Κεφάλαιο 3

Αριθμητική για υπολογιστές

Αριθμητική για υπολογιστές

- Λειτουργίες (πράξεις) σε ακεραίους
  - Πρόσθεση και αφαίρεση
  - Πολλαπλασιασμός και διαίρεση
  - Χειρισμός της υπερχείλισης
- Πραγματικοί αριθμοί κινητής υποδιαστολής (floating-point)
  - Αναπαράσταση και λειτουργίες (πράξεις)

Ακέραια πρόσθεση

- Παράδειγμα: 7 + 6

		(0)		*(0)*		*(1)*		*(1)*		*(0)*			*(Κρατούμενα)*
…		0		0		0		1		1		1	
…		0		0		0		1		1		0	
…	(0)	0	(0)	0	(0)	1	(1)	1	(1)	0	(0)	1	

- Υπερχείλιση (overflow) αν το αποτέλεσμα είναι εκτός του εύρους των τιμών
  - Πρόσθεση ετερόσημων τελεστέων, όχι υπερχείλιση
  - Πρόσθεση θετικών τελεστέων
    - Υπερχείλιση αν το πρόσημο του αποτελέσματος είναι 1
  - Πρόσθεση αρνητικών τελεστέων
    - Υπερχείλιση αν το πρόσημο του αποτελέσματος είναι 0

Ακέραια αφαίρεση

- Πρόσθεση του αντιθέτου του δεύτερου τελεστέου
- Παράδειγμα: 7 – 6 = 7 + (–6)

+7:	0000 0000 … 0000 0111
–6:	1111 1111 … 1111 1010
+1:	0000 0000 … 0000 0001

- Υπερχείλιση αν το αποτέλεσμα είναι εκτός του εύρους των τιμών
  - Αφαίρεση δύο θετικών ή δύο αρνητικών, όχι υπερχείλιση
  - Αφαίρεση θετικού από αρνητικό τελεστέο
    - Υπερχείλιση αν το πρόσημο του αποτελέσματος είναι 0
  - Αφαίρεση αρνητικού από θετικό τελεστέο
    - Υπερχείλιση αν το πρόσημο του αποτελέσματος είναι 1

Χειρισμός της υπερχείλισης

- Μερικές γλώσσες (π.χ., C) αγνοούν την υπερχείλιση
  - Χρησιμοποιούν τις εντολές του MIPS `addu`, `addui`, `subu`
- Άλλες γλώσσες (π.χ., Ada, Fortran) απαιτούν τη δημιουργία μιας εξαίρεσης
  - Χρησιμοποιούν τις εντολές του MIPS `add`, `addi`, `sub`
  - Στην υπερχείλιση, καλείται ο χειριστής εξαιρέσεων
    - Αποθήκευση του PC στο μετρητή προγράμματος εξαιρέσεων (exception program counter – EPC)
    - Άλμα στην προκαθορισμένη διεύθυνση του χειριστή
    - Η εντολή `mfc0` (move from coprocessor reg) μπορεί να ανακτήσει την τιμή του EPC, για να γίνει επιστροφή μετά τη διορθωτική ενέργεια

