



Εισαγωγή στην Πληροφορική και τον Προγραμματισμό Η/Υ

5^ο Μάθημα

Πολλαπλασιασμός στο Δυαδικό
Παράσταση Κλασματικών
Σταθερή & Κινητή Υποδιαστολή

Λεωνίδας Αλεξόπουλος

Αν. Καθηγητής ΕΜΠ

E-mail: leo@mail.ntua.gr

Τηλ: 210 772-1666

Στα προηγούμενα μαθήματα:

- 1- Τι είναι Η/Υ
- 2- Κωδικοποίηση-Επεξεργασία- Αποκωδικοποίηση
- 3- Συστήματα Αρίθμησης: 2δικό, 8δικό, 10δικό, 16δικό
- 4- Μετατροπή από το ένα σύστημα στο άλλο
- 5- Bit/byte/word, KB/MB/GB/TB...
- 6- Η πρόσθεση στο Δυαδικό
- 7- Παράσταση Αρνητικών -1
- 8- Η αφαίρεση στο Δυαδικό

Στη συνέχεια

- Ο πολλαπλασιασμός στο Δυαδικό
- Παράσταση Κλασματικών στο Δυαδικό
 - Σταθερή υποδιαστολή
 - Κινητή Υποδιαστολή / IEEE standard

Προγραμματισμός σε Η/Υ

Εφαρμογές



Γλώσσες Προγραμματισμού

```
1 function rowTotals = rowsum
2 % Add the values in each row and
3 % store them in a new array.
4
5 x = ones(2,10);
6 [n, m] = size(x);
7 rowTotals = zeros(1,n);
8 for j = 1:n
9     % Press Shift+Enter to rename 5 instances of 'j' to 'j'
10    end
11
12 function colsum = addToSum
13 colsum = 0;
14 thisrow = x(i,:);
15 for i = 1:m
16     colsum = colsum + thisrow(i);
17 end
18 end
19
20 end
```

Λειτουργικό Σύστημα

Γλώσσα Μηχανής

Assembly Language	Machine Code
add \$t1, \$t2, \$t3	04CB: 0000 0100 1100 1011
addi \$t2, \$t3, 60	16BC: 0001 0110 1011 1100
and \$t3, \$t1, \$t2	0299: 0000 0010 1001 1001
andi \$t3, \$t1, 5	22C5: 0010 0010 1100 0101
beq \$t1, \$t2, 4	3444: 0011 0100 0100 0100
bne \$t1, \$t2, 4	4444: 0100 0100 0100 0100
j 0x50	F032: 1111 0000 0011 0010
lw \$t1, 16(\$s1)	5A50: 0101 1010 0101 0000
nop	0005: 0000 0000 0000 0101
nor \$t3, \$t1, \$t2	029E: 0000 0010 1001 1110
or \$t3, \$t1, \$t2	029A: 0000 0010 1001 1010
ori \$t3, \$t1, 10	62CA: 0110 0010 1100 1010
ssl \$t2, \$t1, 2	0455: 0000 0100 0101 0101
srl \$t2, \$t1, 1	0457: 0000 0100 0101 0111
sw \$t1, 16(\$t0)	7050: 0111 0000 0101 0000
sub \$t2, \$t1, \$t0	0214: 0000 0010 0001 0100

Μικρολειτουργίες
&
Μικροπρογραμματισμός

Ψηφιακή Λογική

Πολλαπλασιασμός Δυαδικών Ακεραίων

- $0 * 0 = 0$ $0 * 1 = 0$ $1 * 0 = 0$ $1 * 1 = 1$

- Παραδείγματα

$$\begin{array}{r} 1101 \\ \times 1001 \\ \hline 1101 \\ 0000 \\ 0000 \\ \hline 1101 \\ \hline 1110101 \end{array}$$



$$\begin{array}{r} 1101 \\ \times 1001 \\ \hline 1101 \\ 1101 \\ \hline 1110101 \end{array}$$

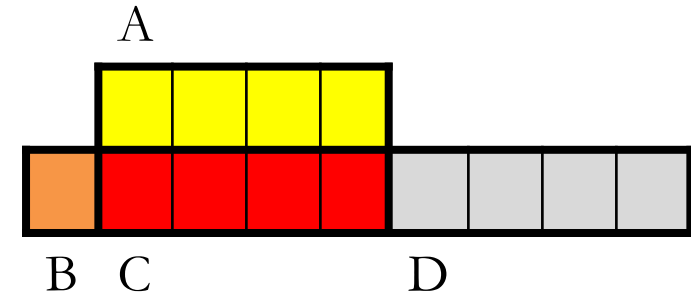
- ο πολλαπλασιασμός δυαδικών αριθμών πραγματοποιείται με **πρόσθεση** και **ολίσθηση**.

