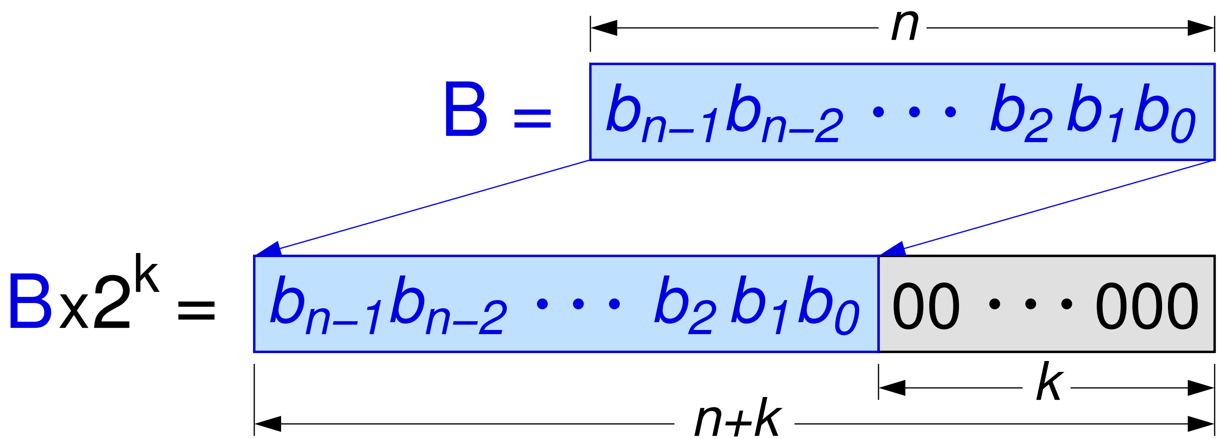
Πολλαπλασιασμός, Διαίρεση, Υπόλοιπο επί/διά δύναμη του 2, Συστροφή

06α (§ 6.1 - 6.2) – 25-30 Οκτ. 2023 – Μανόλης Κατεβαίνης

Πολλαπλασιασμός x δύναμη του 2: Αριστερή Ολίσθηση

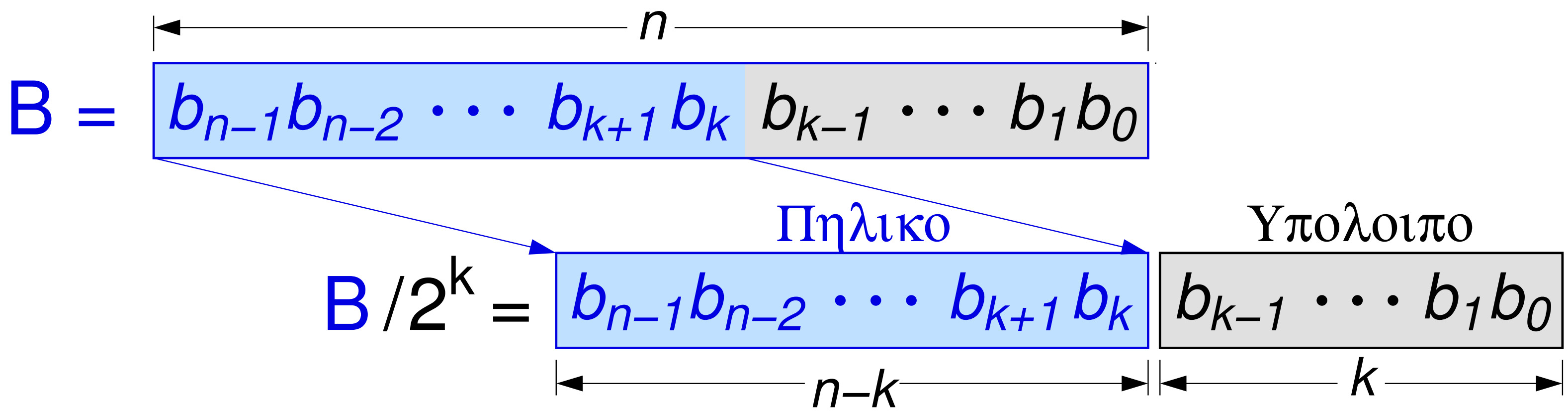


- $B = b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \ldots + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$

$$\Rightarrow B \times 2^k = b_{n-1} \times 2^{n+k-1} + b_{n-2} \times 2^{n+k-2} + \ldots + b_2 \times 2^{k+2} + b_1 \times 2^{k+1} + b_0 \times 2^k +$$
$$+ 0 \times 2^{k-1} + 0 \times 2^{k-2} + \ldots + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$

