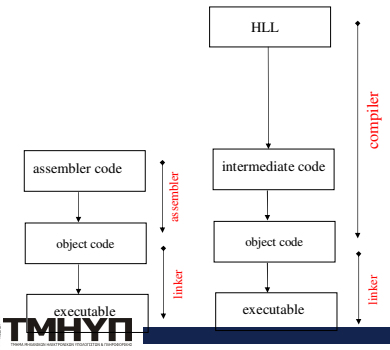
Επιστημονικός Υπολογισμός Ι
ΗΥ 343: ΔΙΑΛΕΞΗ 5

Ε. Γαλλόπουλος
Τμήμα Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών

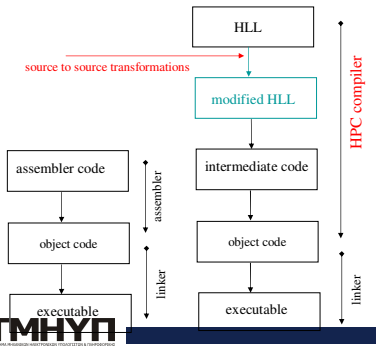
Προσέξτε τη σχέση n, Ω, Mflop/s σε κάποιο «laptop 2ετίας»

Operations	n	Ω	Mflop/s
calling PAPI flops	200	2	0.15
dot product	200	413	13.73
matrix vector	200	82053	252.12
random matrix	200	139967	67.12
chol(a)	200	3201127	789.27
lu(a)	200	5493443	829.53
x=a\y	200	6228144	742.98
condest(a)	200	7126555	173.63
qr(a)	200	13236723	1033.10
matrix multiply	200	16000012	1280.42
inv(a)	200	17398916	853.39
svd(a)	200	27039244	685.65
cond(a)	200	27000896	763.26
hess(a)	200	30180072	1063.27
eig(a)	200	82578728	680.60
[u,s,v]=svd(a)	200	138280160	691.18
pinv(a)	200	170228800	764.50
s=gsvd(a)	200	303512192	765.81
[x,e]=eig(a)	200	198741216	753.79
[u,v,x,c,s]=gsvd(a,b)	200	319475232	789.67

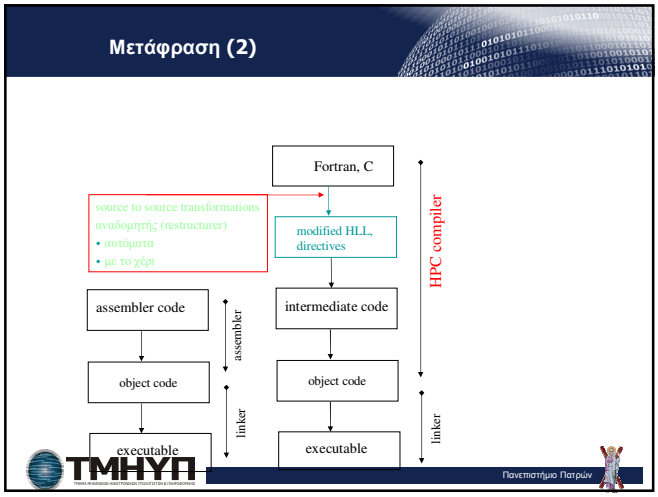
Μετάφραση (1)



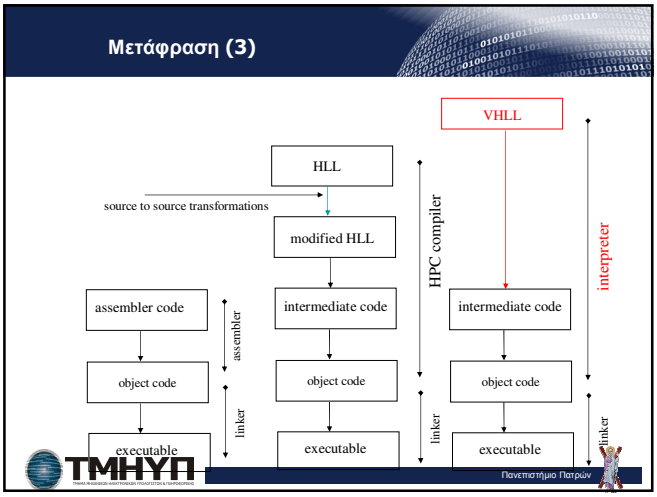
Μετάφραση (2)



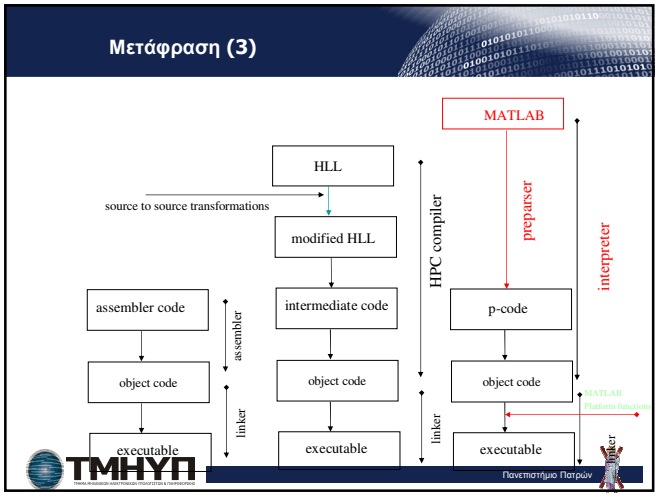
Μετάφραση (2)