Αλγόριθμος πολλαπλασιασμού

Έστω ότι θέλουμε να πολλαπλασιάσουμε 2 αριθμούς A , D μήκους λέξης n -bit.

Κατασκευάζουμε τα χωρία A, B, C, D με το B να έχει μήκος 1-bit ενώ τα A, C, D n -bit π.χ για $n=4$

- **Βήμα 1ο** : Τοποθέτησε τους A , D στα αντίστοιχα κελιά και τοποθέτησε μηδενικά στα B , C .



- **Βήμα 2ο** : Κάνε τα παρακάτω “ n ” φορές (βρόχος)
 - Αν το LSD του D είναι 1, τότε πρόσθεσε τα A , C . Βάλε το αποτέλεσμα στο C και το κρατούμενο (αν υπάρχει) στο B . Αν δεν υπάρχει κρατούμενο, τότε στο B μπαίνει 0.
 - Μετάθεσε όλα τα ψηφία του ενοποιημένου χώρου BCD όλα μια θέση δεξιά (ολίσθηση)
- **Βήμα 3ο** : Μόλις τελειώσουν οι “ n ” φορές, το αποτέλεσμα είναι στο CD .

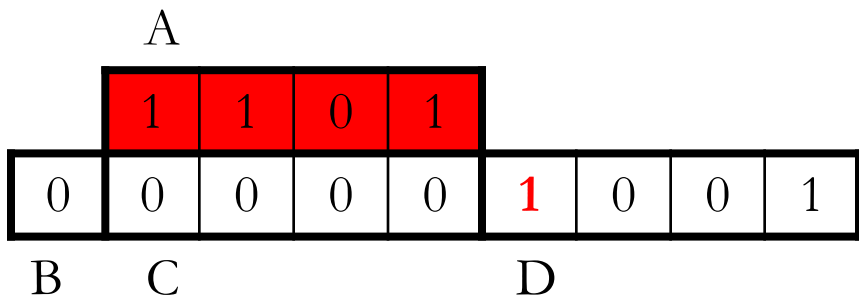
Αλγόριθμος πολλαπλασιασμού, (...συν)

```

    1 1 0 1
  × 1 0 0 1
  -----
    1 1 0 1
   0 0 0 0
  0 0 0 0
 1 1 0 1
  -----
1 1 1 0 1 0 1
    
```

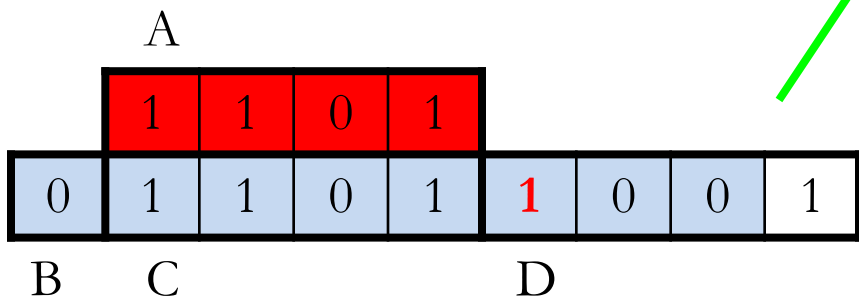
- Παράδειγμα: A=1101, D=1001, προφανώς $n=4$

Βήμα 1:



Βήμα 2:

(α) Το LSD=1 οπότε $A+C \rightarrow C$



(β)

Βήμα 1ο: Τοποθέτησε τους A, D στα αντίστοιχα κελιά και βάλε μηδενικά στα B, C.

Βήμα 2ο: Κάνε τα παρακάτω “n” φορές (βρόχος)

- Αν το LSD του D είναι 1, τότε πρόσθεσε τα A, C. Βάλε το αποτέλεσμα στο C και το κρατούμενο:

(αν υπάρχει) στο B.