Πολλαπλασιασμός

- Ξεκινάμε με τον πολ/σμό μεγάλου μήκους

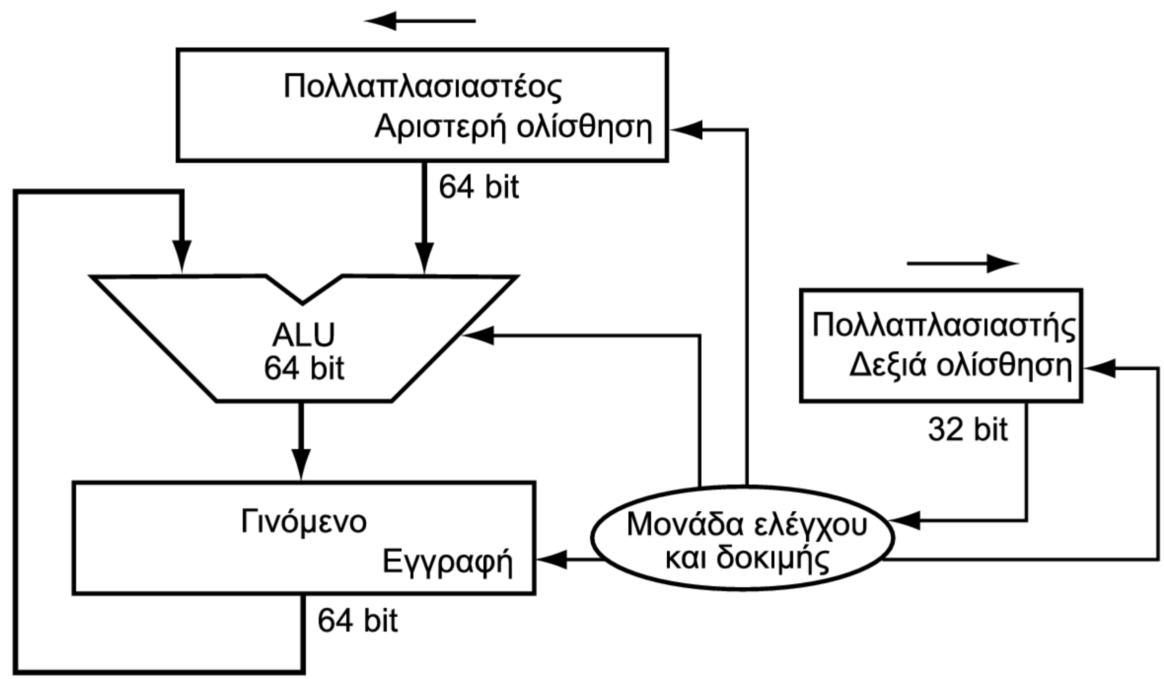
πολλαπλασιαστέος

πολλαπλασιαστής

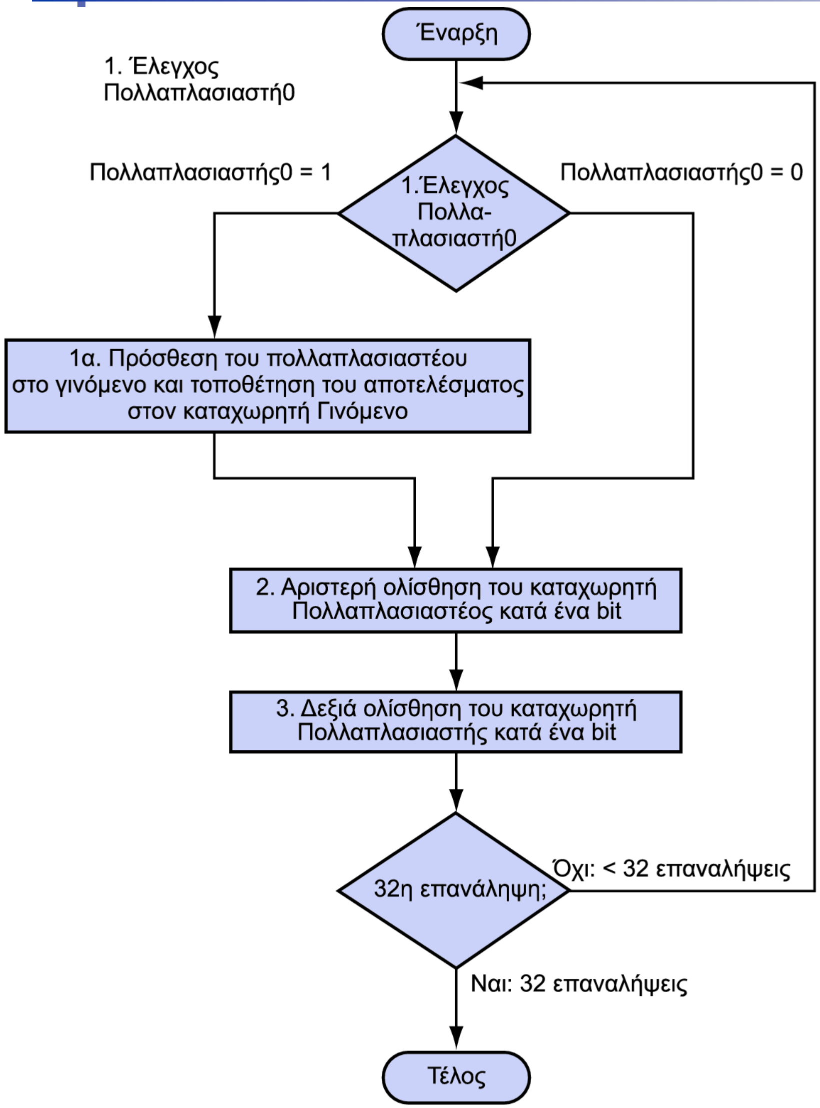
```
      1000
×     1001
----------
      1000
     0000
    0000
   1000
----------
   1001000
```