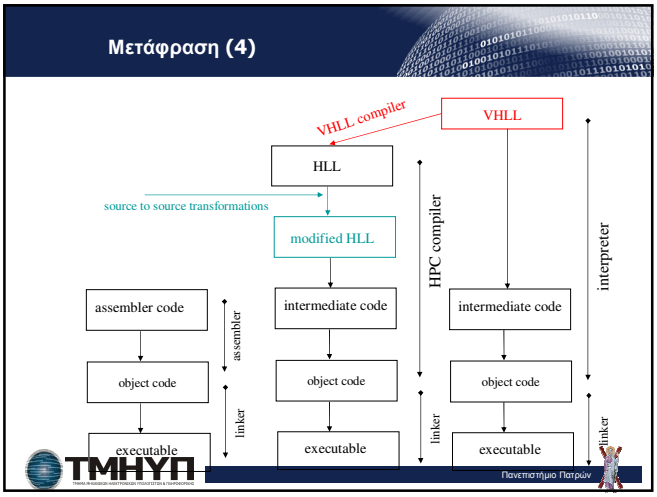
Μετάφραση (3)

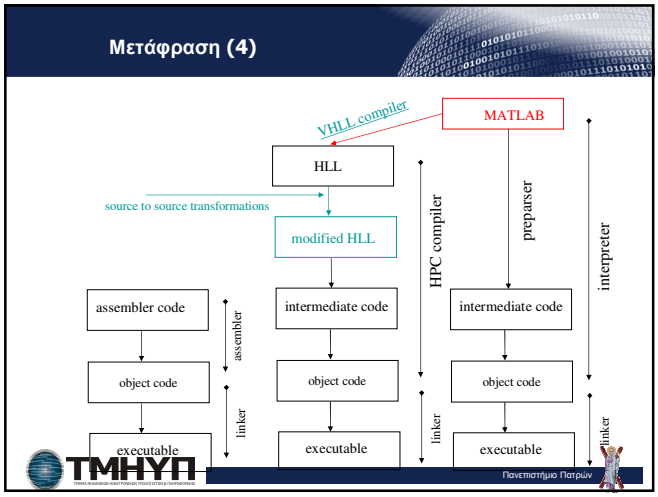


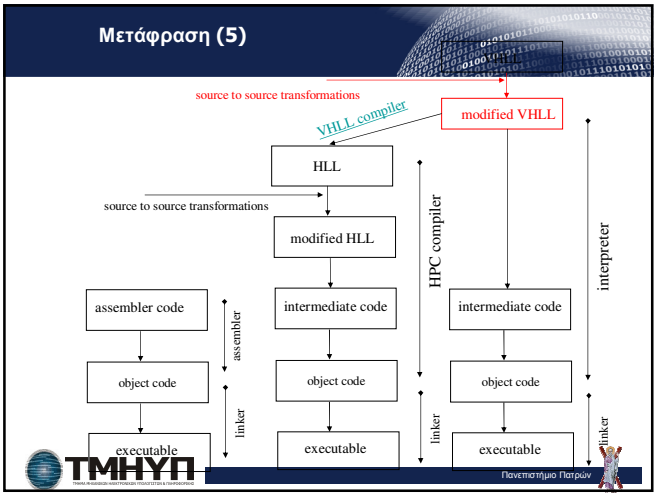
Μετάφραση (3)

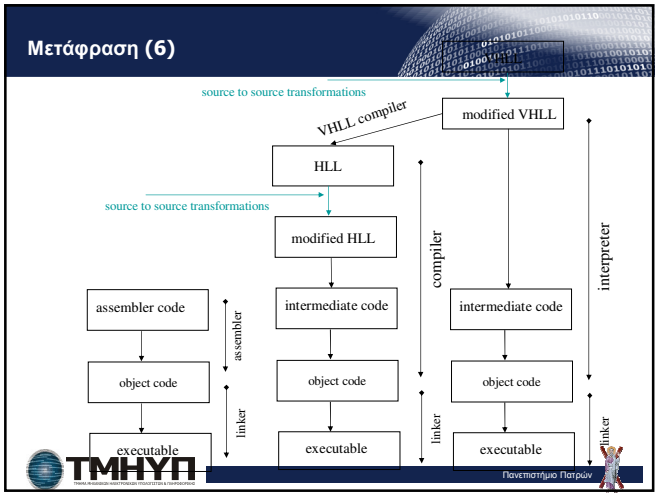


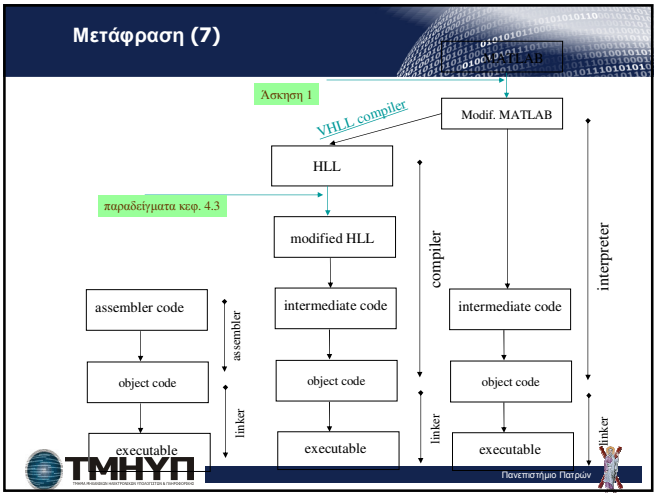
Μετάφραση (4)