- «Αριστερή Ολίσθηση» (“Left Shift”) κατά *k* bits

Διαίρεση διά δύναμη του 2: Δεξιά Ολίσθηση

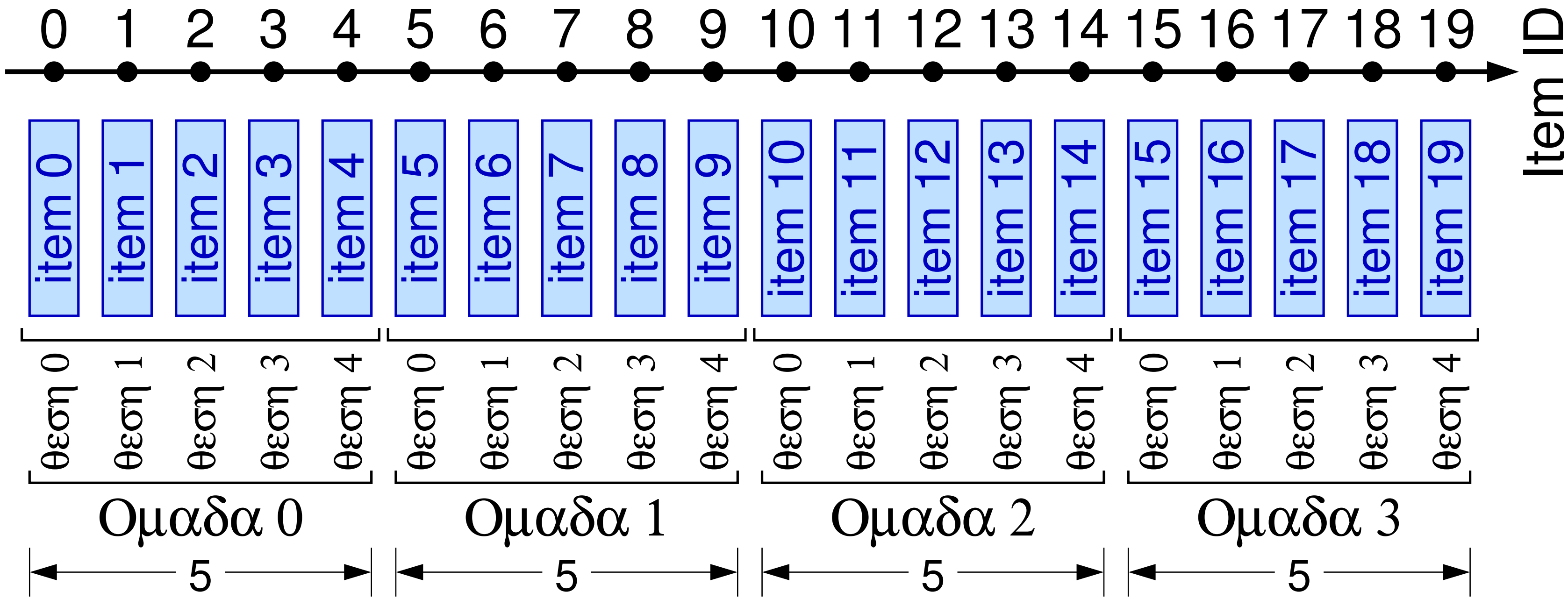


- $B = b_{n-1}\times 2^{n-1} + b_{n-2}\times 2^{n-2} + \ldots + b_{k+1}\times 2^{k+1} + b_k\times 2^k +$
  $+ b_{k-1}\times 2^{k-1} + \ldots + b_1\times 2^1 + b_0\times 2^0$

$\Rightarrow B/2^k = b_{n-1}\times 2^{n-k-1} + b_{n-2}\times 2^{n-k-2} + \ldots + b_{k+1}\times 2^1 + b_k\times 2^0 +$
$+ (b_{k-1}\times 2^{k-1} + \ldots + b_1\times 2^1 + b_0\times 2^0) / 2^k$

