμπληρώματος ως προς δύο

- Υπερχείλιση μπορεί να έχουμε ΜΟΝΟ όταν προσθέτουμε ομόσημους αριθμούς.
  - *Αν το αποτέλεσμα έχει διαφορετικό πρόσημο από τους αριθμούς που προσθέτουμε τότε έχουμε υπερχείλιση*

- Παράδειγμα, υπολογίστε την ακόλουθη πράξη (στα 3 bit)
  - $-4_{10}+-4_{10}$
    - Το 4_{10} =100 => - 4_{10} =011+1=100 (Μπορούμε να το υπολογίσουμε και απευθείας)

$$
\begin{array}{r}
100_2 \\
100_2 \\
\hline
\cancel{1}000_2
\end{array}
$$

Αντί για -8_{10} έχουμε 0_{10}
Υπερχείλιση (Overflow)

Δυαδικές κωδικοποιήσεις με 4 bit

- Binary Coded Decimal (BCD) κωδικοποίηση

Δεκαδικός	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Επιλεγμένες ασκήσεις

- **Άσκηση 1.64**
  Σε ένα *δυαδικά κωδικοποιημένο δεκαδικό* (binary coded decimal, BCD) σύστημα, χρησιμοποιούνται 4 bit για την αναπαράσταση ενός δεκαδικού ψηφίου από το 0 έως το 9.

  Για παράδειγμα, το 37_{10} γράφεται ως 00110111_{BCD}.

  (α) Γράψτε τον αριθμό 289_{10} στο σύστημα BCD.

  (β) Μετατρέψτε τον αριθμό 100101010001_{BCD} σε δεκαδικό.

  (γ) Μετατρέψτε τον αριθμό 01101001_{BCD} σε δυαδικό.

  (δ) Εξηγήστε γιατί το σύστημα BCD ενδέχεται να είναι ένας χρήσιμος τρόπος αναπαράστασης αριθμών.

  – *Εύκολη μετατροπή από το BCD στο δεκαδικό σύστημα*

  – *Ακριβής αναπαράσταση του διψήφιου δεκαδικού κλασματικού μέρους με ένα byte στις χρηματοοικονομικές εφαρμογές*

Επιλεγμένες ασκήσεις

- **Άσκηση 1.64 - Λύση**
  Σε ένα *δυαδικά κωδικοποιημένο δεκαδικό* (binary coded decimal, BCD) σύστημα, χρησιμοποιούνται 4 bit για την αναπαράσταση ενός δεκαδικού ψηφίου από το 0 έως το 9.

  Για παράδειγμα, το 37_{10} γράφεται ως 00110111_{BCD}.

  (α) Γράψτε τον αριθμό 289_{10} στο σύστημα BCD.

  0010 1000 1001

  2 8 9

Επιλεγμένες ασκήσεις

- Άσκηση 1.64 - Λύση
  Σε ένα *δυαδικά κωδικοποιημένο δεκαδικό* (binary coded decimal, BCD) σύστημα, χρησιμοποιούνται 4 bit για την αναπαράσταση ενός δεκαδικού ψηφίου από το 0 έως το 9.

Για παράδειγμα, το 37_{10} γράφεται ως 00110111_{BCD}.

(β) Μετατρέψτε τον αριθμό 100101010001_{BCD} σε δεκαδικό.

$1001=9_{10}$, $0101=5_{10}$, $0001=1_{10}$ $=>951_{10}$

Επιλεγμένες ασκήσεις

- **Άσκηση 1.64 - Λύση**
  Σε ένα *δυαδικά κωδικοποιημένο δεκαδικό* (binary coded decimal, BCD) σύστημα, χρησιμοποιούνται 4 bit για την αναπαράσταση ενός δεκαδικού ψηφίου από το 0 έως το 9.

Για παράδειγμα, το 37_{10} γράφεται ως 00110111_{BCD}.

(γ) Μετατρέψτε τον αριθμό 01101001_{BCD} σε δυαδικό.

Αρχικά μετατρέπουμε το BCD σε δεκαδικό

$0110=6_{10}$, $1001=9_{10}$, οπότε έχουμε τον 69_{10}

Κατόπιν μετατρέπουμε το δεκαδικό σε δυαδικό με όποιο τρόπο θέλουμε

$69_{10} = 64+4+1=1*2^6+1*2^2+1*2^0 = 1000101_2$

Επιλεγμένες ασκήσεις

- **Άσκηση 1.65**
  Σε ένα *δυαδικά κωδικοποιημένο δεκαδικό* (binary coded decimal, BCD) σύστημα, χρησιμοποιούνται 4 bit για την αναπαράσταση ενός δεκαδικού ψηφίου από το 0 έως το 9.

  Για παράδειγμα, το 37_{10} γράφεται ως 00110111_{BCD}.

  Εξηγήστε τα μειονεκτήματα του συστήματος BCD
  σε σύγκριση με τις δυαδικές αναπαραστάσεις αριθμών

Επιλεγμένες ασκήσεις

- **Άσκηση 1.65**
  Σε ένα *δυαδικά κωδικοποιημένο δεκαδικό* (binary coded decimal, BCD) σύστημα, χρησιμοποιούνται 4 bit για την αναπαράσταση ενός δεκαδικού ψηφίου από το 0 έως το 9.

  Για παράδειγμα, το 37_{10} γράφεται ως 00110111_{BCD}.

  Εξηγήστε τα μειονεκτήματα του συστήματος BCD σε σύγκριση με τις δυαδικές αναπαραστάσεις αριθμών

  – *Δεν λειτουργεί σωστά η πρόσθεση*

  – *Σε 1 byte στο BCD σύστημα αποθηκεύονται 100 αριθμοί (0-99), ενώ στο δυαδικό σύστημα 256 αριθμοί*

Επιλεγμένες ασκήσεις

- **Άσκηση 1.66**

  Ένας ιπτάμενος δίσκος πέφτει σε ένα χωράφι με καλαμπόκια στη Νεμπράσκα. Το FBI, που καλείται να ερευνήσει τα συντρίμμια, βρίσκει ένα τεχνικό εγχειρίδιο το οποίο περιέχει μια εξίσωση στο Αρειανό αριθμητικό σύστημα:

  $$325 + 42 = 411$$

  Αν η εξίσωση είναι σωστή, πόσα δάχτυλα σε κάθε ένα από τα δύο χέρια θα περιμένατε να έχουν οι Αρειανοί;

*Σκεφτείτε πως χρησιμοποιούμε εμείς τα δάκτυλα για πράξεις και τι σημαίνει αυτό για το αριθμητικό σύστημα που χρησιμοποιούμε.

Επιλεγμένες ασκήσεις

- **Άσκηση 1.66**

  Ένας ιπτάμενος δίσκος πέφτει σε ένα χωράφι με καλαμπόκια στη Νεμπράσκα. Το FBI, που καλείται να ερευνήσει τα συντρίμμια, βρίσκει ένα τεχνικό εγχειρίδιο το οποίο περιέχει μια εξίσωση στο Αρειανό αριθμητικό σύστημα:

  $325 + 42 = 411$

  Αν η εξίσωση είναι σωστή, πόσα δάχτυλα σε κάθε ένα από τα δύο χέρια θα περιμένατε να έχουν οι Αρειανοί;

*Σκεφτείτε πως χρησιμοποιούμε εμείς τα δάκτυλα για πράξεις και τι σημαίνει αυτό για το αριθμητικό σύστημα που χρησιμοποιούμε.

6αδικό σύστημα :3 δάκτυλα σε κάθε χέρι, ψηφία 0-5

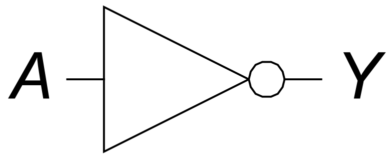
Λογικές Πύλες

- Οι **λογικές πύλες** είναι απλά ψηφιακά κυκλώματα που δέχονται μία η παραπάνω δυαδικές εισόδους και παράγουν μία δυαδική έξοδο
- Η σχέση μεταξύ εισόδων και εξόδου μπορεί να περιγραφτεί από έναν **πίνακα αλήθειας** ή μία **εξίσωση Boole**
- Ο πίνακας αλήθειας έχει **δύο μέρη**, αριστερά οι είσοδοι και δεξιά οι αντίστοιχοι έξοδοι
- Ο πίνακας αληθείας έχει **2^N γραμμές** ώστε να περιέχει **όλους τους δυνατούς συνδυασμούς των τιμών των N εισόδων**
- Η εξίσωση Boole είναι μια **μαθηματική παράσταση με δυαδικές μεταβλητές και λογικούς τελεστές** που προκύπτει απο τον πίνακα αλήθειας

Πύλη NOT

- Η **πύλη NOT** παίρνει μία τιμή και παράγει την αντίστροφη τιμή στην έξοδο
  - *Για είσοδο ‘1’ (TRUE) παράγει έξοδο ‘0’ (FALSE)*
  - *Για είσοδο ‘0’ (FALSE) παράγει έξοδο ‘1’ (TRUE)*
- Αν η είσοδος είναι A και η έξοδος Y τότε το σχηματικό της, η εξίσωση Boole και ο πίνακας αλήθειας φαίνονται δεξιά
  - *Ο κύκλος στο τέλος της πύλης ονομάζεται **φυσαλίδα** και δηλώνει αντιστροφή της εξόδου*
  - *Η γραμμή πάνω από το A στην εξίσωση Boole προφέρεται **NOT***
- Η πύλη NOT είναι επίσης γνωστή ως αντιστροφέας (inverter)

NOT



$$Y = \overline{A}$$

A	Y
0	1
1	0

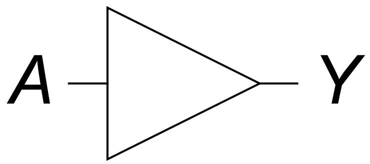
Συμβολισμοί Πύλης NOT

- `Y = ~A` (SystemVerilog)
- `Y = !A`
- `Y = `**`not`**` A` (VHDL)
- $Y = A'$
- $Y = \neg A$
- $Y = \overline{A}$ (βιβλίο)

Απομονωτής (Buffer)

- Ο απομονωτής ή buffer αντιγράφει την είσοδο στην έξοδο
  - *Για είσοδο ‘1’ (TRUE) παράγει έξοδο ‘1’ (TRUE)*
  - *Για είσοδο ‘0’ (FALSE) παράγει έξοδο ‘0’ (FALSE)*
- Αν η είσοδος είναι Α και η έξοδος Υ τότε το σχηματικό της, η εξίσωση Boole και ο πίνακας αλήθειας φαίνονται δεξιά
  - *Στο σχηματικό διαφέρει από την πύλη NOT ως προς τη φυσαλίδα*
- Ο απομονωτής δεν αλλάζει τη λογική τιμή, αλλά ενισχύει το ρεύμα του σήματος, ώστε να διοχετεύει μεγάλες ποσότητες ρεύματος σε έναν κινητήρα ή να διασυνδέει την έξοδό του με εισόδους πολλών πυλών

BUF



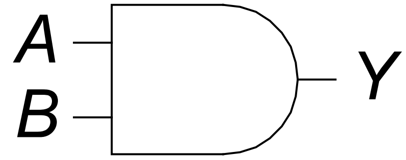
$$Y = A$$

A	Y
0	0
1	1

Πύλη AND (2 εισόδων)

- Η **πύλη AND** παράγει έξοδο *Y* με τιμή ‘1’ (TRUE), αν και μόνο αν, τόσο το *A* όσο και το *B* έχουν την τιμή ‘1’ (TRUE). Διαφορετικά, η έξοδος έχει την τιμή ‘0’ (FALSE)
- Η εξίσωση Boole διαβάζεται:
  - *το Y ισούται με το A **και** το B*

AND



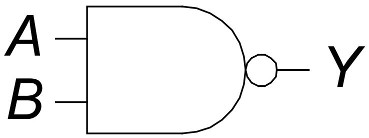
$$Y = AB$$

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

Πύλη NAND (2 εισόδων)

- Η **πύλη NAND** εκτελεί την πράξη **NOT AND**
  - *Στο σχηματικό διαφέρει από την πύλη AND ως προς τη φυσαλίδα που έχει στην έξοδο*
- Η πύλη NAND παράγει έξοδο πάντα με τιμή ‘1’ (TRUE), με εξαίρεση την περίπτωση που και οι δύο είσοδοι έχουν την τιμή TRUE

