Κεφάλαιο 3

Αριθμητική για υπολογιστές

‘Το ελάχιστο θέλησα,
και με τιμώρησαν με το πολύ.’
-- Οδ. Ελύτης

Νταχράν, Σαουδική Αραβία: 25/2/1991

Ιρακινος πύραυλος Scud **σκοτώνει 28** και τραυματίζει 98 στρατιώτες των ΗΠΑ

Σφάλμα λογισμικού στο σύστημα Patriot

Ο χρόνος υπολογιζόταν σε δέκατα του sec

Πολλαπλασιάστηκε επί 1/10 για να έρθει σε sec

1/10 = 0,0001 1001 1001 1001 1001 100 (24 bit)

Σφάλμα: $0{,}1100\ 1100 \times 2^{-23} \approx 9.5 \times 10^{-8}$ s

Σωρευμένο σφάλμα σε 100 ώρες:

$\approx 9{,}5 \times 10^{-8} \times 100 \times 60 \times 60 \times 10 = 0{,}34$ s

Απόσταση Scud = (0,34 s) × (1676 m/s) ≈ **570 m**

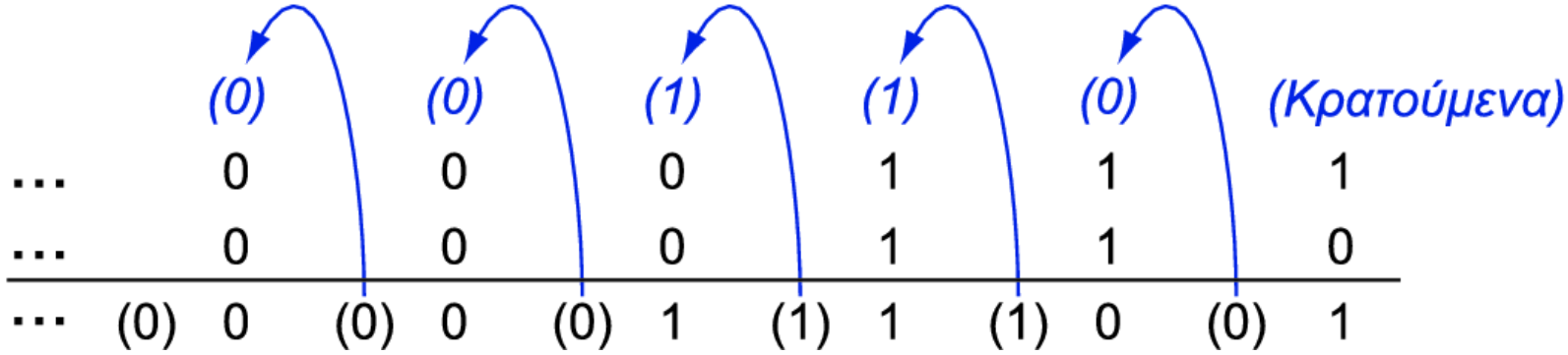
http://www.ima.umn.edu/~arnold/disasters/disasters.html