- «Δεξιά Ολίσθηση» (“Right Shift”) κατά *k* bits

Πηλίκο-Υπόλοιπο: μοίρασμα στοιχείων σε Ομάδες



- Στοιχείο *N:* σε ποιάν Ομάδα και σε ποιά θέση μέσα της;
- Ομάδες μεγέθους *Δ* ⇒ *Δ* είναι ο διαιρέτης, εδώ *Δ*=5
- Ομάδα = ***Πηλίκο*** (*N/Δ*)
- Θέση εντός ομάδας = ***Υπόλοιπο*** (*N/Δ*)

Πηλίκο-Υπόλοιπο: μοίρασμα σε Γραμμές-Στήλες

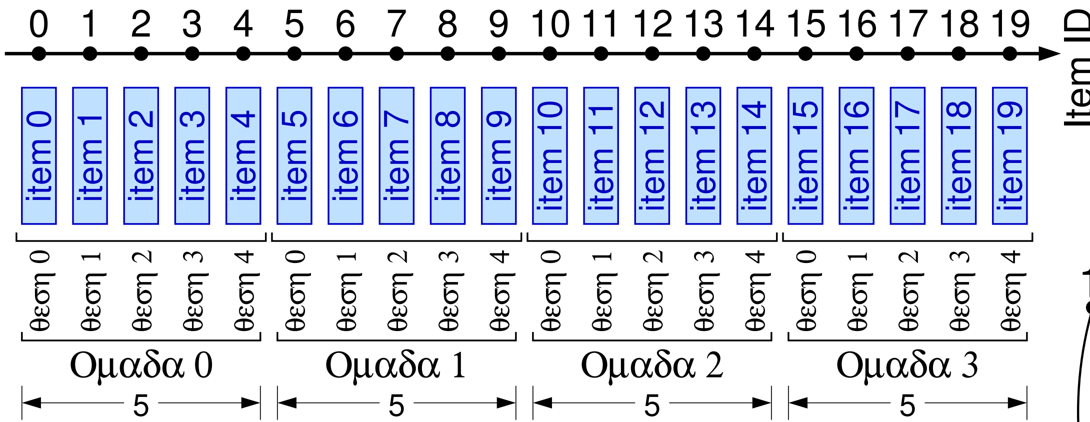
Υπολοιπο →

Πηλικο ↓

	Στηλη 0	Στηλη 1	Στηλη 2	Στηλη 3	Στηλη 4
Γραμμη 0	item 0	item 1	item 2	item 3	item 4
Γραμμη 1	item 5	item 6	item 7	item 8	item 9
Γραμμη 2	item 10	item 11	item 12	item 13	item 14
Γραμμη 3	item 15	item 16	item 17	item 18	item 19

- Γραμμές μεγέθους *Δ* ⇒ *Δ* είναι ο διαιρέτης, εδώ *Δ*=5
- Το στοιχείο *N* είναι στην Ομάδα = Γραμμή = ***Πηλίκο*** (*N/Δ*),
- στη θέση = Στήλη = ***Υπόλοιπο*** (*N/Δ*)

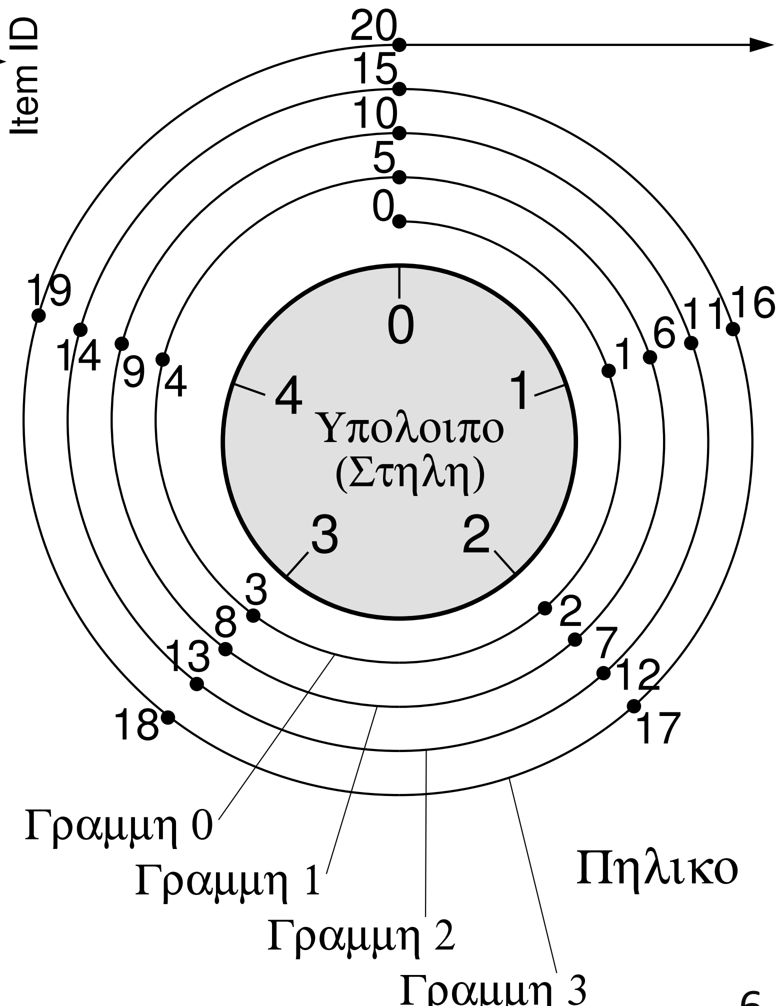
Πηλίκο-Υπόλοιπο: Συστροφή του Άξονα των Αριθμών