NAND



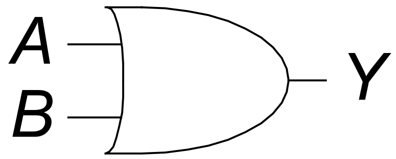
$$Y = \overline{AB}$$

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

Πύλη OR (2 εισόδων)

- Η **πύλη OR** παράγει έξοδο Y με τιμή ‘1’ (TRUE), αν είτε το A είτε το B (είτε και τα δύο) έχουν την τιμή ‘1’ (TRUE). Διαφορετικά, η έξοδος έχει την τιμή ‘0’ (FALSE)
- Η εξίσωση Boole διαβάζεται:
  - *το Y ισούται με το A **ή** το B*

OR



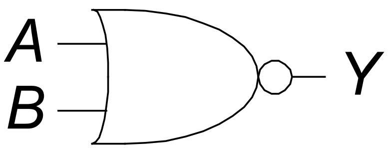
$$Y = A + B$$

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

Πύλη NOR (2 εισόδων)

- Η **πύλη NOR** εκτελεί την πράξη **NOT OR**
  - *Στο σχηματικό διαφέρει από την πύλη OR ως προς τη φυσαλίδα που έχει στην έξοδο*
- Η πύλη NOR παράγει έξοδο με τιμή ‘1’ (TRUE) αν ούτε το A ούτε το B δεν έχουν την τιμή TRUE, δηλαδή αν και τα δύο έχουν την τιμή ‘0’ (FALSE)
- Η NOR παράγει έξοδο ‘1’ (TRUE) όταν και οι 2 είσοδοι είναι ‘0’ και η AND παράγει έξοδο ‘1’ (TRUE) όταν και οι 2 είσοδοι είναι ‘1’

NOR



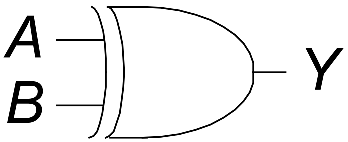
$$Y = \overline{A + B}$$

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

Πύλη XOR (2 εισόδων)

- Η **πύλη XOR (exclusive OR)** παράγει έξοδο με τιμή ‘1’ (TRUE) αν το A ή το B, αλλά όχι και τα δύο ταυτόχρονα, έχουν την τιμή ‘1’ (TRUE)
- Η πράξη XOR συμβολίζεται με το ⊕ δηλαδή το συν μέσα σε έναν κύκλο
- Επίσης, υλοποιεί την αλγεβρική πράξη $Y = (A+B)\bmod 2,\ A,B \in \{0,1\}$
- Μερικές φορές θα την διαβάσετε και ως EOR
- Έχει τιμή 1 όταν οι είσοδοι είναι διαφορετικές

XOR



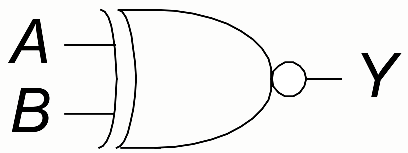
$$Y = A \oplus B$$

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

Πύλη XNOR (2 εισόδων)*

- Η **πύλη XNOR (exclusive NOR)** παράγει έξοδο με τιμή ‘1’ (TRUE) αν το A και το B ταυτόχρονα έχουν την τιμή ‘1’ (TRUE) ή την τιμή ‘0’ (FALSE)
- Η πύλη XNOR εκτελεί την **αντίστροφη πράξη** μίας πύλης XOR
  - *Στο σχηματικό διαφέρει από την πύλη XOR ως προς τη φυσαλίδα που έχει στην έξοδο*
- Μία πύλη XNOR με δύο εισόδους ονομάζεται ενίοτε πύλη ισότητας **(equality gate)** επειδή η έξοδός της έχει την τιμή ‘1’ (TRUE) όταν οι **είσοδοι είναι ίσες**

XNOR



$$Y = \overline{A \oplus B}$$

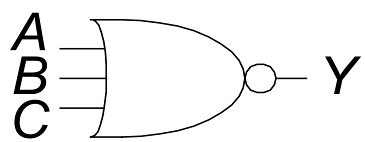
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

*Παράδειγμα 1.15

Παράδειγμα 1.16

- Πύλες NOR με περισσότερες από δύο εισόδους
- Μία πύλη NOR με N εισόδους παράγει έξοδο με τιμή ‘1’ (TRUE) μόνο αν όλες οι είσοδοι έχουν την τιμή ‘0’ (FALSE)

NOR3



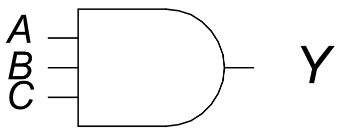
$$Y = \overline{A+B+C}$$

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

Παράδειγμα 1.17

- Πύλες AND με περισσότερες από δύο εισόδους
- Μία πύλη AND με N εισόδους παράγει έξοδο με τιμή ‘1’ (TRUE) μόνο αν όλες οι είσοδοι έχουν την τιμή ‘1’ (TRUE)

AND3



$Y = ABC$

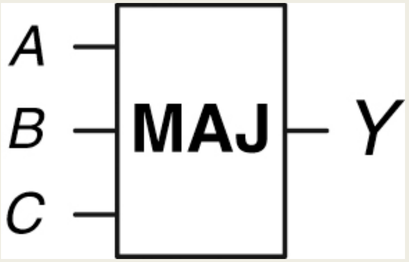
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Πύλες περιττής και άρτιας ισοτιμίας

- Μια πύλη XOR με N εισόδους ονομάζεται ενίοτε **πύλη περιττής ισοτιμίας (odd parity gate)** και παράγει έξοδο με τιμή ‘1’ (TRUE) αν ένα **περιττό πλήθος** εισόδων έχουν την τιμή ’1’ (TRUE)
- Μια πύλη XNOR με N εισόδους ονομάζεται ενίοτε **πύλη άρτιας ισοτιμίας (even parity gate)** και παράγει έξοδο με τιμή ‘1’ (TRUE) αν ένα **άρτιο πλήθος εισόδων** έχουν την τιμή ’1’ (TRUE)
- Υλοποιούνται με δένδρα από N-1 πύλες XOR δύο εισόδων
  - *Στην περίπτωση της πύλης XNOR, η πύλη δύο εισόδων στην έξοδο είναι πύλη XNOR αντί για XOR*
- Χρησιμοποιούνται στην κωδικοποίηση και την κρυπτογραφία

Επιλεγμένες ασκήσεις

- Άσκηση 1.73
  Μια πύλη πλειοψηφίας (majority gate) παράγει έξοδο με τιμή '1' (TRUE), αν και μόνο αν, περισσότερες από τις μισές είσοδοι έχουν την τιμή '1' (TRUE)
  - *Συμπληρώστε τον πίνακα αληθείας για την πύλη πλειοψηφίας με τρεις εισόδους*

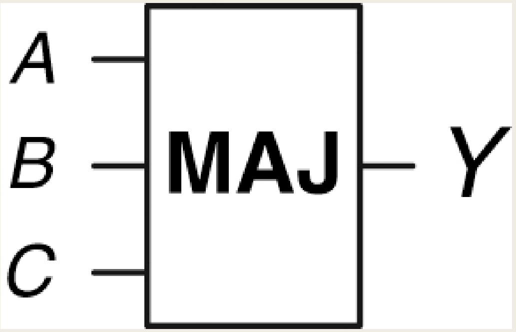


- ## Άσκηση 1.73

  Μια πύλη πλειοψηφίας (majority gate) παράγει έξοδο με τιμή '1' (TRUE), αν και μόνο αν, περισσότερες από τις μισές είσοδοι έχουν την τιμή '1' (TRUE)

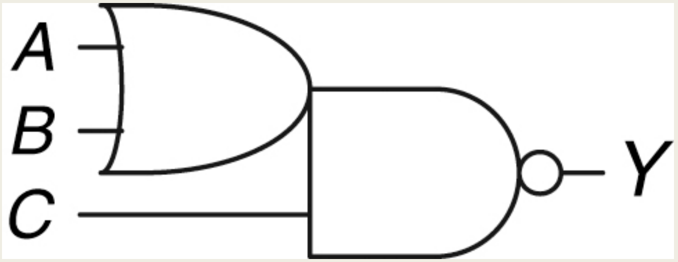
  – *Συμπληρώστε τον πίνακα αληθείας για την πύλη πλειοψηφίας με τρεις εισόδους*

A	*B*	*C*	*Y*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



Επιλεγμένες ασκήσεις

- Άσκηση 1.75
  Μια *πύλη* ***OR-AND-INVERT*** **(OAI)** παράγει έξοδο με τιμή ‘0’ (FALSE) αν το *C* έχει την τιμή ‘1’ (TRUE) και το *A* ή το *B* έχει την τιμή ’1’ (TRUE). Διαφορετικά παράγει έξοδο με τιμή ’1’ (TRUE).
  - *Συμπληρώστε τον πίνακα αληθείας για την πύλη OR-AND-INVERT (OAI)*

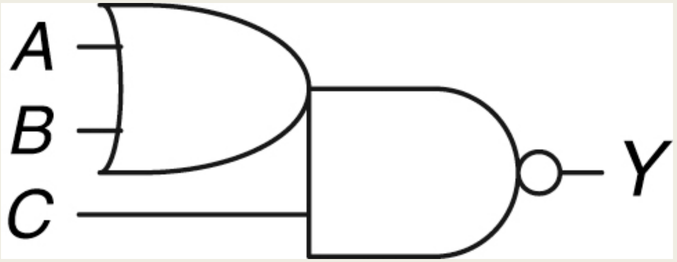


Άσκηση 1.75

Μια *πύλη* *OR-AND-INVERT* (OAI) παράγει έξοδο με τιμή ‘0’ (FALSE) αν το *C* έχει την τιμή ‘1’ (TRUE) και το *A* ή το *B* έχει την τιμή ’1’ (TRUE). Διαφορετικά παράγει έξοδο με τιμή ’1’ (TRUE).

- *Συμπληρώστε τον πίνακα αληθείας για την πύλη OR-AND-INVERT (OAI)*

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



Επιλεγμένες ασκήσεις

- Άσκηση 1.76
  Υπάρχουν 16 διαφορετικοί πίνακες αληθείας για εξισώσεις (συναρτήσεις) Boole με δύο μεταβλητές.
  - *Παραθέστε όλους τους πίνακες.*
  - *Δώστε ένα σύντομο περιγραφικό όνομα.*
- Άσκηση 1.77
  Πόσοι διαφορετικοί πίνακες αληθείας υπάρχουν για συναρτήσεις Boole με *N* μεταβλητές;

Επιλεγμένες ασκήσεις

- **Άσκηση 1.76**
  Υπάρχουν 16 διαφορετικοί πίνακες αληθείας για εξισώσεις (συναρτήσεις) Boole με δύο μεταβλητές.
  - *Παραθέστε όλους τους πίνακες.*
  - *Δώστε ένα σύντομο περιγραφικό όνομα.*
- **Άσκηση 1.77**
  Πόσοι διαφορετικοί πίνακες αληθείας υπάρχουν για συναρτήσεις Boole με *N* μεταβλητές;
  - 2^{2^N}

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

Zero

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

A NOR B

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

$\overline{A}B$

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

NOT A

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

$A\overline{B}$

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

NOT B

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

XOR

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

NAND

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

AND

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

XNOR

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

B

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

$\overline{A} + B$

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

A

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

$A + \overline{B}$

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

OR

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

One

Αριθμητικά Κυκλώματα

- Τα **αριθμητικά κυκλώματα** αποτελούν τα κεντρικά δομικά στοιχεία των υπολογιστών
- Οι υπολογιστές και τα στοιχεία ψηφιακής λογικής εκτελούν πολλές αριθμητικές λειτουργίες ή πράξεις, όπως:
  - *πρόσθεση*
  - *αφαίρεση*
  - *συγκρίσεις*
  - *ολισθήσεις*
  - *πολλαπλασιασμό*
  - *διαίρεση*

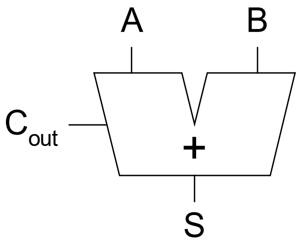
Στην ενότητα αυτή θα επικεντρωθούμε στα αριθμητικά κυκλώματα που εκτελούν τις πράξεις της **πρόσθεσης** και της **αφαίρεσης**

