

ΗΥ 134

Εισαγωγή στην Οργάνωση και
στον Σχεδιασμό Υπολογιστών Ι

Διάλεξη 10

Πολλαπλασιασμός και Διαίρεση Ακεραίων

Νίκος Μπέλλας

Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων

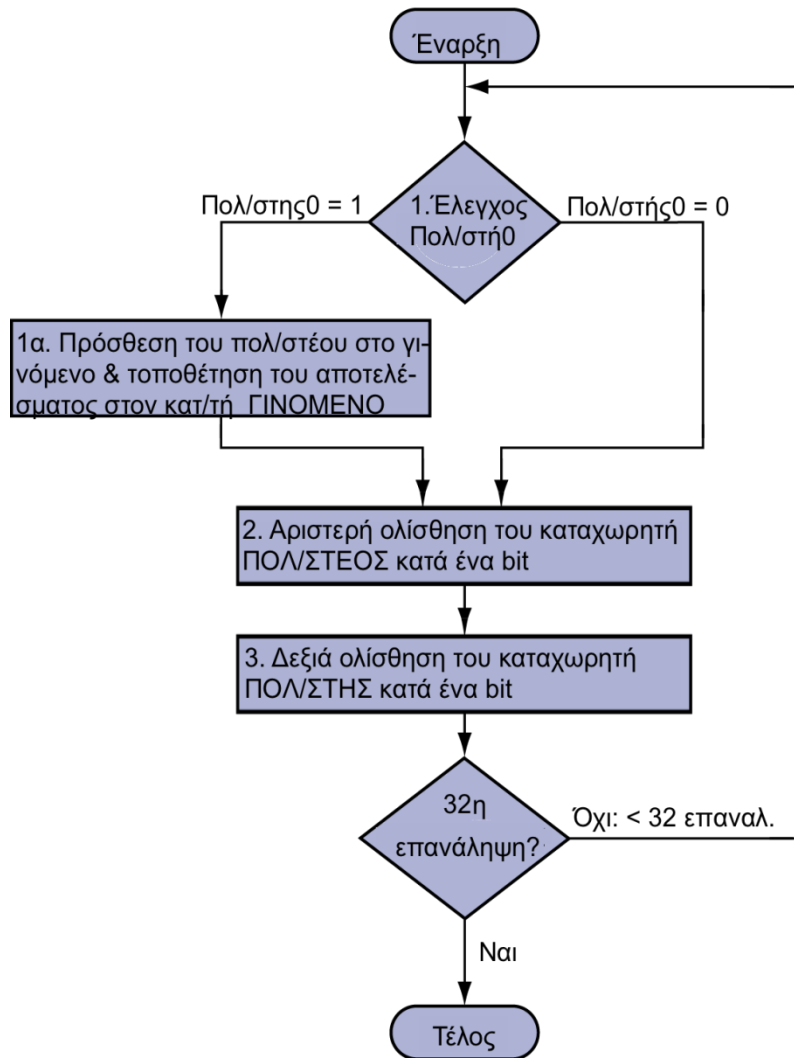
Πολλαπλασιασμός Ακεραίων

Μη προσημασμένοι ακέραιοι: ακριβώς όπως και στον πολλαπλασιασμό στο δεκαδικό σύστημα

πολλαπλασιαστέος	→	1000	(8)
πολλαπλασιαστής	×	1001	(9)
		<hr/>	
		1000	
		0000	
		0000	
		1000	
γινόμενο	→	<hr/>	
		01001000	(72)