(αν δεν υπάρχει) τότε στο B μπαίνει 0

- Μετάθεσε όλα τα ψηφία του ενοποιημένου χώρου BCD μια θέση δεξιά (ολίσθηση)

Βήμα 3ο: Μόλις τελειώσουν οι “n” φορές, το αποτέλεσμα είναι στο CD.

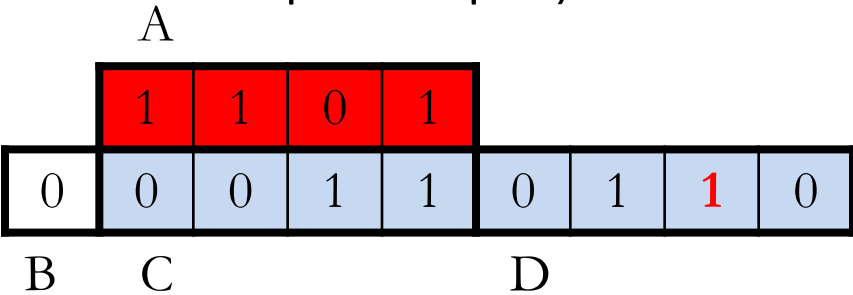
$i=1 \leq n=4.$

Αλγόριθμος πολλαπλασιασμού, (...συν)

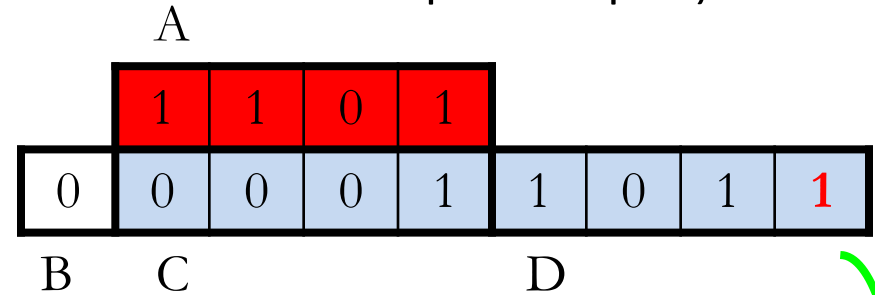
```

    1 1 0 1
  × 1 0 0 1
  -----
    1 1 0 1
   0 0 0 0
  0 0 0 0
 1 1 0 1
  -----
1 1 1 0 1 0 1
  
```

(β) Μετάθεση όλων των ψηφίων του BCD του κατά μια θέση δεξιά



(β) Μετάθεση όλων των ψηφίων BCD κατά μια θέση δεξιά

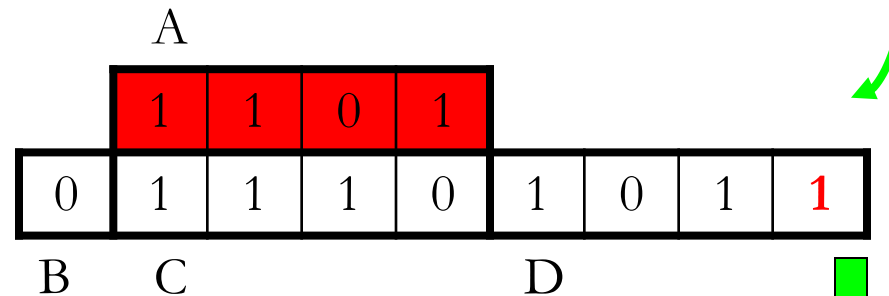
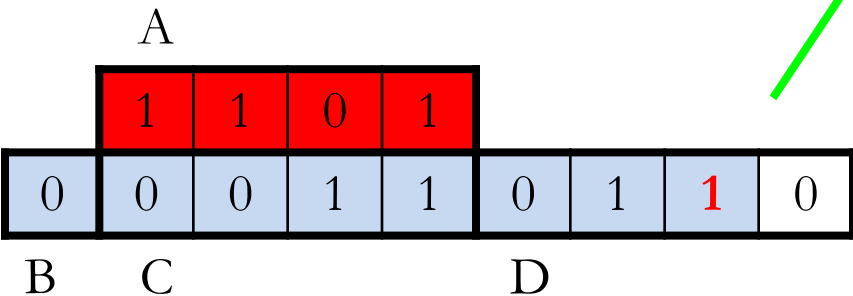