Μετάφραση (4)
VHLL compiler
MATLAB
HLL
source to source transformations
modified HLL
HPC compiler
preparser
interpreter
assembler code
intermediate code
intermediate code
assembler
object code
object code
object code
linker
executable
executable
executable
linker
linker
ΤΜΗΥΠ
Πανεπιστήμιο Πατρών


Μετάφραση (5)
source to source transformations
VHLL compiler
modified VHLL
HLL
source to source transformations
modified HLL
HPC compiler
interpreter
assembler code
intermediate code
intermediate code
assembler
object code
object code
object code
linker
executable
executable
executable
linker
linker
ΤΜΗΥΠ
Πανεπιστήμιο Πατρών


Μετάφραση (6)
source to source transformations
VHLL compiler
modified VHLL
HLL
source to source transformations
modified HLL
compiler
interpreter
assembler code
intermediate code
intermediate code
assembler
object code
object code
object code
linker
executable
executable
executable
linker
linker
ΤΜΗΥΠ
Πανεπιστήμιο Πατρών

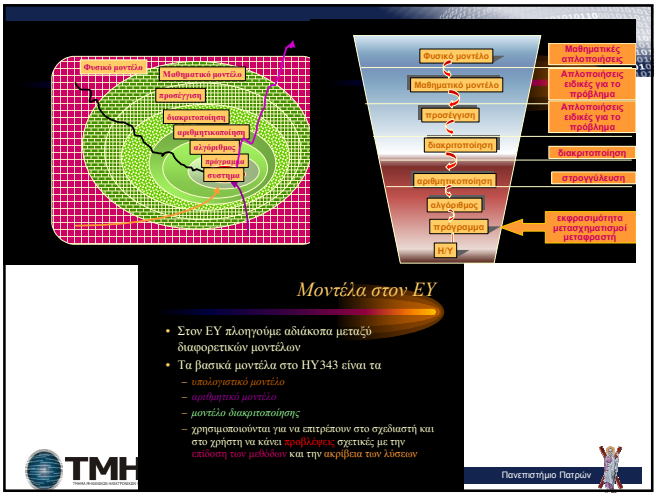

Μετάφραση (7)
Άσκηση 1
VHLL compiler
Modif. MATLAB
HLL
παραδείγματα κεφ. 4.3
modified HLL
compiler
interpreter
assembler code
intermediate code
intermediate code
assembler
object code
object code
object code
linker
executable
executable
executable
linker
linker
ΤΜΗΥΠ
Πανεπιστήμιο Πατρών

Παράδειγμα

Δίδονται $A_1, \ldots, A_s$ όπου κάθε A_j είναι $n_j \times n_{j-1}$
και θέλουμε να υπολογίσουμε το $B = A_s\, A_{s-1} \cdots A_1$
ΕΚΔΟΧΗ ΑρΔε:

```
B=A_1; for j=2:s, B = mxmul(A_j,B); end
```

- Αν $n_0 = 1$ και $n_j = n$ for $j = 1:s$, τότε
  - Αριστερά προς Δεξιά $T = O(s\, n^3)$
  - Δεξιά προς Αριστερά $T = O(s\, n^2)$
- Μεγάλη διαφορά κόστους (O(n)) που εξαρτάται από τη σειρά των υπολογισμών
  - Επίλυση με *δυναμικό προγραμματισμό*
- Ακόμα πιο δύσκολο να λάβουμε υπόψη τη *δομή* των A_j
- Ακόμα πιο ενδιαφέρον και «ανατροπές» σε ιεραρχική μνήμη
  - block Householder (κεφ. 6)



Μοντέλο αριθμητικής

- **Επιπτώσεις της αναπαράστασης των αριθμών με πεπερασμένο αριθμό ψηφίων.**
- **Αριθμητική κινητής υποδιαστολής**

Floating point arithmetic is by nature inexact, and it is not difficult to misuse it so that the computed answers consist almost entirely of "noise". One of the principal components of numerical analysis is to determine how accurate the results of certain numerical methods will be