Αριθμητική για υπολογιστές

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

Ακέραια πρόσθεση

- Παράδειγμα: 7 + 6



- 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

Αφαίρεση ακεραίων

- 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

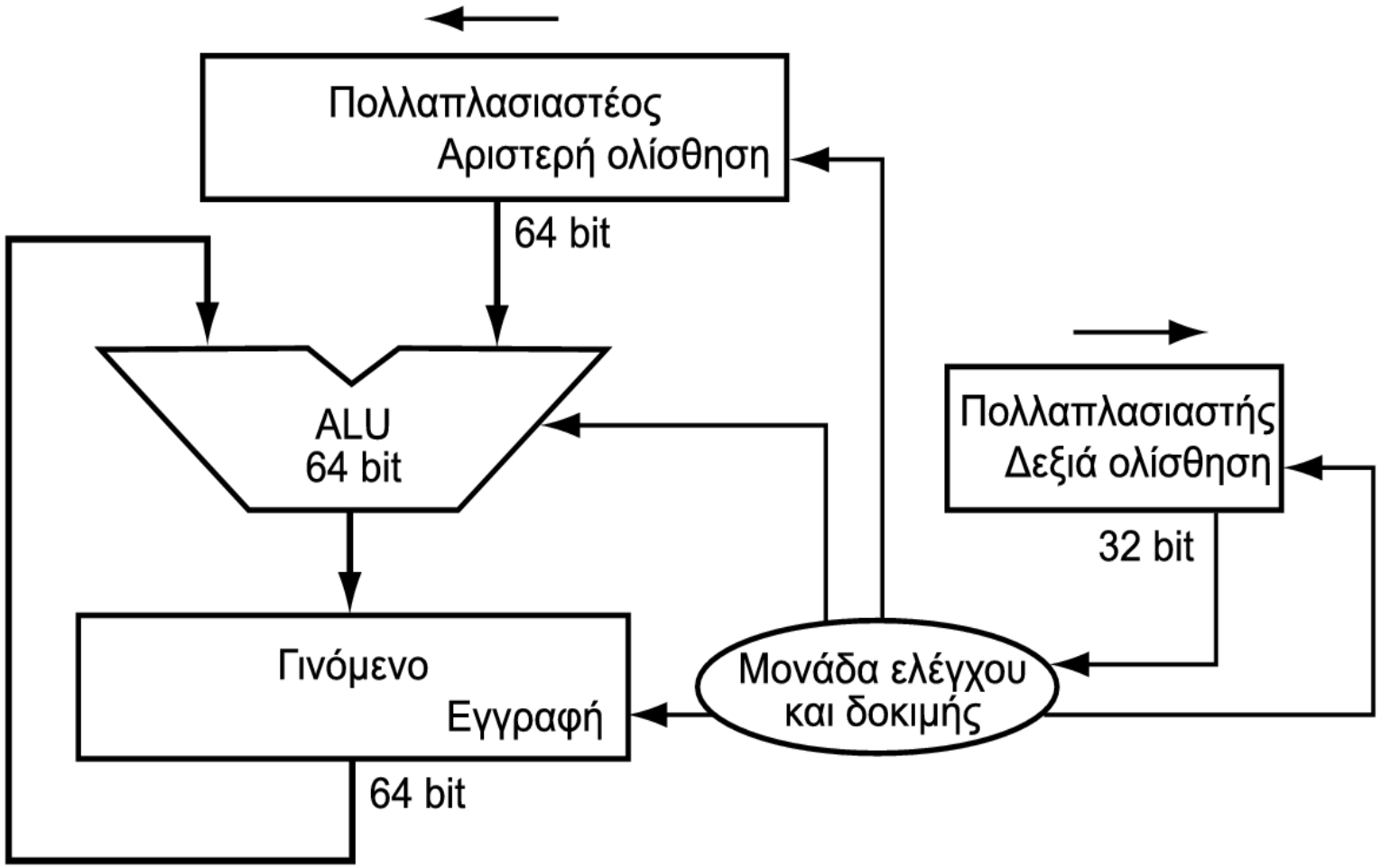
πολλαπλασιαστέος

πολλαπλασιαστής

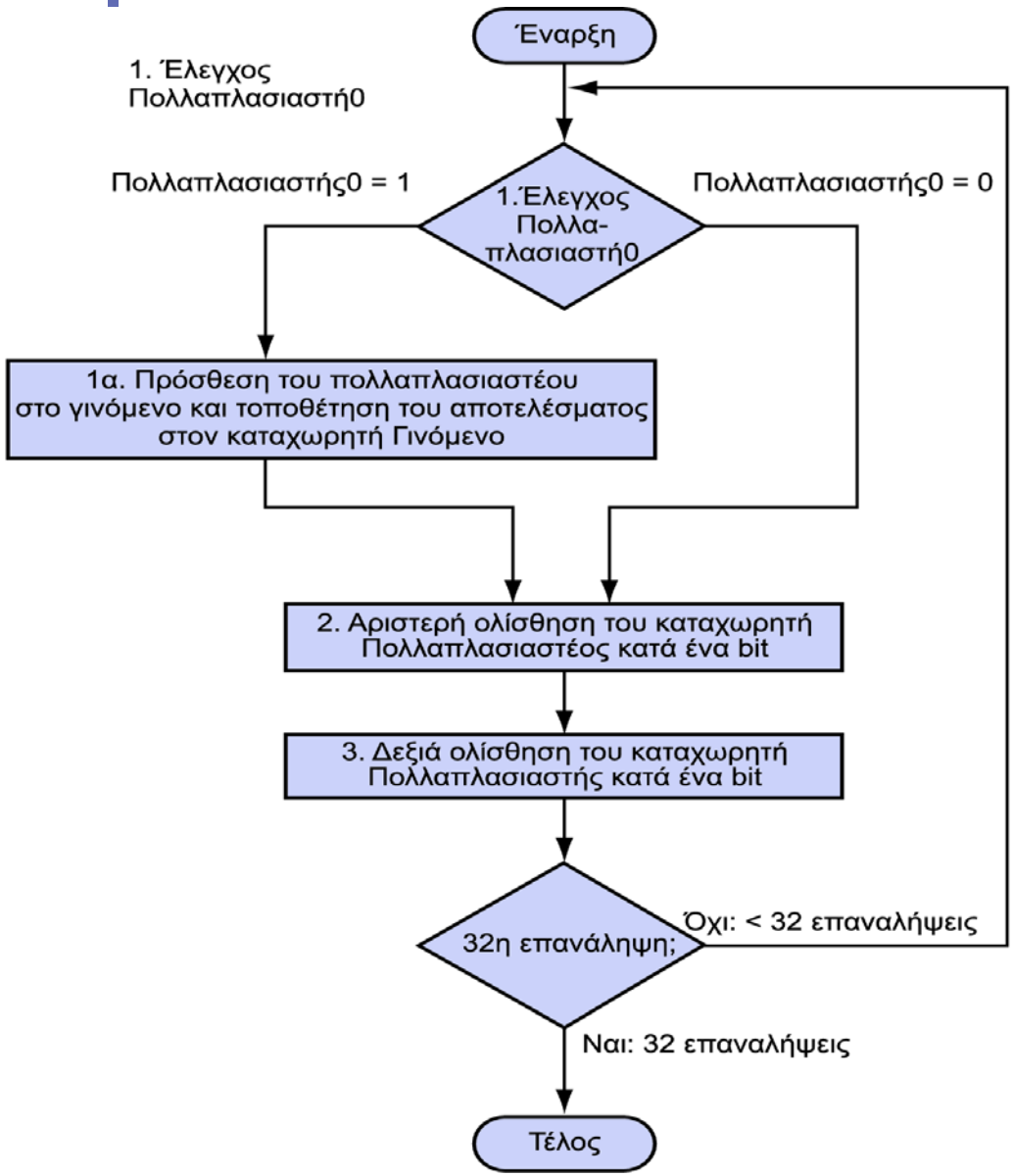
γινόμενο

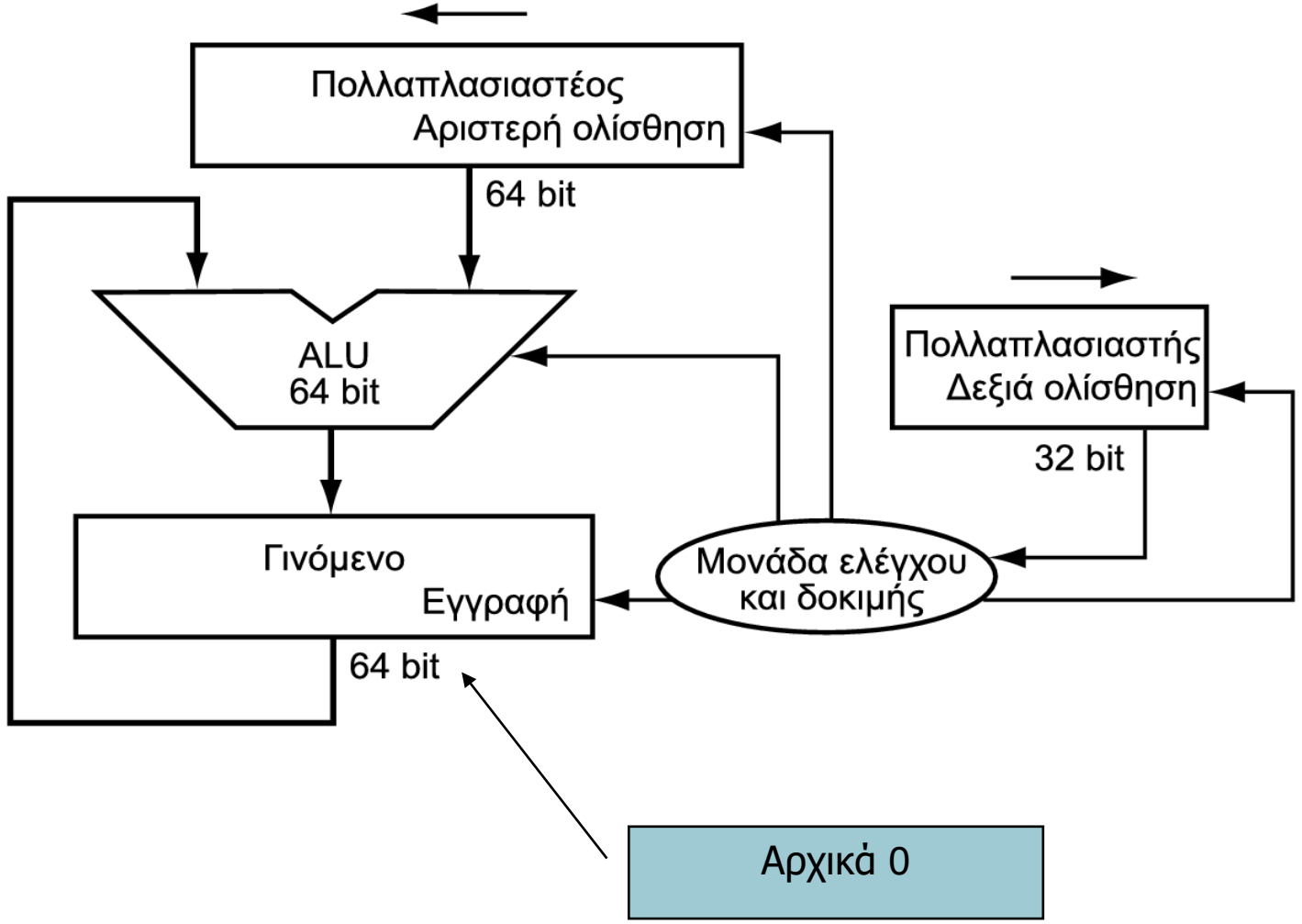
```
  1000
  1001
 ------
  1000
 0000
0000
1000
------
1001000
```