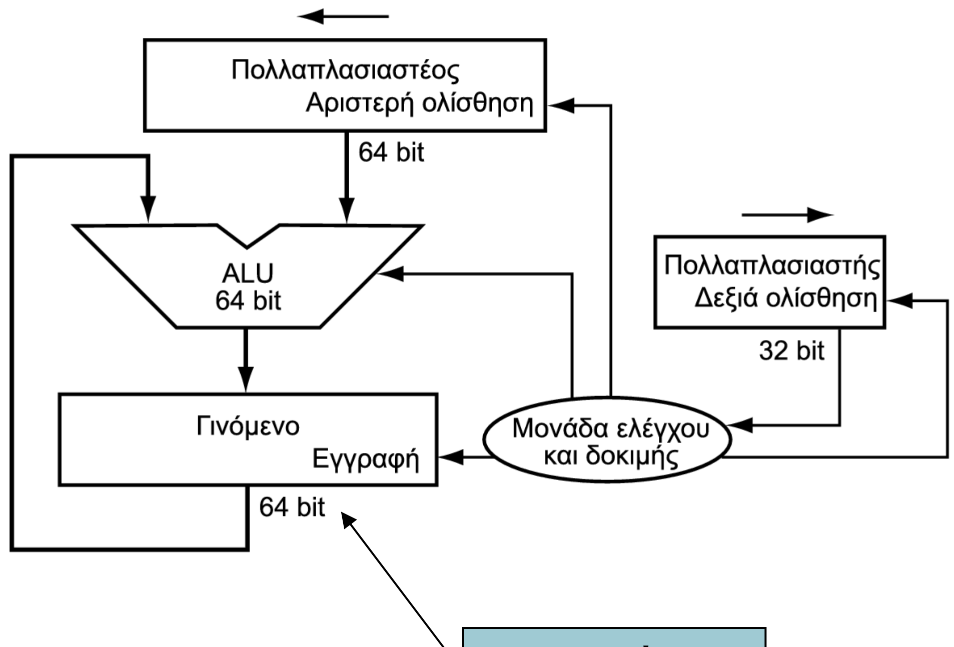
γινόμενο

Το μήκος του γινομένου είναι το άθροισμα των μηκών των τελεστέων



Υλικό πολλαπλασιασμού

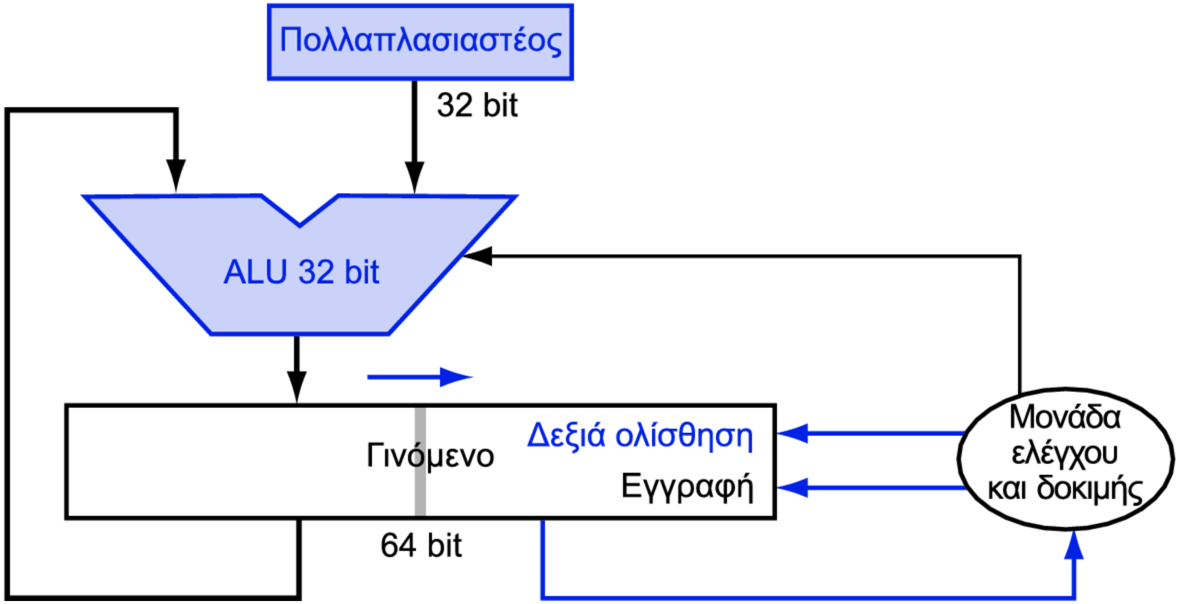




Αρχικά 0

Βελτιστοποιημένος πολλαπλασιαστής

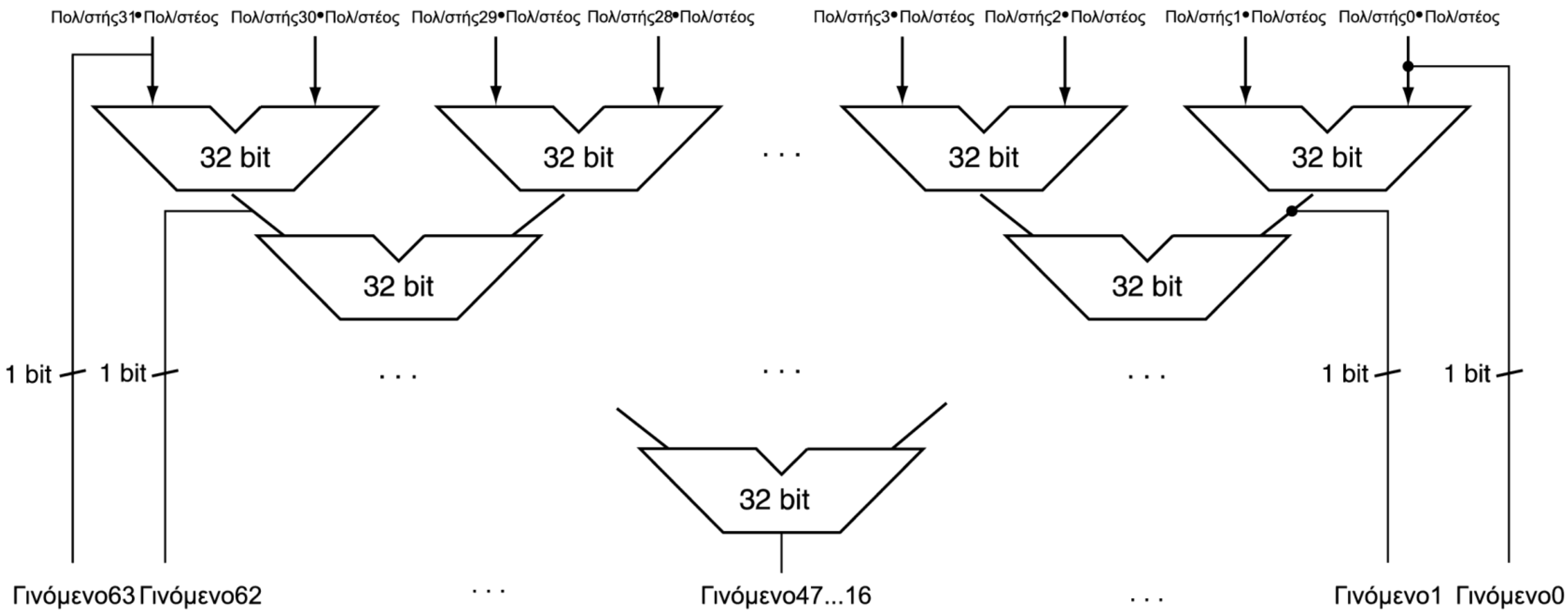
- Εκτέλεση βημάτων παράλληλα: πρόσθεση/ολίσθηση



- Ένας κύκλος ανά πρόσθεση μερικού γινομένου
  - Είναι εντάξει, αν η συχνότητα εμφάνισης του πολλαπλασιασμού είναι χαμηλή

Ταχύτερος πολλαπλασιαστής

- Χρησιμοποιεί πολλούς αθροιστές
  - Συμβιβασμός κόστους/απόδοσης

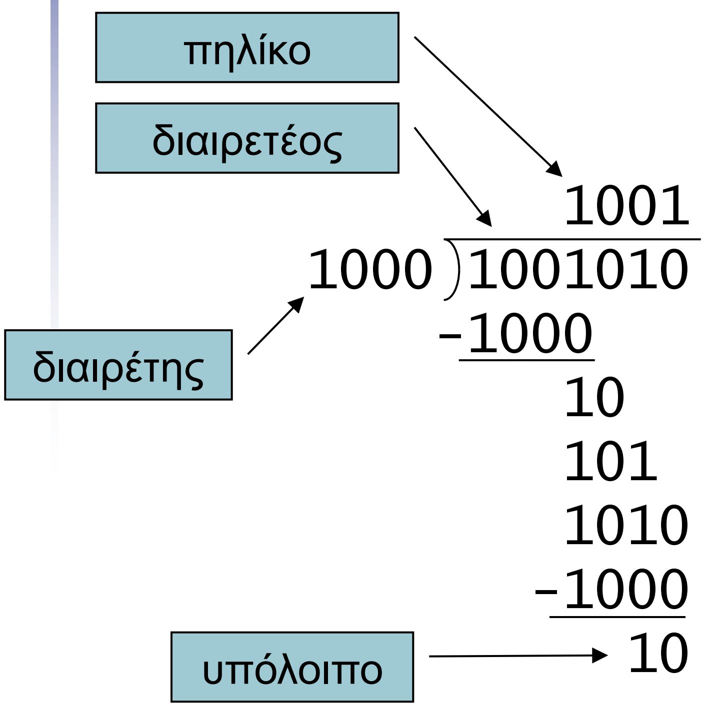


- Μπορεί να υλοποιηθεί με διοχέτευση (pipeline)
  - Πολλοί πολλαπλασιασμοί εκτελούνται παράλληλα

Πολλαπλασιασμός στον MIPS

- Δύο καταχωρητές των 32 bit για το γινόμενο
  - HI: τα περισσότερο σημαντικά 32 bit
  - LO: τα λιγότερο σημαντικά 32 bit
- Εντολές
  - `mult rs, rt  /  multu rs, rt`
    - γινόμενο των 64 bit στους HI/LO
  - `mfhi rd  /  mflo rd`
    - Μεταφορά από (move from) του HI/LO στον rd
    - Μπορούμε να ελέγξουμε την τιμή του HI για να δούμε αν το γινόμενο ξεπερνά τα 32 bit
  - `mul rd, rs, rt`
    - Τα λιγότερο σημαντικά 32 bit του γινομένου –> rd