$i=3 \leq n=4.$

$i=4 \leq n=4.$

(α) Το LSD=0 οπότε δεν αλλάζει τίποτα

(α) Το LSD=1 οπότε $A+C \rightarrow C$

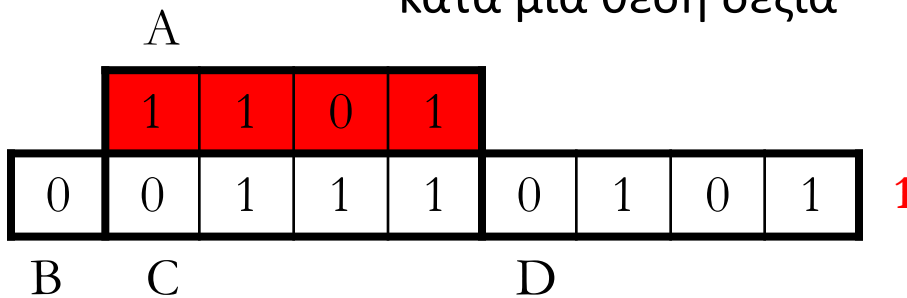


Αλγόριθμος πολλαπλασιασμού, (...συν)

$$\begin{array}{r}
 1101 \\
 \times 1001 \\
 \hline
 1101 \\
 0000 \\
 0000 \\
 1101 \\
 \hline
 1110101
 \end{array}$$

(β) Μετάθεση όλων των ψηφίων του BCD

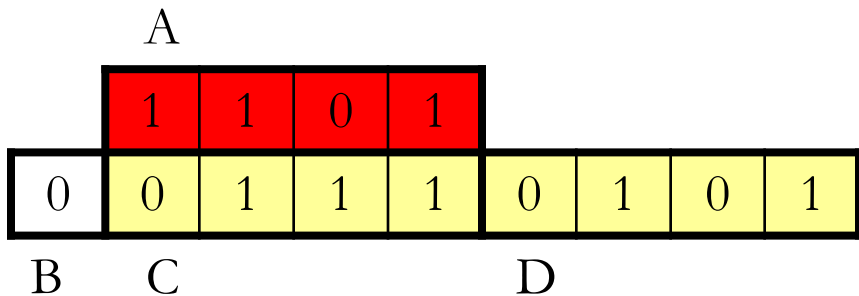
κατά μια θέση δεξιά



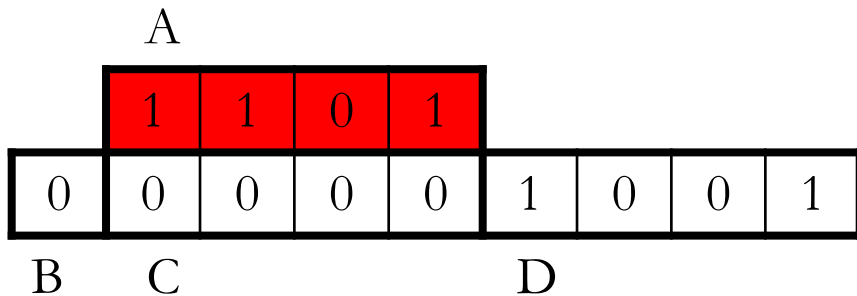
$$\begin{array}{r}
 1101 \\
 \times 1001 \\
 \hline
 1101 \\
 1101 \\
 \hline
 1110101
 \end{array}$$

Βήμα 4:

Το αποτέλεσμα είναι στο CD



Αλγόριθμος πολλαπλασιασμού



Παράδειγμα:

Κάντε τον πολλαπλασιασμό

$$5 \times 6 = ?$$

Με όσο το δυνατόν λιγότερα bits

Βήμα 1ο: Τοποθέτησε τους **A**, **D** στα αντίστοιχα κελιά και βάλε μηδενικά στα **B**, **C**.

Βήμα 2ο: Κάνε τα παρακάτω "**n**" φορές (βρόχος)

- Αν το LSD του **D** είναι 1, τότε πρόσθεσε τα **A**, **C**. Βάλε το αποτέλεσμα στο **C** και το κρατούμενο:

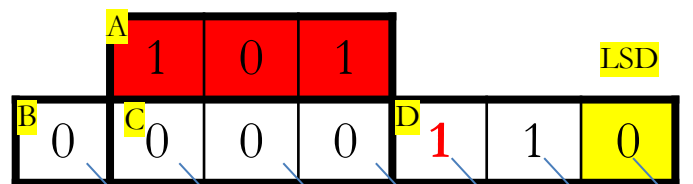
(αν υπάρχει) στο **B**.

