ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

Παράσταση πληροφορίας

Παράσταση Αριθμών

Ι. Βογιατζής

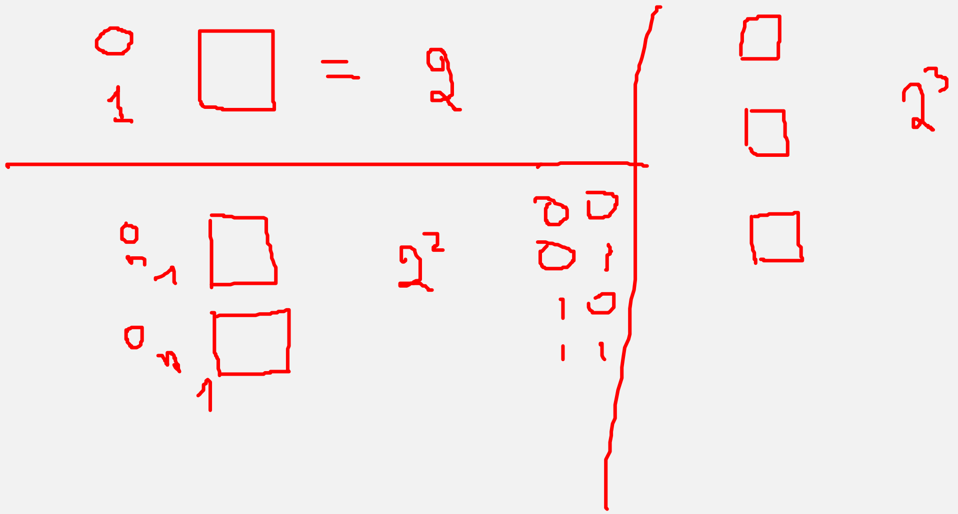
Η έννοια του bit

bits and Bytes (1)

- Το **bit** (μπιτ) (συμβολίζεται ως **b**) είναι ένα δυαδικό ψηφίο το οποίο μπορεί να πάρει τις τιμές 0 ή 1.
- Οι υπολογιστές εργάζονται με το **δυαδικό σύστημα αρίθμησης** και χρησιμοποιούν δυαδικά ψηφία για να συμβολίζουν **εντολές** και **δεδομένα**.
- Η συντομογραφία bit προέρχεται από τη σύντμηση των λέξεων **BI**nary digi**T**.

bits and Bytes (1)

- Το **bit** (μπιτ) (συμβολίζεται ως **b**) είναι ένα δυαδικό ψηφίο το οποίο μπορεί να πάρει τις τιμές 0 ή 1.
- Οι υπολογιστές εργάζονται με το **δυαδικό σύστημα αρίθμησης** και χρησιμοποιούν δυαδικά ψηφία για να συμβολίζουν **εντολές** και **δεδομένα**.
- Η συντομογραφία bit προέρχεται από τη σύντμηση των λέξεων **BI**nary digi**T**.



bits and Bytes (2)

- Το **Byte** είναι **μονάδα μέτρησης** ποσότητας πληροφορίας στα υπολογιστικά συστήματα.

1 Byte ισοδυναμεί με 8 bits

- Το Byte μπορεί να αντιπροσωπεύσει τιμές από 0 έως και 255 στο δεκαδικό σύστημα (2^8 = 256 τιμές).
- Το byte είναι και η βασική μονάδα μέτρησης (χώρου και πληροφορίας) στα υπολογιστικά συστήματα. Πολλαπλάσιά του είναι τα:

μονάδα	πλήθος μπάιτ	προσέγγιση
1 kilobyte (KB)	2^{10}=1.024	10^3
1 megabyte (MB)	2^{20}=1.048.576	10^6
1 gigabyte (GB)	2^{30}=1.073.741.824	10^9
1 terabyte (TB)	2^{40}	10^{12}
1 petabyte (PB)	2^{50}	10^{15}
1 exabyte (EB)	2^{60}	10^{18}

Πόσα Kbyte έχει ένα Mbyte?