Υ π ο λ ο ι π ο

Π η λ ι κ ο

	Στηλη 0	Στηλη 1	Στηλη 2	Στηλη 3	Στηλη 4
Γραμμη 0	item 0	item 1	item 2	item 3	item 4
Γραμμη 1	item 5	item 6	item 7	item 8	item 9
Γραμμη 2	item 10	item 11	item 12	item 13	item 14
Γραμμη 3	item 15	item 16	item 17	item 18	item 19



- Σαν να τυλίγουμε τον άξονα των αριθ-μών σε ένα καρούλι με περίμετρο = Δ, όσο το μέγεθος της ομάδας (Διαιρέτης)

Εφαρμογές ($\Delta=2^k$): Μνήμες-υποδιαιρέσεις, αρν. αριθ.

- Όταν ο διαιρέτης Δ = δύναμη του 2, η διαίρεση είναι τετριμένη: → *bit selection*

Υπολοιπο

Πηλικο

	Στηλη 0	Στηλη 1	Στηλη 2	Στηλη 3	Στηλη 4
Γραμμη 0	item 0	item 1	item 2	item 3	item 4
Γραμμη 1	item 5	item 6	item 7	item 8	item 9
Γραμμη 2	item 10	item 11	item 12	item 13	item 14
Γραμμη 3	item 15	item 16	item 17	item 18	item 19

- Εφαρμογές:
- Bytes μέσα στις Λέξεις της μνήμης
- Λέξεις μέσα σε Γραμμές (Blocks) Κρυφής μνήμης (Cache)
- Λέξεις μέσα σε Σελίδες (Pages) Εικονικής Μν. (Virtual M.)
- Λέξεις μέσα στα Chips που απαρτίζουν Μεγάλη Μνήμη
- Συστροφή (modulo arithmetic): ιδέα για αναπαρ. αρνητικών