(αν δεν υπάρχει) τότε στο **B** μπαίνει 0

- Μετάθεσε όλα τα ψηφία του ενοποιημένου χώρου **BCD** μια θέση δεξιά (ολίσθηση)

Βήμα 3ο: Μόλις τελειώσουν οι "**n**" φορές, το αποτέλεσμα είναι στο **CD**.

Αλγόριθμος πολλαπλασιασμού: 5Χ6



1

End n=1

2

3

End n=2

4

5

End n=3

6

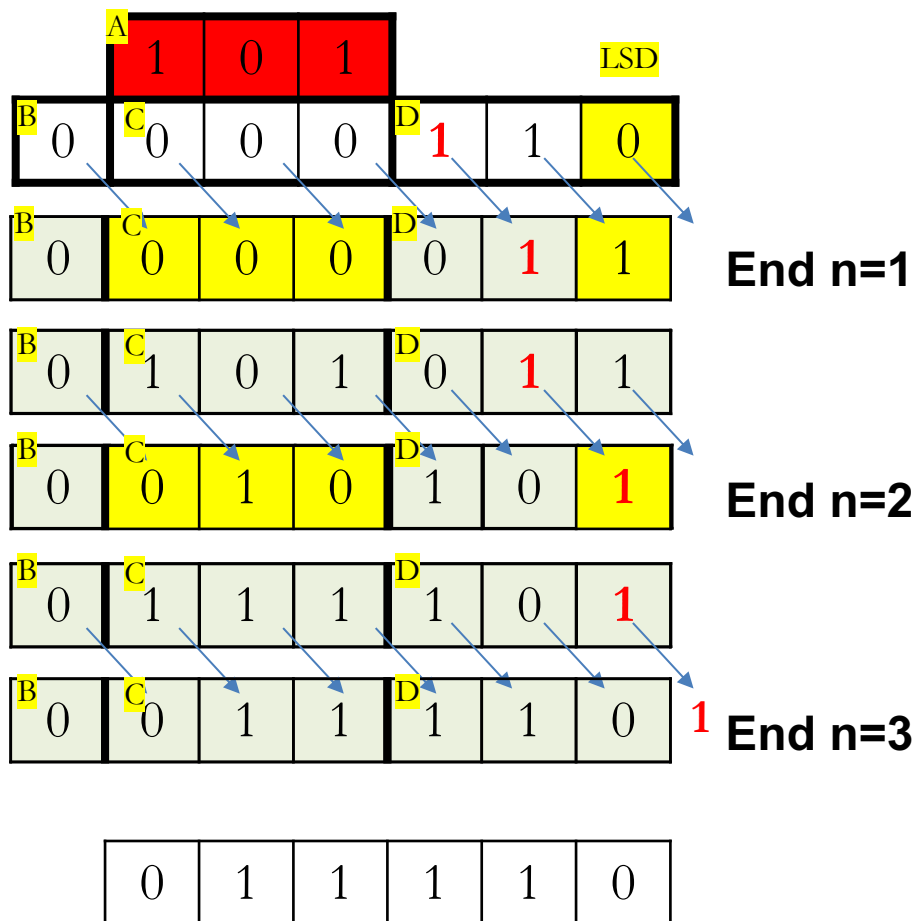
Βήμα 1ο: Τοποθέτησε τους **A**, **D** στα αντίστοιχα κελιά και βάλε μηδενικά στα **B**, **C**.

Βήμα 2ο: Κάνε τα παρακάτω "**n=3**" φορές

- Αν το LSD του **D** είναι 1, τότε πρόσθεσε τα **A**, **C**. Βάλε το αποτέλεσμα στο **C** και το κρατούμενο: (αν υπάρχει) στο **B** (αν δεν υπάρχει) τότε στο **B** μπαίνει
- Μετάθεσε όλα τα ψηφία του ενοποιημένου χώρου **BCD** μια θέση δεξιά (ολίσθηση)

Βήμα 3ο: Μόλις τελειώσουν οι "**n=3**" φορές, το αποτέλεσμα είναι στο **CD**.

Αλγόριθμος πολλαπλασιασμού: 5Χ6



Βήμα 1ο: Τοποθέτησε τους **A**, **D** στα αντίστοιχα κελιά και βάλε μηδενικά στα **B**, **C**.