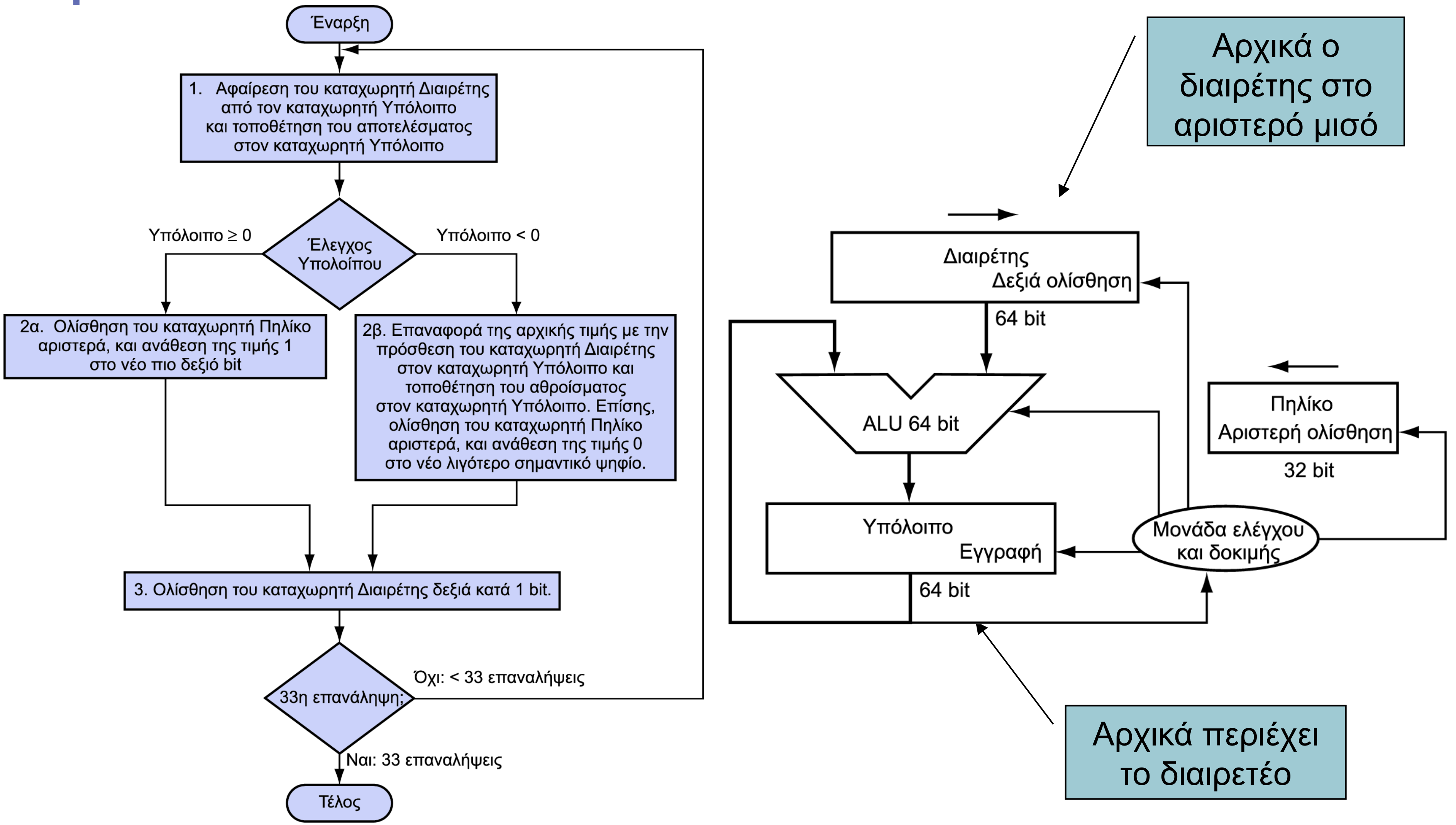
Διαίρεση



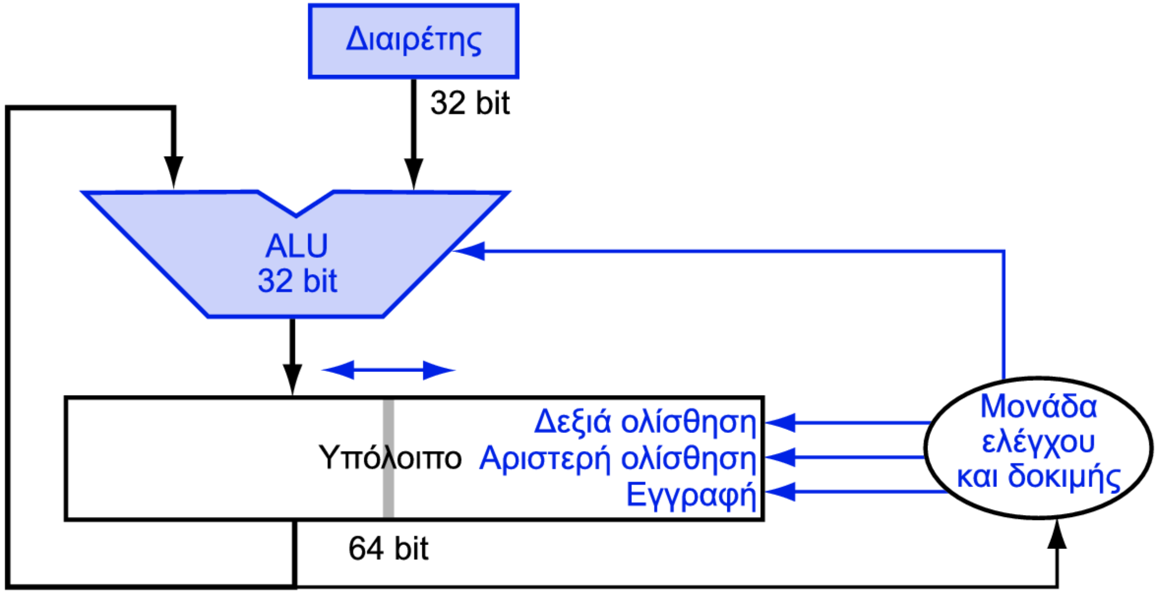
τελεστέοι των *n* bit δίνουν πηλίκο και υπόλοιπο των *n* bit

- Έλεγχος για μηδενικό διαιρέτη
- Διαίρεση μεγάλου μήκους
  - Αν διαιρέτης ≤ από τα bit του διαιρετέου
    - 1 bit στο πηλίκο, αφαίρεση
  - Αλλιώς
    - 0 bit στο πηλίκο, κατέβασμα του επόμενου bit του διαιρετέου
- Διαίρεση με επαναφορά (restoring division)
  - Κάνε την αφαίρεση και αν το υπόλοιπο γίνει < 0, πρόσθεσε πίσω το διαιρέτη
- Προσημασμένη διαίρεση
  - Κάνε τη διαίρεση με τις απόλυτες τιμές
  - Ρύθμισε το πρόσημο του πηλίκου και του υπολοίπου όπως απαιτείται

Υλικό διαίρεσης



Βελτιστοποιημένος διαιρέτης



- Ένας κύκλος για κάθε αφαίρεση μερικού υπολοίπου
- Μοιάζει πολύ με πολλαπλασιαστή!
  - Το ίδιο υλικό μπορεί να χρησιμοποιηθεί και για τις δύο πράξεις