Αθροιστές του ενός bit

- Προσθέτουν τα ψηφία της ίδιας στήλης (ή αλλιώς ίδιου βάρους) μαζί με το bit κρατούμενου (εάν υπάρχει)
- Υπάρχουν δύο αθροιστές του ενός bit
  - *Ο **ημιαθροιστής (half-adder)**, όταν **δεν** υπάρχει κρατούμενο εισόδου*
  - *Ο **πλήρης αθροιστής (full-adder)**, όταν υπάρχει **κρατούμενο εισόδου C_{in} (carry in)***
- Παράγουν στην έξοδο
  - *Το **άθροισμα S (sum)***
  - *Το **κρατούμενο εξόδου C_{out} (carry out)***
- Σε έναν αθροιστή πολλών bit, το κρατούμενο εξόδου C_{out} μεταφέρεται ως κρατούμενο εισόδου C_{in}, στην επόμενη στήλη στα αριστερά (στο επόμενο βάρος), όταν υπάρχει
- **Δεν διαφοροποιείται η πράξη της πρόσθεσης στο δυαδικό σύστημα μεταξύ μη προσημασμένων αριθμών και προσημασμένων αριθμών σε αναπαράσταση συμπληρώματος ως προς δύο**
  - ***αλλάζει μόνο η ερμηνεία των αριθμών και η εμφάνιση της υπερχείλισης***

Αθροιστές του ενός bit

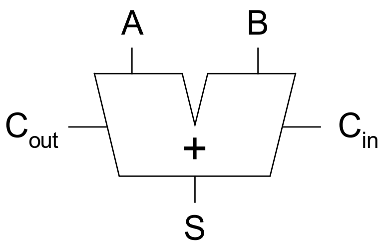
Half Adder



A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = A \oplus B$$
$$C_{out} = AB$$

Full Adder

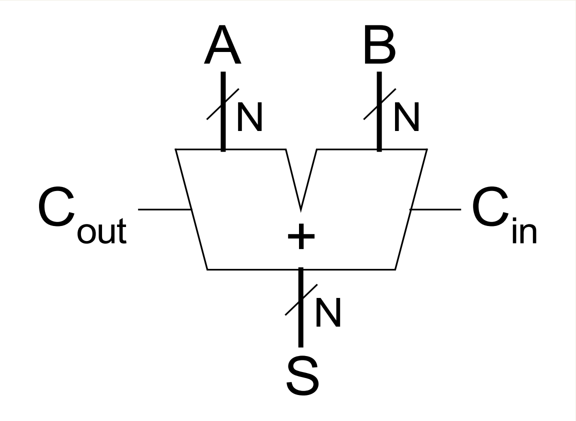


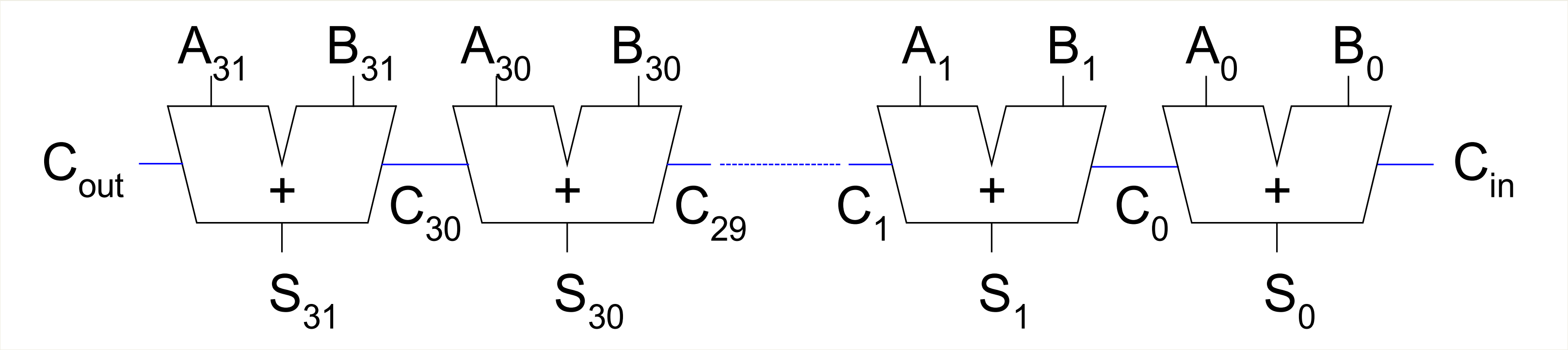
C_{in}	A	B	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S = A \oplus B \oplus C_{in}$$
$$C_{out} = AB + AC_{in} + BC_{in}$$

Αθροιστές των *N* bit

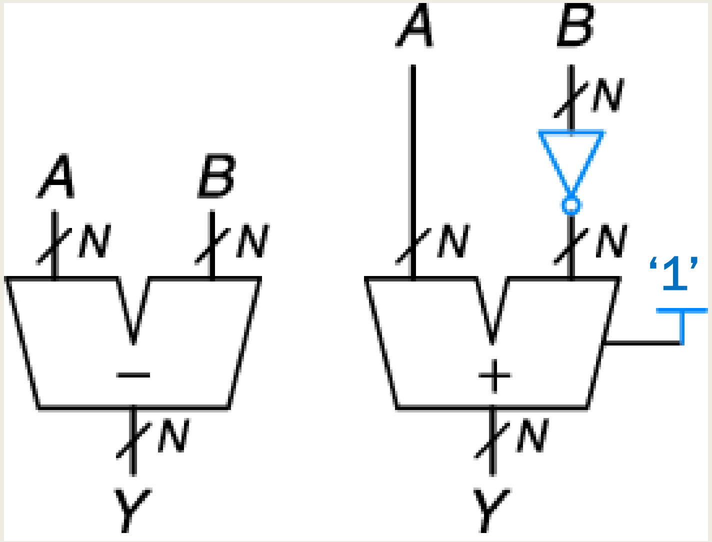
- Αθροιστής κυμάτωσης κρατούμενου (ripple-carry adder)
  - *N* ***πλήρεις αθροιστές*** *του ενός bit συνδεδεμένοι αλυσιδωτά*
  - *Το κρατούμενο* ***διαδίδεται*** *μέσω της αλυσίδας κρατουμένου*
  - *Η έξοδος C_{out} του ενός βάρους χρησιμεύει ως είσοδος C_{in} του επόμενου βάρους*
  - *Γενικά είναι αργός (χρησιμοποιείται μόνο στα FPGA)*



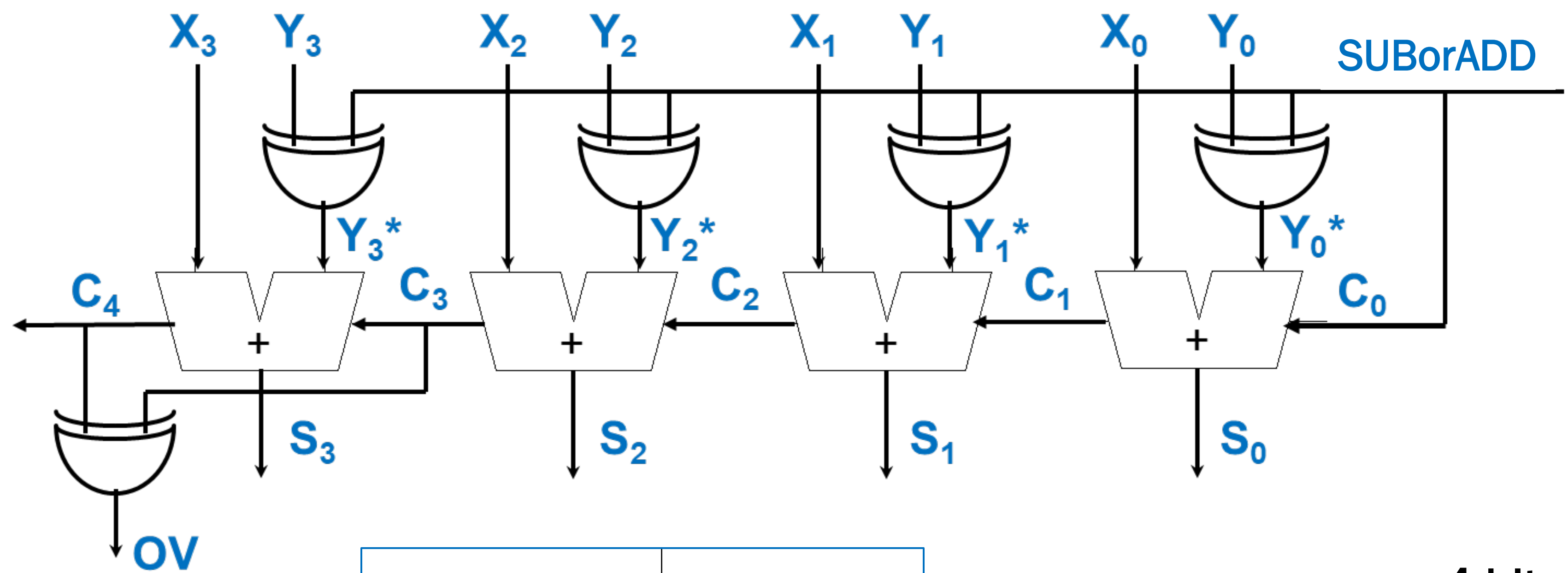


Αφαιρέτες των *N* bit

- **Αφαιρέτης κυμάτωσης κρατούμενου (ripple-carry subtracter)**
  - *N **πλήρεις αθροιστές** του ενός bit συνδεδεμένοι αλυσιδωτά*
  - *Το κρατούμενο **διαδίδεται** μέσω της αλυσίδας κρατουμένου*
  - *Η έξοδος C_{out} του ενός βάρους χρησιμεύει ως είσοδος C_{in} του επόμενου βάρους*
  - ***Αντιστρέφονται** τα ψηφία του αφαιρετέου*
  - *Η είσοδος C_{in} του μικρότερου βάρους **αρχικοποιείται στο 1***
  - *Γενικά είναι αργός (χρησιμοποιείται μόνο στα FPGA)*



Αθροιστής/αφαιρέτης με επιλογή και υπερχείλιση προσημασμένων



4 bit

C_3	X_3	Y_3*	C_4	S_3
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

OV

$X_3=Y_3*=1$ & $S_3=0$

$X_3=Y_3*=0$ & $S_3=1$

Πρόσθεση
SUBorADD = 0

$C_3\ C_2\ C_1\ C_0=0$

$$\begin{array}{r} X_3\ X_2\ X_1\ X_0 \\ +\ Y_3\ Y_2\ Y_1\ Y_0 \\ \hline \cancel{C_4}\ S_3\ S_2\ S_1\ S_0 \end{array}$$

Αφαίρεση
SUBorADD = 1

$C_3\ C_2\ C_1\ C_0=1$

$$\begin{array}{r} X_3\ X_2\ X_1\ X_0 \\ +\ \overline{Y_3}\ \overline{Y_2}\ \overline{Y_1}\ \overline{Y_0} \\ \hline \cancel{C_4}\ S_3\ S_2\ S_1\ S_0 \end{array}$$

Τάση Τροφοδοσίας και Γείωση

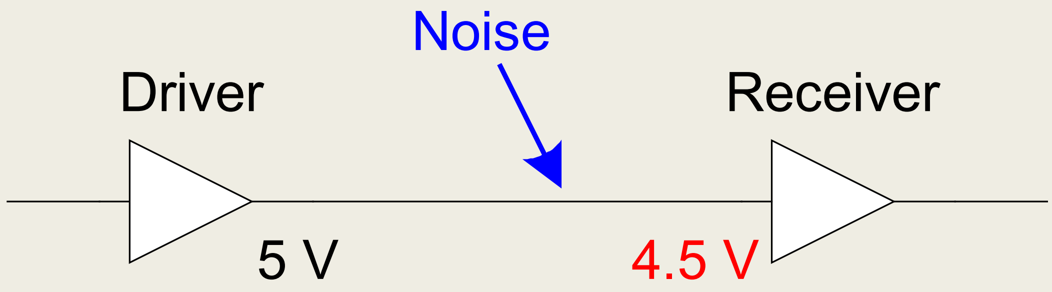
- Η χαμηλότερη τάση σε ένα σύστημα ονομάζεται **γείωση** (ground) ή GND
  - *Θεωρούμε ότι η χαμηλότερη τάση είναι 0V*
- Η υψήλοτερη τάση σε ένα σύστημα, δηλαδή η τάση τροφοδοσίας, προέρχεται από την τροφοδοσία και συμβολίζεται με V_{DD}
  - *Στις δεκαετίες του 1970 και 1980 είχε την τιμή 5V*
  - *Κάθως η τεχνολογία των τσιπ εξελίσσεται η υψηλότερη τάση μειώνεται συνεχώς (από τα 5V στο 1V)*