Βήμα 2ο: Κάνε τα παρακάτω "**n=3**" φορές

- Αν το LSD του **D** είναι 1, τότε πρόσθεσε τα **A**, **C**. Βάλε το αποτέλεσμα στο **C** και το κρατούμενο: (αν υπάρχει) στο **B** (αν δεν υπάρχει) τότε στο **B** μπαίνει
- Μετάθεσε όλα τα ψηφία του ενοποιημένου χώρου **BCD** μια θέση δεξιά (ολίσθηση)

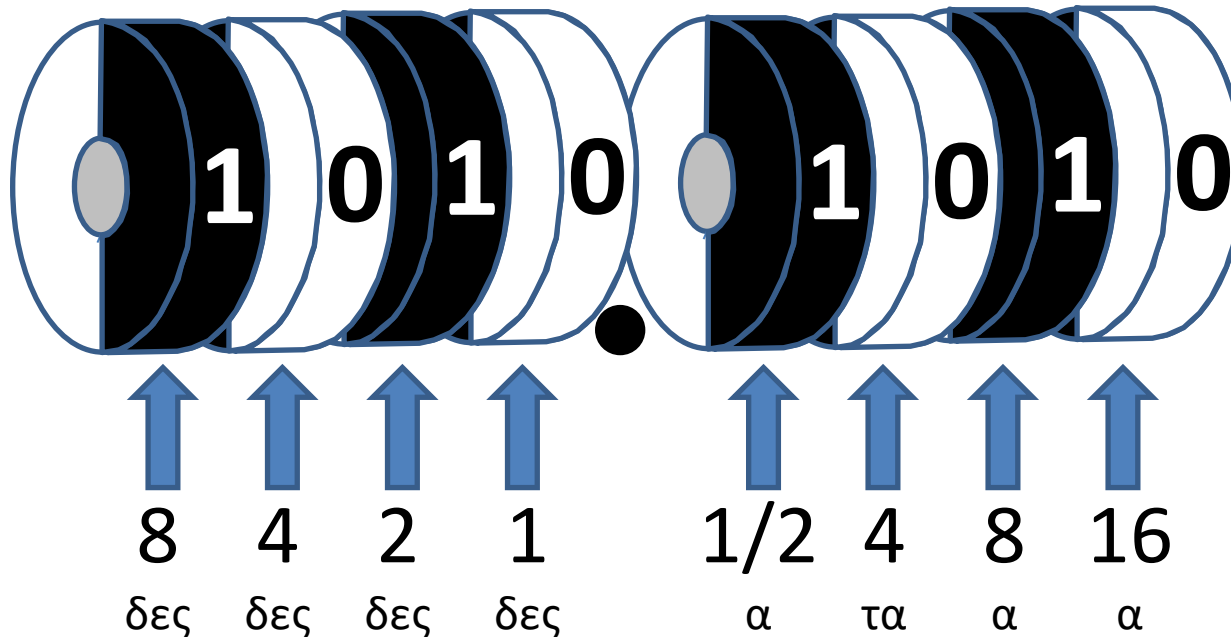
Βήμα 3ο: Μόλις τελειώσουν οι "**n=3**" φορές, το αποτέλεσμα είναι στο **CD**.

Αριθμητική Σταθερής Υποδιαστολής (fixed point arithmetic)

Για μήκος λέξης n bit με παραχωρούμε κάποια bit σε κλασματικό και κάποια bit στο ακέραιο μέρος.

1010.1010

(από 3^ο μάθημα)



$$\begin{aligned} & \left({}^{n-1}D \ {}^{n-2}D \ \dots \ {}^1D \ {}^0D \ . \ {}^{-1}D \ {}^{-2}D \ \dots \ {}^{-m}D \right)_B = \\ & = {}^{n-1}D * B^{n-1} + \dots + {}^0D * B^0 + {}^{-1}D * B^{-1} + \dots + {}^{-m}D * B^{-m} \end{aligned}$$

Αρνητικοί Κλασματικοί: Πρόσημο-μέτρο & c2

- *πρόσημο-μέτρο*: το **MSD** δηλώνει το πρόσημο (π.χ. **0**: θετικό, **1**: αρνητικό). Παράδειγμα:

$$13/32 = 1/4 + 1/8 + 1/32$$

Sign ... 1/2 1/4 1/8 1/16 1/32 ...