Το μήκος του γινομένου είναι το άθροισμα των μηκών των τελεστέων

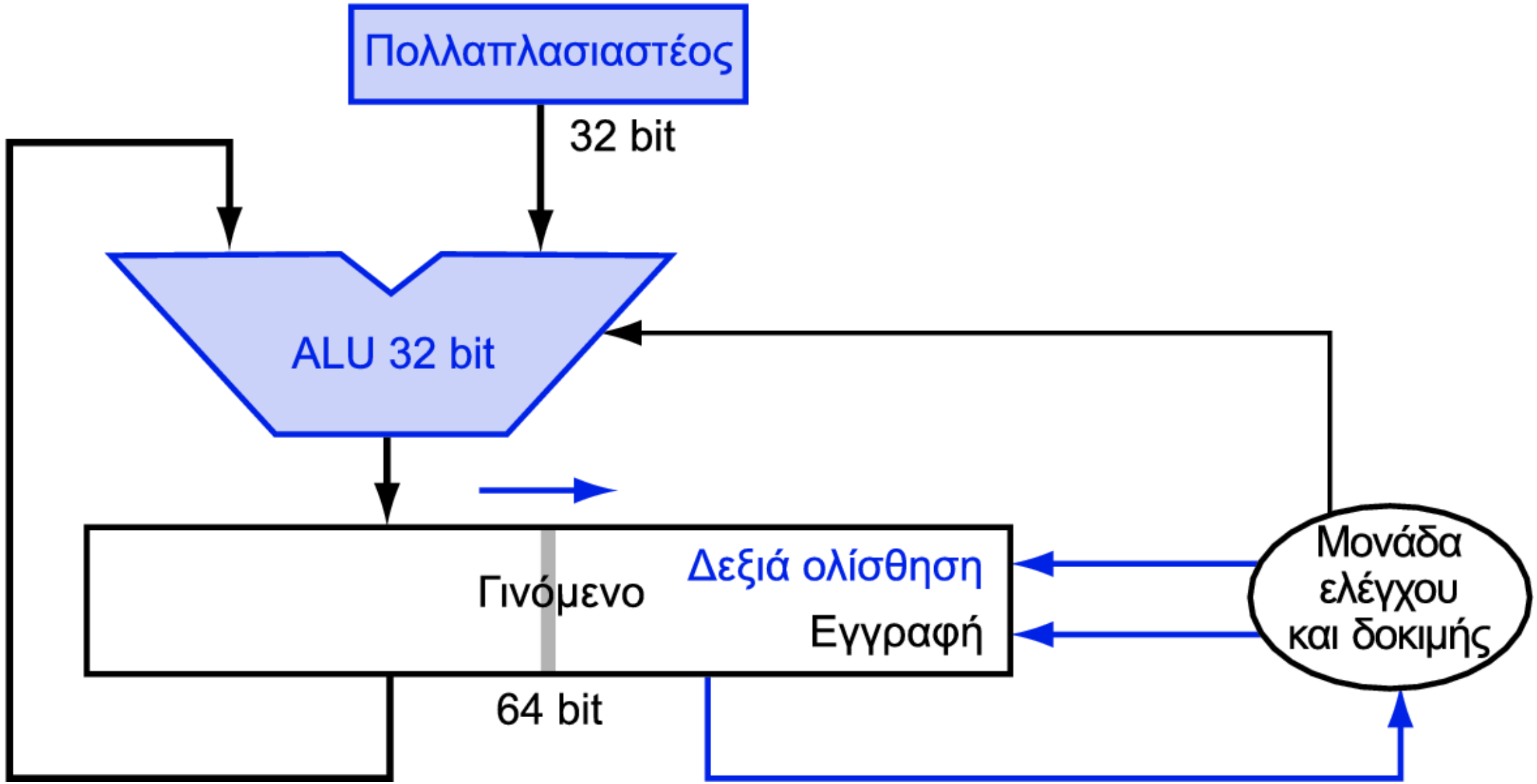


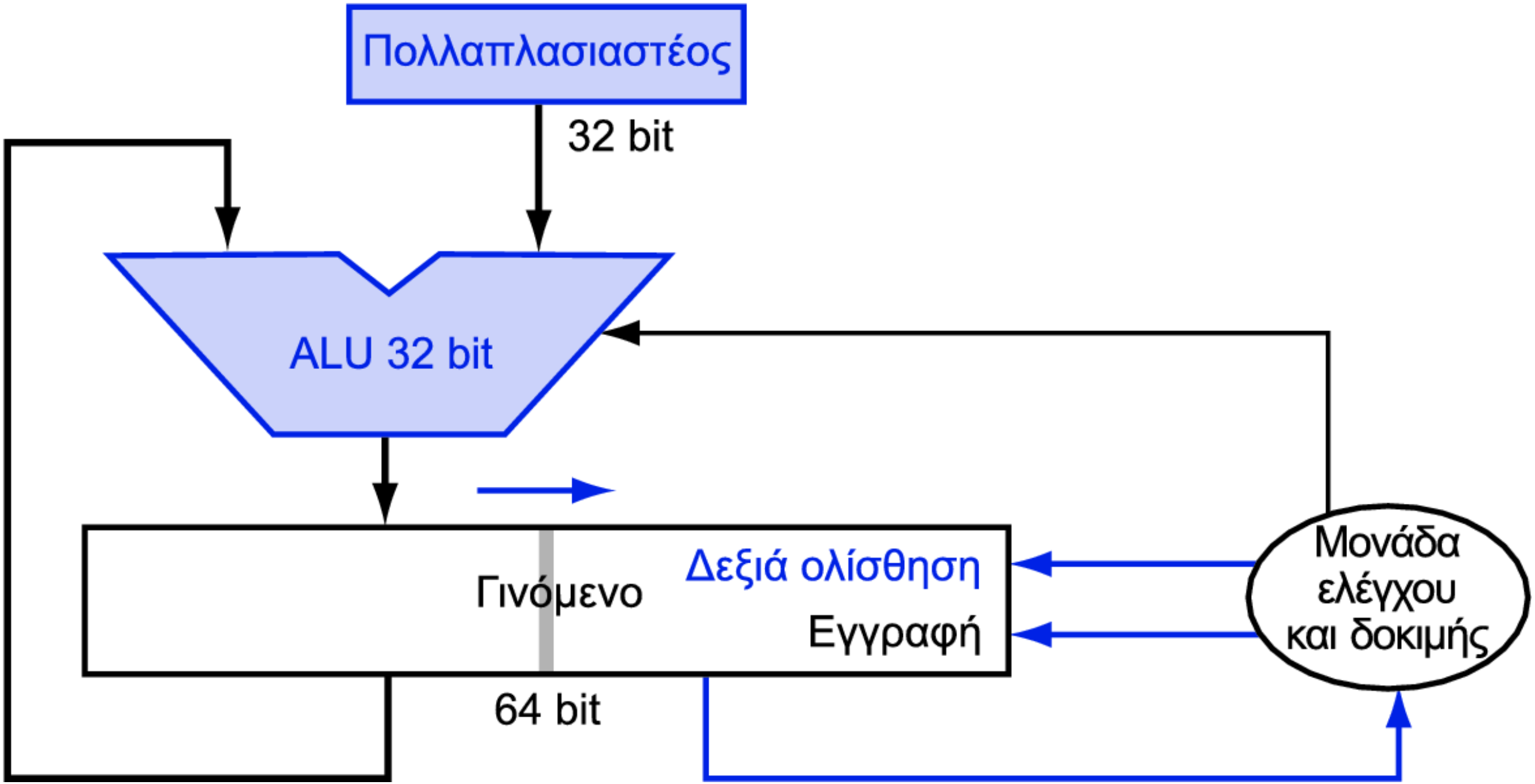
Υλικό πολλαπλασιασμού





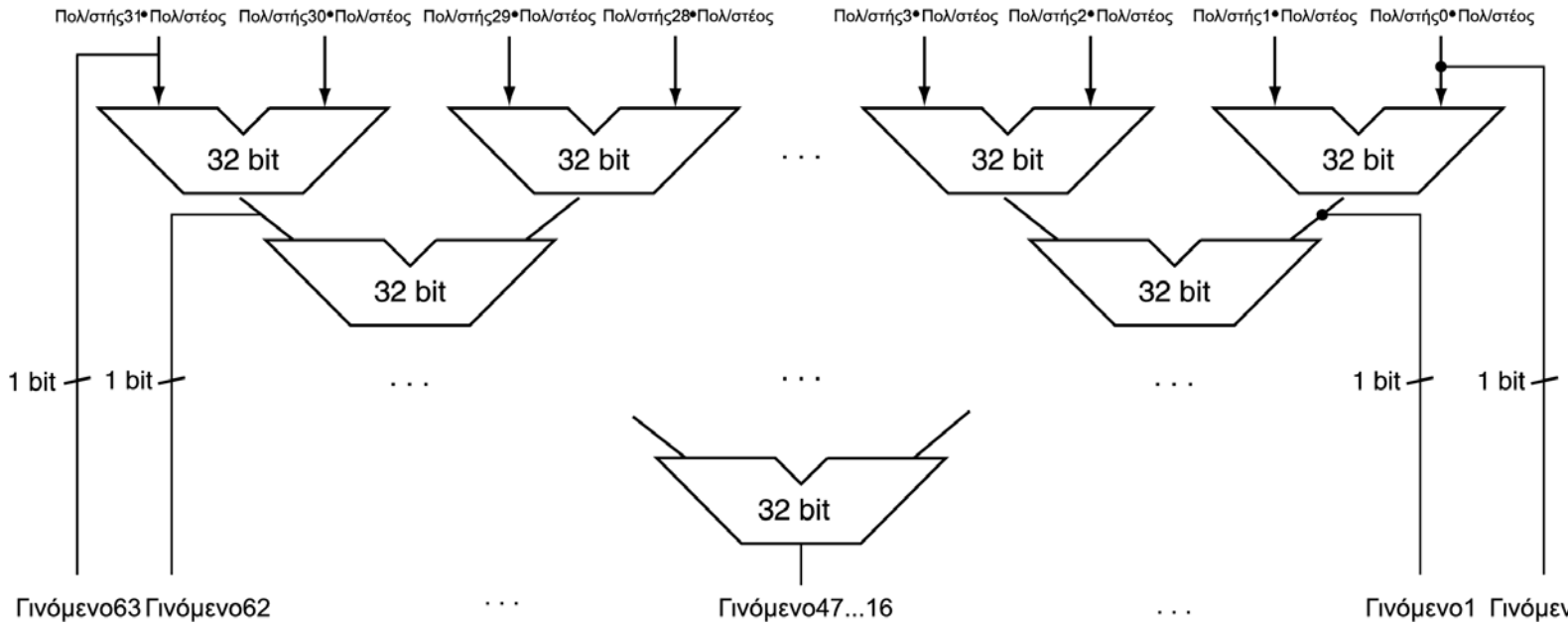
Βελτιστοποιημένος πολλαπλασιαστής



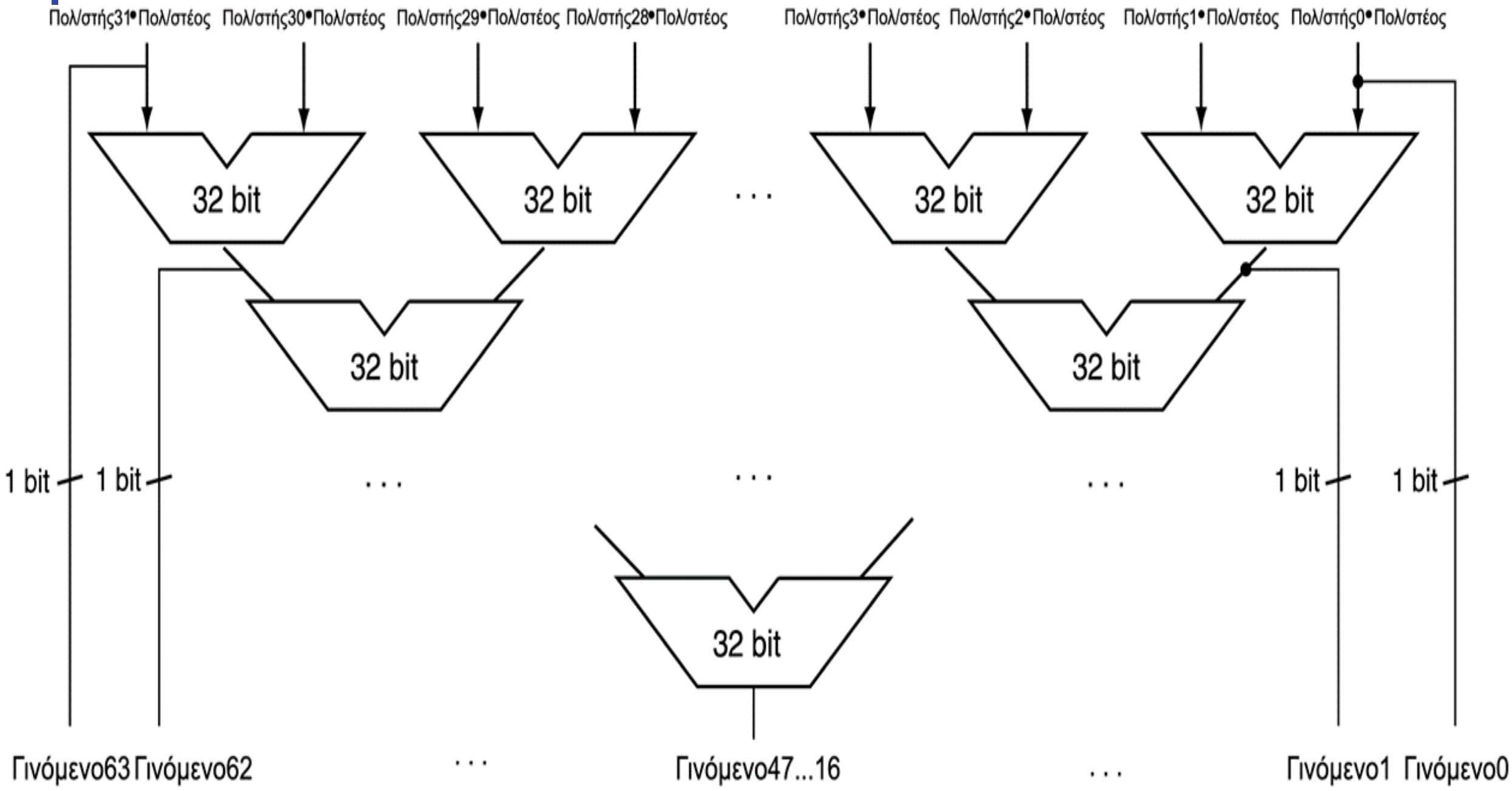

Πολλαπλασιαστέος
32 bit
ALU 32 bit
Γινόμενο
Δεξιά ολίσθηση
Εγγραφή
64 bit
Μονάδα
ελέγχου
και δοκιμής

Ταχύτερος πολλαπλασιαστής

- Χρησιμοποιεί πολλούς αθροιστές
  - Συμβιβασμός κόστους/απόδοσης



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


Πολ/στής31•Πολ/στέος
Πολ/στής30•Πολ/στέος
Πολ/στής29•Πολ/στέος
Πολ/στής28•Πολ/στέος
Πολ/στής3•Πολ/στέος
Πολ/στής2•Πολ/στέος
Πολ/στής1•Πολ/στέος
Πολ/στής0•Πολ/στέος
32 bit
32 bit
32 bit
32 bit
32 bit
32 bit
32 bit
1 bit
1 bit
1 bit
1 bit
Γινόμενο63
Γινόμενο62
Γινόμενο47...16
Γινόμενο1
Γινόμενο0