Λογικά επίπεδα

- Οι μεταβλητές (όπως η τάση στην έξοδο ενός ηλεκτρονικού κυκλώματος) αναπαρίστανται με συνεχείς φυσικές τιμές
- Ενα ψηφιακό σύστημα όμως χρησιμοποιεί μεταβλητές που παίρνουν **διακριτές τιμές**
- Επομένως ένας κατασκευαστής ψηφιακών συστημάτων πρέπει να βρεί μία αντιστοίχιση **που να συνδέει τις συνεχείς τιμές με τις διακριτές τιμές**
- Για παράδειγμα: Η αντιστοίχιση ενός δυαδικού σήματος A με την τάση σε μία έξοδο μίας πύλης (κατά τη θετική λογική)
  - *Τα 0V αντιστοιχούν στο A=0 (LOW)*
  - *Τα 5V αντιστοιχούν στο A=1 (HIGH)*
- Λόγω ανοχής στον θόρυβο, θα πρέπει και κάποιες κοντινές τάσεις να αντιστοιχούν στο A=0 ή A=1
  - *π.χ. τα 4,9V στα 5V αντιστοιχούν στο A=1*
- Τί συμβαίνει όμως με ποιο ενδιάμεσες τιμές τάσεων (π.χ. 3,2V);

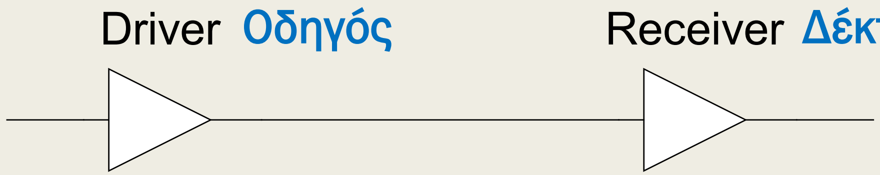
Τι είναι ο θόρυβος (noise);

- Η ύπαρξη θορύβου είναι αρκετά συχνό φαινόμενο ανάμεσα στον οδηγό και στον δέκτη
- Οτιδήποτε μπορεί να επηρεάσει ή να υποβαθμίσει ένα σήμα
  - *Π.χ., αντίσταση, θόρυβος τάσης τροφοδοσίας, ηλεκτρομαγνητική αλληλεπίδραση μεταξύ γειτονικών συρμάτων, κλπ.*
- Για παράδειγμα: Η πύλη «οδηγός» έχει έξοδο 5V. Λόγω της αντίστασης απ' το μακρύ σύρμα που συνδέει την έξοδο του οδηγού με την είσοδο του δέκτη τελικά ο δέκτης θα λάβει 4.5V

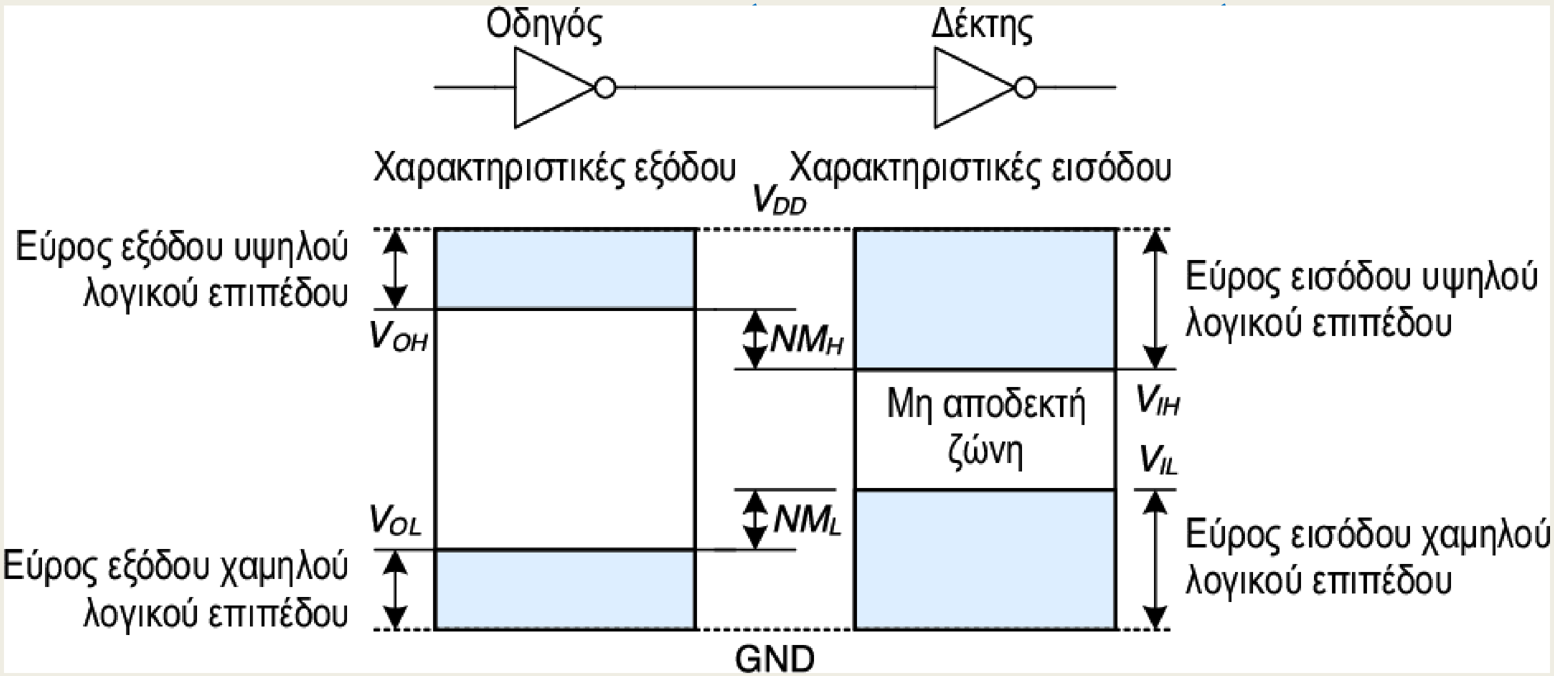


Λογικά Επίπεδα

- Η αντιστοίχηση μιας συνεχούς μεταβλητής σε μία διακριτή δυαδική μεταβλητή πραγματοποιείται με τον ορισμό των λογικών επιπέδων
- Ορίζονται διαφορετικά λογικά επίπεδα για εισόδους και εξόδους, ώστε να παρέχεται προστασία από τον θόρυβο
  - *V_{OH}: ελάχιστη τάση στην έξοδο ενός οδηγού που αντιστοιχεί στην τιμή ‘1’ (HIGH)*
  - *V_{OL}: μέγιστη τάση στην έξοδο ενός οδηγού που αντιστοιχεί στην τιμή ‘0’ (LOW)*
  - *V_{IH}: ελάχιστη τάση στην είσοδο ενός δέκτη που αντιστοιχεί στην τιμή ‘1’ (HIGH)*
  - *V_{IL}: μέγιστη τάση στην είσοδο ενός δέκτη που αντιστοιχεί στην τιμή ‘0’ (LOW)*



Περιθώρια θορύβου (noise margins)



Υψηλό Περιθώριο Θορύβου: $NM_H = V_{OH} - V_{IH}$

Χαμηλό Περιθώριο Θορύβου : $NM_L = V_{IL} - V_{OL}$

Περιθώρια θορύβου

- Για να ερμηνευθεί σωστά η έξοδος του οδηγού στην είσοδο του δέκτη, πρέπει να επιλέξουμε τέτοιες τιμές λογικών επιπέδων ώστε να ισχύουν οι ανισότητες:
  - $V_{OL} < V_{IL}$ *και* $V_{OH} > V_{IH}$
- Άρα, ακόμα και αν η τάση στην έξοδο του οδηγού έχει μεταβληθεί από κάποια ποσότητα θορύβου, η είσοδος του δέκτη θα εξακολουθεί να είναι σε θέση να ανιχνεύει το σωστό λογικό επίπεδο.
- Το **περιθώριο θορύβου (noise margin)** είναι εκείνη η ποσότητα του θορύβου που θα μπορούσε να προστεθεί στην έξοδο του οδηγού στη χειρότερη περίπτωση, η οποία επιτρέπει ωστόσο στο σήμα να ερμηνευθεί σωστά ως μια έγκυρη είσοδος του δέκτη
- Αν για κάποιον λόγο, π.χ. την παρουσία θορύβου ή την ύπαρξη κάποιου ελαττωματικού εξαρτήματος, η είσοδος του δέκτη έχει τιμή που εμπίπτει στη **μη αποδεκτή ζώνη (forbidden zone)** μεταξύ V_{IL} και V_{IH}, τότε η συμπεριφορά της πύλης είναι απρόβλεπτη.

Παράδειγμα 1.18

- Θεωρήστε το κύκλωμα της Εικόνας.
  - *Το V_{O1} είναι η τάση εξόδου του οδηγού, ενώ το V_{I2} είναι η τάση εισόδου του δέκτη*
  - *Και οι δύο buffer έχουν V_{DD} = 5V, V_{IL} = 1,35V, V_{IH} = 3,15V, V_{OL} = 0,33V και V_{OH} = 3,84V.*
  - *Ποια είναι τα περιθώρια θορύβου για το χαμηλό και το υψηλό λογικό επίπεδο του buffer;*
  - *Μπορεί το κύκλωμα να ανέχεται 1 V θορύβου μεταξύ των V_{O1} και V_{I2};*

Παράδειγμα 1.18

- Θεωρήστε το κύκλωμα της Εικόνας.
  - *Το V_{O1} είναι η τάση εξόδου του οδηγού, ενώ το V_{I2} είναι η τάση εισόδου του δέκτη*
  - *Και οι δύο buffer έχουν V_{DD} = 5V, V_{IL} = 1,35V, V_{IH} = 3,15V, V_{OL} = 0,33V και V_{OH} = 3,84V.*
  - *Ποια είναι τα περιθώρια θορύβου για το χαμηλό και το υψηλό λογικό επίπεδο του buffer;*
  - *Μπορεί το κύκλωμα να ανέχεται 1 V θορύβου μεταξύ των V_{O1} και V_{I2};*
- Τα περιθώρια θορύβου του αντιστροφέα είναι τα εξής:
  - ***$NM_L = V_{IL} - V_{OL}$ = 1,35V − 0,33 V) = 1,02 V***
  - ***$NM_H = V_{OH} - V_{IH}$ = 3,84V – 3,15 V) = 0,69 V***
  - *Το κύκλωμα μπορεί να ανεχθεί 1V θορύβου όταν η έξοδος είναι LOW (NM_L = 1,02 V), αλλά **όχι** όταν η έξοδος είναι HIGH (NM_H = 0,69 V).*
  - *Εάν $V_{O1} = V_{OH}$ = 3,84V, τότε εξαιτίας του θορύβου V_{I2} = 3,84V – 1V = 2,84V < V_{IH} = 3,15V, άρα ο δέκτης ενδέχεται να μην ανιχνεύσει μια σωστή είσοδο HIGH, αλλά μία λανθασμένη είσοδο LOW*

Λογικές οικογένειες

- Από το 1970 μέχρι και το 1990 υπήρχαν τέσσερις λογικές οικογένειες που κυριάρχησαν στην αγορά
  - *Λογική τρανζίστορ-τρανζίστορ(Transistor-Transistor Logic, TTL)*
  - *Λογική συμπληρωματικού ημιαγωγού οξειδίου μετάλλου (Complementary Metal-Oxide-Semiconductor Logic, CMOS)*
  - *Λογική TTL χαμηλής τάσης (Low Voltage TTL Logic, LVTTL)*
  - *Λογική CMOS χαμηλής τάσης(Low Voltage CMOS Logic, LVCMOS).*