Πόσα bits έχει ένα Kbyte?

Παράσταση αριθμών

Παράσταση θετικών αριθμών

- Έστω ότι ένα υπολογιστικό σύστημα χρησιμοποιεί 4 bits για την αναπαράσταση των θετικών αριθμών.
- Οι αριθμοί που μπορούν να παρασταθούν είναι:

Αυτό συνήθως το ονομάζουμε μήκος λέξης!!

0000	0	1000	+8
0001	+1	1001	+9
0010	+2	1010	+10
0011	+3	1011	+11
0100	+4	1100	+12
0101	+5	1101	+13
0110	+6	1110	+14
0111	+7	1111	+15

Παράσταση αρνητικών αριθμών
Παράσταση μέτρου

- Έστω ότι ένα υπολογιστικό σύστημα χρησιμοποιεί 4 bits για την αναπαράσταση των αριθμών.
- Το πιο σημαντικό bit (MSB) χρησιμοποιείται για τον <u>προσδιορισμό του προσήμου</u>.

Γενικά αναπαρίστανται οι αριθμοί $[-(2^{n-1}-1), +(2^{n-1}-1)]$

-7 $+7$

- Οι αριθμοί που μπορούν να παρασταθούν είναι:

0000	0	1000	-0
0001	+1	1001	-1
0010	+2	1010	-2
0011	+3	1011	-3
0100	+4	1100	-4
0101	+5	1101	-5
0110	+6	1110	-6
0111	+7	1111	-7

Παράσταση αρνητικών αριθμών
Παράσταση συμπληρώματος ως προς 1

- Έστω ότι ένα υπολογιστικό σύστημα χρησιμοποιεί 4 bits για την αναπαράσταση των αριθμών.
- Το **συμπλήρωμα ως προς ένα** μιας δυαδικής ακολουθίας λαμβάνεται αλλάζοντας όλα τα bit από μηδέν σε ένα και αντιστρόφως.
- Το πιο σημαντικό bit (MSB) χρησιμοποιείται για τον προσδιορισμό του προσήμου.

Γενικά αναπαρίστανται οι αριθμοί $[-(2^{n-1}-1), +(2^{n-1}-1)]$

$[-(2^3-1), +(2^3-1)]$

$[-7, +7]$

- Οι αριθμοί που μπορούν να παρασταθούν είναι:

A

0000	0	1000	-7
0001	+1	1001	-6
0010	+2	1010	-5
0011	+3	1011	-4
0100	+4	1100	-3
0101	+5	1101	-2
0110	+6	1110	-1
0111	+7	1111	-0

Παράσταση αρνητικών αριθμών
Παράσταση συμπληρώματος ως προς 2

- Έστω ότι ένα υπολογιστικό σύστημα χρησιμοποιεί 4 bits για την αναπαράσταση των αριθμών.
- Το **συμπλήρωμα ως προς δύο** μιας δυαδικής ακολουθίας λαμβάνεται αλλάζοντας όλα τα bit από μηδέν σε ένα και αντιστρόφως και στη συνέχεια προσθέτοντας 1.
- Το πιο σημαντικό bit (MSB) χρησιμοποιείται για τον προσδιορισμό του προσήμου.

Γενικά αναπαρίστανται οι αριθμοί $[-2^{n-1}, +(2^{n-1}-1)]$

- Οι αριθμοί που μπορούν να παρασταθούν είναι:

0000	0	1000	-8
0001	+1	1001	-7
0010	+2	1010	-6
0011	+3	1011	-5
0100	+4	1100	-4
0101	+5	1101	-3
0110	+6	1110	-2
0111	+7	1111	-1

1011
0100
+ 0001
0101

-2 .
+2 = 0010
1101
+ 0001
1110

Παράδειγμα

8

00010001 +17

Προσοχή: θεωρούμε συγκεκριμένο αριθμό bits

για n=6 bits, $X_{10} = -17 \Rightarrow$

010001

10000
\+ 1
10001

Εύρεση συμπληρώματος ως προς 2 του αριθμού 17 :