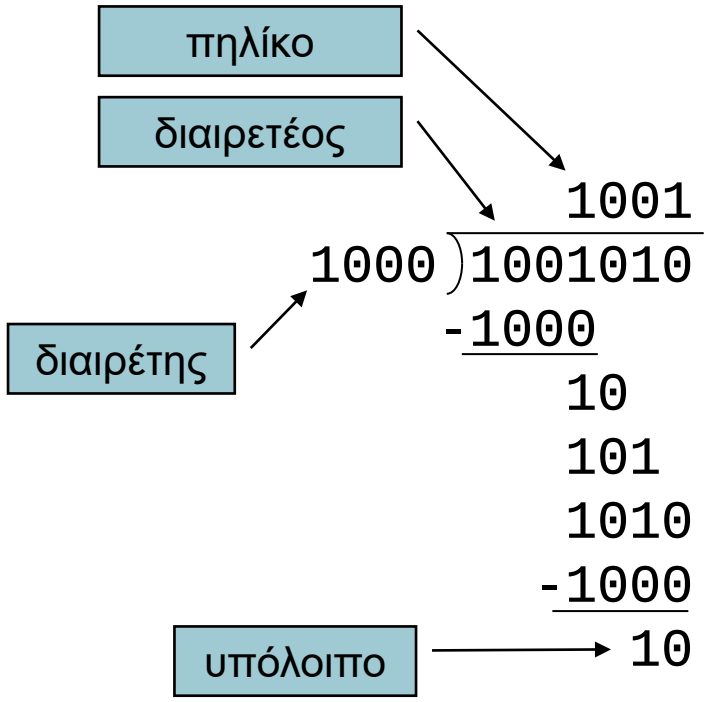
Ταχείες λειτουργίες σ2

- Επέκταση παράστασης
- Πολλαπλασιασμός x2
- Διαίρεση /2

Πολλαπλασιασμός στον MIPS

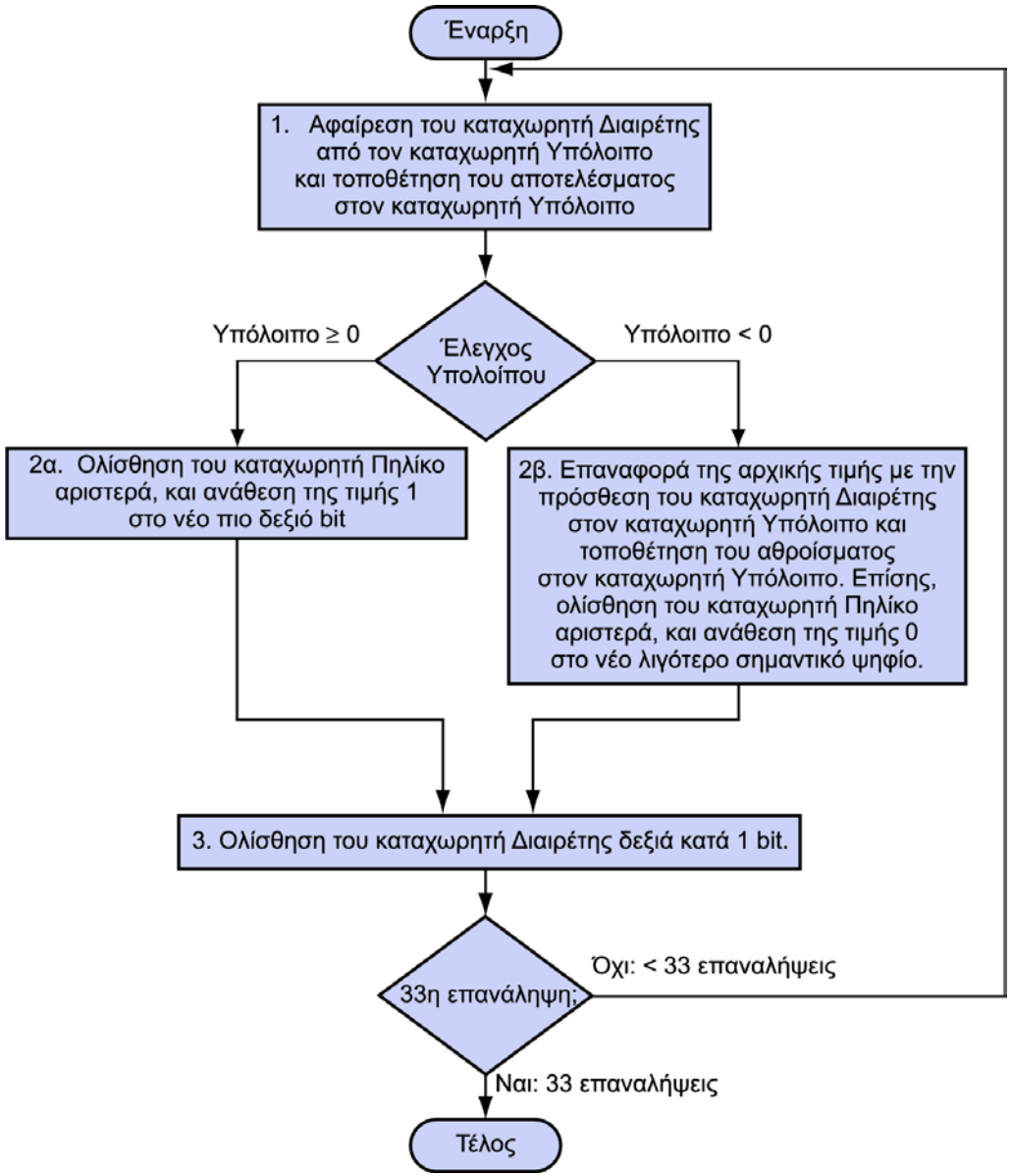
- Δύο καταχωρητές των 32 bit για το γινόμενο
  - HI: τα περισσότερο σημαντικά 32 bit
  - LO: τα λιγότερο σημαντικά 32 bit
- Εντολές
  - mult rs, rt / multu rs, rt
    - γινόμενο των 64 bit στους HI/LO
  - mfhi rd / mflo rd
    - Μεταφορά από (move from) του HI/LO στον rd
    - Μπορούμε να ελέγξουμε την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