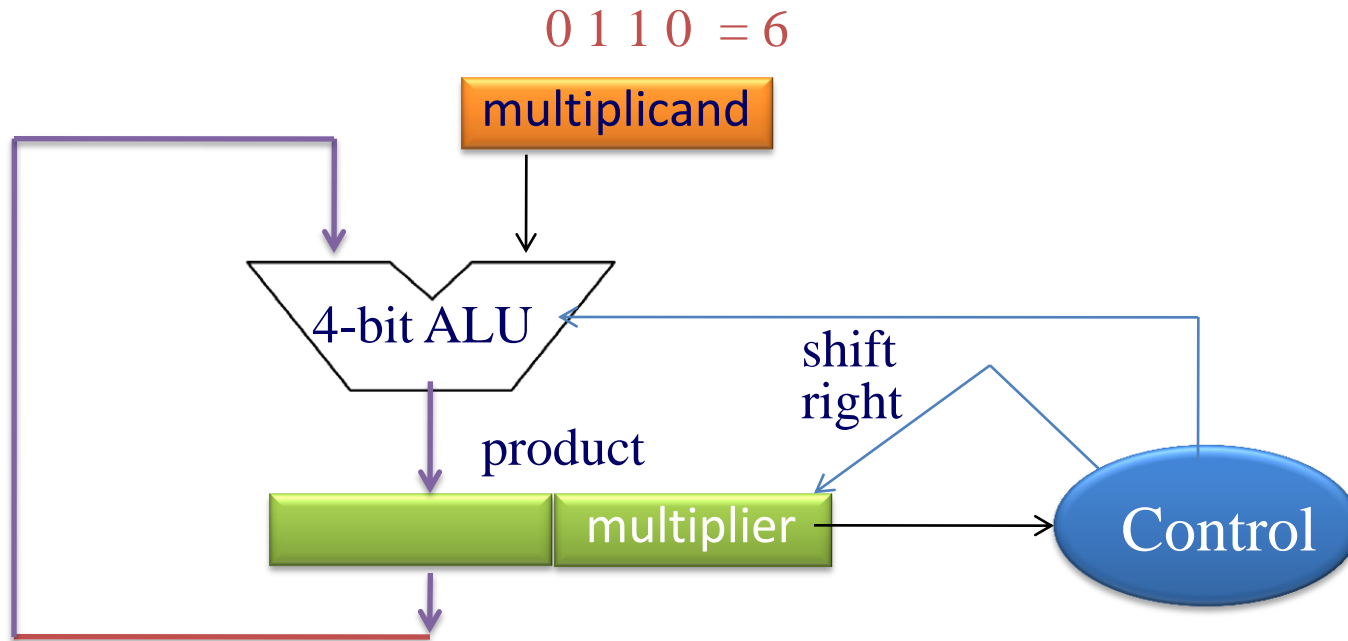
Το μήκος του γινομένου είναι όσο το άθροισμα των μηκών των τελεστών.

Απλή υλοποίηση ενός 32-bit πολλαπλασιαστή απρόσημων αριθμών



Υπάρχει όμως και πιο αποδοτικός τρόπος

Σειριακός πολλαπλασιαστής απρόσημων αριθμών



	0 0 0 0	0 1 0 1	= 5
add	0 1 1 0	0 1 0 1	
	0 0 1 1	→ 0 0 1 0	
add	0 0 1 1	0 0 1 0	
	0 0 0 1	→ 1 0 0 1	
add	0 1 1 1	1 0 0 1	
	0 0 1 1	→ 1 1 0 0	
add	0 0 1 1	1 1 0 0	
	0 0 0 1	→ 1 1 1 0	= 30

Εισαγωγή στους Η/Υ (HY134)

Προσημασμένος πολλαπλασιασμός

- Ο απλούστερος τρόπος να πολλαπλασιάσουμε δύο προσημασμένους αριθμούς n και m bits:
 1. είναι να τους μετατρέψουμε πρώτα σε θετικούς αριθμούς
 2. να κάνουμε τον πολλαπλασιασμό όπως μεταξύ απρόσημων
 3. να μετατρέψουμε το γινόμενο σε αρνητικό (συμπλήρωμα ως προς 2), εάν ένας από τους πολλαπλασιαστέο ή πολλαπλασιαστή είναι αρνητικός
- Το γινόμενο έχει μέγεθος $(n-1)+(m-1) + 1 = n+m-1$ bits
 - $m-1$ είναι το μέγεθος του πολλαπλασιαστέου (χωρίς το πρόσημο)

$$(-8) \times (+10) \text{ με } m=n=5$$

1. πολλ/σμός $(+8) \times (+10)$

2.

	01000	(+8)
	<u>01010</u>	(+10)
	00000	
	01000	
	00000	
	01000	
	<u>00000</u>	
	001010000	(+80)

3. Μετατροπή σε
 $-80 = (110110000)_2$
χρησιμοποιώντας $4+4+1=9$
bits

Προσημασμένος πολλαπλασιασμός

- Ο πολλαπλασιασμός με προσημασμένους μπορεί να γίνει και απευθείας με την κατάλληλη επέκταση προσήμου του πολλαπλασιαστέου στα μερικά γινόμενα (βλ. παράδειγμα Α)
- Επίσης έχουμε διαφορετικό χειρισμό στην περίπτωση αρνητικού πολλ/στή.
- Στην περίπτωση αυτή, αφαιρούμε τον πολλ/στέο από τον πολλ/στη στο ν-οστό βήμα (βλ. παράδειγμα Β)