David Knuth στο The Art of Computer Programming, vol. 2

- Οι παρακάτω πράξεις είναι ισοδύναμες και το αποτέλεσμα ίσο με **10**, αλλά σε α.κ.υ. IEEE διπλής ακρίβειας (π.χ. στη **MATLAB**)

$$10^{20} + 20 - 10 - 10^{20} = \quad 0$$

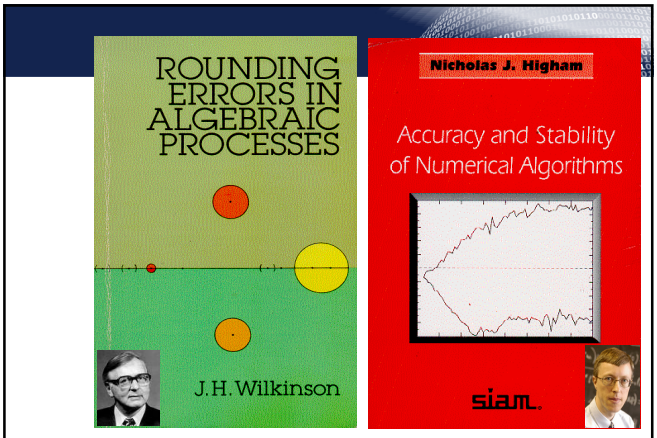
$$10^{20} + 20 - 10^{20} - 10 = \quad -10$$

$$10^{20} - 10 - 10^{20} + 20 = \quad 20$$

- Προσέξτε ότι $10^{20} > 2^{52}$ επομένως τα παραπάνω πρέπει να αναμένονται!
- Ερώτηση: Από τους **24 (=4!)** τρόπους υπολογισμού παραπάνω, ποιοί επιστρέφουν σωστό αποτέλεσμα;

Παράδειγμα

- Πόσες επαναλήψεις θα εκτελέσει ο παρακάτω βρόχος?
- >> d= 0; while (d ~= 1.0), d = d+0.1; end
- Το αποτέλεσμα θα έχουν
  - >> sqrt(10)-sqrt(2)*sqrt(5)
  - >> sqrt(100)-sqrt(20)*sqrt(5)
  - >> exp(10)-exp(5)*exp(5)
  - >> exp(10)-exp(5)/exp(-5)



Attractive mathematics does not protect one from the rigors of digital computation

[J.H. Wilkinson, "von Neumann Lecture", SIAM Meeting, Boston, 1970]

Μοντέλο αριθμητικής

Αναπαράσταση αριθμών στον Η/Υ ώστε να διευκολύνεται η «διαχείρισή τους» - πεπερασμένη ακρίβεια
- Ακέραιοι
- Αναπαραστάσεις πραγματικών:
  - Fixed point και floating point arithmetic
- Αριθμητικές πράξεις, λογικές πράξεις, I/O, ...

- Κεντρικά θέματα
  - Αριθμοί κινητής υποδιαστολής (α.κ.υ.)
    - Αναπαράσταση, αριθμητική, διαχείριση
  - Σφάλματα στρογγύλευσης
    - Διάδοση, συσσώρευση
    - Πρόβλεψη, έλεγχος

Επιστημονική γραφή

- Για (μεγάλους ή μικρούς) αριθμούς συνηθίζεται η «Επιστημονική Γραφή Αριθμών» ή «εκθετική γραφή» δηλ. στη μορφή $a \times 10^e$.
- Π.χ. το **350** μπορεί να γραφτεί ως 3.5×10^2, ή 35×10^1, ή 350×10^0.
- Για να υπάρχει μοναδικότητα στην αναπαράσταση, συμφωνείται συνήθως να χρησιμοποιούμε *κανονκοποιημένη* (normalized) επιστημόνική γραφή:
  - Επιλέγεται $1 \le |a| < 10$
- Αν χρησιμοποιούμε κανονικοποιημένη αναπαράσταση, το μέγεθος του εκθέτη **e** καθορίζει άμεσα την τάξη μεγέθους του αριθμού.
- Αν ο εκθέτης είναι αρνητικός, ο αριθμός θα είναι μεταξύ **0** και **1** σε απόλυτη τιμή.
- Αντίστοιχα μπορούμε να σκεφτούμε και την αναπαράσταση α.κ.υ.

Στοιχεία για την α.κ.υ.

- Οι α.κ.υ. συνήθως γράφονται όπως και η «Επιστημονική

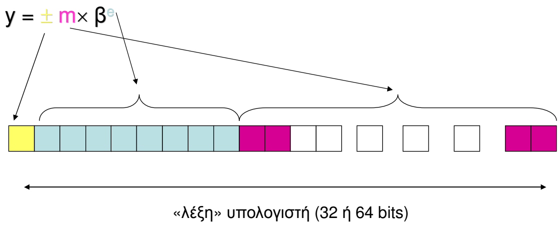
  $y = \pm m \times \beta^e$

  β: βάση

  e: εκθέτης, ακέραιος που αναπαρίσταται με $L_{εκθ}$ ψηφία στη βάση β, $e \in [e_{min}, e_{max}]$

  m = □.□□□□□□ --- □□ : ουρά, αναπαρίσταται σε μορφή (sign-magnitude) με **t** ψηφία (βάση β)
- Όλοι οι παραπάνω α.κ.υ. είναι εκ κατασκευής ρητοί
- Μπορούμε να συμβολίσουμε το σύστημα με $F(\beta, t, e_{min}, e_{max})$
  - Η έκφραση αυτή δεν δίνει όλα τα στοιχεία ....
  - Έχουν υπάρξει συστήματα με:
    - β = 2, 10, 16, ακόμα και 3 (υπολογιστές **Setun** στις ΕΣΣΔ, 1958) !!!! Το τριαδικό (ternary) σύστημα – **trits** αντί για **bits**!