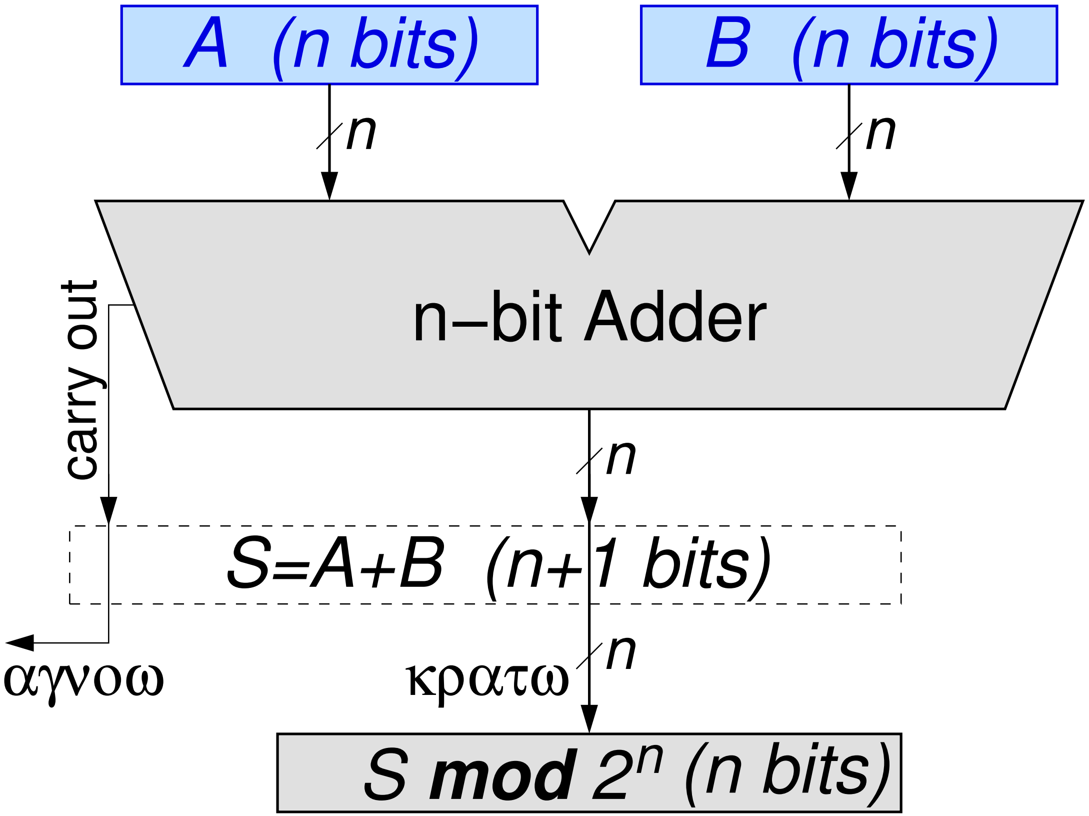
Γιατί μετράμε τα *N* στοιχεία από το *0* έως το *N-1*

- Γιά να ισχύει η προηγούμενη ιδιότητα πηλίκου-υπολοίπου ***πρέπει*** τα *N* στοιχεία να τα μετράμε από το *0* έως το *N-1*
- Παράδειγμα «σωστής» μέτρησης:
  - Ξενοδοχείο με 500 δωμάτια σε 5 επίπεδα:
  - Όροφος *0*: Δωμάτια *000* έως *099* (Ισόγειο)
  - Όροφος *1*: Δωμάτια *100* έως *199*
  - Όροφος *2*: Δωμάτια *200* έως *299*
  - Όροφος *3*: Δωμάτια *300* έως *399*
  - Όροφος *4*: Δωμάτια *400* έως *499*

Παράδειγμα «λάθος» μέτρησης, από το *1* έως το *N*

- 2100 χρόνια σε 21 αιώνες από 100 χρόνια ανά αιώνα:
  - 1^{ος} αιώνας: έτη *1* έως και *100*
  - …
  - 19^{ος} αιώνας: έτη *1801* έως και *1900*
  - 20^{ος} αιώνας: έτη *1901* έως και *2000*
  - 21^{ος} αιώνας: έτη *2001* έως και *2100*
- Ο αιώνας “*A*” περιέχει 99 χρόνια που η αρίθμησή τους ξεκινά με “*A-1*”, και ένα έτος με αριθμό “*A00*” (!)

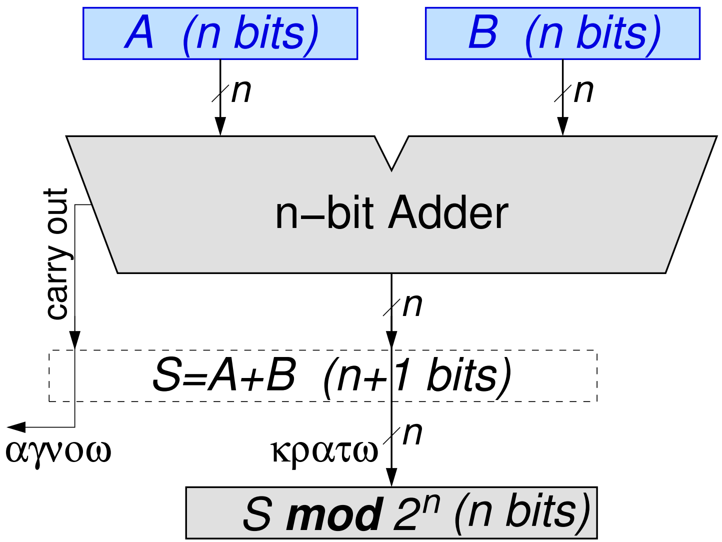
Πρόσθεση κρατώντας μόνον *n* bits: *Συστροφή*