Ταχύτερη διαίρεση

- Δεν μπορεί να χρησιμοποιηθεί παράλληλο υλικό όπως στον πολλαπλασιαστή
  - Η αφαίρεση εκτελείται υπό συνθήκη, ανάλογα με το πρόσημο του υπολοίπου
- Ταχύτεροι διαιρέτες (π.χ. διαίρεση SRT) δημιουργούν πολλά bit του πηλίκου σε κάθε βήμα
  - Και πάλι απαιτούνται πολλά βήματα

Διαίρεση στο MIPS

- Χρήση των καταχωρητών HI/LO για το αποτέλεσμα
  - HI: υπόλοιπο 32 bit
  - LO: πηλίκο 32 bit
- Εντολές
  - `div rs, rt` / `divu rs, rt`
  - Όχι έλεγχος για υπερχείλιση ή διαίρεση με το 0
    - Το λογισμικό πρέπει να εκτελεί τους ελέγχους αν αυτό απαιτείται
  - Χρήση των `mfhi`, `mflo` για προσπέλαση του αποτελέσματος

Κινητή υποδιαστολή

- Αναπαράσταση για μη ακεραίους αριθμούς
  - Περιλαμβάνει και πολύ μικρούς και πολύ μεγάλους αριθμούς
- Όπως η επιστημονική σημειογραφία (scientific notation)
  - -2.34×10^{56} ← κανονικοποιημένος
  - $+0.002 \times 10^{-4}$ ← μη κανονικοποιημένος
  - $+987.02 \times 10^{9}$ ← μη κανονικοποιημένος
- Σε δυαδικό
  - $\pm 1.xxxxxxx_2 \times 2^{yyyy}$
- Οι τύποι `float` και `double` της C

Πρότυπο κινητής υποδιαστολής

- Ορίζεται από το IEEE Std 754-1985
- Αναπτύχθηκε ως λύση στην απόκλιση των αναπαραστάσεων
  - Ζητήματα φορητότητας (portability) για τον κώδικα επιστημονικών εφαρμογών
- Πλέον είναι σχεδόν οικουμενικά αποδεκτό
- Δύο αναπαραστάσεις κινητής υποδιαστολής (floating point)
  - Απλή ακρίβεια – single precision (32 bit)
  - Διπλή ακρίβεια – double precision (64 bit)

Μορφή κινητής υποδιαστολής IEEE

	single: 8 bit double: 11 bit	single: 23 bit double: 52 bit
S	Εκθέτης	Κλάσμα

$$x = (-1)^S \times (1 + \text{Κλάσμα}) \times 2^{(\text{Εκθέτης} - \text{Πόλωση})}$$

- Εκθέτης (exponent) – Κλάσμα (fraction)
- S: bit προσήμου (0 ⇒ μη αρνητικός, 1 ⇒ αρνητικός)
- Κανονικοποίηση του σημαντικού (significand): $1.0 \leq |\text{significand}| < 2.0$
  - Έχει πάντα ένα αρχικό bit 1 πριν την υποδιαστολή, και συνεπώς δε χρειάζεται ρητή αναπαράστασή του («κρυμμένο» bit)
  - Το σημαντικό (significand) είναι το κλάσμα (fraction) μαζί με το κρυμμένο “1”
- Εκθέτης: αναπαράσταση «με υπέρβαση» (excess): πραγματικός εκθέτης + πόλωση (bias)
  - Εγγυάται ότι ο εκθέτης είναι απρόσημος
  - Απλή ακρίβεια: Πόλωση = 127 – Διπλή ακρίβεια: Πόλωση = 1023

IEEE 754 (απλή ακρίβεια)

s	e(8)	f(23)
S	Εκθέτης	Κλάσμα

Συνθήκη	Τιμή
$0 < e < 255$	$(-1)^s \times 2^{e-127} \times 1,f$ (κανονικοποιημένη μορφή)
$e = 0; f \neq 0$	$(-1)^s \times 2^{-126} \times 0,f$ (υπο-κανονική μορφή)
$e = 0; f = 0$	$(-1)^s \times 0.0$ (μηδέν)
$s = 0; e = 255; f = 0$	+INF (θετικό άπειρο)
$s = 1; e = 255; f = 0$	-INF (αρνητικό άπειρο)
$s = x; e = 255; f \neq 0$	NaN (μη-αριθμός)

IEEE 754 (διπλή ακρίβεια)

s	e(11)	f(52)
S	Εκθέτης	Κλάσμα

Συνθήκη	Τιμή
$0 < e < 2047$	$(-1)^{s} \times 2^{e-1023} \times 1.f$ (κανονικοποιημένη μορφή)
$e = 0; f \neq 0$	$(-1)^{s} \times 2^{-1022} \times 0.f$ (υπο-κανονική μορφή)
$e = 0; f = 0$	$(-1)^{s} \times 0.0$ (μηδέν)
$s = 0; e = 2047; f = 0$	+INF (θετικό άπειρο)
$s = 1; e = 2047; f = 0$	-INF (αρνητικό άπειρο)

Μετατροπές ΑΚΥ

- 2 → 10 http://www.ajdesigner.com/fl_ieee_754_word/ieee_32_bit_word.php
- 10 → 2
- http://babbage.cs.qc.cuny.edu/IEEE-754.old/Decimal.html

Εύρος απλής ακρίβειας

- Οι εκθέτες 00000000 και 11111111 δεσμεύονται
- Μικρότερη τιμή
  - Εκθέτης: 00000001
    $\Rightarrow$ πραγματικός εκθέτης = 1 – 127 = –126
  - Κλάσμα: 000…00 $\Rightarrow$ σημαντικό = 1.0
  - $\pm 1.0 \times 2^{-126} \approx \pm 1.2 \times 10^{-38}$
- Μεγαλύτερη τιμή
  - Εκθέτης: 11111110
    $\Rightarrow$ πραγματικός εκθέτης = 254 – 127 = +127
  - Κλάσμα: 111…11 $\Rightarrow$ σημαντικό ≈ 2.0
  - $\pm 2.0 \times 2^{+127} \approx \pm 3.4 \times 10^{+38}$