α.κ.υ.: Επιστημονική γραφή σε πεπερασμένο χώρο



Βασική ιδιότητα: Κατανομή των α.κ.υ. (α.κ.υ. ως α.κ.ορντεόν)

- Οι αριθμοί δεν είναι όλοι ισοκατανεμημένοι μεταξύ

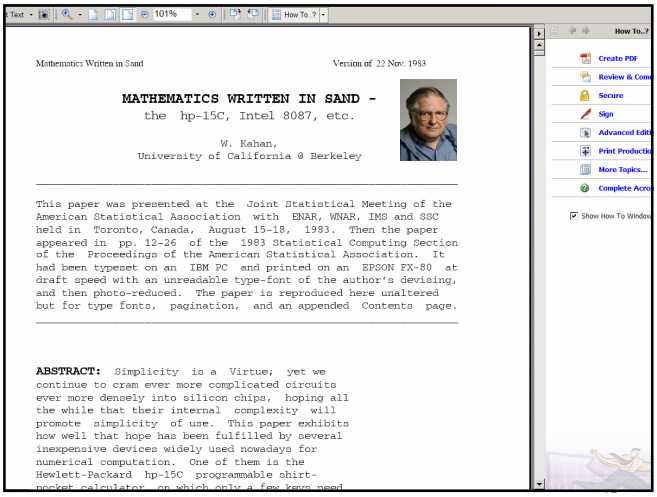
- Π.χ. αν

  $x_1 = m \times \beta^e, x_2 = (m+2^{-t+1}) \times \beta^e \Rightarrow x_2 - x_1 = 2^{-t+1} \beta^e$

  $y_1 = m \times \beta^{e+1}, y_2 = (m+2^{-t+1}) \times \beta^{e+1} \Rightarrow y_2 - y_1 = 2^{-t+1} \beta^{e+1}$
- ... αυτό επιτρέπει την αναπαράσταση μεγάλου εύρους τιμών

Ιστορικά στοιχεία

- Το **1955** η **IBM** εισήγαγε το μοντέλο **704** με βασικό αρχιτέκτονα τον **Gene Amdahl**;
  - Το **IBM-704** ήταν το πρώτο εμπορικό σύστημα με υλοποίησης α.κ.υ. στο υλικό
  - Επιδόσεις περί τα **5 kFLOPS.**
- Από τότε το μοντέλο των α.κ.υ. Βρίσκεται στο κέντρο της Επιστήμης και Τεχνολογίας των Υπολογιστών!
  - Πολλές ιδέες, πολλές αναπαραστάσεις
    - Πολύ μπλέξιμο!
- **… The simplest and best, though harder to attain, solution to the problem of environmental parameters is to standardize floating-point hardware, so that the values of the parameters become universal constants.**
  - **[Webb Miller, The Engineering of Numerical Software, 1984.]**



Mathematics Written in Sand — Version of 22 Nov 1983

MATHEMATICS WRITTEN IN SAND -
the hp-15C, Intel 8087, etc.

W. Kahan,
University of California @ Berkeley

This paper was presented at the Joint Statistical Meeting of the American Statistical Association with ENAR, WNAR, IMS and SSC held in Toronto, Canada, August 15-18, 1983. Then the paper appeared in pp. 12-26 of the 1983 Statistical Computing Section of the Proceedings of the American Statistical Association. It had been typeset on an IBM PC and printed on an EPSON FX-80 at draft speed with an unreadable type-font of the author's devising, and then photo-reduced. The paper is reproduced here unaltered but for type fonts, pagination, and an appended Contents page.

ABSTRACT: Simplicity is a Virtue; yet we continue to cram ever more complicated circuits ever more densely into silicon chips, hoping all the while that their internal complexity will promote simplicity of use. This paper exhibits how well that hope has been fulfilled by several inexpensive devices widely used nowadays for numerical computation. One of them is the Hewlett-Packard hp-15C programmable shirt-pocket calculator on which only a few keys need

Τυποποίηση

- Διαδικασία ξεκίνησε στις αρχές του **1980**
  - Για να αντιμετωπιστούν οι μεγάλες ασυμβατότητες μεταξύ διαφορετικών συστημάτων της εποχής στην αριθμητική τους.
- Τυποποίηση από το **Institute for Electrical and Electronic Engineers (IEEE)**
  - Πρότυπο **754 (1985)**
  - Σημαντική η συμβολή **W. Kahan (UC Berkeley), Intel, Apple, …**
  - Για «φανατικούς»: Δείτε την ιστοσελίδα του **Kahan** στο **UCB.**
- Έχει πλέον υιοθετηθεί από όλα τα σημαντικά υπολογιστικά συστήματα.
- Η τυποποίηση αφορά την αναπαράσταση ΚΑΙ ΠΟΛΛΑ ΑΛΛΑ (πράξεις, μετατροπές, …)
- Πειραματιστείτε στη http://babbage.cs.qc.edu/IEEE-754/32bit.html

Προβληματισμός στα GPUs

GPU Floating-Point Paranoia

Karl E. Hillesland
University of North Carolina at Chapel Hill *

Anselmo Lastra
University of North Carolina at Chapel Hill *

1 Introduction

Up until the late eighties, each computer vendor was left to develop their own conventions for floating-point computation as they saw fit. As a result, programmers needed to familiarize themselves with the peculiarities of each system in order to write effective software and evaluate numerical error. In 1987, a standard was established for floating-point computation to alleviate this problem, and CPU vendors now design to this standard [IEEE 1987].