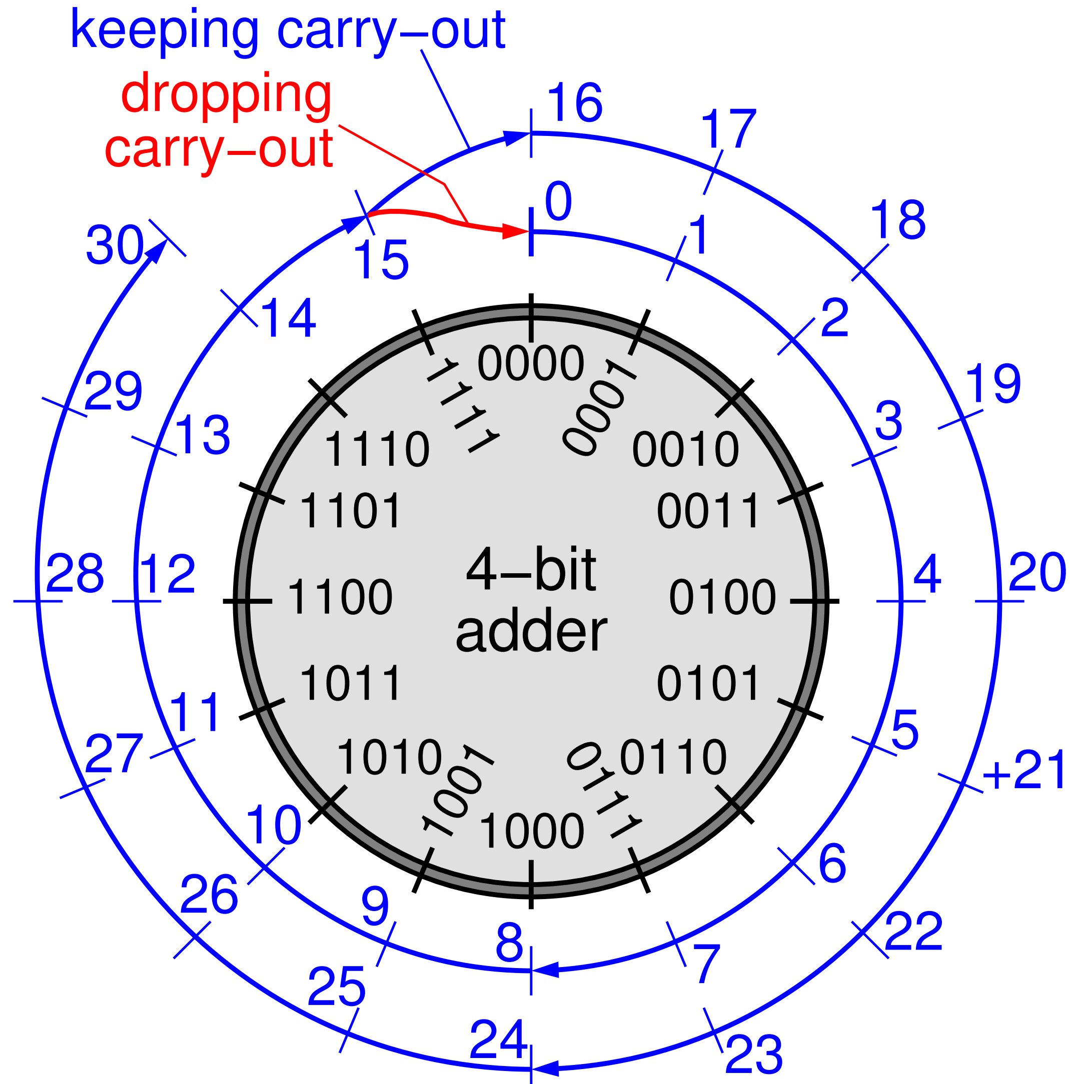


- Δύο αιτίες:
- Δεν χωρά το πρόσ-θετο bit σε μνήμη / καταχωρητές
- Για «πολύ μεγάλα» αποτελέσματα, η αγνόηση του carry-out ισοδυναμεί με «μείον 2^n»