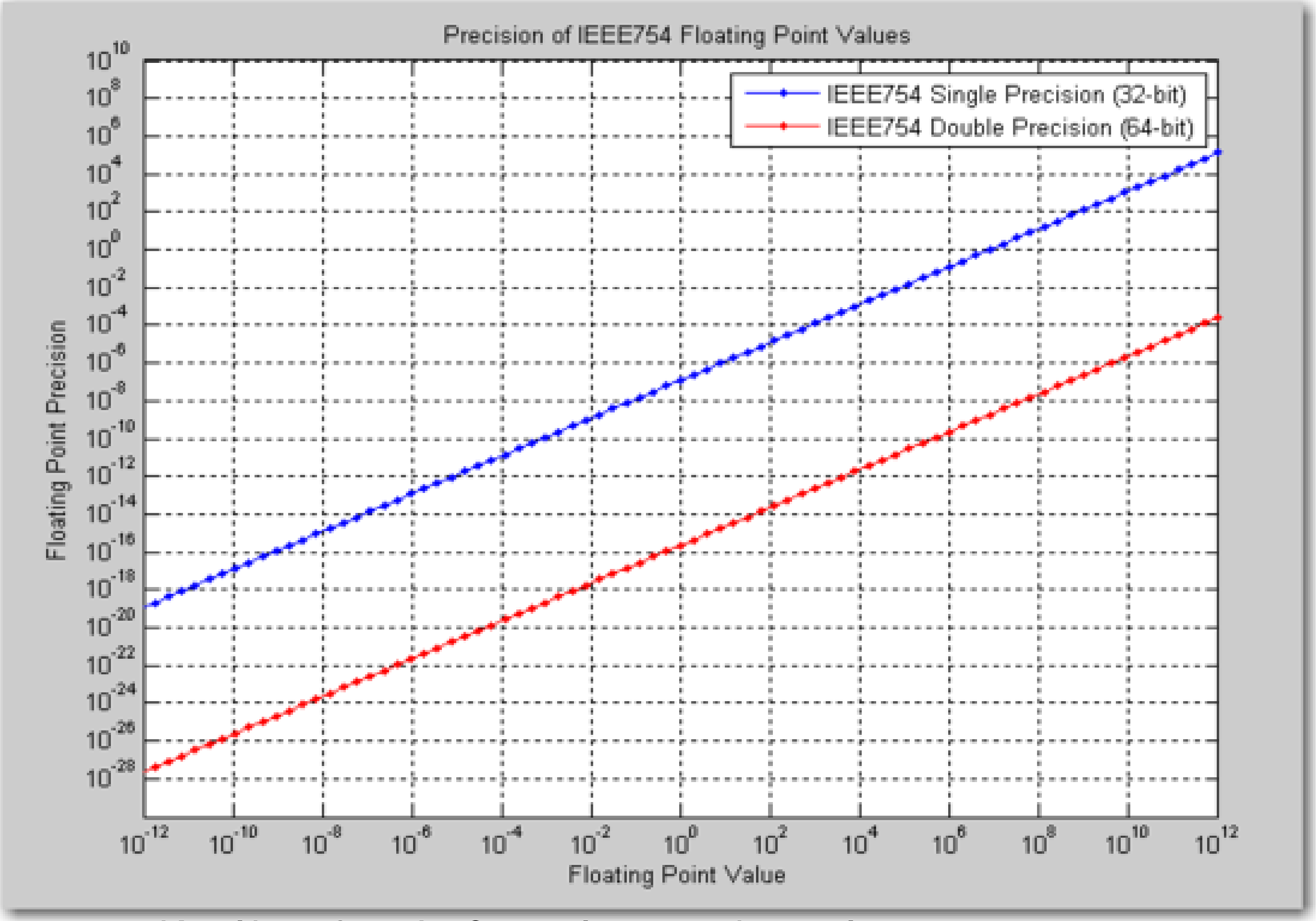
Εύρος διπλής ακρίβειας

- Οι εκθέτες 0000…00 και 1111…11 δεσμεύονται
- Μικρότερη τιμή
  - Εκθέτης: 00000000001
    $\Rightarrow$ πραγματικός = 1 – 1023 = –1022
  - Κλάσμα: 000…00 $\Rightarrow$ σημαντικό = 1.0
  - $\pm 1.0 \times 2^{-1022} \approx \pm 2.2 \times 10^{-308}$
- Μεγαλύτερη τιμή
  - Εκθέτης: 11111111110
    $\Rightarrow$ πραγματικός εκθέτης = 2046 – 1023 = +1023
  - Κλάσμα: 111…11 $\Rightarrow$ σημαντικό ≈ 2.0
  - $\pm 2.0 \times 2^{+1023} \approx \pm 1.8 \times 10^{+308}$

Ακρίβεια κινητής υποδιαστολής

- Σχετική ακρίβεια
  - Όλα τα bit του κλάσματος είναι σημαντικά
  - Απλή: περίπου 2^{-23}
    - Ισοδύναμο με $23 \times \log_{10}2 \approx 23 \times 0.3 \approx 6$ δεκαδικά ψηφία ακρίβειας
  - Διπλή: περίπου 2^{-52}
    - Ισοδύναμο με $52 \times \log_{10}2 \approx 52 \times 0.3 \approx 16$ δεκαδικά ψηφία ακρίβειας

Ακρίβεια ΑΚΥ



Παράδειγμα κινητής υποδιαστολής

- Αναπαράσταση του –0.75
  - $-0.75 = (-1)^1 \times 1.1_2 \times 2^{-1}$
  - S = 1
  - Κλάσμα = $1000\ldots00_2$
  - Εκθέτης = –1 + Πόλωση
    - Απλή: –1 + 127 = 126 = 01111110_2
    - Διπλή: –1 + 1023 = 1022 = 01111111110_2
- Απλή: 1011111101000…00
- Διπλή: 1011111111101000…00

Παράδειγμα κινητής υποδιαστολής

- Ποιος αριθμός αναπαρίσταται από τον απλής ακρίβειας κινητής υποδιαστολής αριθμό;

  11000000101000…00

  - S = 1
  - Κλάσμα = $01000\ldots00_2$
  - Εκθέτης = $10000001_2 = 129$
- $x = (-1)^1 \times (1 + .01_2) \times 2^{(129 - 127)}$

  $= (-1) \times 1.25 \times 2^2$

  $= -5.0$

Μη κανονικοποιημένοι (denormals)

- Εκθέτης = 000...0 $\Rightarrow$ το «κρυμμένο» bit είναι 0

$$x = (-1)^S \times (0 + \text{Κλάσμα}) \times 2^{-\text{Πόλωση}}$$

- Μικρότεροι από τους κανονικοποιημένους
  - επιτρέπουν βαθμιαία ανεπάρκεια (gradual underflow), με μειούμενη ακρίβεια
- Denormal με κλάσμα = 000...0

$$x = (-1)^S \times (0 + 0) \times 2^{-\text{Πόλωση}} = \pm\, 0.0$$

Δύο αναπαραστάσεις του 0.0!