- $(-8) \times (+10)$ με $m=n=5$

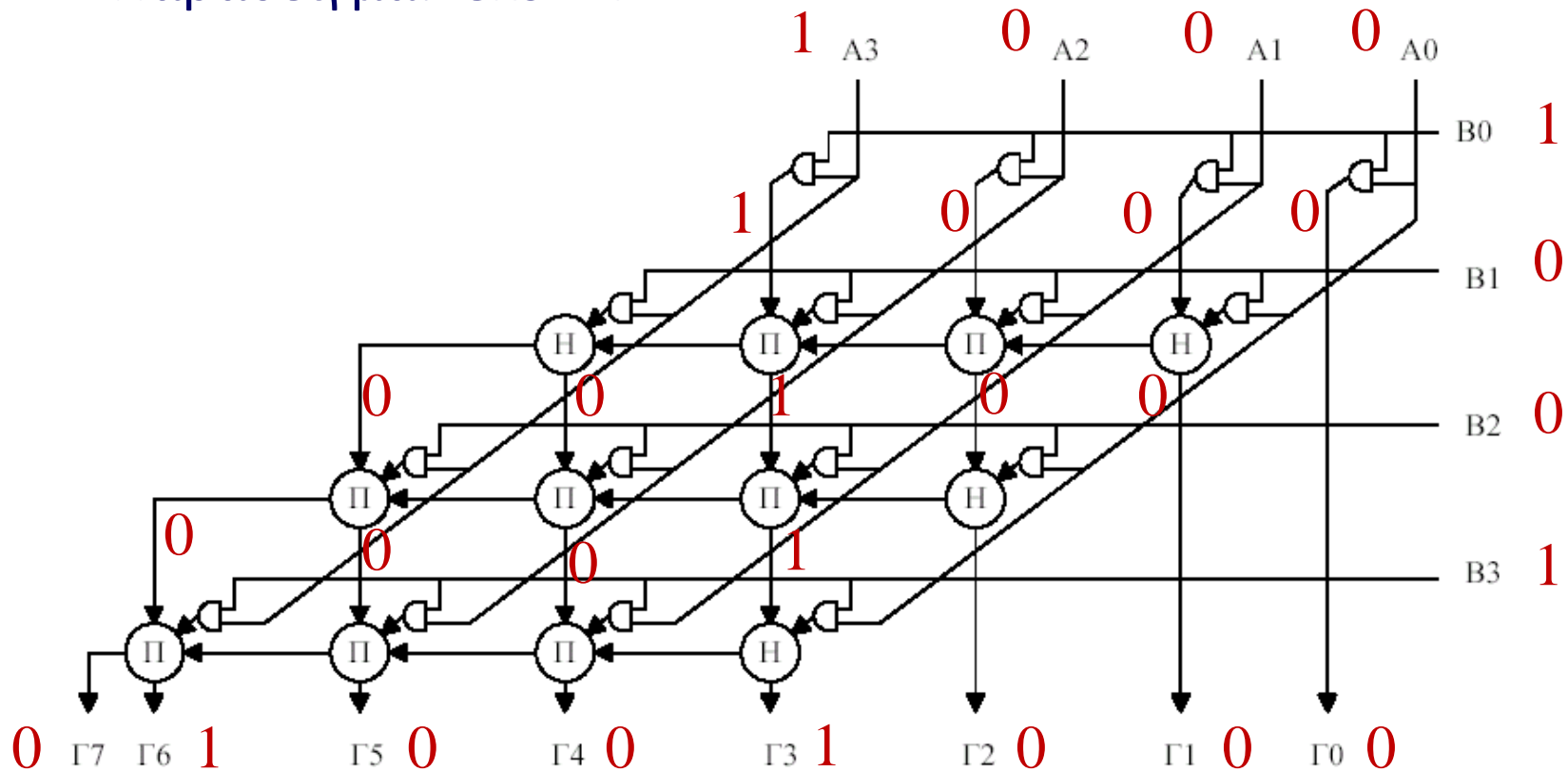
$$\begin{array}{r} 11000 \quad (-8) \\ 01010 \quad (+10) \\ \hline 00000 \\ 11111000 \\ 00000 \\ 111000 \\ 000000 \\ \hline 110110000 \quad (-80) \end{array}$$