Today there is an interest in the use of graphics processing units, or GPUs, for non-graphics applications such as scientific computing. GPUs have floating-point representations similar to, and sometimes matching, the IEEE standard. However, we have found that GPUs do not adhere to IEEE standards for floating-point operations, nor do they give the information necessary to establish bounds on error for these operations. Another complication is that this behavior seems to be in a constant state of flux due to the depen-

Operation	R300/arbfp	NV30/fp30
Addition	[-1.000, 0.000]	[-1.000, 0.000]
Subtraction	[-1.000, 1.000]	[-0.750, 0.750]
Multiplication	[-0.989, 0.125]	[-0.782, 0.625]
Division	[-2.869, 0.094]	[-1.199, 1.375]

Table 1: Floating-Point Error in ULPs (Units in Last Place). Note that the R300 has a 16 bit significand, whereas the NV30 has 23 bits. Therefore one ULP on an R300 is equivalent to 2^7 ULPs on an NV30. Division is implemented by a combination of reciprical and multiply on these systems. Cg version 1.2.1. ATI driver 6.14.10.6444. NVIDIA driver 56.72.

Schryer [Schryer 1981]. By testing all combinations of these numbers, we include all the test cases in Paranoia, as well as cases that push the limits of round-off error and cases where the most work must be performed, such as extensive carry propagation. Table 1 gives results for some example systems.

IEEE floating-point standard

- http://en.wikipedia.org/wiki/IEEE_754-1985
- Το πρότυπο IEEE για δυαδική α.κ.υ. **(IEEE 754)** είναι το ευρύτερα χρησιμοποιούμενο πρότυπο για α.κ.υ. και υλοποιείται σε πάρα πολλά **CPU** και **FPU**.
- Στο πρότυπο ορίζονται:
  - **Formats** αναπαράστασης για α.κ.υ. (και «αρνητικό μηδέν» και «υποκανονικοποιημένοι αριθμοί»)
  - Ειδικές τιμές (άπειρα, **NaN**)
  - Σύνολο πράξεων που μπορούν να εκτελεστούν επί α.κ.υ.
  - **(4)** τρόποι στρογγύλευσης **(rounding modes)**
  - **(5)** εξαιρέσεις (πότε συμβαίνουν, τι αποτέλεσμα έχουν)
- IEEE 754-2008 **(previously known as *IEEE 754r*) was published in** August 2008 **and is a significant revision to, and replaces, the** IEEE 754-1985 floating point standard**. The revision extended the previous standard where it was necessary, added decimal arithmetic and formats, tightened up certain areas of the original standard which were left undefined, and merged in** IEEE 854 **(the radix-independent floating-point standard).** http://en.wikipedia.org/wiki/IEEE_754
- Το πρότυπο **IEEE 754-2008** στοχεύει ώστε μια υλοποίηση που το ικανοποιεί πλήρως να μπορεί να γραφεί εξ ολοκλήρου στο λογισμικό, ή στο υλικό, ή σε οποιοδήποτε συνδυασμό τους.
- Τμήματα του προτύπου μπορεί να υλοποιούνται σε επίπεδο Λ/Σ, βιβλιοθηκών μεταφραστή ή γενικών βιβλιοθηκών.
- Η συμμόρφωση με το πρότυπο υπολογίζεται στο σύνολο του υλικού και λογισμικού που το ικανοποιεί και όχι τμηματικά.

Χαρακτηριστικά κωδικοποίησης

- δυαδική λέξη με **w bits (β=2)** που παριστά α.κ.υ.
- **1 bit** πρόσημο ουράς
- **L bits** εκθέτη
- **t bits** ουρά
  - ουρά: □.□□□□ --- □
    - κανονικοποιημένη αναπαράσταση => πρώτο ψηφίο μη μηδενικό
      - ✓ εξοικονομείται μια θέση «κρύβοντας» το πρώτο **bit** και αποθηκεύοντας μόνον τα **t-1 bits** μετά την υποδιαστολή – τακτική κρυμμένου **bit**
      - ✓ πρέπει να «ζωντανεύει» όταν γίνονται πράξεις ή αποκωδικοποίηση
  - **Παράδειγμα:** «θετικός» αριθμός κινητής υποδ. με

Βάση β=2,
Κανονικοποιημένη ουρά: m = 1.□□□□□□□ – t bits

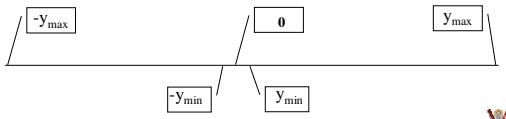
	Single	Single-Ext	Double	Double-Ext	Quad-Precision
e(max)	+127	1023	+1023	+16383	+16383
e(min)	-126	1022	-1022	-16382	-16382
πόλωση P	+127	+1023	+1023	+16383	+16383
(#bits ουράς) = t	24	≥ 32	53	≥ 64	113
μήκος α.κ.υ.	32	≥ 43	64	80	128
bits s	1	1	1	1	1
bits e	8	11	11	15	15
bits για κλάσμα	23	≥ 32	52	64	112