1	0	1	1	0	1	...	= -13/32 = 0.40625
---	---	---	---	---	---	-----	--------------------

- *συμπλήρωμα ως προς 2*: Παράδειγμα: -13/32=?

-1 1/2 1/4 1/8 1/16 1/32 ...

0	0	1	1	0	1	...	= 13/32
---	---	---	---	---	---	-----	---------

1	1	0	0	1	0	...	} ←
---	---	---	---	---	---	-----	-----

+					1	...	}
						1	

1	1	0	0	1	1	...	= c ₂ (13/32) = -13/32
---	---	---	---	---	---	-----	-----------------------------------

Αριθμητική Σταθερής Υποδιαστολής σε c2

Για μήκος λέξης n bit με παραχωρούμε κάποια bit σε κλασματικό και κάποια bit στο ακέραιο μέρος ενώ το MSD είναι -2^{m-1} όπου m τα bits του ακεραίου.

ΠΑΡΑΔΕΙΓΜΑ Σταθερής Υποδιαστολής σε c2

- Για $n=8$ σε $c2(.)$ με κανένα κλασματικό μέρος το μέγιστο εύρος τιμών είναι από -128 έως $+127$
- Για $n=8$ σε $c2(.)$ με παραχώρηση 2 ψηφίων σε κλασματικό το εύρος του ακεραίου μέρους των τιμών είναι: -32 έως $+31$ ενώ τα δεκαδικά που μπορεί να παρασταθούν είναι μέχρι τέταρτα.
- ***ΘΑ ΠΡΟΤΙΜΟΥΣΑΜΕ ΝΑ ΕΧΟΥΜΕ ΤΗΝ ΕΥΕΛΙΞΙΑ ΤΟ ΜΕΓΕΘΟΣ ΤΟΥ ΚΛΑΣΜΑΤΙΚΟΥ ΜΕΡΟΥΣ ΝΑ ΕΙΝΑΙ ΜΕΤΑΒΛΗΤΟ***

Πρόβλημα για εξάσκηση

- Σε Η/Υ που
 - χρησιμοποιεί τη σύμβαση $c_2(\cdot)$,
 - το μήκος λέξης είναι 8 bit, και
 - τα 3 bit είναι για το κλασματικό μέρος,
- Να παρασταθούν ως δυαδικοί οι αριθμοί 5.5_{10} και 4.25_{10}
- Να γίνει η πράξη $5.5_{10} - 4.25_{10}$ στο δυαδικό
- Να εξετασθεί (από πλευράς Η/Υ) η ορθότητα της πράξης.

Κινητή Υποδιαστολή

$$2032.21 = 2.03221 \times 10^{+3}$$

$$101.11 = 1.0111 \times 2^{+2}$$

- **Αριθμητική Κινητής Υποδιαστολής (floating point arithmetic):**
Κάθε αριθμός εκφράζεται με τη παρακάτω παράσταση όπου:

$$\overbrace{\left(\begin{array}{c} \text{sign} \\ \pm \\ m_1 \ m_2 \ \cdots \ m_{n_m-1} \end{array} \right)}^{\text{mantissa}} \cdot 2^{\overbrace{(e_1 e_2 \cdots e_{n_e})}_2^{\text{exponent}}}$$

s: πρόσημο

$m_1 \dots m_{n_m-1}$: συντελεστής (mantissa), και

$(e_1 \dots e_{n_e})_2$: ο εκθέτης (exponent) σε πολωμένη μορφή.

Πόλωση - Πλεονασμός κατά K

- Στο δυαδικό σύστημα ένας 8-bit αριθμός περιέχει 2^8 καταστάσεις που αντιστοιχούν σε αριθμούς από το 0 \rightarrow 255
- *Πόλωση: Ορίζουμε τον αριθμό «x» ως «x+Πόλωση»*
πχ για Πόλωση=128
 - Το «000» θα αντιστοιχούσε σε «128»
 - Το «-128» θα αντιστοιχούσε σε «0» (δηλ. Min)
 - Το «+127» θα αντιστοιχούσε σε «255» (δηλ. max)
- Π.χ Να παρασταθεί ο αριθμός 5 σε 8-bit δυαδικό σύστημα με πόλωση 128