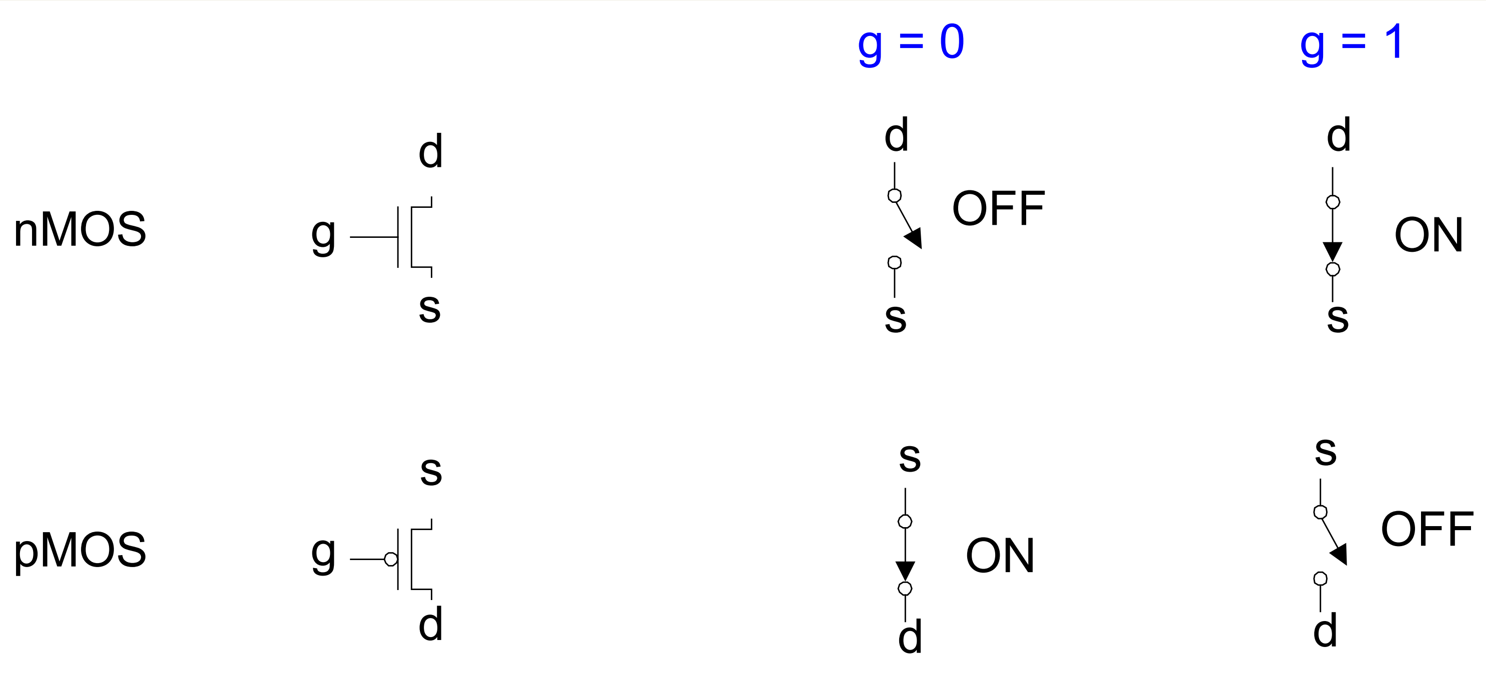
Λογική οικογένεια	V_{DD}	V_{IL}	V_{IH}	V_{OL}	V_{OH}
TTL	5(4.75-5.25)	0.8	2.0	0.4	2.4
CMOS	5(4.5-6)	1.35	3.15	0.33	3.84
LVTTL	3.3(3-3.6)	0.8	2.0	0.4	2.4
LVCMOS	3.3(3-3.6)	0.9	1.8	0.36	2.7

Τρανζίστορ

- Οι πρώτοι ηλεκτρονικοί υπολογιστές χρησιμοποιούσαν ηλεκτρονόμους ή λυχνίες κενού.
- Οι σύγχρονοι υπολογιστές χρησιμοποιούν ηλεκτρικούς διακόπτες οι οποίοι είναι επίσης γνωστοί σαν τρανζίστορ
  - *Τρανζίστορ διπολικής επαφής*
  - *Τρανζίστορ φαινομένου πεδίου ημιαγωγών μετάλλου οξειδίου (MOSFET* - metal-oxide-semiconductor field-effect transistor*)*
- Τα τρανζίστορ MOSFET χρησιμοποιούνται σαν δομικά στοιχεία σχεδόν σε όλα τα ψηφιακά συστήματα.

Τρανζίστορ MOSFET ως διακόπτης

- Οι λογικές πύλες κατασκευάζονται από τρανζίστορ φαινομένου πεδίου ημιαγωγών μετάλλου οξειδίου (MOSFET) που λειτουργούν ως διακόπτες
- Υπάρχουν δύο είδη τρανζίστορ MOSFET: nMOS και pMOS
  - *Η τάση στην **πύλη (gate)** ρυθμίζει τη ροή του ρεύματος μεταξύ της **πηγής (source)** και της **υποδοχής (drain)***

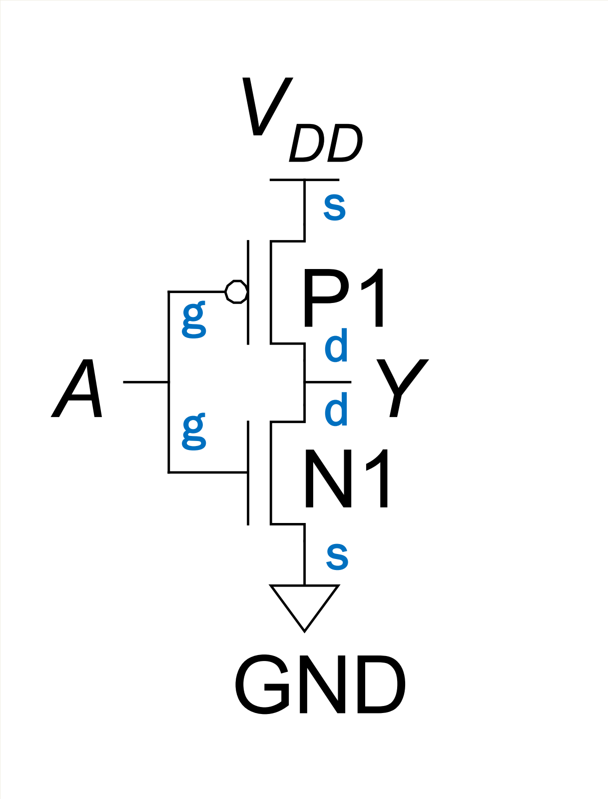


Λειτουργία τρανζίστορ MOSFET

- Το τρανζίστορ nMOS
  - *δεν άγει (είναι OFF) όταν η πύλη (gate) είναι ‘0’*
    (έχει σαν τάση αντίστοιχη της λογικής τιμής ‘0’)
  - *άγει (είναι ON) όταν η πύλη (gate) είναι ‘1’*
    (έχει σαν τάση αντίστοιχη της λογικής τιμής ‘1’)
- Το τρανζίστορ pMOS
  - *δεν άγει (είναι OFF) όταν η πύλη είναι ‘1’*
  - *άγει (είναι ON) όταν η πύλη είναι ‘0’*
- Όταν το τρανζίστορ δεν άγει (είναι OFF) έχει πολύ μεγάλη αντίσταση αλλά όχι άπειρη
  - *Υπάρχει ένα μικρό ρεύμα διαρροής I_{DD}*
- Όταν το τρανζίστορ άγει (είναι ON) μεταφέρει «καλά» από την πηγή (source) στην υποδοχή (drain)
  - *εάν είναι nMOS το ‘0’, συνεπώς συνδέεται η πηγή (s) στο GND*
  - *εάν είναι pMOS το ‘1’, συνεπώς συνδέεται η πηγή (s) στο V_{DD}*

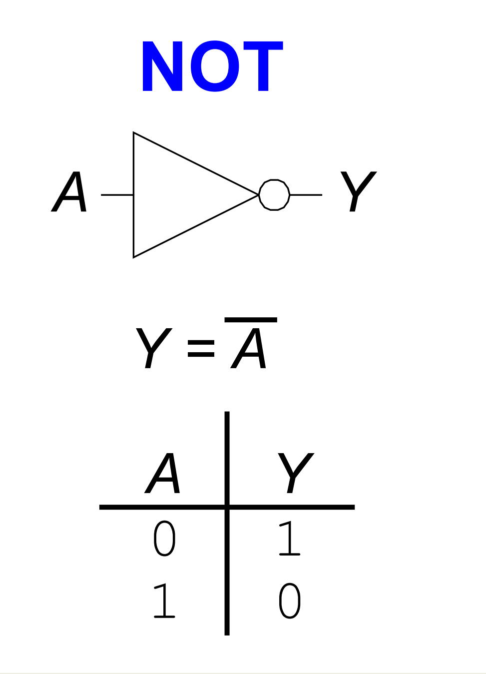
Πύλη NOT σε τεχνολογία CMOS

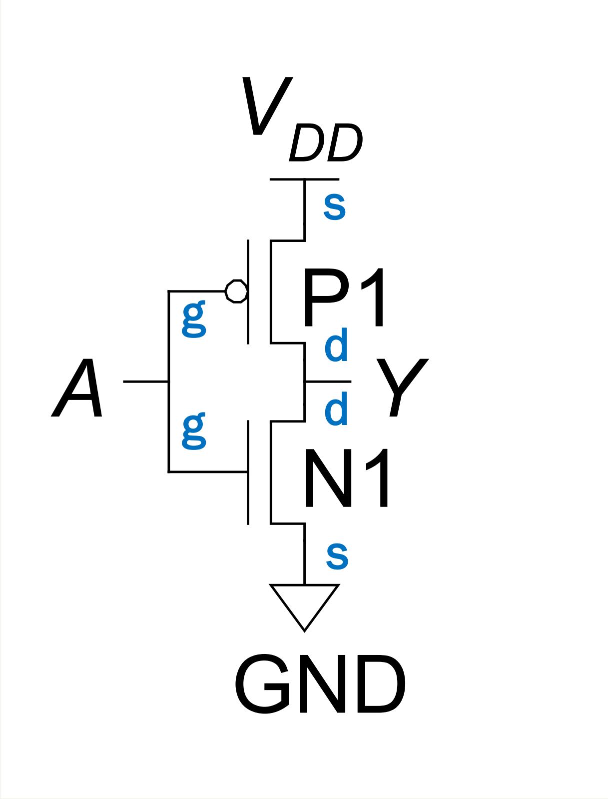
- Στην τεχνολογία **CMOS (complementary MOS)** χρησιμοποιούνται **μαζί** και τα δύο είδη τρανζίστορ MOSFET, δηλαδή **nMOS και pMOS**
- Πώς δημιουργείται μία **πύλη NOT** σε τεχνολογία CMOS;
- Σχηματικό διάγραμμα:
  - *Στο τρανζίστορ nMOS (N1) συνδέεται η πηγή (s) στο GND (τρίγωνο)*
  - *Στο τρανζίστορ pMOS (P1) συνδέεται η πηγή (s) στο V_{DD} (γραμμή)*
  - *Οι πύλες (g) των N1 και P1 συνδέονται στην είσοδο A*
  - *Οι υποδοχές (d) των N1 και P1 συνδέονται στην έξοδο Y*