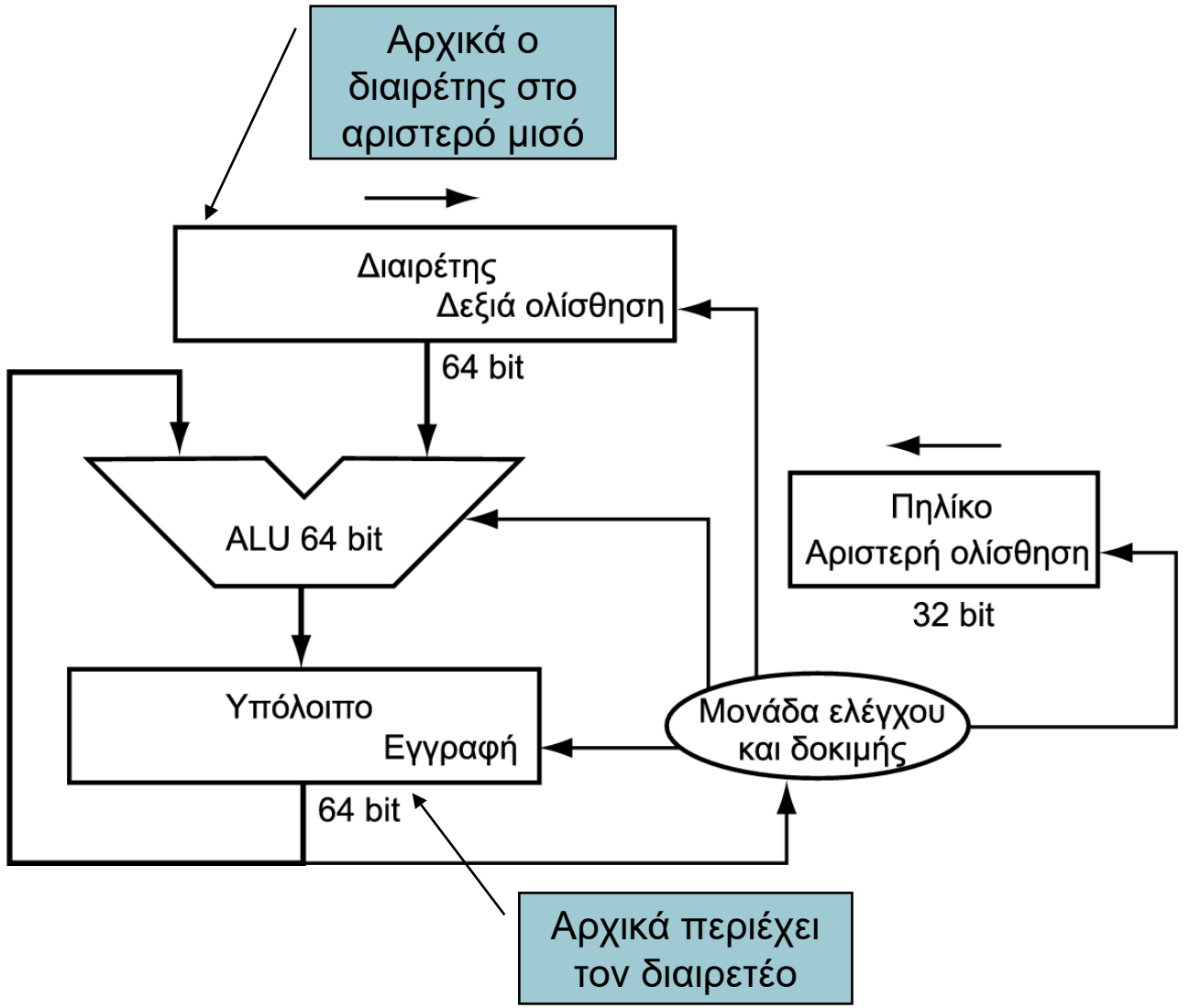
Διαίρεση



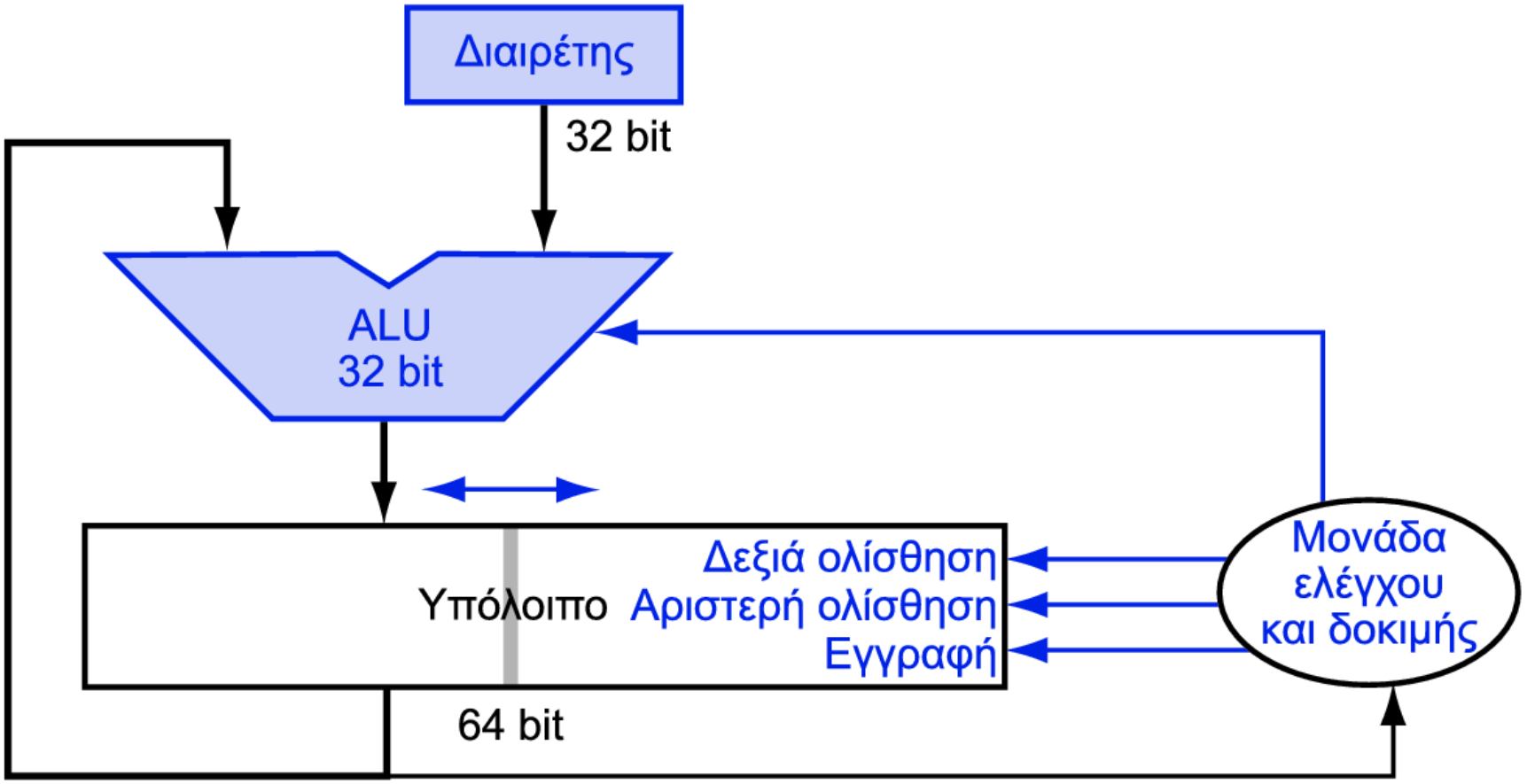
- Έλεγχος για μηδενικό διαιρέτη
- Διαίρεση μεγάλου μήκους
  - Αν διαιρέτης ≤ από τα bit του διαιρετέου
    - '1' στο πηλίκο, αφαίρεση
  - Αλλιώς
    - '0' στο πηλίκο, κατέβασμα του επόμενου bit του διαιρετέου
- Διαίρεση με επαναφορά (restoring division)
  - Κάνε την αφαίρεση και αν το υπόλοιπο γίνει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

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 στον 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 ακέραιους αριθμούς
  - Περιλαμβάνει και πολύ μικρούς και πολύ μεγάλους αριθμούς
- Όπως η επιστημονική σημειογραφία (scientific notation)
  - $-2{,}34 \times 10^{56}$
  - $+0{,}002 \times 10^{-4}$
  - $+987{,}02 \times 10^{9}$
- 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
- Αναπαραστάσεις κινητής υποδιαστολής (float point)
  - Απλή ακρίβεια – single precision (32 bit)
  - Διπλή ακρίβεια – double precision (64 bit)
  - 4πλή ακρίβεια – quad precision (128 bit)

Μορφή κινητής υποδιαστολής IEEE

single: 8 bit
double: 11 bit

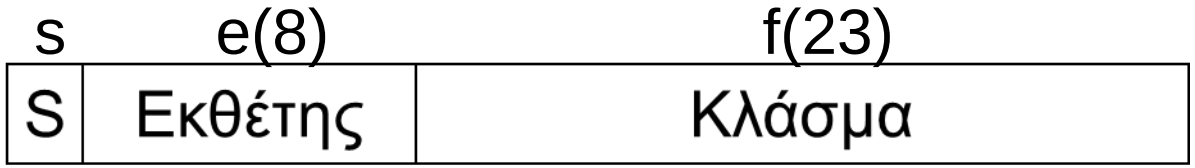
single: 23 bit
double: 52 bit

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 **δεν χρειάζεται** ρητή αναπαράστασή του («κρυμμένο» bit)
- 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) 32-bit



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)
$s = 0; e = 255; f = 0$	**+∞** (θετικό άπειρο)
$s = 1; e = 255; f = 0$	**-∞** (αρνητικό άπειρο)
$s = x; e = 255; f \neq 0$	NaN (μη-αριθμός)

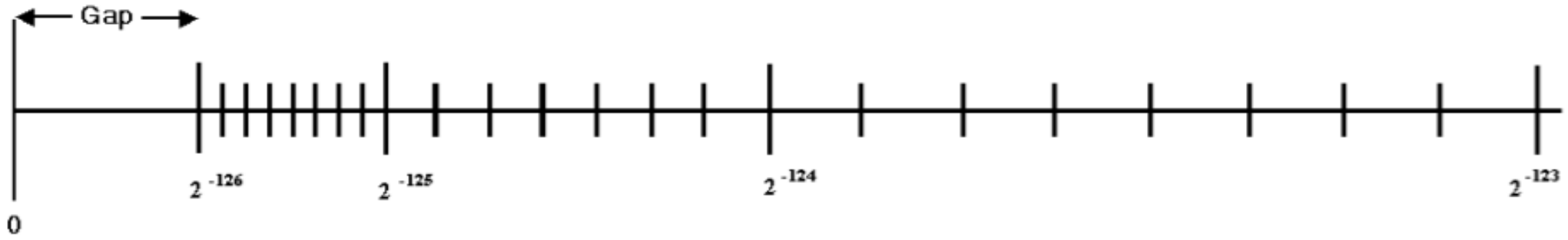
Εύρος απλής ακρίβειας

- V_{Max}: 0 11111110 11111111111111111111111
- V_{Max}: $(2^{254-127})$ (1,11111111111111111111111)
- V_{Max}: (2^{127}) $(2 - 2^{-23}) \approx \mathbf{2^{128}}$