X = 17 → 010001

101110 (Αντιστροφή όλων των bit)

101111 (Πρόσθεση του 1)

17 | 2
1 8 | 2
0 4 | 2
0 2 | 2
0 1 | 2
1 0

Παράδειγμα

για n=4 bits,   Τι παριστάνει το 0011 σε:

- Παράσταση μέτρου
- Παράσταση συμπληρώματος ως προς 1
- Παράσταση συμπληρώματος ως προς 2
- Μη προσημασμένη αναπαράσταση

Παράδειγμα

για n=4 bits,   Τι παριστάνει το 1011 σε:

- Παράσταση μέτρου
- Παράσταση συμπληρώματος ως προς 1
- Παράσταση συμπληρώματος ως προς 2
- Μη προσημασμένη αναπαράσταση

Ερωτήσεις

1. Ποιους αριθμούς μπορώ να παραστήσω χρησιμοποιώντας 6 bits με παράσταση συμπληρώματος ως προς 2?

   $[-2^{n-1}, +(2^{n-1}-1)] = [-2^5, +2^5-1] = [-32, +31]$

2. Μπορώ να παραστήσω τον αριθμό -60 με τη συγκεκριμένη μορφή αναπαράστασης?
   - ΟΧΙ

3. Ποια θα ήταν η παράσταση του αριθμού 23 και ποια του -13?

Ερωτήσεις

Ποια θα ήταν η παράσταση του αριθμού 23 και ποια του -13 χρησιμοποιώντας 6 bits με παράσταση συμπληρώματος ως προς 2?

Ερωτήσεις

Ποια θα ήταν η παράσταση του αριθμού 23 και ποια του -13 χρησιμοποιώντας 6 bits με παράσταση συμπληρώματος ως προς 2?

Αλγεβρικό Άθροισμα
με παράσταση συμπληρώματος ως προς 2 (1)

Το αλγεβρικό άθροισμα δύο αριθμών στην παράσταση συμπληρώματος του 2 προκύπτει ως το δυαδικό άθροισμα των δύο αριθμών, αγνοώντας το τυχόν κρατούμενο:

+12	001100	+12	001100
+17	010001	- 17	101111
-----	---------	-----	---------
29	011101	- 5	111011

Αλγεβρικό Άθροισμα
με παράσταση συμπληρώματος ως προς 2 (2)

Ας δούμε, όμως κι αυτές τις πράξεις:

+20	010100	-20	101100
+17	010001	-20	101100
-----	---------	-----	---------
-27!	100101	+24!	011000

$$[-2^{n-1}, +(2^{n-1}-1)] = [-2^5, +2^5-1] = [-32, +31]$$

+12	001100	+12	001100
+17	010001	- 17	101111
-----	---------	-----	---------
29	011101	- 5	111011

+20	010100	-20	101100	-1	111111
+17	010001	-20	101100	-1	111111
-----	---------	-----	---------	-----	---------
-27!	100101	+24!	011000	-2	111110

Αριθμητική Υπολογιστών

- **Αριθμητική Πεπερασμένης Ακρίβειας**: Καθώς η ποσότητα της διαθέσιμης μνήμης για την αποθήκευση ενός αριθμού είναι καθορισμένη, οι αριθμοί μπορούν να αναπαρασταθούν με έναν καθορισμένο αριθμό ψηφίων.
- Έστω ότι για την αναπαράσταση θετικών ακεραίων αριθμών διατίθενται **τρία ψηφία του δεκαδικού συστήματος**.
- Με αυτό τον περιορισμό **δεν** μπορούμε να εκφράσουμε:
  - Αριθμούς μεγαλύτερους από 999
  - Αρνητικούς αριθμούς
  - Κλάσματα
  - Μιγαδικούς αριθμούς

Πεπερασμένη Ακρίβεια Υπολογισμών

Οι αριθμοί πεπερασμένης ακρίβειας δεν είναι κλειστοί ως προς τις πράξεις :

