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!

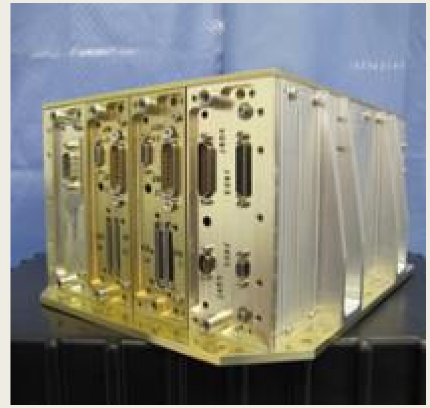
- Ψηφιακές συσκευές καταναλωτών (consumer electronics):

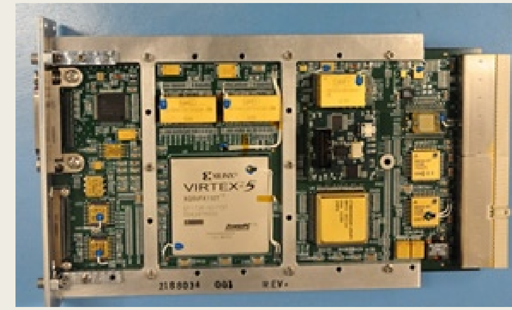
1960

1980

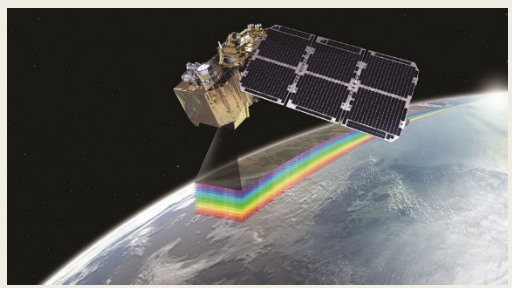
Τα ψηφιακά συστήματα βρίσκονται παντού!

- Μερικά παραδείγματα από την αεροδιαστημική:

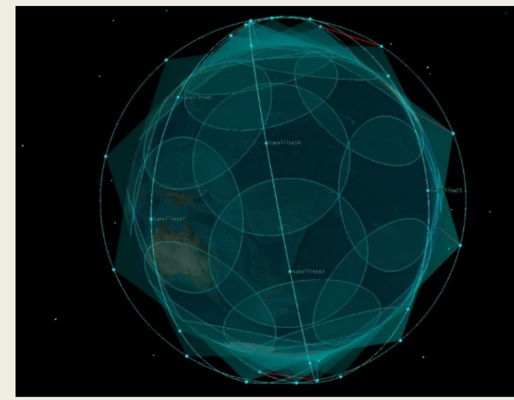




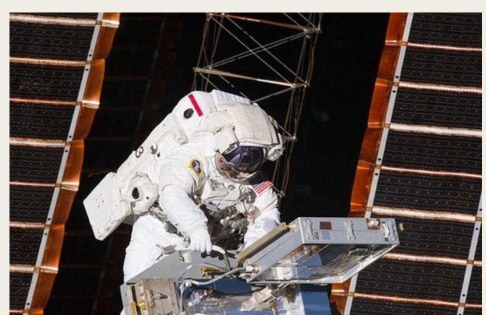
Adaptive image & video processing/compression