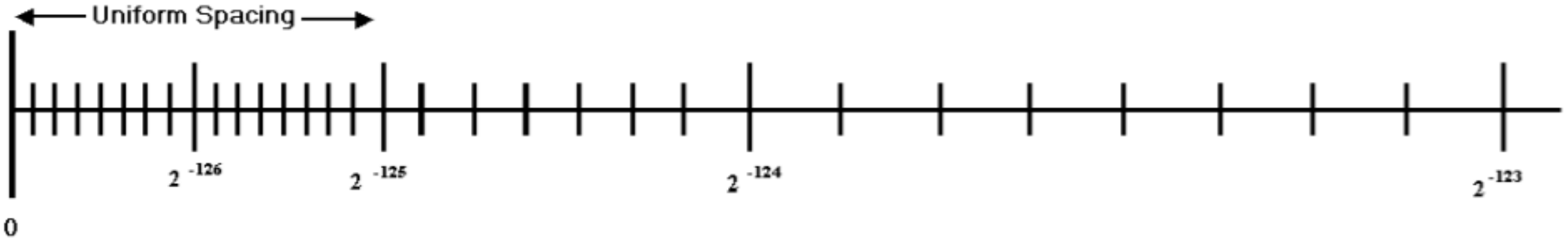
- V_{Min}: 0 00000001 00000000000000000000000
- V_{Min}: 0 00000001 00000000000000000000000
- V_{Min}: (2^{1-127}) (1,00000000000000000000000)
- V_{Min}: 2^{-126}

- V_{MIN}: 0 00000000 00000000000000000000001
- V_{MIN}: (2^{-126}) $(2^{-23}) = \mathbf{2^{-149}}$

Κανονική/Υποκανονική παράσταση

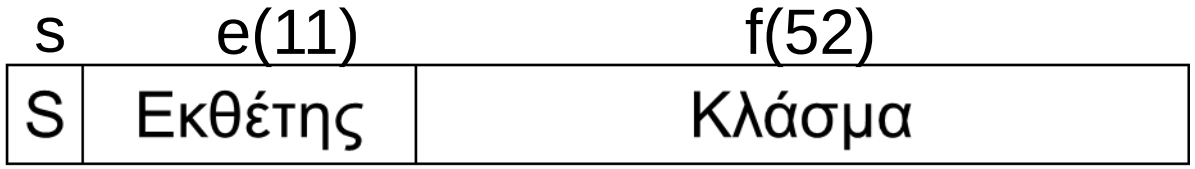


(a) 32-bit Format Without Denormalized Numbers



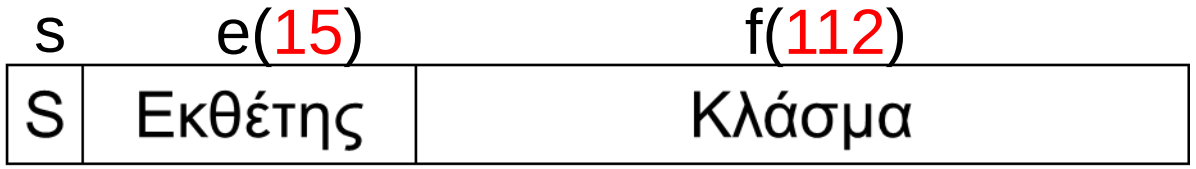
(b) 32-bit Format With Denormalized Numbers

IEEE 754 (διπλή ακρίβεια) 64-bit



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)
$s = 0; e = 2047; f = 0$	$+\infty$ (θετικό άπειρο)
$s = 1; e = 2047; f = 0$	$-\infty$ (αρνητικό άπειρο)
$s = x; e = 2047; f \neq 0$	NaN (μη-αριθμός)

IEEE 754 (4πλή ακρίβεια) 128-bit

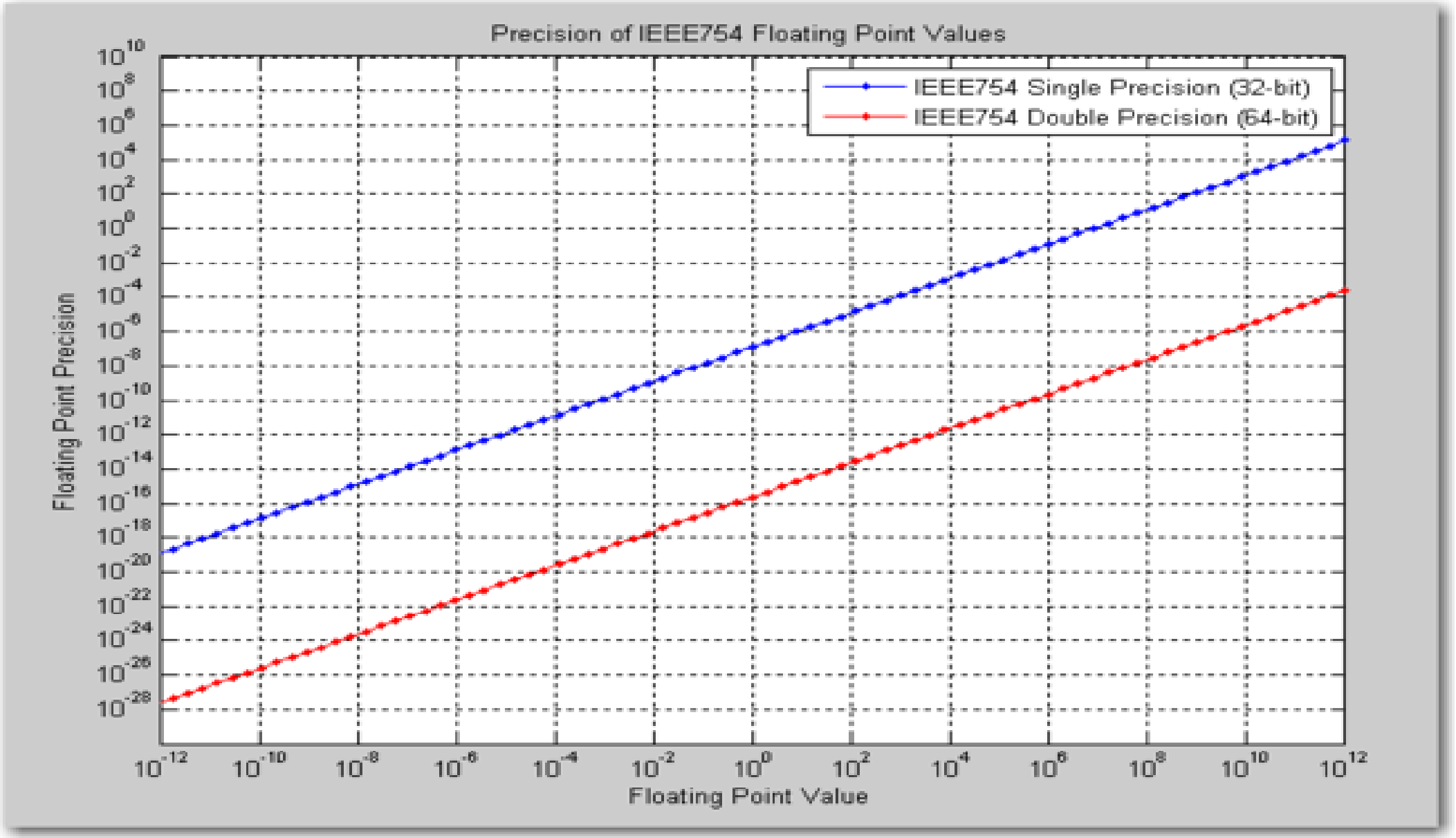


Συνθήκη	Τιμή
$0 < e < 32767$	$(-1)^s \times 2^{e-16383} \times 1.f$ (κανονικοποιημένη μορφή)
$e = 0; f \neq 0$	$(-1)^s \times 2^{-16382} \times 0.f$ (υπο-κανονική μορφή)
$e = 0; f = 0$	$(-1)^s \times 0.0$ (μηδέν)
$s = 0; e = 32767; f = 0$	**+∞** (θετικό άπειρο)
$s = 1; e = 32767; f = 0$	**-∞** (αρνητικό άπειρο)
$s = x; e = 32767; f \neq 0$	NaN (μη-αριθμός)

Εργαλεία μετατροπής ΑΚΥ

- https://babbage.cs.qc.cuny.edu/IEEE-754/
- https://www.h-schmidt.net/FloatConverter/IEEE754.html

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: 10111111010000000000000000000000
- Διπλή: 10111111111010000…00

Παράδειγμα κινητής υποδιαστολής

- Ποιος είναι ο αριθμός
  1 10000001 01000000000000000000000

  - S = 1
  - Κλάσμα = $01000\ldots00_2$
  - Εκθέτης = $10000001_2 = 129$

- $x = (-1)^1 \times (1 + ,01_2) \times 2^{(129 - 127)}$
  $= (-1) \times 1,25 \times 2^2$
  $= -5,0$