- 600 + 600 = 1200 (πολύ μεγάλος)
- 003 – 005 = - 2 (αρνητικός)
- 050 x 050 = 2500 (πολύ μεγάλος)
- 007 / 002 = 3.5 (όχι ακέραιος)

Μπορεί να έχουμε:

- **σφάλμα υπερχείλισης (overflow)**, δηλαδή το αποτέλεσμα είναι μεγαλύτερο από τον μεγαλύτερο αριθμό του συνόλου,
- **σφάλμα ανεπάρκειας/υποχείλισης (underflow)**, όταν το αποτέλεσμα είναι μικρότερο από τον μικρότερο αριθμό του συνόλου.

Ανακεφαλαίωση – Παράσταση Ακεραίων Αριθμών Μήκος λέξης 4 ψηφίων

Μόνο θετικοί Αριθμοί		Αρνητικοί - Παράσταση μέτρου		Αρνητικοί - ΠΣ2	
0000	0	0000	0	0000	0
0001	1	0001	1	0001	1
0010	2	0010	2	0010	2
0011	3	0011	3	0011	3
0100	4	0100	4	0100	4
0101	5	0101	5	0101	5
0110	6	0110	6	0110	6
0111	7	0111	7	0111	7
1000	8	1000	-0	1000	-8
1001	9	1001	-1	1001	-7
1010	10	1010	-2	1010	-6
1011	11	1011	-3	1011	-5
1100	12	1100	-4	1100	-4
1101	13	1101	-5	1101	-3
1110	14	1110	-6	1110	-2
1111	15	1111	-7	1111	-1

Παράσταση πραγματικών αριθμών
Παράσταση σταθερής υποδιαστολής (1)

Υπολογιστής με μήκος λέξης τέσσερα ψηφία, μόνο θετικοί αριθμοί:

0000	000.0	0.0
0001	000.1	0.5
0010	001.0	1.0
0011	001.1	1.5
0100	010.0	2.0
0101	010.1	2.5
0110	011.0	3.0
0111	011.1	3.5
1000	100.0	4.0
1001	100.1	4.5
1010	101.0	5.0
1011	101.1	5.5
1100	110.0	6.0
1101	110.1	6.5
1110	111.0	7.0
1111	111.1	7.5

Παράσταση πραγματικών αριθμών
Παράσταση σταθερής υποδιαστολής (2)

Υπολογιστής με μήκος λέξης τέσσερα ψηφία, ένα για υποδιαστολή, ΠΣ2:

0000	000.0	0.0
0001	000.1	0.5
0010	001.0	1.0
0011	001.1	1.5
0100	010.0	2.0
0101	010.1	2.5
0110	011.0	3.0
0111	011.1	3.5
1000	100.0	-4.0
1001	100.1	-3.5
1010	101.0	-3.0
1011	101.1	-2.5
1100	110.0	-2.0
1101	110.1	-1.5
1110	111.0	-1.0
1111	111.1	-0.5

Παράσταση πραγματικών αριθμών
Παράσταση σταθερής υποδιαστολής (3)

Υπολογιστής με μήκος λέξης τέσσερα ψηφία, δύο για υποδιαστολή, ΠΣ2:

0000	00.00	0.00
0001	00.01	0.25
0010	00.10	0.50
0011	00.11	0.75
0100	01.00	1.00
0101	01.01	1.25
0110	01.10	1.50
0111	01.11	1.75
1000	10.00	-2.00
1001	10.01	-1.75
1010	10.10	-1.50
1011	10.11	-1.25
1100	11.00	-1.00
1101	11.01	-0.75
1110	11.10	-0.50
1111	11.11	-0.25

Παράσταση πραγματικών αριθμών
Ερωτήματα

- Μπορώ να παραστήσω πιο μεγάλους ή πιο μικρούς αριθμούς;
- Θα μπορούσα να παραστήσω με κάποιον άλλον τρόπο την υποδιαστολή, ώστε να «κινείται»;
- Θέματα για προβληματισμό:
  - Τι γίνεται για πολύ μεγάλους αριθμούς;
  - Τι γίνεται για αποτελέσματα της μορφής 0/0?