⇒ δυνατότητα αναπαράστασης & πράξεων με αρνητικούς

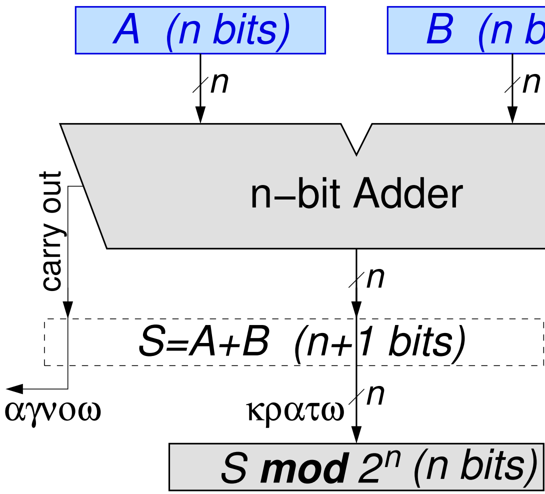
Carry-out – Συστροφή



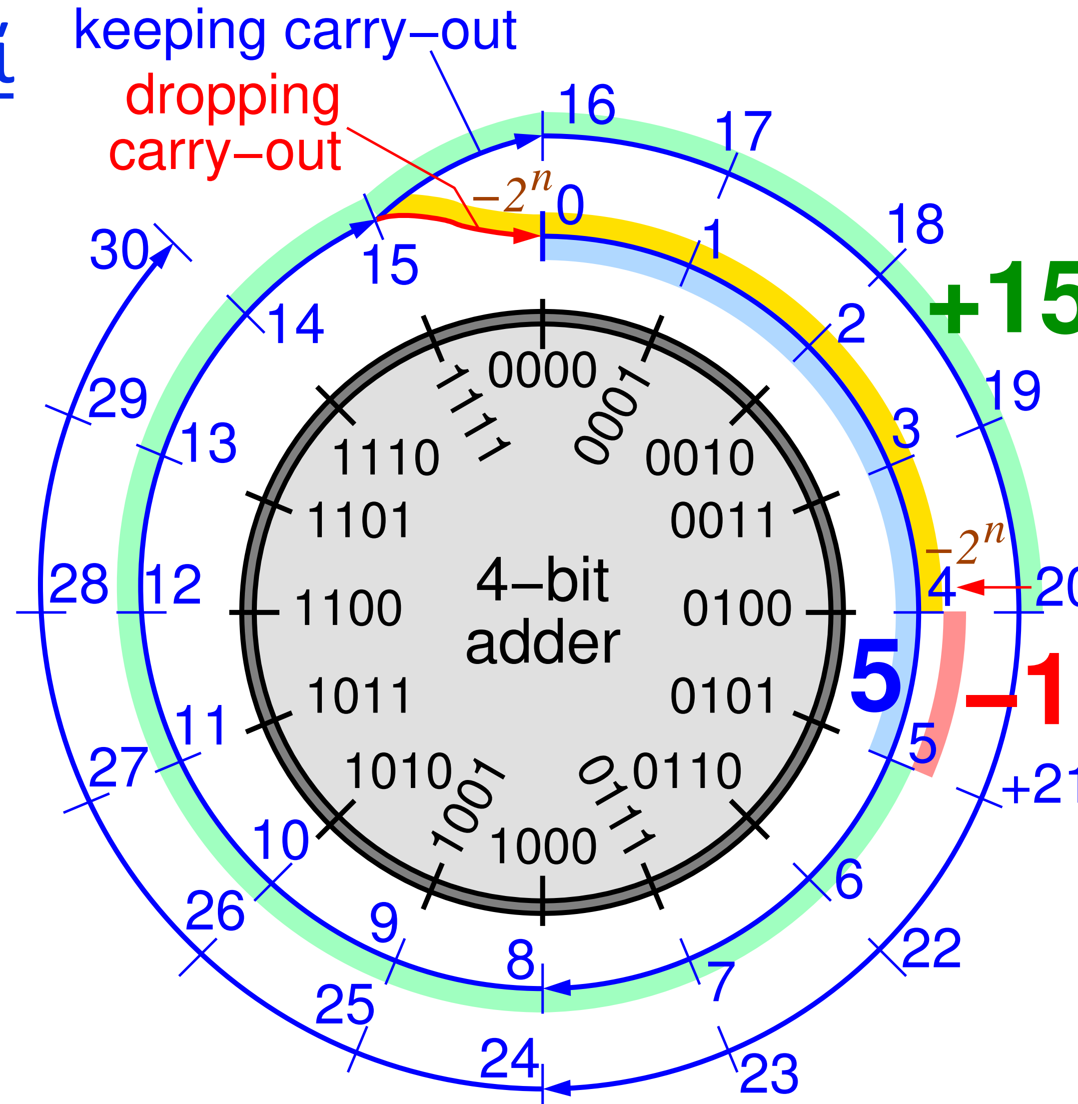


- Χωρίς το carry-out: n least-significant bits = υπόλοιπο διαίρεσης δια 2^n (εδώ: n=4)