Άπειρο και μη-αριθμοί (NaN)

- Εκθέτης = 111...1, Κλάσμα = 000...0
  - ±Άπειρο
  - Μπορεί να χρησιμοποιηθεί σε επόμενους υπολογισμούς, για αποφυγή 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
  - **$1,002 \times 10^{2}$**

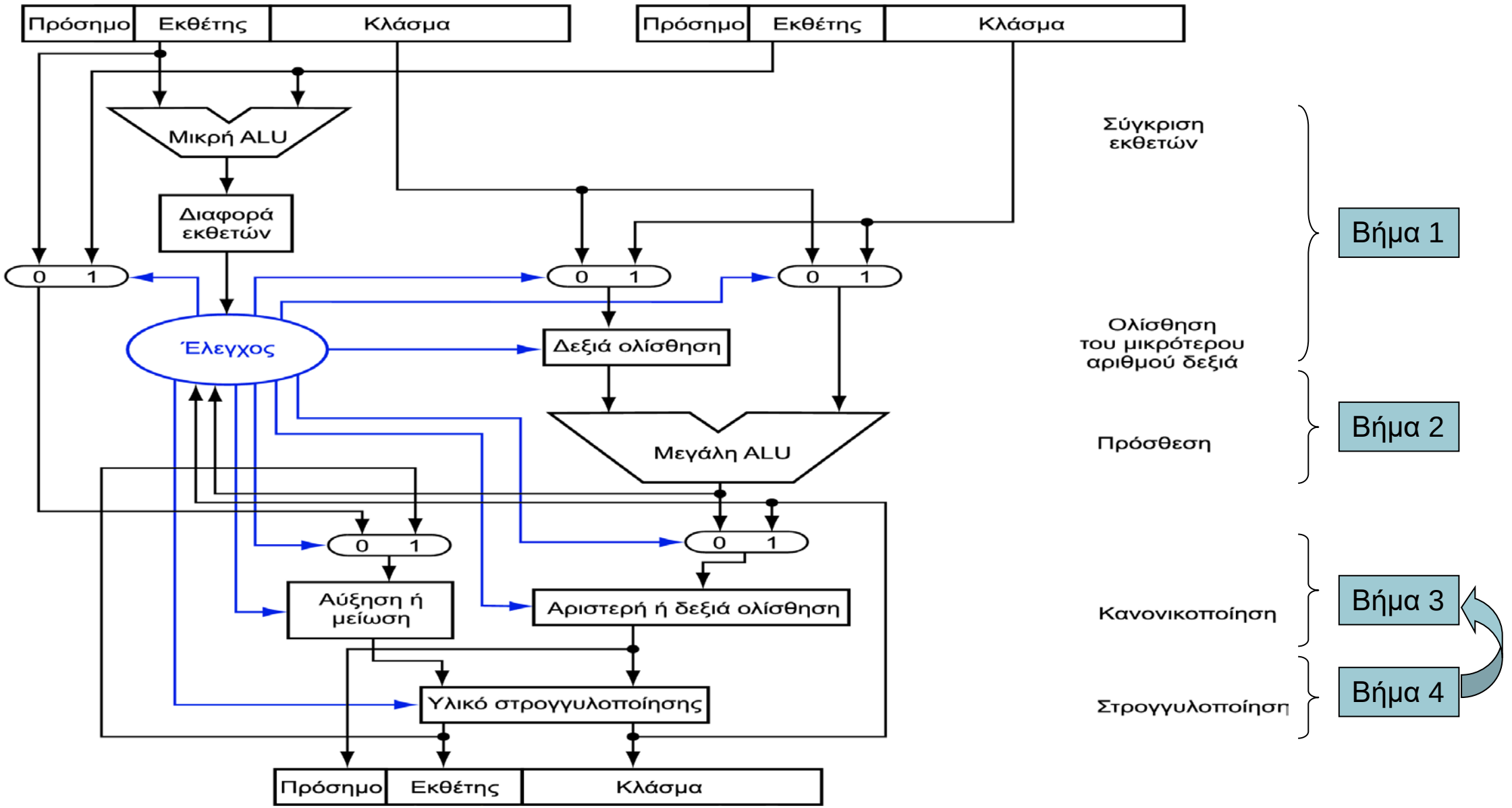
Πρόσθεση κινητής υποδιαστολής

- 4-bit αριθμοί $1.000_2 \times 2^{-1} + -1,110_2 \times 2^{-2}$ (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 επανα-κανονικοποιήση, αν είναι απαραίτητο
  - $1,000_2 \times 2^{-4}$ (καμία αλλαγή) = 0,0625

Υλικό αθροιστή κινητής υποδιαστολής

- Πολύ πιο πολύπλοκος από τον ακέραιο 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 υλοποιηθεί με διοχέτευση

Αθροιστής κινητής υποδιαστολής



Πολλαπλασιασμός κιν. υποδιαστολής

- Ένα δεκαδικό παράδειγμα με 4 ψηφία
  - $1{,}110 \times 10^{10} \times 9{,}200 \times 10^{-5}$
- 1. Πρόσθεση εκθετών
  - Για πολωμένους εκθέτες, αφαίρεση της πόλωσης από το άθροισμα
  - Νέος εκθέτης = 10 + (–5) = 5
- 2. Πολλαπλασιασμός σημαντικών
  - $1.110 \times 9{,}200 = 10.212 \Rightarrow 10{,}212 \times 10^{5}$
- 3. Κανονικοποίηση αποτελέσματος & έλεγχος για υπερχείλιση ή ανεπάρκεια
  - $1{,}0212 \times 10^{6}$
- 4. Στρογγυλοποίηση και επανακανονικοποίηση
  - $1{,}021 \times 10^{6}$
- 5. Καθορισμός προσήμου του αποτελέσματος
  - $+1{,}021 \times 10^{6}$

Πολλαπλασιασμός κιν. υποδιαστολής

- Ένα δυαδικό παράδειγμα με 4 bit
  - $1{,}000_2 \times 2^{-1} \times -1{,}110_2 \times 2^{-2}$ (0,5 × -0,4375)
- 1. Πρόσθεση εκθετών
  - Χωρίς πόλωση: -1 + -2 = -3
  - Με πόλωση: (-1 + 127) + (-2 + 127) = -3 + 254 - 127 = -3 + 127
- 2. Πολλαπλασιασμός σημαντικών
  - $1{,}000_2 \times 1{,}110_2 = 1{,}1102 \Rightarrow 1{,}110_2 \times 2^{-3}$
- 3. Κανονικοποίηση αποτελέσματος και έλεγχος υπερχείλισης/ανεπάρκειας
  - $1{,}110_2 \times 2^{-3}$ (καμία αλλαγή) χωρίς υπερχείλιση/ανεπάρκεια
- 4. Στρογγυλοποίηση και επανακανονικοποίηση
  - $1{,}110_2 \times 2^{-3}$ (καμία αλλαγή)
- 5. Καθορισμός προσήμου: +ve × -ve ⇒ -ve
  - $-1{,}110_2 \times 2^{-3} = -0{,}21875$

Υλικό κινητής υποδιαστολής

- 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 στον MIPS

- Επιπρόσθετος συνεπεξεργαστής επεκτείνει την αρχιτεκτονική
- Ξεχωριστοί **καταχωρητές ΚΥ**
  - 32 απλής ακρίβειας: $f0, $f1, … $f31
  - Ζεύγη για διπλή ακρίβεια: $f0/$f1, $f2/$f3, …
- **Εντολές ΚΥ** επενεργούν μόνο σε καταχωρητές ΚΥ
  - Τα προγράμματα δεν εκτελούν ακέραιες πράξεις σε δεδομένα ΚΥ
- Εντολές φόρτωσης και αποθήκευσης καταχωρητών 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 (t) ή ψευδή (f) κωδικό συνθήκης ΚΥ
  - bc1t, bc1f π.χ. bc1t TargetLabel

Παράδειγμα ΚΥ: βαθμοί °F σε °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

- Κώδικας MIPS:

```
f2c: lwc1  $f16, const5($gp)
     lwc1  $f18, const9($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 3 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

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