Παράσταση πραγματικών αριθμών
Παράσταση Κινητής Υποδιαστολής (1)

- Κινητή υποδιαστολή

$$x = sign \times fraction \times base^{exponent-bias}$$

- **sign:** το πρόσημο του αριθμού
- **fraction** (**mantissa, magnitude, significand, σημαντικό μέρος**) κανονικοποιείται, ώστε το πρώτο bit να είναι πάντα 1.
  Το bit αυτό δεν αποθηκεύεται (για λόγους οικονομίας) και ονομάζεται κρυφό bit.
- **base** (**radix)** η βάση του αριθμητικού συστήματος στο οποίο αναπαρίσταται ο αριθμός (2 για το δυαδικό 10 για το δεκαδικό)
- **exponent (characteristic)** είναι ο εκθέτης της βάσης
- **bias (πόλωση)** είναι ένας ακέραιος αριθμός

Παράσταση πραγματικών αριθμών
Παράσταση Κινητής Υποδιαστολής (2)

Πρόσημο	Εκθέτης	Κλασματικό μέρος

1 bit αρκεί (0 θετικό 1 αρνητικό)

Κ bit που συνήθως περιγράφουν έναν πολωμένο εκθέτη

Λ bit που το καθένα έχει το βάρος μιας αρνητικής δύναμης του 2.

Πόλωση=2^{K-1}-1

Στο IEEE 754 έχουμε 2 τέτοιες αναπαραστάσεις:
Α) Απλής ακρίβειας (Κ=8, Λ=23, σύνολο 32 bit, Πόλωση=127)
Β) Διπλής ακρίβειας (Κ=11, Λ=52, σύνολο 64 bit, Πόλωση=1023)

Παράσταση πραγματικών αριθμών
Το πρότυπο IEEE 754

S Πρόσημο	E Εκθέτης	Σ Σημαντικό
Sign	Exponent	Mantissa
1 bit	8 bits	23 bits

αριθμός $X = (-1)^{S} \cdot (1+\Sigma) \cdot 2^{E-127}$,

πόλωση = 127 (2^7 - 1)

(S:πρόσημο, «0»: θετικός, «1»: αρνητικός)

Το πρότυπο IEEE 754 – Παραδείγματα (1)

S	E Εκθέτης	Σ Σημαντικό
0	1000 0010	1100 0000 0000 0000 0000 000

Ποιος αριθμός είναι αυτός;

εκθέτης (πολωμένος) E=1000 0010 = 130

Σ = 1100 0000 0000 0000 0000 000 = 0.75

$\alpha = (-1)^{S} \cdot (1+\Sigma) \cdot 2^{E-127}$

$\alpha = (-1)^{0} \cdot (1+0.75) \cdot 2^{130-127}$

$\alpha = +(1.75) \cdot 2^{3}$

$\alpha = +14$

Το πρότυπο IEEE 754 – Παραδείγματα (2)

S	E Εκθέτης	Σ Σημαντικό
0	0111 1110	1000 0000 0000 0000 0000 000

εκθέτης (πολωμένος) E=0111 1110=126, Σ=0.5

$\boldsymbol{\alpha = (-1)^{0} \cdot (1+0.5) \cdot 2^{126-127} = +(1.5) \cdot 2^{-1} = +0.75}$

S	E Εκθέτης	Σ Σημαντικό
0	1000 0000	1000 0000 0000 0000 0000 000

εκθέτης (πολωμένος) E=1000 0000=128, Σ=0.5

$\boldsymbol{\alpha = (-1)^{0} \cdot (1+0.5) \cdot 2^{128-127} = +(1.5) \cdot 2^{+1} = +3}$

Το πρότυπο IEEE 754 – Παραδείγματα (3)

- Μετατρέψτε τον αριθμό 85.125 σε παράσταση κινητής υποδιαστολής απλής ακρίβειας *(…κατά το πρότυπο IEEE 754)*