Πεπερασμένη αναπαράσταση

- Ακρότατες (θετικές) τιμές

$$y_{min} = m_{min} \times \beta^{e_{min}}$$
$$y_{max} = m_{max} \times \beta^{e_{max}}$$

Σύνολο $G = \{0\} \cup \{x \in R: y_{min} \le |x| \le y_{max}\}$



Ακρότατες τιμές

- Οι ακρότατες τιμές που μπορούν να αναπαρασταθούν σε κανονικοποιημένη μορφή είναι:

$$y_{min} = (1.0\text{---}0) \times \beta^{e_{min}} = \beta^{e_{min}}$$

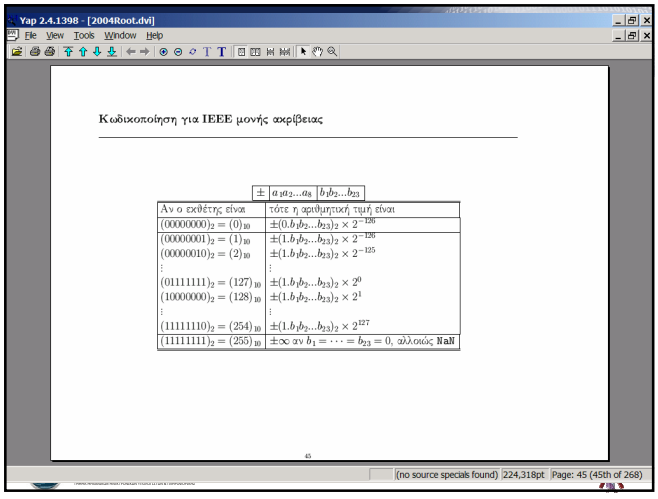
$$y_{max} = (1.1\text{---}1) \times \beta^{e_{max}}$$
$$= (\beta\text{-}1)\,(1+\beta^{-1} + \ldots + \beta^{-t+1}) \times \beta^{e_{max}}$$
$$= (\beta\text{-}1) \times \beta^{e_{max}-t+1}\,(\beta^t\text{-}1)/(\beta\text{-}1) = \beta^{e_{max}-t+1}\,(\beta^t\text{-}1)$$

$$y_{min} \le |y| \le y_{max}$$

- Στην **IEEE** διπλής ακρίβειας, για παράδειγμα,
- $y_{min} = 2^{-1022} \approx 2.2251 \times 10^{-308}$. Στη **MATLAB** `realmin`
- $Y_{max} = 2^{1024-53}\,(2^{53}\text{-}1) \approx 1.7977 \times 10^{+308}$. Στη **MATLAB** `realmax`

Παράδειγμα σε μικρογραφία

- w=12, L=7, t=4+1
- S EEE EEEE MMMM
- $P = 2^{L-1} - 1 = 63$
- $F = (-1)^S \times 2^{e-P} \times (1.MMM)_2$
- Ο εκθέτης φαίνεται να κυμαίνεται μεταξύ (000000)_2 και (1111111)_2 αλλά αυτές οι ακραίες τιμές εξοτοιούνται για να συμβολίσουν ειδικούς αριθμούς και δεν χρησιμοποιείται η πραγματική τους τιμή.
- $e_{max} = (2^L\text{-}1)\text{-}1\text{-}P = 63$
- $e_{min} = 0+1-P = -62$
- E=1111111 και MMMM=0000 αναπαριστούν τα ± Inf
- E=1111111 και MMMM μη μηδενικό αναπαριστούν NaN
- E=0000000 είναι το 0 και οι μη κανονικοποιημένοι αριθμοί $2^{emin} \times (0.MMMM)_2$
- Inf = 0 111 1111 0000 $= (70)_{16}$
- realmax=0 111 1110 1111 $= (7ef)_{16}$ $= 2^{63} \times (1+1\text{-}2^{-5})$ $= 2^{64} \times (1\text{-}1/64)$
- 0 = 0 000 0000
- realmin=0 000 0001 0000 $= (010)_{16}$ $= 2^{1-P} \times (1+0)$ $= 2^{-62}$
- Max υποκ. =0 000 0000 1111 $= (00f)_{16}$ $= 2^{1-P} \times (1\text{-}2^{-t})$ $= 2^{-62} \times (1\text{-}2^{-7})$
- Min υποκ.=0 000 0000 0001 $= (001)_{16}$ $= 2^{1-P} \times (0+2^{-(t-1)})$ $= 2^{-62} \times (0+2^{-6}) = 2^{-66}$
- -Inf = 1 111 1111 0000 $= (ff0)_{16}$
- $\varepsilon_M = 2^{-(1-t)} = 1/16$ $= 0\ 011\ 1011\ 0000$ $= (3b0)_{16}$