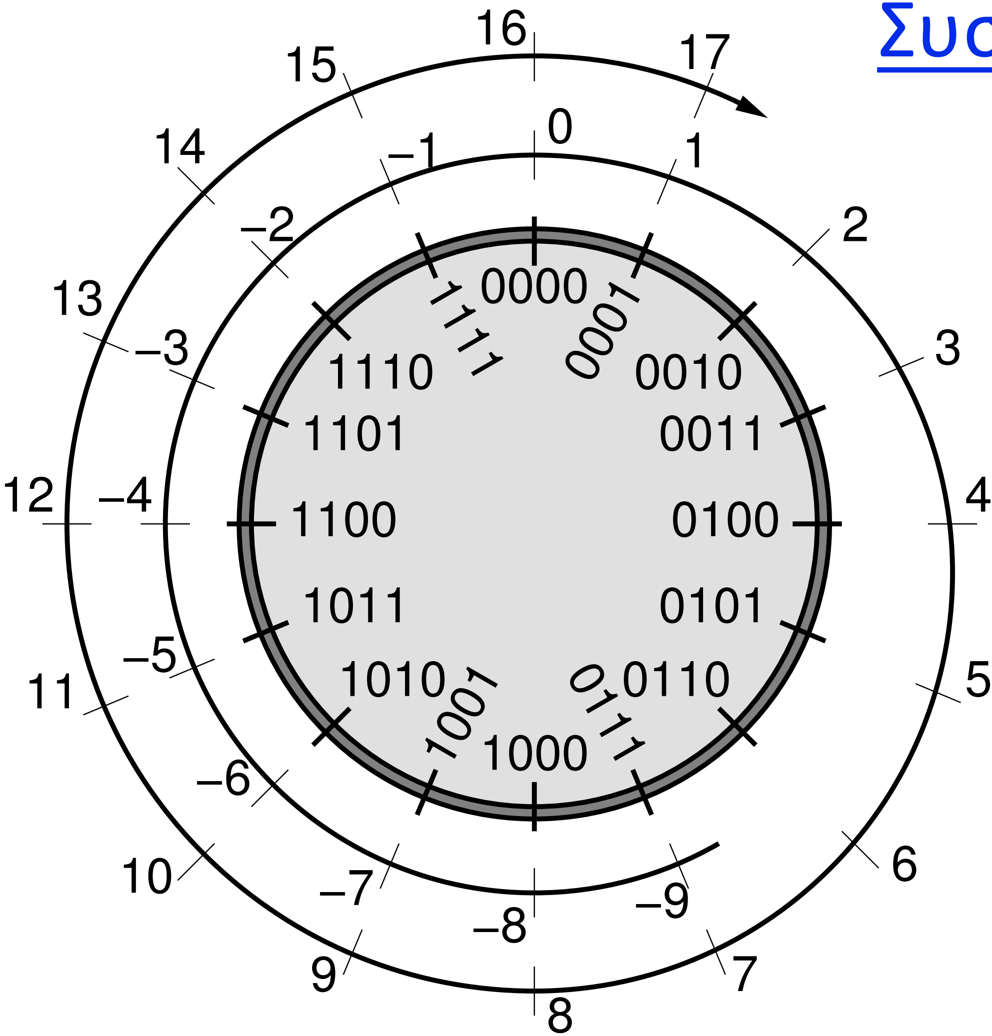
Συστροφή – Αρνητικοί



- 5 + 15 – 2^4 = 5 + (-1)
- Μεγάλοι θετικοί με συστροφή μοιάζουν με αρνητικούς!



Συστροφή και Αρνητικοί Αρ.



- Πρόσθεση <u>*mod*</u> 2^n = συστροφή
- Όπως επεκτείνουμε προς τα δεξιά τον άξονα των αριθμών, μπορούμε να τον επεκτείνουμε και αριστερά, προς τους αρνητικούς αριθμούς