Earth observation and hyperspectral imaging

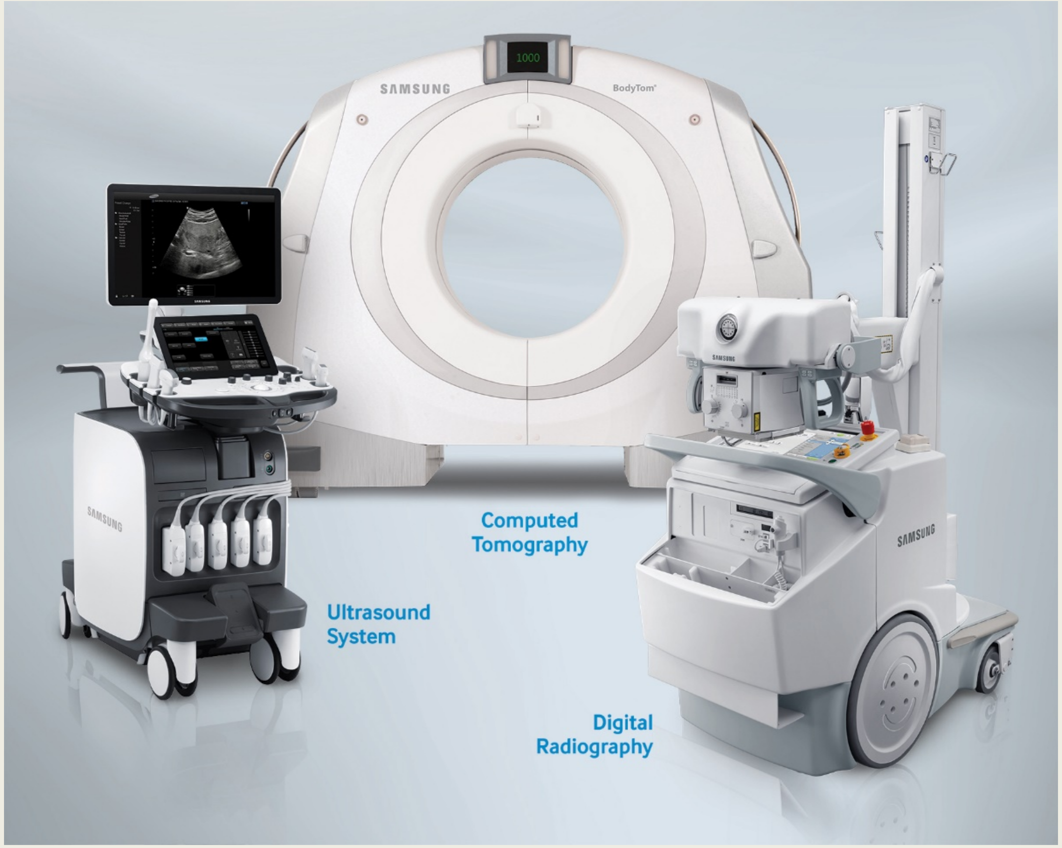


Internet of space and navigation



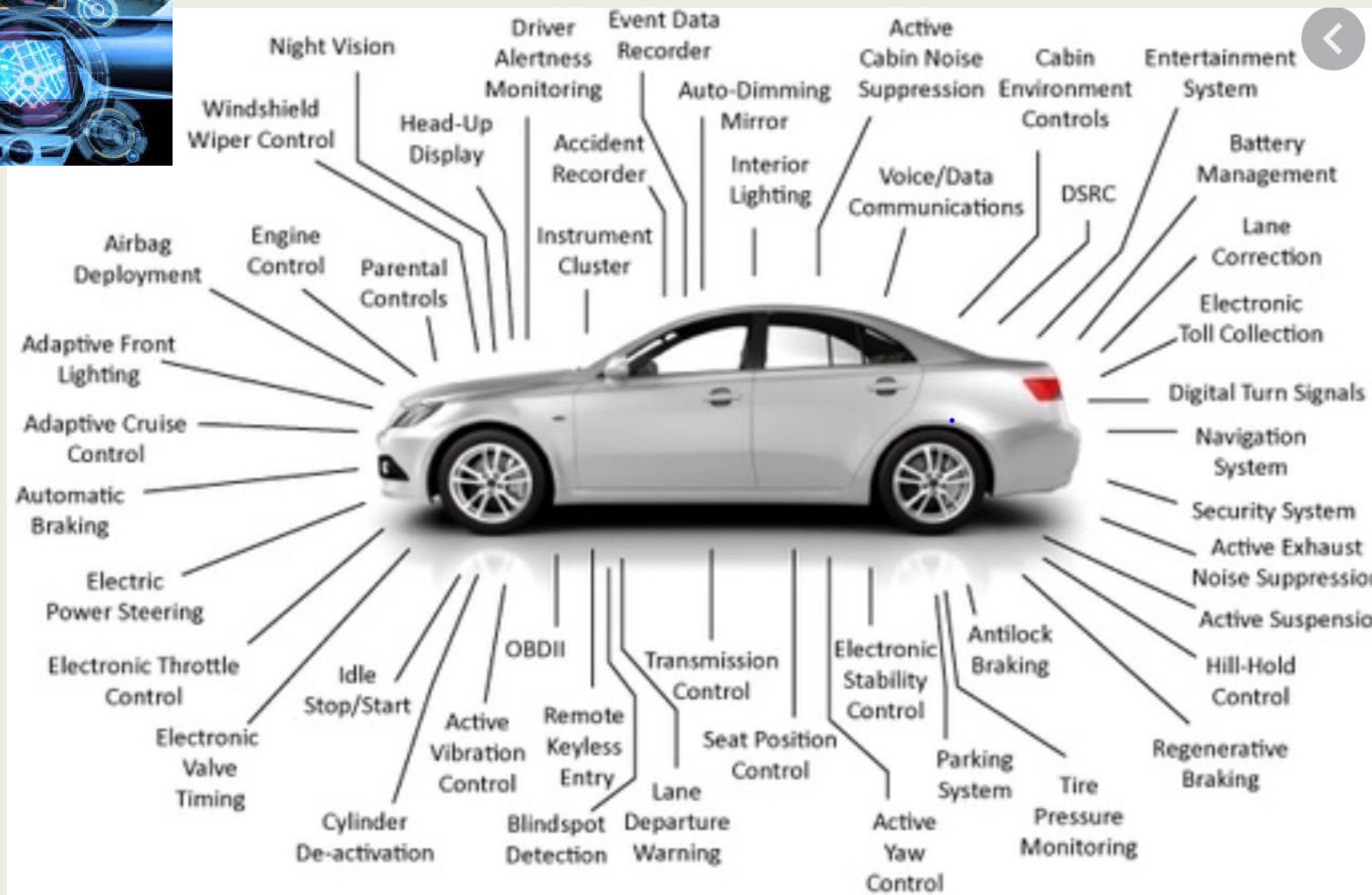
Τα ψηφιακά συστήματα βρίσκονται παντού!