$\pm$ | $a_1 a_2 \ldots a_8$ | $b_1 b_2 \ldots b_{23}$

Αν ο εκθέτης είναι	τότε η αριθμητική τιμή είναι
$(00000000)_2 = (0)_{10}$	$\pm(0.b_1b_2\ldots b_{23})_2 \times 2^{-126}$
$(00000001)_2 = (1)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2 \times 2^{-126}$
$(00000010)_2 = (2)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2 \times 2^{-125}$
⋮	⋮
$(01111111)_2 = (127)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2 \times 2^{0}$
$(10000000)_2 = (128)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2 \times 2^{1}$
⋮	⋮
$(11111110)_2 = (254)_{10}$	$\pm(1.b_1b_2\ldots b_{23})_2 \times 2^{127}$
$(11111111)_2 = (255)_{10}$	$\pm\infty$ αν $b_1 = \cdots = b_{23} = 0$, αλλοιώς NaN

MATLAB 7.* single precision

Maximum and Minimum Single-Precision Values. The MATLAB functions realmax and realmin, when called with the argument 'single', return the maximum and minimum values that you can represent with the single data type:

```
str = 'The range for single is:\n\t%g to %g and\n\t %g to %g'; sprintf(str, -realmax('single'), -realmin('single'), ... realmin('single'), realmax('single'))
ans = The range for single is: -3.40282e+038 to -1.17549e-038 and 1.17549e-038 to 3.40282e+038
```

Numbers larger than realmax('single') or smaller than -realmax('single') are assigned the values of positive and negative infinity respectively:

realmax('single') + .0001e+038 ans = Inf -realmax('single') - .0001e+038 ans = -Inf

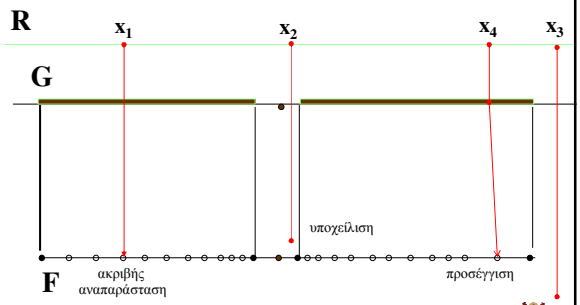
Creating Single-Precision Data. Because MATLAB stores numeric data as a double by default, you need to use the single conversion function to create a single-precision number:
x = single(25.783);

Προσεγγίσεις → Στρογγύλευση + ειδικοί αριθμοί

- Οι α.κ.υ. είναι ένα (μικροσκοπικό) πεπερασμένο υποσύνολο των πραγματικών.
- Πρέπει να τους χρησιμοποιήσουμε για να αναπαραστήσουμε όλους τους αριθμούς!
- προσεγγίσεις – **quantization**

- Απεικόνιση πραγματικών στο F: συνάρτηση fl: R → F
- Τρείς περιπτώσεις
  - $x \in F$
    $\Rightarrow y = fl(x)$
  - $x \notin G$
    $\Rightarrow$ υπερχείλιση ή υποχείλιση
  - $x \in G$ και $x \notin F$
    $\Rightarrow$ προσέγγιση του x με στοιχείο $fl(x) \in F$

Απεικόνιση πραγματικών σε ακυ



Μη κανονικοποιημένοι α.κ.υ.

- Συνήθως οι α.κ.υ. IEEE είναι κανονικοποιημένοι
- Το πρότυπο IEEE επιτρέπει το αποτέλεσμα πράξεων μεταξύ α.κ.υ. να είναι α.κ.υ. που είναι μικρότεροι του ελάχιστου κανονικοποιημένου (realmin). Αυτοί είναι πολύ μικροί α.κ.υ. Που δεν είναι κανονικοποιημένοι.
  - Πρόκειται για α.κ.υ. που έχουν πρώτο ψηφίο 0 (μηδέν), δηλ.
  - $0.* * * \text{---} * \times 2^{e_{min}}$
  - Δηλ. $\{(0.0\text{---}1)_2, (0.0\text{---}10)_2, \ldots, (0.1\text{----}1)_2\} \times 2^{e_{min}}$
  - Επομένως, ο ελάχιστος (μη κανονικοποιημένος) θετικός αναπαραστήσιμος α.κ.υ. θα είναι ο

$$0.0\text{---}1 \times 2^{e_{min}}$$

  - Σε α.κ.υ. IEEE διπλής ακρίβειας, $\approx 4.9407 \times 10^{-324}$
  - Αν το διαιρέσουμε με y>1 επιστρέφεται 0.

Έψιλον της μηχανής και ulp

- **eps** μηχανής: Είναι η απόσταση του 1 από τον αμέσως επόμενο α.κ.υ., έστω 1^+ δηλ. $\varepsilon_M = 1^+ - 1$
- Στο σύστημα α.κ.υ. IEEE $\varepsilon_M = 2^{-t+1} = 2u$
  - ε_M διπλής ακρίβειας = $2^{-52} \approx 10^{-16}$
  - Το ε_M υπάρχει χρησιμοποιείται σε αλγορίθμους γιατί δείχνει τη διακριτότητα του συστήματος α.κ.υ. Δείτε π.χ. τη συνάρτηση `rank` της **MATLAB**.
  - Το ε_M ορίστηκε βάσει του 1. Η γενίκευσή του σε μεγαλύτερους αριθμούς ονομάζεται **ulp (units in the last place)**
  - Αν $x = m \times 2^E$ τότε $ulp(x) = \varepsilon_M \times 2^E$