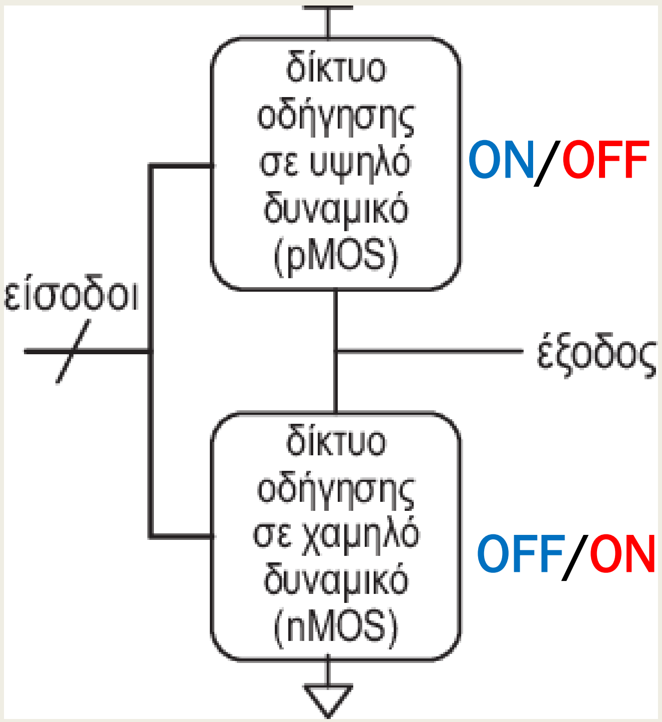
CMOS Πύλη NOT

A	P1	N1	Y
0	ON	OFF	1
1	OFF	ON	0





Γενική μορφή λογικής πύλης CMOS



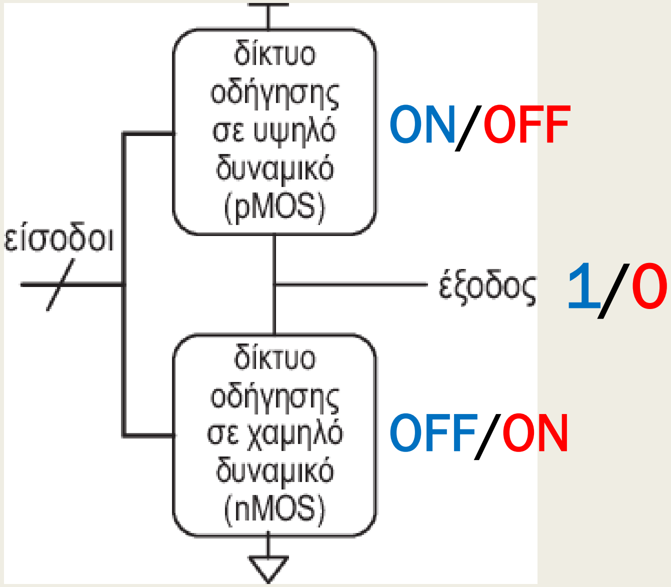
Δίκτυο οδήγησης της εξόδου σε υψηλό δυναμικό (ώστε όταν είναι ON να είναι η έξοδος ‘1’) με τρανζίστορ pMOS. Το δίκτυο pMOS τοποθετείται ανάμεσα στην τάση V_{DD} και την έξοδο

Δίκτυο οδήγησης της εξόδου σε χαμηλό δυναμικό (ώστε όταν είναι ON να είναι η έξοδος ‘0’) με τρανζίστορ nMOS. Το δίκτυο nMOS τοποθετείται ανάμεσα στην έξοδο και την τάση 0V (GND)

- Τα δίκτυα αποτελούνται από τρανζίστορ συνδεδεμένα παράλληλα ή σε σειρά.
  - *Όταν τα τρανζίστορ είναι συνδεδεμένα παράλληλα, το δίκτυο είναι ON αν οποιοδήποτε από τα τρανζίστορ είναι ON.*
  - *Όταν τα τρανζίστορ είναι συνδεδεμένα σε σειρά, το δίκτυο είναι ON μόνο αν όλα τα τρανζίστορ είναι ON.*

Γενική μορφή λογικής πύλης CMOS

- Σε μια CMOS πύλη που λειτουργεί σωστά σε κάθε δεδομένη στιγμή, το ένα από τα δύο δίκτυα πρέπει να είναι ON και το άλλο OFF, έτσι ώστε η έξοδος να μην βραχυκυ-κλώνει ή να παραμένει μετέωρη.

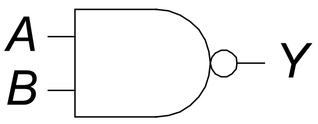


- Κανόνας των ***συμπληρωμάτων αγωγιμότητας***
  - *Όταν τα τρανζίστορ nMOS είναι συνδεδεμένα σε **σειρά (πράξη AND)**, τα τρανζίστορ pMOS πρέπει να είναι συνδεδεμένα παράλληλα*
  - *Όταν τα τρανζίστορ nMOS είναι συνδεδεμένα **παράλληλα (πράξη OR)**, τα τρανζίστορ pMOS πρέπει να είναι συνδεδεμένα σε σειρά.*
- Ξεκινάμε τη σχεδίαση πάντα από το δίκτυο nMOS υλοποιώντας τις απαιτούμενες πράξεις AND και OR
  - ***Η έξοδος πάντα αντιστρέφεται (πράξη NOT) εκ κατασκευής***

CMOS πύλη NAND-2

- Σχηματικό διάγραμμα:
  - *Στο δίκτυο nMOS τα 2 τρανζίστορ συνδέονται στη σειρά, ώστε να γίνει η πράξη AND*
  - *Στο δίκτυο pMOS τα 2 τρανζίστορ συνδέονται παράλληλα, λόγω του κανόνα των συμπληρωμάτων αγωγιμότητας*

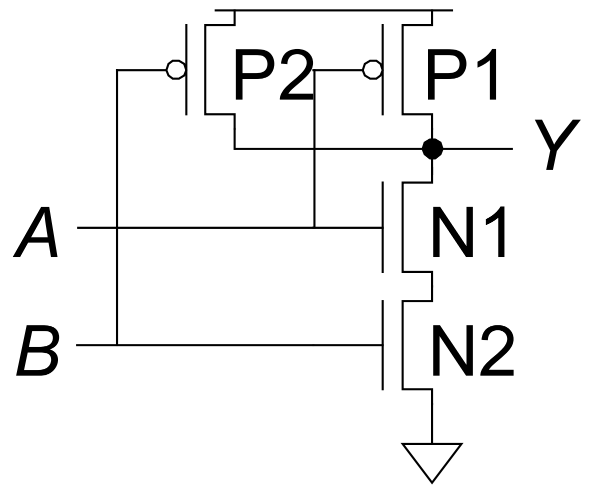
NAND



$Y = \overline{AB}$

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

A	*B*	P1	P2	N1	N2	*Y*
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

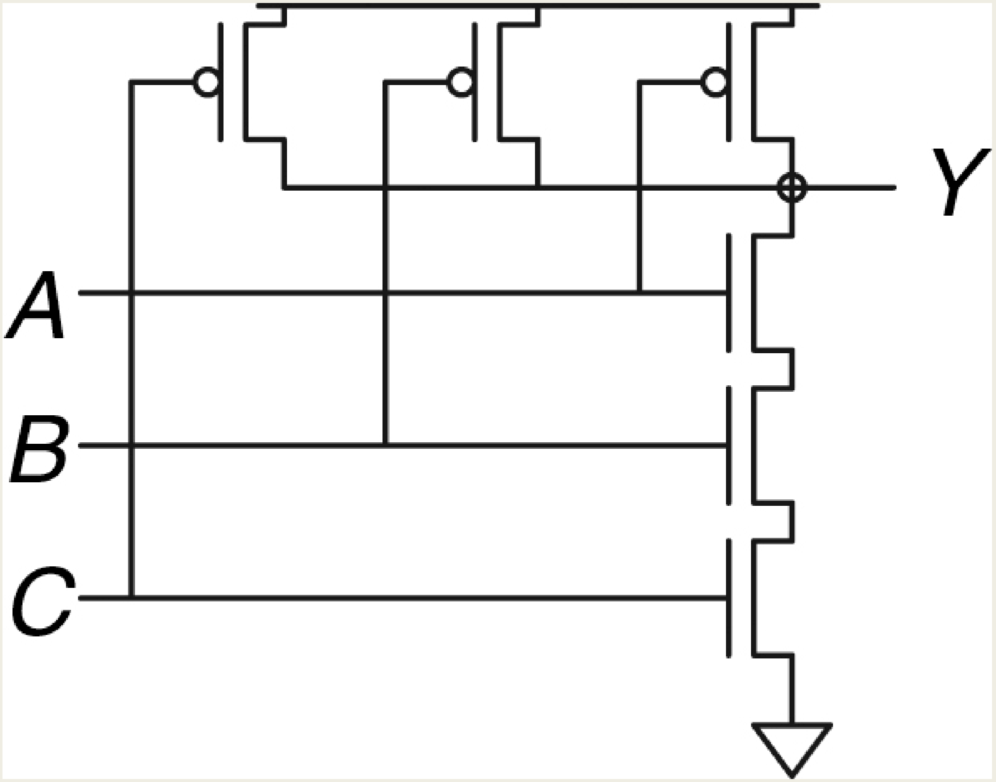


CMOS πύλη NAND-3*

- Σχηματικό διάγραμμα:
  - *Στο δίκτυο nMOS τα 3 τρανζίστορ συνδέονται στη σειρά (πράξη AND)*
  - *Στο δίκτυο pMOS τα 3 τρανζίστορ συνδέονται παράλληλα*

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

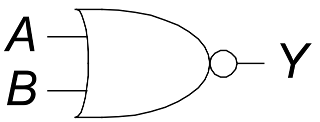
*Παράδειγμα 1.20



CMOS πύλη NOR-2*

- Σχηματικό διάγραμμα:
  - *Στο δίκτυο nMOS τα 2 τρανζίστορ συνδέονται παράλληλα, ώστε να γίνει η πράξη OR*
  - *Στο δίκτυο pMOS τα 2 τρανζίστορ συνδέονται σε σειρά, λόγω του κανόνα των συμπληρωμάτων αγωγιμότητας*

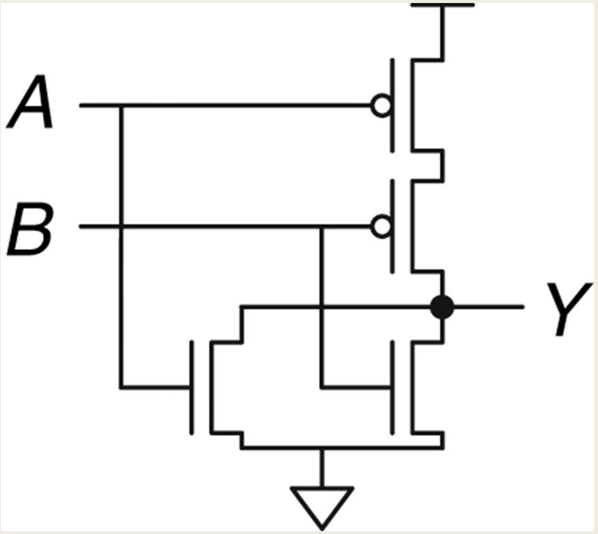
NOR



$Y = \overline{A + B}$

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

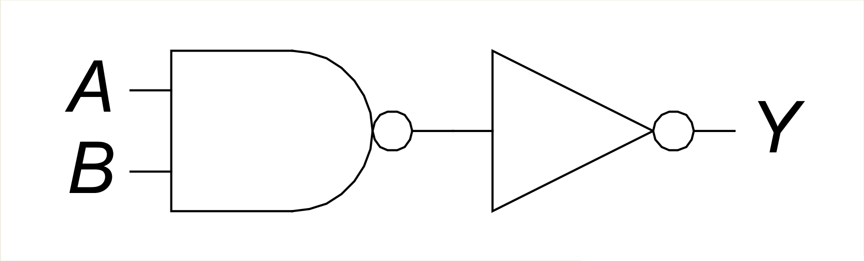
A	*B*	P1	P2	N1	N2	*Y*
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	0
1	0	OFF	ON	ON	OFF	0
1	1	OFF	OFF	ON	ON	0

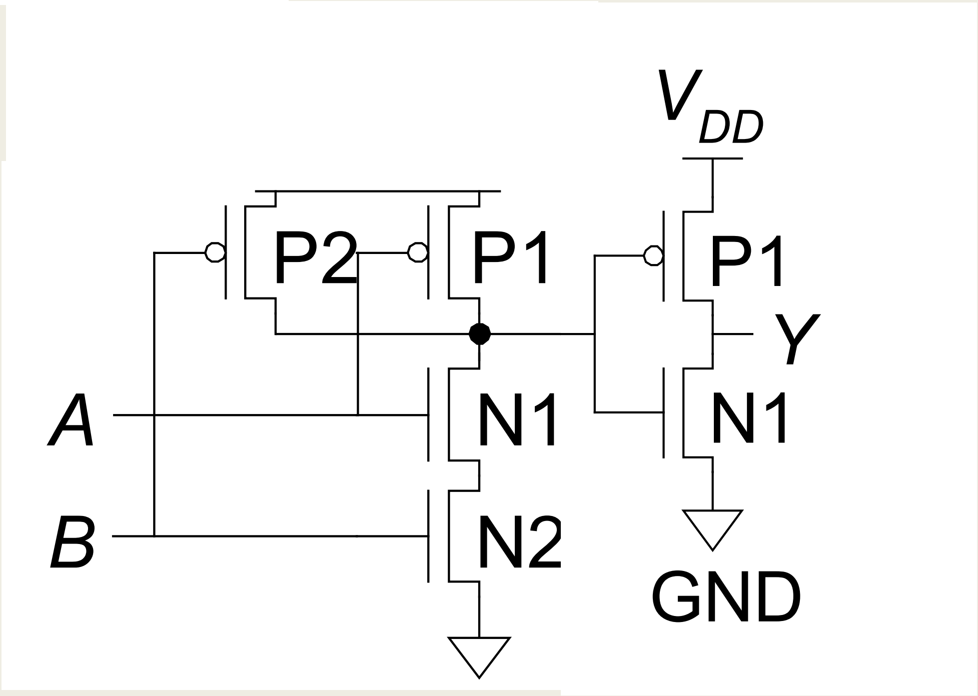


*Παράδειγμα 1.21

CMOS Πύλη AND-2*

- Πώς θα κατασκευάσετε μια πύλη AND?