- Μερικά παραδείγματα από την ιατρική:



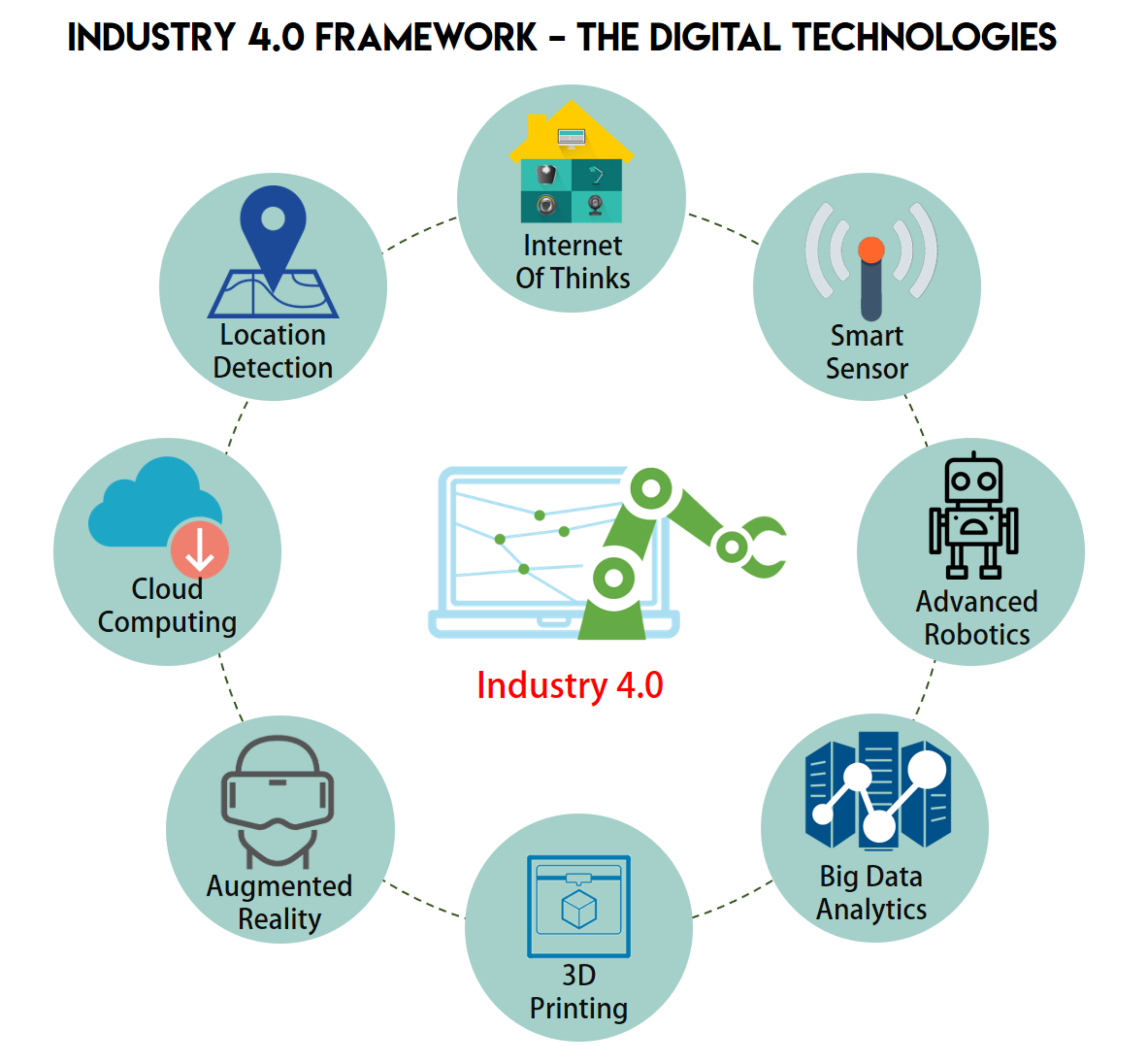
Τα ψηφιακά συστήματα βρίσκονται παντού!