Παράδειγμα Πόλωσης

Να παρασταθεί ο αριθμός 5 σε 8-bit δυαδικό σύστημα με πόλωση 128

- Το «5» αντιστοιχεί σε 133 (=5+Πόλωση)
- 133 --> 1000 0101

Παράδειγμα Πόλωσης

Να παρασταθεί ο αριθμός -20 σε 8-bit δυαδικό σύστημα με πόλωση 128

- Το «-20» αντιστοιχεί σε 108 ($= -20 + \text{Πόλωση}$)
- -20 --> 0110 1100

Κινητή Υποδιαστολή: το πρότυπο IEEE

Τύπος Δεδομένων	Bits	Ελάχιστη	Μέγιστη
Ακέραιες Απρόσημες Τιμές			
unsigned char	8	0	255
unsigned short	16	0	65535
unsigned int	32	0	4294967295
unsigned long	32	0	4294967295
unsigned long long	64	0	18446744073709551615
Ακέραιες Προσημασμένες Τιμές			
char	8	-128	127
short	16	-32768	32767
int	32	-2147483648	2147483647
long	32	-2147483648	2147483647
long long	64	-9223372036854775808	9223372036854775807
Δεκαδικές Τιμές (κινητής υποδιαστολής)			
float	32	$\pm 1.17549e-038$	$\pm 3.40282e+038$
double	64	$\pm 2.22507e-308$	$\pm 1.79769e+308$

Μεταβλητές στη C

Όταν είναι επιθυμητό να αποθηκευθεί στη μνήμη του Η/Υ το αποτέλεσμα μιας πράξης ή ένας αριθμός χρησιμοποιούνται οι μεταβλητές. Αυτές μοιάζουν με τις γνωστές μεταβλητές των μαθηματικών, όμως στους Η/Υ η μεταβλητή συνδέεται με 3 πράγματα:

- Την ονομασία της
- Τον τύπο δεδομένων της
- Την τιμή της μεταβλητής

Αυτό φαίνεται και στη δήλωση μίας μεταβλητής. Π.χ.:

`double x = 12.34;`

Τιμή της μεταβλητής μαζί με τον τύπο δεδομένων της καθορίζει το δυαδικό περιεχόμενο της μνήμης.

Παράδειγμα

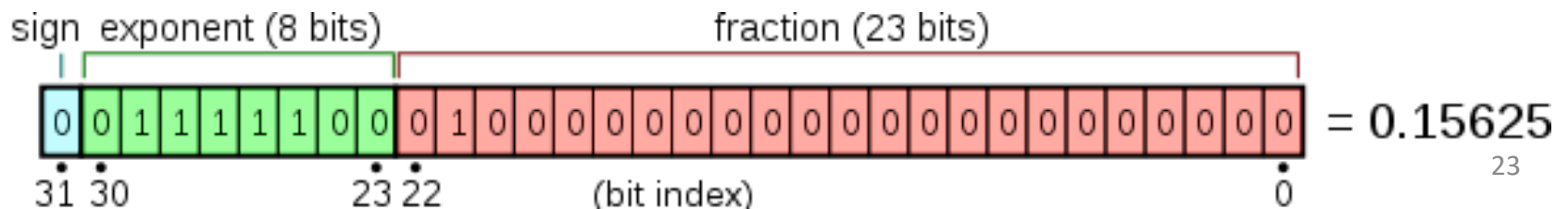
- Να παρασταθεί ο αριθμός 5,75 σε μορφή κινητής υποδιαστολής με λέξη 32 bits και αφιερώνει:

→ 1 bit για το πρόσημο

→ 8 bits για τον πολωμένο εκθέτη κατά 127 (n_e)
και

→ 23 bits για το συντελεστή

(single precision format)



Παράδειγμα 5,75

- $5.75 \rightarrow 5 (101) + 0,75 (.11)$
- $5.75 \rightarrow 101.11$ ή 1.0111×2^2
- 2 με πόλωση 127 ισοδυναμεί με 129
- Mantissa:

0	1	1	1
---	---	---	---

Exponent (8 bits)

1	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Sign (1 bit) Exponent (8 bits)

0

Mantissa (23bits)

0	0	0	0	0
---	---	---	---	---	----	----	----

Παράδειγμα float $x = -27.75$ (single precision format)

- Να παρασταθεί ο αριθμός $-27,75$ σε μορφή κινητής υποδιαστολής με λέξη 32 bits και αφιερώνει:
 - 1 bit για το πρόσημο
 - 8 bits για τον πολωμένο εκθέτη κατά 127 (n_e) και
 - 23 bits για το συντελεστή



END