Άπειρο και μη-αριθμοί (NaN)

- Εκθέτης = 111...1, Κλάσμα = 000...0
  - ±Άπειρο
  - Μπορεί να χρησιμοποιηθεί σε επόμενους υπολογισμούς, για αποφυγή της ανάγκης του ελέγχου υπερχείλισης
- Εκθέτης = 111...1, Κλάσμα ≠ 000...0
  - Όχι αριθμός (Not-a-Number – NaN)
  - Δείχνει ένα άκυρο ή απροσδιόριστο αποτέλεσμα
    - π.χ., 0.0 / 0.0
  - Μπορεί να χρησιμοποιηθεί σε επόμενους υπολογισμούς

Πρόσθεση κινητής υποδιαστολής

- Ένα δεκαδικό παράδειγμα με 4 ψηφία
  - $9.999 \times 10^{1} + 1.610 \times 10^{-1}$
- 1. Ευθυγράμμιση υποδιαστολών
  - Ολίσθηση αριθμού με το μικρότερο εκθέτη
  - $9.999 \times 10^{1} + 0.016 \times 10^{1}$
- 2. Πρόσθεση σημαντικών
  - $9.999 \times 10^{1} + 0.016 \times 10^{1} = 10.015 \times 10^{1}$
- 3. Κανονικοποίηση αποτελέσματος & έλεγχος υπερχείλισης/ανεπάρκειας
  - 1.0015×10^{2}
- 4. Στρογγυλοποίηση και επανακανονικοποιήση αν είναι απαραίτητο
  - 1.002×10^{2}

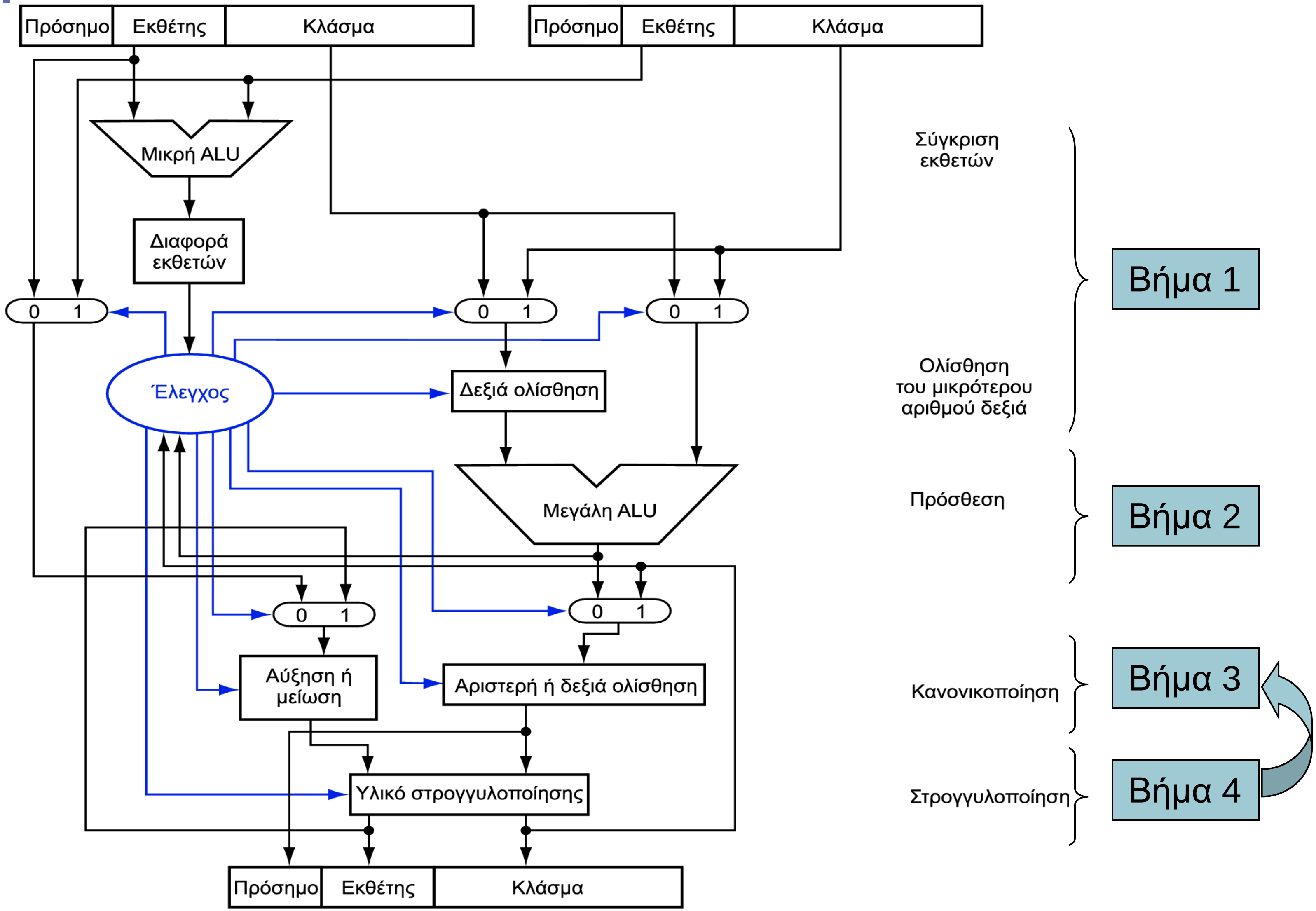
Πρόσθεση κινητής υποδιαστολής

- Τώρα ένα δυαδικό παράδειγμα με 4 ψηφία
  - $1.000_2 \times 2^{-1} + -1.110_2 \times 2^{-2}$ (0.5 + –0.4375)
- 1. Ευθυγράμμιση υποδιαστολών
  - Ολίσθηση αριθμού με το μικρότερο εκθέτη
  - $1.000_2 \times 2^{-1} + -0.111_2 \times 2^{-1}$
- 2. Πρόσθεση συντελεστών
  - $1.000_2 \times 2^{-1} + -0.111_2 \times 2^{-1} = 0.001_2 \times 2^{-1}$
- 3. Κανονικοποίηση αποτελέσματος και έλεγχος υπερχείλισης/ανεπάρκειας
  - $1.000_2 \times 2^{-4}$, χωρίς υπερχείλιση/ανεπάρκεια
- 4. Στρογγυλοποίηση και επανακανονικοποιήση αν είναι απαραίτητο
  - $1.000_2 \times 2^{-4}$ (καμία αλλαγή) $= 0.0625$

Υλικό αθροιστή κιν. υποδ.

- Πολύ πιο πολύπλοκο από του ακέραιου αθροιστή
- Για να γίνει σε έναν κύκλο πρέπει να έχει πολύ μεγάλη διάρκεια
  - Πολύ μεγαλύτερη από τις ακέραιες λειτουργίες
  - Το πιο αργό ρολόι θα επιβάρυνε όλες τις εντολές
- Ο αθροιστής κινητής υποδιαστολής συνήθως παίρνει πολλούς κύκλους
  - Μπορεί να υπολοποιηθεί με διοχέτευση

Υλικό αθροιστή κιν.υποδ.