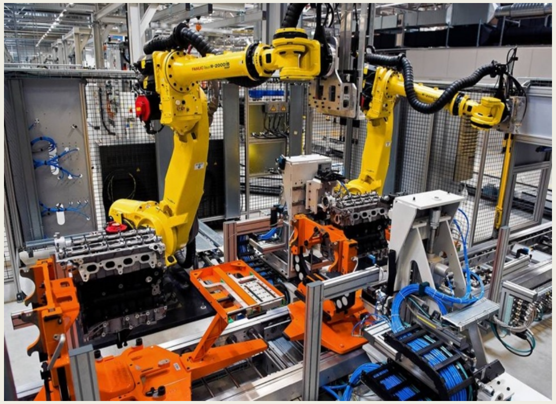
- Μερικά παραδείγματα από την αυτοκινητοβιομηχανία (AI, IoT):



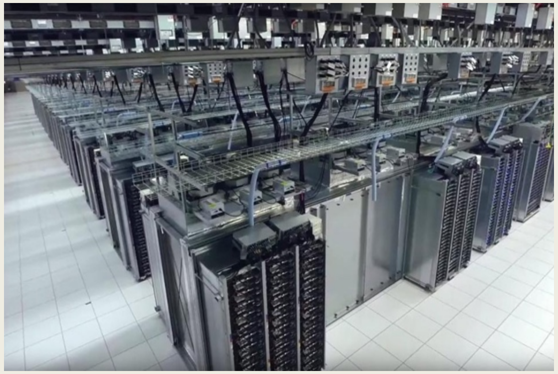
Τα ψηφιακά συστήματα βρίσκονται παντού!

- Μερικά παραδείγματα από την ψηφιακή βιομηχανία:

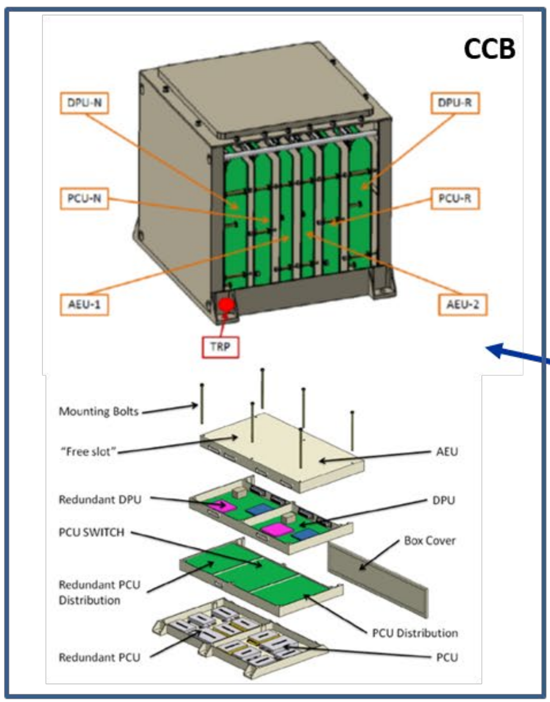


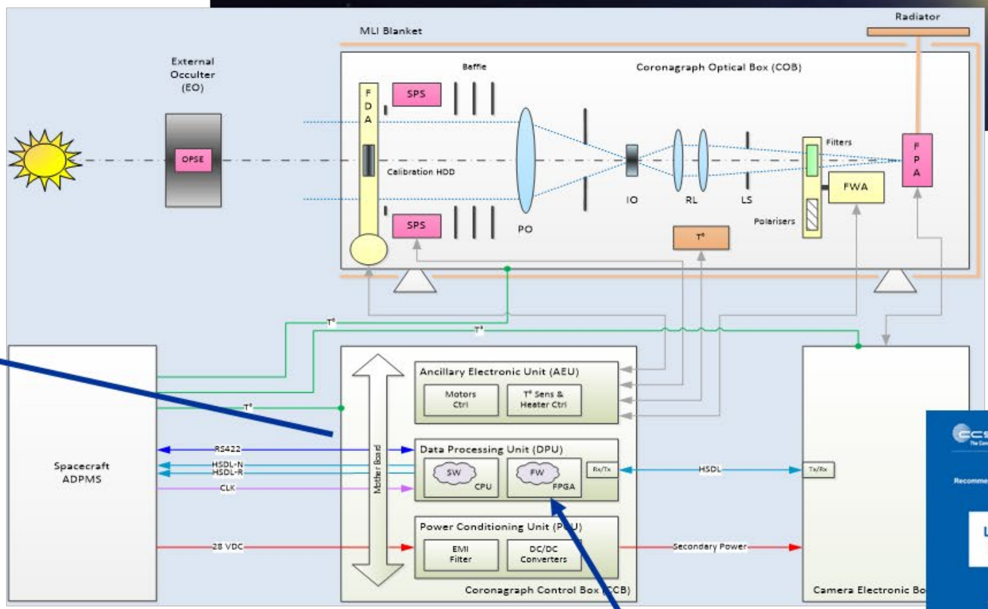


Robotics automation and machine learning @ industry



Big data analytics and machine learning @ data centers





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

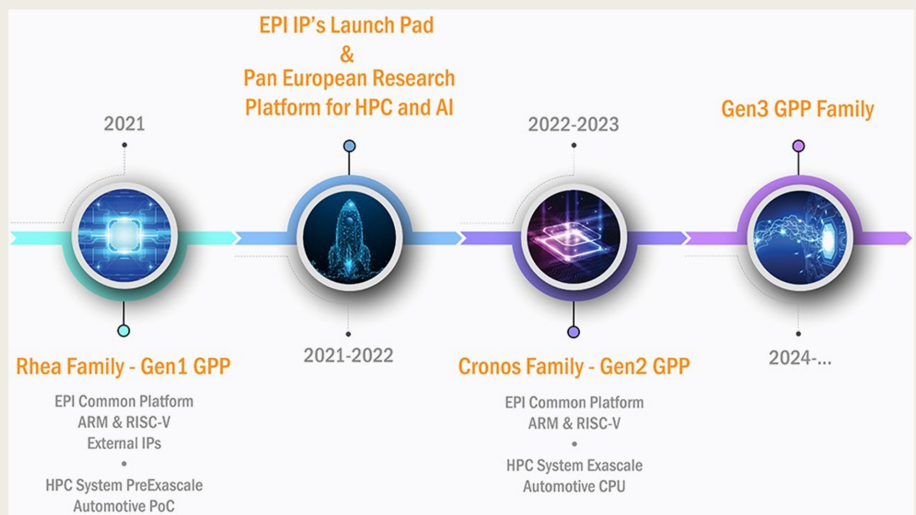
CAL@DI

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

CAL@DI

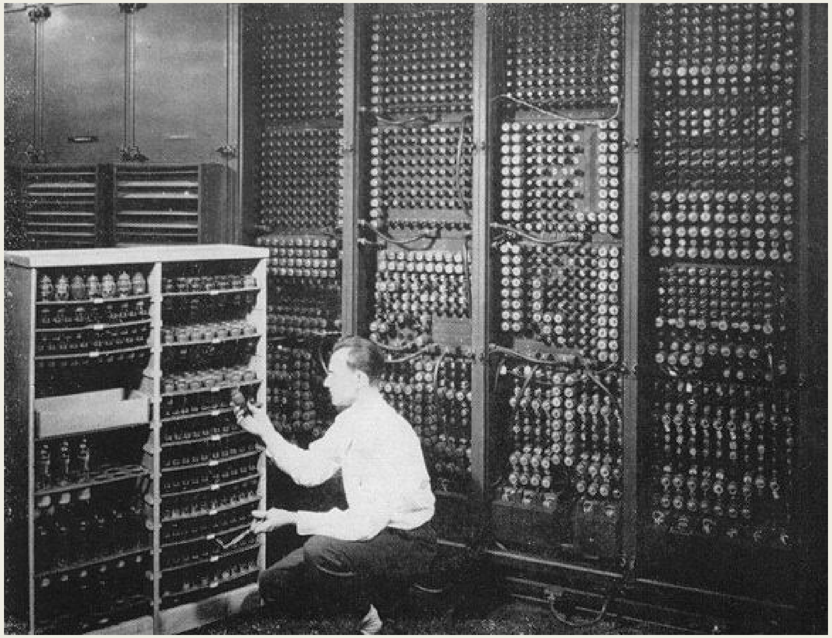
dscal
DIGITAL SYSTEMS & COMPUTER ARCHITECTURE LABORATORY