+ 85.125

(Μετατροπή σε δυαδικό)

85 = 1 0 1 0 1 0 1 0.125 = 0.0 0 1

(Όλο μαζί)

1 0 1 0 1 0 1 . 0 0 1

(Κανονικοποίηση)

1 . 0 1 0 1 0 1 0 0 1

(Μετατροπή)

1.010101001×2^{6}

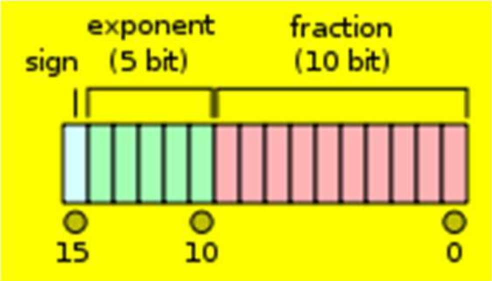
Πόλωση εκθέτη = $127 + 6 = (133)_{10} = (10000101)_2$

(Τελική Μορφή) 01000010101010100100000000000000

Παράσταση πραγματικών αριθμών
Το πρότυπο IEEE 754-2008

Sign	Exponent	Mantissa
Πρόσημο	Εκθέτης	Σημαντικό
1 bit	5 bits	10 bits

- IEEE 754-2008 standard 16-bit base 2 format (half-precision floating point).
- Υπάρχει ένα υπαινισσόμενο (implicit) bit με τιμή 1, *εκτός αν ο εκθέτης (exponent) είναι όλο μηδέν*.
- Εμφανίζονται 10 bits, αλλά η ακρίβεια είναι 11 bits.
- (δεκαδικά ψηφία $\log_{10}(2^{11}) \approx 3.311$).



Το πρότυπο IEEE 754 – Ειδικές περιπτώσεις

Κωδικοποίηση εκθέτη

- *Οι εκθέτες 00000_2 και 11111_2 έχουν ειδική σημασία.*
- Χρησιμοποιείται μια αναπαράσταση στην οποία η πόλωση (offset ή bias) είναι 15. Exponent bias = $01111_2 = 15$
- Ελάχιστος εκθέτης $E_{min} = 00001_2 - 01111_2 = -14$
- Μέγιστος εκθέτης $E_{max} = 11110_2 - 01111_2 = 15$
- Για να πάρουμε τον πραγματικό εκθέτη, πρέπει να αφαιρεθεί το bias από τον αποθηκευμένο εκθέτη.

Το πρότυπο IEEE 754-2008 – Παραδείγματα (1)

Exponent	Mantissa zero	Mantissa non-zero	Equation
00000_2	±zero	denormalized numbers	$(-1)^{signbit} \times 2^{-14} \times 0.significantbits_2$
00001_2, …, 11110_2	normalized value		$(-1)^{signbit} \times 2^{exponent-15} \times 1.significantbits_2$
11111_2	±infinity	NaN (quiet, signalling)	

0 01111 0000000000 = $(-1)^0 \times 2^{15-15} \times 1.0 = 2^0 = 1$
0 01111 0000000001 = $1 + 2^{-10}$ = 1.0009765625 (next float after 1)
1 10000 0000000000 = −2
0 11110 1111111111 = 65504 (max half precision)
0 00001 0000000000 = $2^{-14} \approx 6.10352 \times 10^{-5}$ (min positive normal)
0 00000 1111111111 = $2^{-14} - 2^{-24} \approx 6.09756 \times 10^{-5}$ (max subnormal)

Το πρότυπο IEEE 754-2008 – Παραδείγματα (2)

Exponent	**Mantissa zero**	**Mantissa non-zero**	**Equation**
00000_2	±zero	denormalized numbers	$(-1)^{\text{signbit}} \times 2^{-14} \times 0.\text{significantbits}_2$
00001_2, …, 11110_2	normalized value		$(-1)^{\text{signbit}} \times 2^{\text{exponent}-15} \times 1.\text{significantbits}_2$
11111_2	±infinity	NaN (quiet, signalling)	