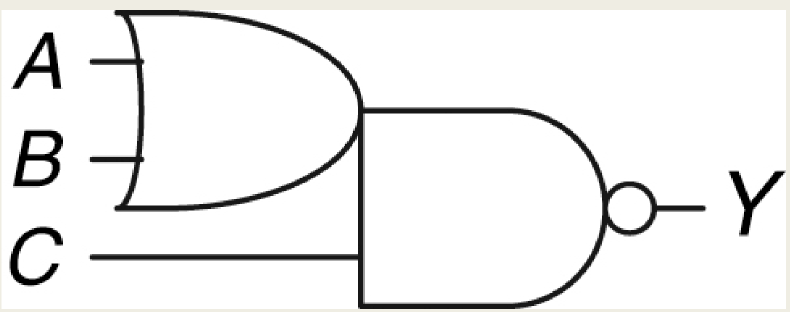


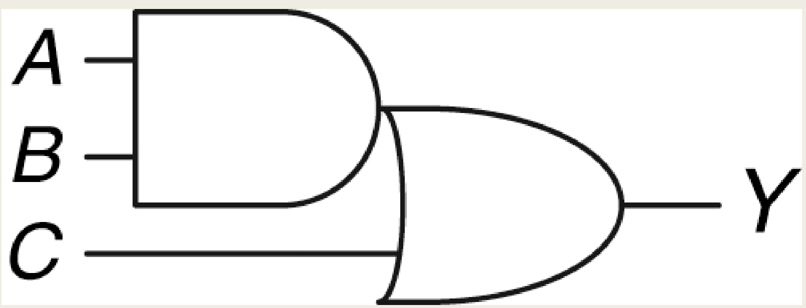


*Παράδειγμα 1.22

Επιλεγμένες ασκήσεις

- Άσκηση 1.84
  Σχεδιάστε ένα σχηματικό διάγραμμα για τις ακόλουθες πύλες CMOS. Χρησιμοποιήστε τον ελάχιστο δυνατό αριθμό τρανζίστορ.
  - *(α) Πύλη NAND-4 με τέσσερις εισόδους*
  - *(β) Πύλη OR-AND-INVERT με τρεις εισόδους*
  - *(γ) Πύλη AND-OR με τρεις εισόδους*

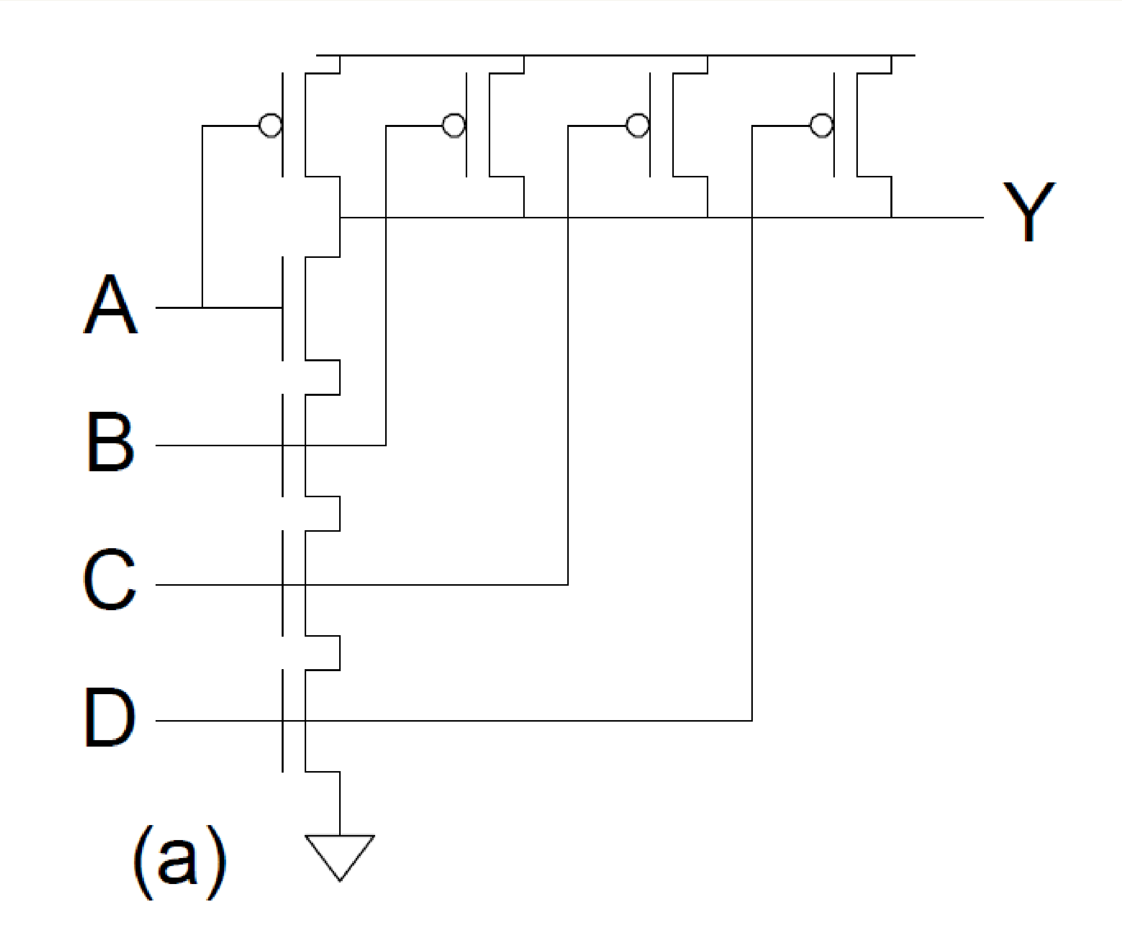




- ## Άσκηση 1.84

  Σχεδιάστε ένα σχηματικό διάγραμμα για τις ακόλουθες πύλες CMOS. Χρησιμοποιήστε τον ελάχιστο δυνατό αριθμό τρανζίστορ.

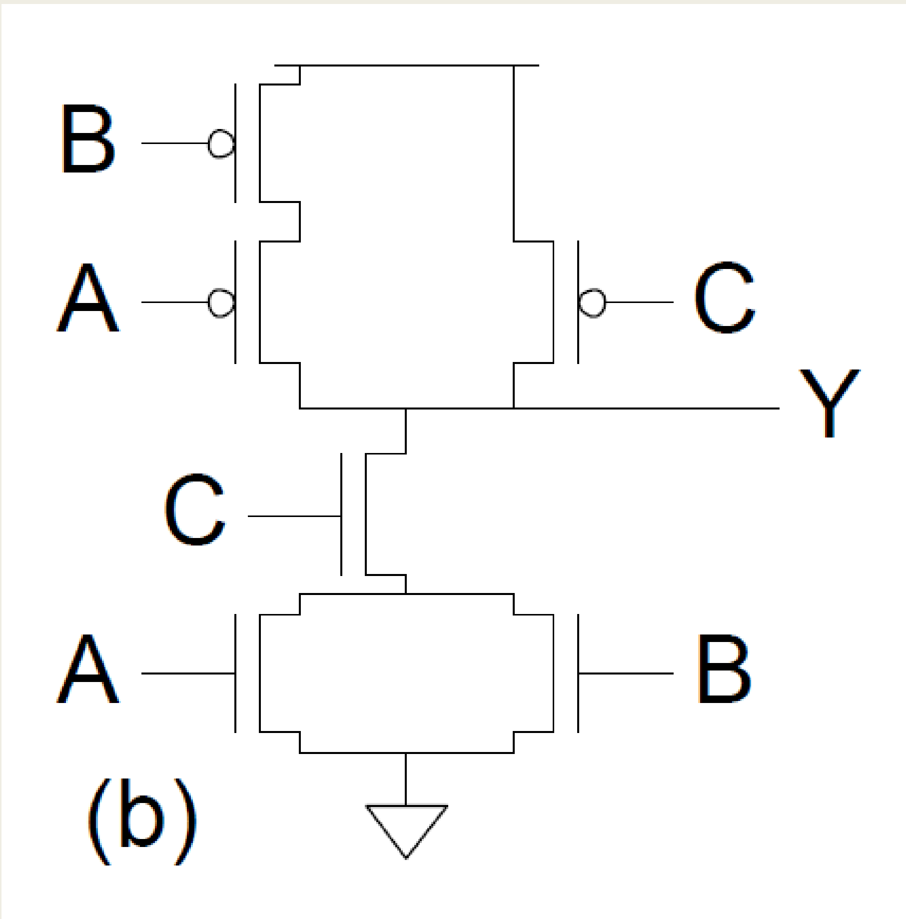
  - *(α) Πύλη NAND-4 με τέσσερις εισόδους*

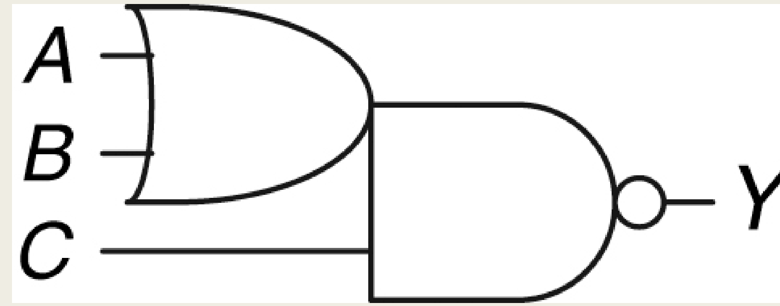


■ Άσκηση 1.84

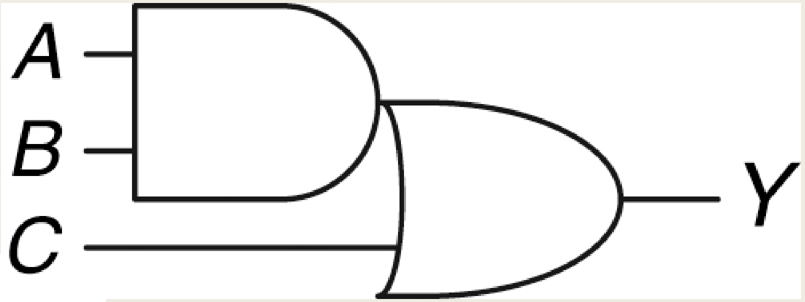
Σχεδιάστε ένα σχηματικό διάγραμμα για τις ακόλουθες πύλες CMOS. Χρησιμοποιήστε τον ελάχιστο δυνατό αριθμό τρανζίστορ.