Περιεχόμενα κεφαλαίου 1

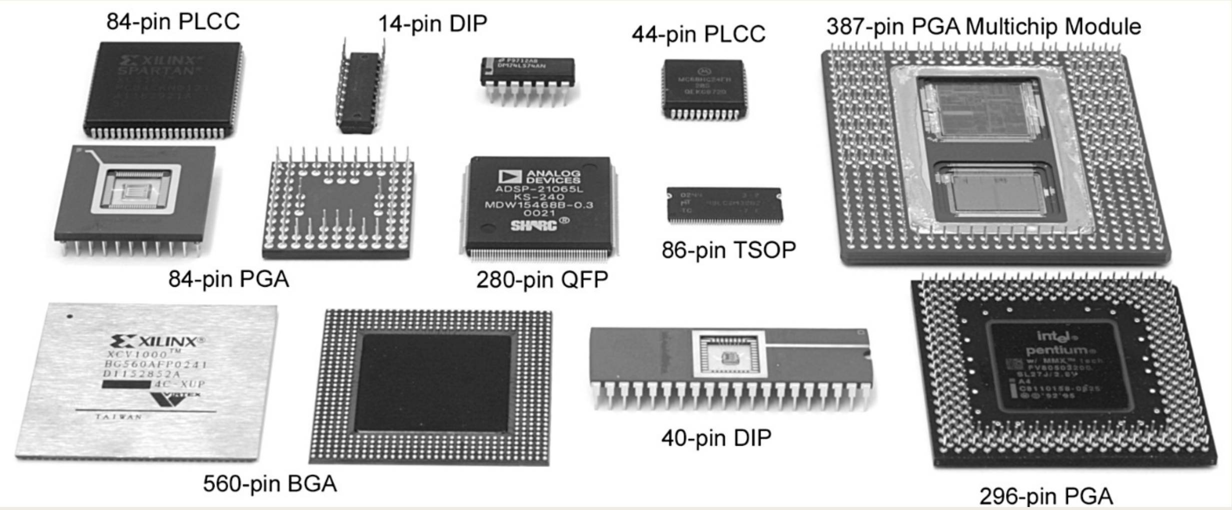
- Εξέλιξη ψηφιακής τεχνολογίας (Εισαγωγή στην Πληροφορική)
- Περί μικροεπεξεργαστών (Εισαγωγή στην Πληροφορική)
- Διαχείριση πολυπλοκότητας υπολογιστικών συστημάτων (Εισαγωγή στην Πληροφορική)
- Ψηφιακή «αφαίρεση» (Εισαγωγή στην Πληροφορική)
- Αναπαράσταση αριθμών σε δυαδική μορφή
- Πράξεις (+, -) μεταξύ 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.

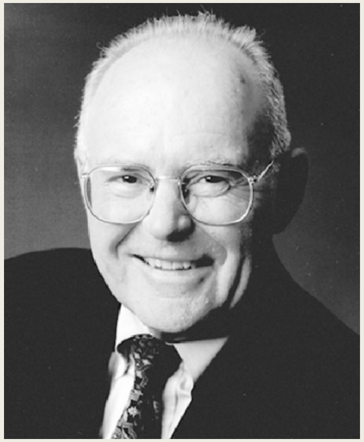
- ENIAC Αντικατάσταση λυχνίας (19.000 πιθανές βλάβες)