- $(-8) \times (-10)$ με $m=n=5$

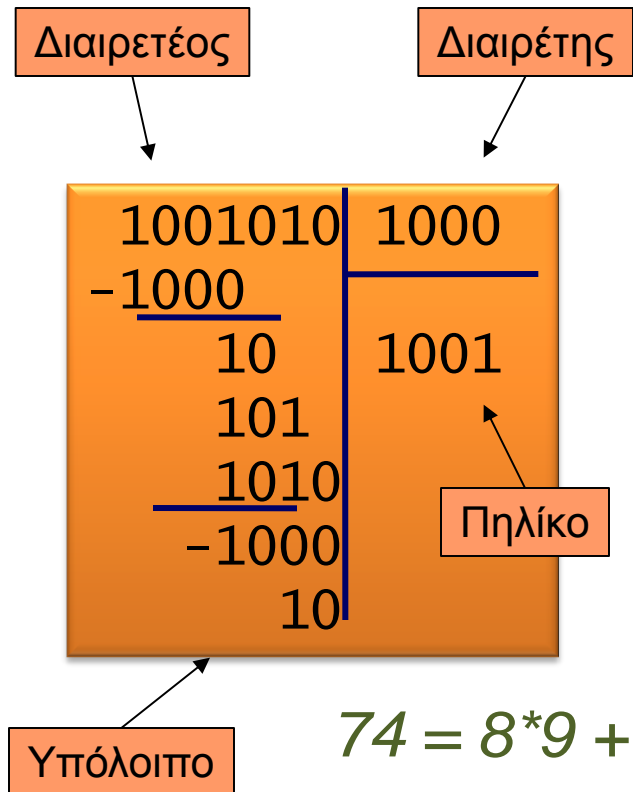
$$\begin{array}{r} 11000 \quad (-8) \\ 10110 \quad (-10) \\ \hline 00000 \\ 11111000 \\ 11111000 \\ 00000 \\ 001000 \quad (+8) \\ \hline 001010000 \quad (+80) \end{array}$$

Παράλληλος Πολλαπλασιαστής

- Χρήση ενός δέντρου (tree) από αθροιστές
- Όλες οι προσθέσεις γίνονται παράλληλα
- Παράδειγμα: $8 \times 9 = 72$