Υλικό αριθμητικής κιν. υποδ.

- Ο πολλαπλασιαστής ΚΥ έχει παρόμοια πολυπλοκότητα με τον αθροιστή ΚΥ
  - Αλλά χρησιμοποιεί πολλαπλασιαστή για τα σημαντικά αντί για αθροιστή
- Το υλικό αριθμητικής κιν. υποδ. συνήθως εκτελεί
  - Πρόσθεση, αφαίρεση, πολλαπλασιασμό, διαίρεση, αντίστροφο, τετραγωνική ρίζα
  - Μετατροπή ΚΥ $\leftrightarrow$ ακέραιο
- Οι λειτουργίες συνήθως διαρκούν πολλούς κύκλους
  - Μπορούν να υπολοποιηθούν με διοχέτευση

Εντολές ΚΥ στο MIPS

- Το υλικό ΚΥ είναι ο συνεπεξεργαστής (coprocessor) 1
  - Επιπρόσθετος επεξεργαστής που επεκτείνει την αρχιτεκτονική συνόλου εντολών
- Ξεχωριστοί καταχωρητές ΚΥ
  - 32 απλής ακρίβειας: \$f0, \$f1, … \$f31
  - Ζευγάρια για διπλή ακρίβεια: \$f0/\$f1, \$f2/\$f3, …
    - Η έκδοση 2 του συνόλου εντολών MIPS υποστηρίζει 32 × 64 bit καταχωρητές ΚΥ
- Εντολές ΚΥ επενεργούν μόνο σε καταχωρητές ΚΥ
  - Γενικά τα προγράμματα δεν εκτελούν ακέραιες πράξεις σε δεδομένα ΚΥ, ή αντίστροφα
  - Περισσότεροι καταχωρητές με ελάχιστη επίδραση στο μέγεθος του κώδικα
- Εντολές φόρτωσης και αποθήκευσης ΚΥ
  - `lwc1`, `ldc1`, `swc1`, `sdc1`
    - π.χ., `ldc1 $f8, 32($sp)`

Εντολές ΚΥ στον MIPS

- Αριθμητική απλής ακρίβειας
  - `add.s`, `sub.s`, `mul.s`, div.s
    - π.χ., `add.s $f0, $f1, $f6`
- Αριθμητική διπλής ακρίβειας
  - `add.d`, `sub.d`, `mul.d`, `div.d`
    - π.χ., `mul.d $f4, $f4, $f6`
- Σύγκριση απλής και διπλής ακρίβειας
  - `c.`*`xx`*`.s`, `c.`*`xx`*`.d` (*xx* είναι eq, `lt`, `le`, …)
  - Δίνει τη τιμή 1 ή 0 σε bit κωδικών συνθήκης ΚΥ (FP condition-code bit)
    - π.χ. `c.lt.s $f3, $f4`
- Διακλάδωση σε αληθή ή ψευδή κωδικό συνθήκης ΚΥ
  - `bc1t`, `bc1f`
    - π.χ., `bc1t TargetLabel`

Παραδειγμα ΚΥ: βαθμοί °F σε °C

- Κώδικας C:

```
float f2c (float fahr) {
  return ((5.0/9.0)*(fahr - 32.0));
}
```

  - `fahr` στον \$f12, αποτέλεσμα στον \$f0, οι σταθερές στο χώρο της καθολικής μνήμης

- Μεταγλωττισμένος κώδικας MIPS:

```
f2c: lwc1   $f16, const5($gp)
     lwc2   $f18, const9($gp)
     div.s  $f16, $f16, $f18
     lwc1   $f18, const32($gp)
     sub.s  $f18, $f12, $f18
     mul.s  $f0,  $f16, $f18
     jr     $ra
```

Ακριβής αριθμητική

- Το IEEE Std 754 καθορίζει πρόσθετο έλεγχο της στρογγυλοποίησης
  - Επιπλέον bit ακρίβειας (guard, round, sticky)
  - Επιλογή τρόπων στρογγυλοποίησης (rounding modes)
  - Επιτρέπει στον προγραμματιστή να ρυθμίσει με λεπτομέρεια την αριθμητική συμπεριφορά ενός υπολογισμού
- Δεν υλοποιούν όλες τις επιλογές όλες οι μονάδες ΚΥ
  - Οι περισσότερες γλώσσες προγραμματισμού και βιβλιοθήκες ΚΥ χρησιμοποιούν απλώς τις προκαθορισμένες λειτουργίες
- Συμβιβασμός μεταξύ πολυπλοκότητας του υλικού, απόδοσης, και απαιτήσεων της αγοράς

Διερμηνεία των δεδομένων

ΓΕΝΙΚΗ εικόνα

- Τα bit δεν έχουν έμφυτη σημασία
  - Η διερμηνεία εξαρτάται από τις εντολές που εφαρμόζονται
- Αναπαράσταση των αριθμών στους υπολογιστές
  - Πεπερασμένο εύρος και ακρίβεια
  - Πρέπει να λαμβάνονται υπόψη στα προγράμματα

Δεξιά ολίσθηση και διαίρεση

- Η αριστερή ολίσθηση κατά *i* θέσεις πολλαπλασιάζει έναν ακέραιο με 2^i
- Η δεξιά ολίσθηση διαιρεί με το 2^i ;
  - Μόνο σε απρόσημους ακεραίους
- Για προσημασμένους ακεραίους
  - Αριθμητική δεξιά ολίσθηση: επανάληψη του προσήμου
  - π.χ., –5 / 4
    - 11111011_2 >> 2 = 11111110_2 = –2
    - Στρογγυλοποιεί προς το –∞
  - σύγκριση 11111011_2 >>> 2 = 00111110_2 = +62

Ποιος νοιάζεται για την ακρίβεια ΚΥ;

- Σημαντική για επιστημονικό κώδικα
- Το σφάλμα της διαίρεσης ΚΥ του Intel Pentium (FDIV bug)
  - Η αγορά αναμένει ακρίβεια
  - Δείτε Colwell, *The Pentium Chronicles*

Συμπερασματικές παρατηρήσεις

- Οι αρχιτεκτονικές συνόλου εντολών υποστηρίζουν αριθμητική
  - Προσημασμένων και απρόσημων ακεραίων
  - Κινητής υποδιαστολής για τους πραγματικούς
- Πεπερασμένο εύρος και ακρίβεια
  - Οι λειτουργίες μπορεί να οδηγήσουν σε υπερχείλιση (overflow) και ανεπάρκεια (underflow)
- Αρχιτεκτονική συνόλου εντολών MIPS
  - Εντολές πυρήνα: οι 54 πιο συχνά χρησιμοποιούμενες
    - 100% του SPECINT, 97% του SPECFP
  - Άλλες εντολές: λιγότερο συχνές