- Ολοκληρωμένα κυκλώματα



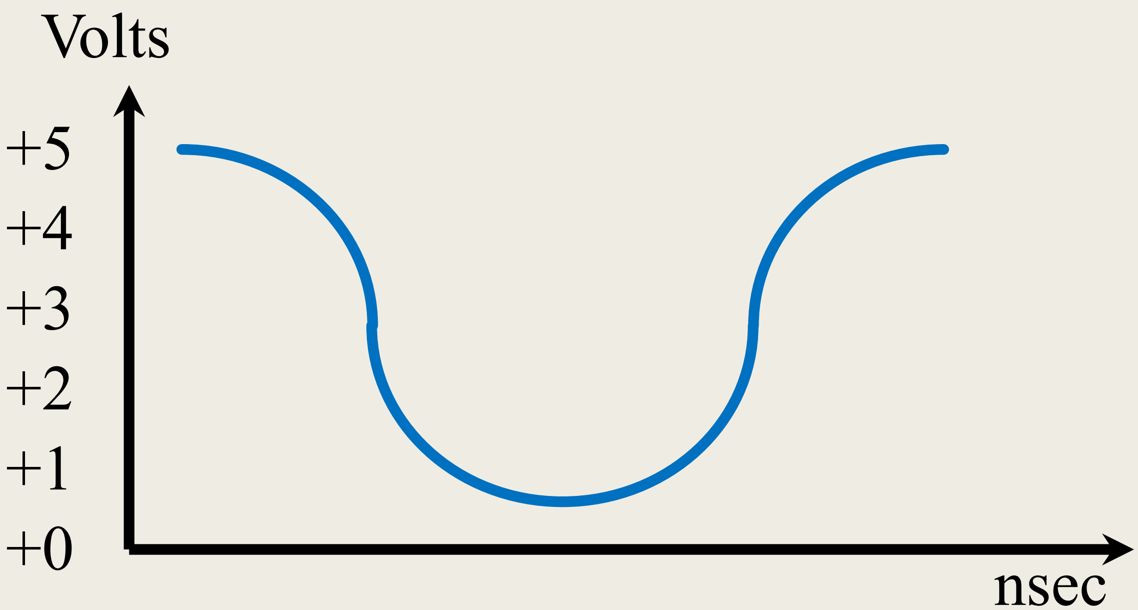
Νόμος του Moore



- 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

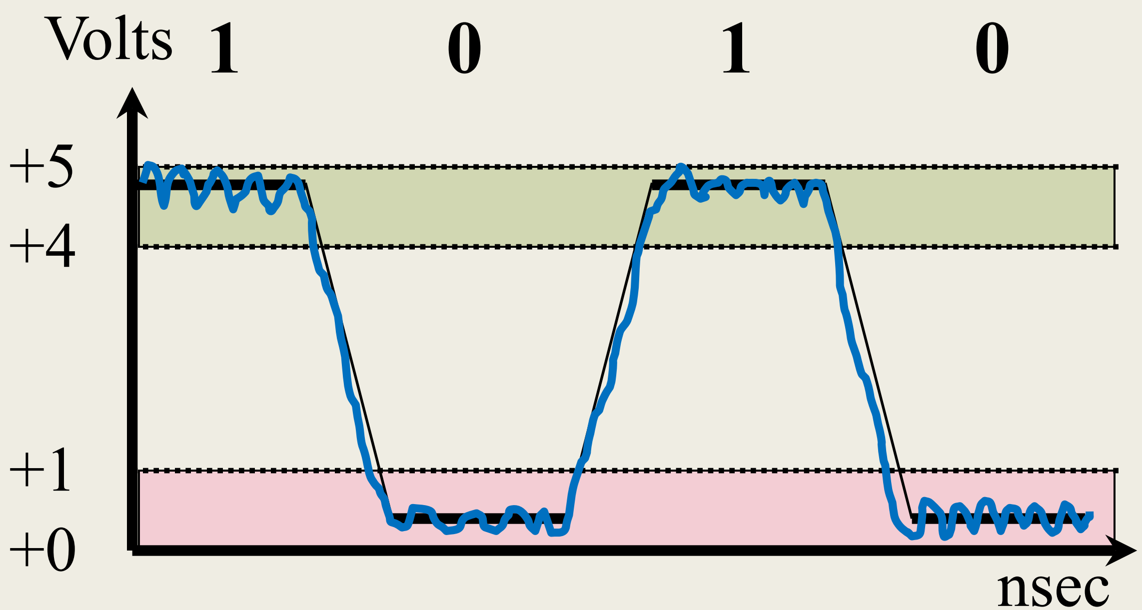
Αναλογικά συστήματα

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

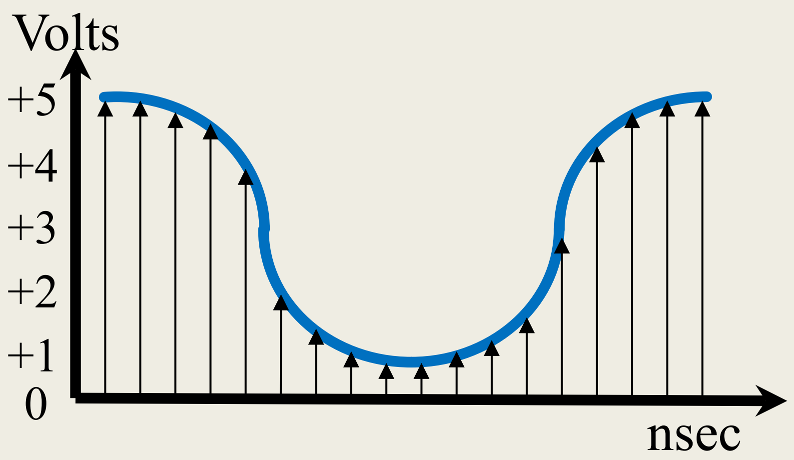


Ψηφιακά συστήματα

- 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 A και B*
    - 1η Ζύγιση, Από 32 σε A και B : Έστω ελαφρύτερη η A
    - 2η Ζύγιση. Από 16 σε A και B : Έστω ελαφρύτερη η A
    - 3η Ζύγιση. Από 8 σε A και B : Έστω ελαφρύτερη η A
    - 4η Ζύγιση. Από 4 σε A και B : Έστω ελαφρύτερη η A
    - 5η Ζύγιση. Από 2 σε A και B : Έστω ελαφρύτερη η A
    - 6η Ζύγιση. Από 1 σε A και B : Έστω ελαφρύτερη η A (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 Α και Β*
    - 1η Ζύγιση, Από 22 σε Α και Β και 20 στην άκρη: Έστω ελαφρύτερη η Α
    - 2η Ζύγιση. Από 8 σε Α και Β και 6 στην άκρη: Έστω ελαφρύτερη η Α
    - 3η Ζύγιση. Από 3 σε Α και Β και 2 στην άκρη : Έστω ελαφρύτερη η Α
    - 4η Ζύγιση. Από 1 σε Α και Β και 1 στην άκρη : Αν Α=Β τότε κάλπικο αυτό που είναι στην άκρη. Αλλιώς κάλπικο αυτό που είναι το ελαφρύτερο από Α και Β

    (υπάρχουν και άλλες διαιρέσεις π.χ. στην $1^{η}$ 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* $\boldsymbol{84_{10}}$ *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.

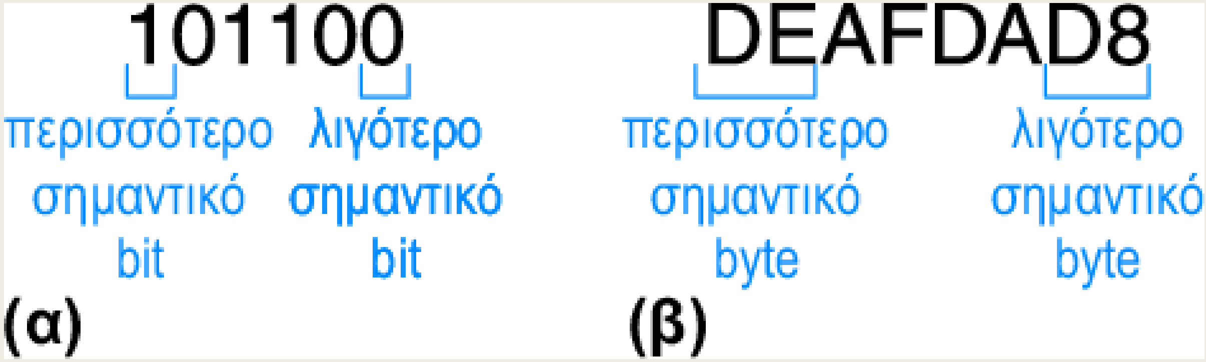
  (β) 63_{10} (δ) 845_{10}

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

Δεκαδικός	Μη προσημα σμένος
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)

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

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

```
  11 1     ← κρατούμενο
   1101
+  0101
-------
   0010
```

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 5ο 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	Προσημα σμένος προσήμου μεγέθους
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Δεκαδικός	Προσημα σμένος 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:*

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

- 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 X οδηγούμαστε στον –X)
  - *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

- <u>Αναπαράσταση ενός αρνητικού αριθμού με χρήση συμπληρώματος ως προς δυο</u>
- 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)
         (143+173=316)
```

Carry

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) $-16_{10} - 21_{10}$

$16_{10}=010000_2$	$21_{10}=010101_2$
$-16_{10}=101111_2 + 1$	$-21_{10}=101010_2 + 1$
110000_2	$=101011_2$

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

  110000_2
$+101011_2$

1011011

Overflow?
NAI

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}.

  (α) 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}.

  (α) 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*