0 00000 0000000001 = $2^{-24} \approx 5.96046 \times 10^{-8}$ (min positive subnormal)
0 00000 0000000000 = 0
1 00000 0000000000 = −0
0 11111 0000000000 = infinity
1 11111 0000000000 = −infinity
0 01101 0101010101 = 0.333251953125 ≈ 1/3

Το IEEE 754-2008 – Παράσταση Πραγματικών

- Ακέραιοι μεταξύ 0 και 2048 αναπαριστώνται ακριβώς
- Ακέραιοι μεταξύ 2049 και 4096 στρογγυλοποιούνται σε πολ/σια του 2
- Ακέραιοι μεταξύ 4097 και 8192 στρογγυλοποιούνται σε πολ/σια του 4
- Ακέραιοι μεταξύ 8193 και 16384 στρογγυλοποιούνται σε πολ/σια του 8
- Ακέραιοι μεταξύ 16385 και 32768 στρογγυλοποιούνται σε πολ/σια του 16
- Ακέραιοι μεταξύ 32769 και 65519 στρογγυλοποιούνται σε πολ/σια του 32
- Ακέραιοι από 65520 και πάνω στρογγυλοποιούνται σε "infinity"

Αναπαράσταση αριθμών και σφάλματα

- **Ο περιορισμός στο μέγεθος της λέξης της μνήμης έχει ως συνέπεια να** μπορούν να αποθηκευτούν μόνο **στρογγυλοποιημένες, προσεγγιστικές τιμές** πραγματικών αριθμών.
- Επομένως, **οι υπολογιστές στην πραγματικότητα κάνουν λάθη σε κάθε έναν από τους υπολογισμούς** που πραγματοποιούν.
    - **Μικρά**, αλλά **πολλά**.
- Όλα αυτά τα μικρά λάθη μπορούν να **συσσωρευτούν** και να αυξηθούν μέσα σε κάθε υπολογισμό που γίνεται και τελικά να δώσουν ένα αποτέλεσμα που είναι αρκετά διαφορετικό από το ακριβές.

Υπολογισμοί και σφάλματα (1)

- Στις 25/2/1991, στην πόλη Νταχράν της Σ. Αραβίας, κατά την διάρκεια του πόλεμου στον κόλπο, μία Αμερικάνικη συστοιχία πυραύλων τύπου Patriot, απέτυχε να εντοπίσει και να εξουδετερώσει έναν Ιρακινό πύραυλο τύπου Scud.
- Ο Scud χτύπησε ένα στόχο του Αμερικάνικου στρατού, σκοτώνοντας 28 στρατιώτες και τραυματίζοντας 100.
- Αίτιο της αποτυχίας ήταν τα λάθη στρογγυλοποιήσεων στον υπολογισμό του χρόνου.
- Ειδικότερα: ο χρόνος χωριζόταν σε δέκατα του second και χρησιμοποιούσε ένα καταχωρητή 24 δυαδικών θέσεων για αποθήκευση.

Υπολογισμοί και σφάλματα (2)

Ο αριθμός 0.1 αναπαρίσταται στον υπολογιστή ως εξής:

$$1/10 = 1/2^4+1/2^5+1/2^8+1/2^9+ 1/2^{12}+ 1/2^{13} + ...$$

Δηλαδή η δυαδική αναπαράσταση του 1/10 είναι

0.0001100110011001100110011001100....

Ο καταχωρητής των Patriot αποθήκευε μόνο το

0.00011001100110011001100

παράγοντας ένα λάθος

0.0000000000000000000000011001100...

στο δυαδικό σύστημα, ή περίπου 0.000000095 στο δεκαδικό.

Έπειτα από 100 ώρες σε λειτουργία η συστοιχία μέτρησε 100X3600X10 = 3600000 δέκατα του second παράγοντας λάθος 0.000000095×3600000 = 0.342sec.

Ο Scud ταξίδευε με ταχύτητα 1,676m/sec και σε 0.342s ήταν 573m μακρύτερα από το αναμενόμενο.

Απορίες?