Διαίρεση

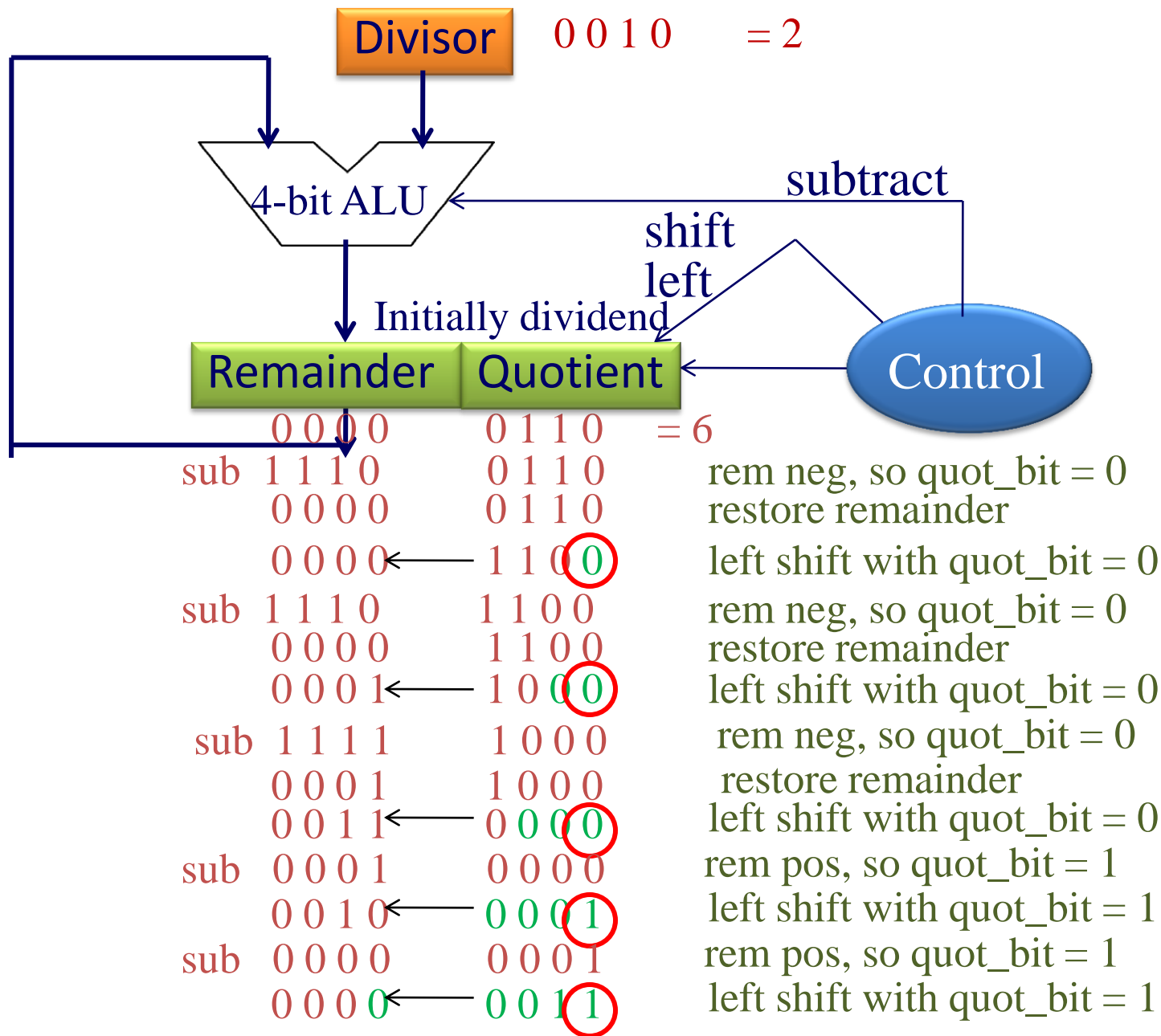


$$74 = 8 * 9 + 2$$

- Έλεγχος για διαιρέτη 0
- Ο κλασικός αλγόριθμος
 - Αν ο διαιρέτης χωράει στο διαιρετέο
 - 1 στο πηλίκο, αφαίρεση
 - Αλλιώς
 - 0 στο πηλίκο, κατεβαίνει επόμενο ψηφίο
- Προσημασμένη διαίρεση
 - Κάνουμε διαίρεση χρησιμοποιώντας απόλυτες τιμές
 - Διορθώνουμε το πρόσημο του πηλίκου και του υπολοίπου αναλόγως

- Βασική σχέση της Διάρησης:
Διαιρετέος = Πηλίκο * Διαιρέτης + Υπόλοιπο

Υλοποίηση Διαίρεσης σε Hardware



Προσημασμένη Διαίρεση

- Μετατρέπουμε τον διαιρέτη και διαιρετέο από αρνητικούς σε θετικούς και κάνουμε απρόσημη διαίρεση
- Στο τέλος, κάνουμε αρνητικό το πηλίκο, εάν τα πρόσημα του διαιρέτη και διαιρετέου είναι διαφορετικά
- Ας δούμε την διαίρεση $(+-7)/(+-2)$
 - $7/2$: Πηλίκο = 3, Υπόλοιπο = 1 ($7 = 3*2 + 1$)
 - $-7/2$: Πηλίκο = -3, Υπόλοιπο = -1 ($-7 = (-3)*2 + (-1)$)
 - $7/-2$: Πηλίκο = -3, Υπόλοιπο = 1 ($-7 = (-3)*(-2) + 1$)
 - $-7/-2$: Πηλίκο = 3, Υπόλοιπο = -1 ($-7 = 3*(-2) + (-1)$)

Πολλαπλασιασμός στον MIPS

- Δύο νέοι καταχωρητές 32-bit για το γινόμενο
 - HI: Περισσότερο-σημαντικά 32 bits
 - LO: Λιγότερο-σημαντικά 32-bits
- Εντολές
 - `mult $rs, $rt / multu $rs, $rt`
 - γινόμενο 64-bit σε καταχωρητές HI/LO
 - `mghi $rd / mflo $rd`
 - Μετακίνηση από HI/LO στο rd
 - Μπορεί να ελέγξει την τιμή του HI για να δει αν το γινόμενο υπερχειλίζει τα 32 bits
 - `mul $rd, $rs, $rt`
 - γινόμενο 64-bit σε καταχωρητές HI/LO ΚΑΙ λιγότερο-σημαντικά 32 bits του γινομένου στον \$rd

Διαίρεση στον MIPS

- Εντολές

$\text{div } \$rs, \$rt \Leftrightarrow Lo = \$rs / \$rt ; Hi = \$rs \text{ mod } \$rt$

$\text{divu } \$rs, \$rt \Leftrightarrow Lo = (\$rs / \$rt)_u ; Hi = (\$rs \text{ mod } \$rt)_u$

- Ο Πολλαπλασιασμός και η Διαίρεση αγνοούν τυχόν υπερχείλιση

- Το software θα πρέπει να ελέγχει για υπερχείλιση
- Το software θα πρέπει να ελέγχει τον διαιρέτη ώστε να είναι διάφορος του 0.