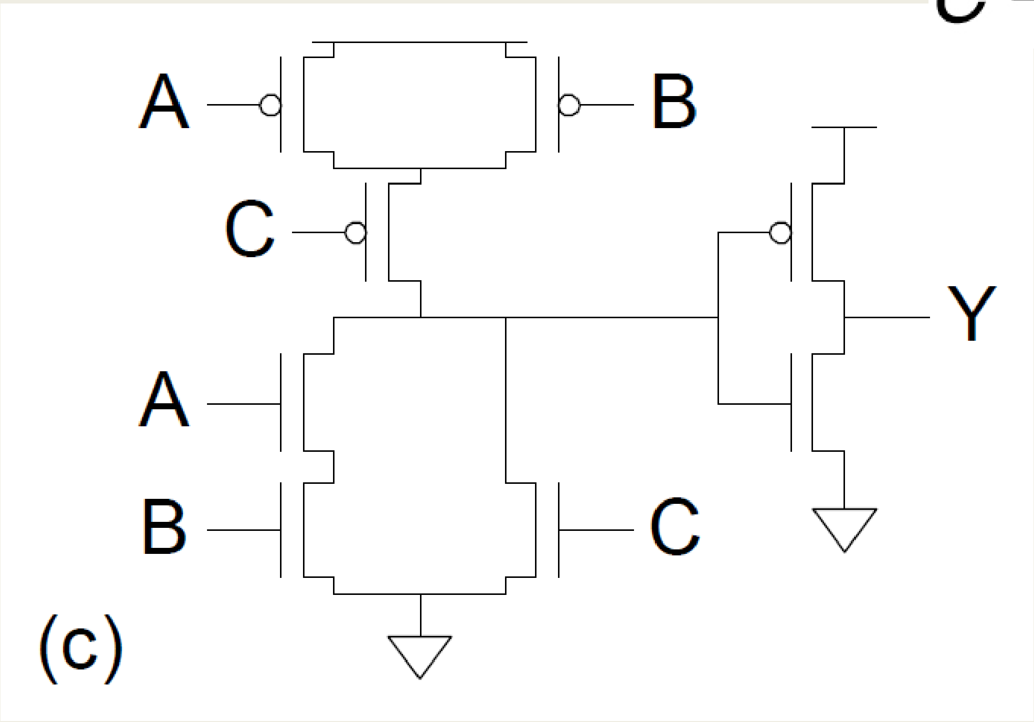
– *(β) Πύλη OR-AND-INVERT με τρεις εισόδους*





- **Άσκηση 1.84**
  Σχεδιάστε ένα σχηματικό διάγραμμα για τις ακόλουθες πύλες CMOS. Χρησιμοποιήστε τον ελάχιστο δυνατό αριθμό τρανζίστορ.
  - *(γ) Πύλη AND-OR με τρεις εισόδους*

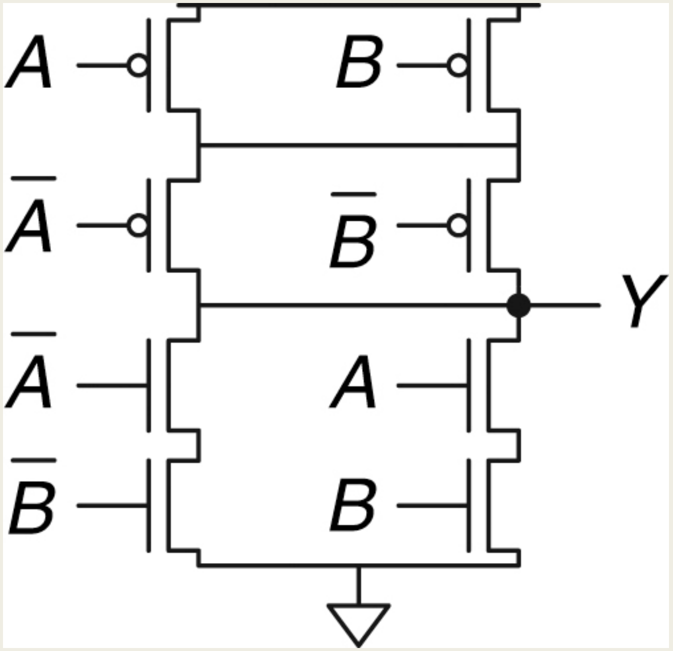




Επιλεγμένες ασκήσεις

- Άσκηση 1.87
  Συμπληρώστε τον πίνακα αληθείας για τη συνάρτηση που εκτελείται από την πύλη της Εικόνας
- Ο πίνακας θα πρέπει να έχει δύο εισόδους, τα *A* και *B*.
- Ποιο είναι το όνομα της συνάρτησης;

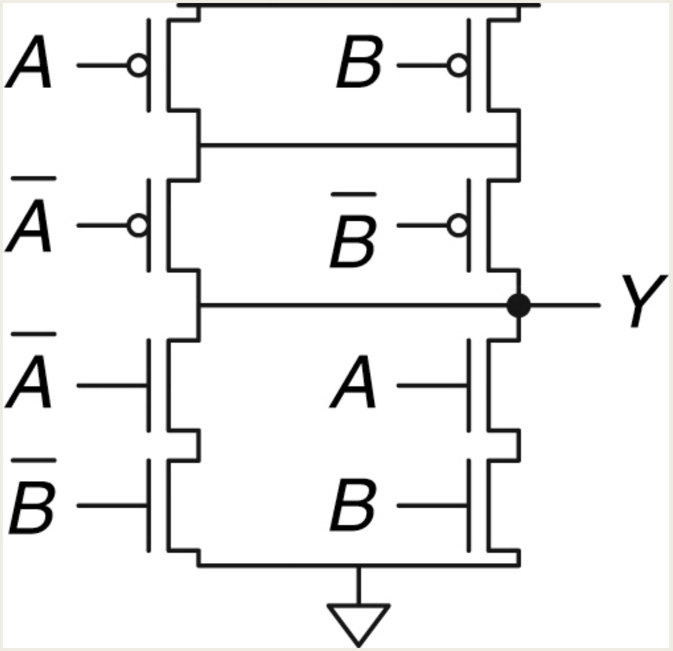
A	*B*	*Y*
0	0	
0	1	
1	0	
1	1	



Επιλεγμένες ασκήσεις

- **Άσκηση 1.87**
  Συμπληρώστε τον πίνακα αληθείας για τη συνάρτηση που εκτελείται από την πύλη της Εικόνας
- Ο πίνακας θα πρέπει να έχει δύο εισόδους, τα *A* και *B*.
- Ποιο είναι το όνομα της συνάρτησης; XOR

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



Κατανάλωση Ισχύος

- Η ποσότητα της ενέργειας που καταναλώνεται ανά μονάδα χρόνου λέγεται **κατανάλωση ισχύος**
- Τα ψηφιακά συστήματα καταναλώνουν ***δυναμική*** και ***στατική*** ισχύς
  - ***Δυναμική ισχύς:*** *η κατανάλωση ισχύος καθώς τα σήματα μεταβάλλονται μεταξύ του 0 και 1*

    $$P_{\text{dynamic}} = \frac{1}{2} C V_{DD}^{2} f$$

    όπου C η μέση χωρητικότητα, V_{DD} η τάση τροφοδοσίας και f η συχνότητα λειτουργίας του τσιπ
  - ***Στατική ισχύς:*** *η κατανάλωση ισχύος όταν τα τρανζίστορ είναι OFF λόγω του* ***ρεύματος διαρροής*** I_{DD} *στο τσιπ*

    $$P_{\text{static}} = I_{DD} V_{DD}$$

Παράδειγμα 1.23

- Ένα κινητό τηλέφωνο διαθέτει μπαταρία των 6 watt-hour (W-hr) και λειτουργεί στα 1,2 V
- Υποθέστε ότι, όταν το κινητό τηλέφωνο βρίσκεται σε χρήση, λειτουργεί στα 300 MHz, η μέση ποσότητα χωρητικότητας στο τσιπ η οποία εναλλάσσεται σε οποιαδήποτε δεδομένη στιγμή είναι ίση με 10 nF (10^{-8} Farad) και εκπέμπει επίσης 3 W ισχύος από την κεραία του.
- Όταν το τηλέφωνο δεν βρίσκεται σε χρήση, η δυναμική ενέργεια μειώνεται σχεδόν στο μηδέν επειδή είναι απενεργοποιημένη η επεξεργασία του σήματος. Όμως, το τηλέφωνο αντλεί 40 mA του ρεύματος ηρεμίας ανεξάρτητα από το αν χρησιμοποιείται ή όχι.
- Υπολογίστε τη διάρκεια ζωής της μπαταρίας του τηλεφώνου (α) αν δεν χρησιμοποιείται, και (β) αν χρησιμοποιείται συνεχώς.

Παράδειγμα 1.23 (Λύση-1/2)

$$P_{\text{static}} = I_{DD}\, V_{DD}$$

- Η στατική ισχύς είναι $P_{\text{static}} = (40\text{ mA})(1{,}2\text{ V}) = 48\text{ mW}$.
- (α) Αν το τηλέφωνο δεν χρησιμοποιείται, τότε αυτή είναι και η μόνη κατανάλωση ισχύος, άρα η διάρκεια ζωής της μπαταρίας είναι (6 Whr)/(0.048 W) = 125 ώρες (περίπου 5 μέρες).

Παράδειγμα 1.23 (Λύση2/2)

$$P_{\text{dynamic}} = \frac{1}{2} C V_{DD}{}^{2} f$$

- Η δυναμική ισχύς είναι $P_{\text{dynamic}} = (0{,}5)(10^{-8}\ \text{F})(1{,}2\ \text{V})^2(3 \times 10^8\ \text{Hz}) = 2{,}16\ \text{W}$.
- (β) Αν το τηλέφωνο χρησιμοποιείται, τότε η συνολική ενεργή ισχύς που καταναλώνεται (η δυναμική ισχύς μαζί με τη στατική ισχύ και την ισχύ εκπομπής) είναι $2{,}16\ \text{W} + 0{,}048\ \text{W} + 3\ \text{W} = 5{,}2\ \text{W}$
- Άρα η διάρκεια ζωής της μπαταρίας είναι 6 W-hr/5,2 W = 1,15 ώρες.

Άσκηση επανάληψης - 1

Ένας επεξεργαστής των 4-bit μετά το τέλος των αριθμητικών του πράξεων (πρόσθεση ή αφαίρεση σε συμπλήρωμα ως προς 2) ενημερώνει και τις αντίστοιχες σημαίες N (Negative), Z (Zero), C (Carry), V (oVerflow). N=1, όταν το αποτέλεσμα είναι ένας αρνητικός προσημασμένος αριθμός. Z=1, όταν το αποτέλσμα είναι μηδέν. C=1, όταν υπάρχει στο αποτέλεσμα υπερχείλιση μη προσημασμένων αριθμών. V=1, όταν υπάρχει στο αποτέλεσμα υπερχείλιση προσημασμένων αριθμών.

Ποιες είναι οι τιμές των σημαιών NZCV μετά την εκτέλεση της πράξης A-B, όπου A=0x8 και B=0x8; (Τα A και B δίδονται στο 16-δικό σύστημα αρίθμησης). Επιλέξτε τη σωστή απάντηση

Α)NZCV="1001"

Β)NZCV="0111"

Γ)NZCV="1001"

Δ)NZCV="1010"

Ε)Δεν ξέρω/Δεν απαντώ

Άσκηση επανάληψης - 1

Ένας επεξεργαστής των 4-bit μετά το τέλος των αριθμητικών του πράξεων (πρόσθεση ή αφαίρεση σε συμπλήρωμα ως προς 2) ενημερώνει και τις αντίστοιχες σημαίες N (Negative), Z (Zero), C (Carry), V (oVerflow). N=1, όταν το αποτέλεσμα είναι ένας αρνητικός προσημασμένος αριθμός. Z=1, όταν το αποτέλσμα είναι μηδέν. C=1, όταν υπάρχει στο αποτέλεσμα υπερχείλιση μη προσημασμένων αριθμών. V=1, όταν υπάρχει στο αποτέλεσμα υπερχείλιση προσημασμένων αριθμών.

Ποιες είναι οι τιμές των σημαιών NZCV μετά την εκτέλεση της πράξης A-B, όπου A=0x8 και B=0x8; (Τα A και B δίδονται στο 16-δικό σύστημα αρίθμησης). Επιλέξτε τη σωστή απάντηση

Α)NZCV="1001"

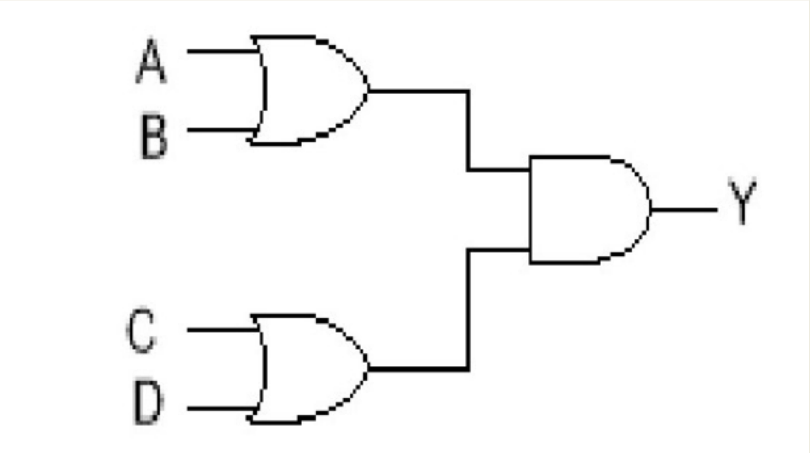
Β)NZCV="0111” (σωστό)

Γ)NZCV="1001"

Δ)NZCV="1010"

Ε)Δεν ξέρω/Δεν απαντώ

Άσκηση επανάληψης - 2



Βρείτε το κύκλωμα CMOS που υλοποιεί το ανωτέρω κύκλωμα

Άσκηση επανάληψης - 2