Επανάληψη 2: Αρίθμηση Ακεραίων, Προσθαφαιρέτες

Φθινόπωρο 2020 – Μανόλης Κατεβαίνης

Αριθμοί σε Βάσεις 10, 2, 8, 16

- Δεκαδικός (Decimal) – ψηφία $d_i \in \{0, 1, 2, 3, 4, 5,6, 7, 8, 9\}$:
  - $d_{n-1}\ldots d_2 d_1 d_0 = d_{n-1}\times 10^{n-1} + \ldots + d_2\times 10^2 + d_1\times 10^1 + d_0\times 10^0$
- Δυαδικός (Binary) – ψηφία $b_i \in \{0, 1\}$:
  - $b_{n-1}\ldots b_2 b_1 b_0 = b_{n-1}\times 2^{n-1} + \ldots + b_2\times 2^2 + b_1\times 2^1 + b_0\times 2^0$
- Οκταδικός (Octal) – ψηφία $c_i \in \{0, 1, 2, 3, 4, 5,6, 7\}$:
  - $c_{n-1}\ldots c_2 c_1 c_0 = c_{n-1}\times 8^{n-1} + \ldots + c_2\times 8^2 + c_1\times 8^1 + c_0\times 8^0$
- Δεκαεξαδικός (Hexadecimal – ενίοτε και “hex”):
  - ψηφία $h_i \in \{0, 1, 2, 3, 4, 5,6, 7, 8, 9, A, B, C, D, E, F\}$
  - $h_{n-1}\ldots h_2 h_1 h_0 = h_{n-1}\times 16^{n-1} + \ldots + h_2\times 16^2 + h_1\times 16^1 + h_0\times 16^0$

Dec	Binary	Oct	Hex
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Dec	Binary	Oct	Hex
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F

Dec	Binary	Oct	Hex
32	100000	40	20
33	100001	41	21
34	100010	42	22
…	…	…	…
39	100111	47	27
40	101000	50	28
41	101001	51	29
42	101010	52	2A
…	…	…	…
46	101110	56	2E
47	101111	57	2F
48	110000	60	30
…	…	…	…
62	111110	76	3E
63	111111	77	3F
64	1000000	100	40

Πόσοι και Ποιοί ακέραιοι (unsigned) χωρούν σε *n* bits

- Πόσοι ακέραιοι; *n* bits $\rightarrow 2^n$ συνδυασμοί $\rightarrow 2^n$ αριθμοί
- Ποιοί ακέραιοι; Από 0 έως και 2^n-1, δηλ. 2^n το πλήθος
- Ο μεγαλύτερος από αυτούς είναι ο: 11…111, δηλαδή ο:

$$1\times2^{n-1} + 1\times2^{n-2} + \ldots + 1\times2^2 + 1\times2^1 + 1\times2^0 = 2^n-1$$

- Απόδειξη βάσει της ταυτότητας:

$$(\alpha^n-\beta^n) = (\alpha-\beta)\cdot(\alpha^{n-1}\beta^0 + \alpha^{n-2}\beta^1 + \ldots + \alpha^1\beta^{n-2} + \alpha^0\beta^{n-1})$$

γιά $\alpha=2$ και $\beta=1$

Παράσταση N σε βάση **B**: Ύπαρξη, Μοναδ., Εύρεση

- Γιά δοθέντα ακέραιο $N \geq 0$ και (ακέραια) βάση $\mathbf{B} \geq 2$, αναζητούμε ψηφία (ακεραίους) d_i με $0 \leq d_i \leq \mathbf{B}-1$, ούτως ώστε:
  - $N = d_{n-1} \times \mathbf{B}^{n-1} + \ldots + d_2 \times \mathbf{B}^2 + d_1 \times \mathbf{B}^1 + d_0 \times \mathbf{B}^0$

$\Rightarrow (N/\mathbf{B}) = d_{n-1} \times \mathbf{B}^{n-2} + \ldots + d_2 \times \mathbf{B}^1 + d_1 \times \mathbf{B}^0 + (d_0/\mathbf{B})$

- έστω $\Pi = (d_{n-1} \times \mathbf{B}^{n-2} + \ldots + d_2 \times \mathbf{B}^1 + d_1 \times \mathbf{B}^0) \longrightarrow$ ακέραιος
- και $Y = d_0$ όπου ξέρουμε ότι $0 \leq d_0 \leq \mathbf{B}-1$ άρα και: $0 \leq Y \leq \mathbf{B}-1$

$\Rightarrow (N/\mathbf{B}) = \Pi + (Y/\mathbf{B})$ [πηλίκο-υπόλοιπο διαίρεσης]

- Άρα Π και $Y = d_0$ υπάρχουν και είναι μοναδικά

$\Rightarrow$ βρήκαμε το d_0 και Π, και από Π βρίσκουμε d_1 κ.ο.κ.

Εφαρμογή 1: Μετατροπή από Δεκαδικό σε Δυαδικό

- $N = 39_{10}$ να γραφτεί στο δυαδικό
- Χρειάζονται 6 bits: $2^6=64$ αριθμοί – από το 0 έως και το 63
  - 5 bits δεν αρκούν: αναπαριστούν $2^5=32$ αρ. – από 0 έως και το 31
- $N = b_5\times2^5 + b_4\times2^4 + b_3\times2^3 + b_2\times2^2 + b_1\times2^1 + b_0\times2^0$
- $N/2 = 39/2 = 19 + (1/2) \Rightarrow \Pi_1 = 19 =(b_5b_4b_3b_2b_1)$, $Y_0 = 1 = b_0$
- $\Pi_1/2 = 19/2 = 9 + (1/2) \Rightarrow \Pi_2 = 9 =(b_5b_4b_3b_2)$, $Y_1 = 1 = b_1$
- $\Pi_2/2 = 9/2 = 4 + (1/2) \Rightarrow \Pi_3 = 4 =(b_5b_4b_3)$, $Y_2 = 1 = b_2$
- $\Pi_3/2 = 4/2 = 2 + (0/2) \Rightarrow \Pi_4 = 2 =(b_5b_4)$, $Y_3 = 0 = b_3$
- $\Pi_4/2 = 2/2 = 1 + (0/2) \Rightarrow \Pi_5 = 1 =(b_5)$, $Y_4 = 0 = b_4$

$\Rightarrow$ Ο αριθμός N στο δυαδικό = 100111_2

Άλλος τρόπος: Ποιές δυνάμεις του 2 χωράνε;

- $N = \underline{39}_{10}$ να γραφτεί στο δυαδικό
- $N = 39 = 32 + 7 = 32 + 4 + 3 = 32 + 4 + 2 + 1 =$

  $= 2^5 + 2^2 + 2^1 + 2^0 =$

  $= \mathit{1}\times2^5 + \mathit{0}\times2^4 + \mathit{0}\times2^3 + \mathit{1}\times2^2 + \mathit{1}\times2^1 + \mathit{1}\times2^0$
- Άρα: $N = \underline{100111}_2$ στο δυαδικό

Ομοίως:

- $M = \underline{708}_{10}$ να γραφτεί στο δυαδικό
- $M = 708 = 512 + 196 = 512 + 128 + 68 = 512 + 128 + 64 + 4$

  $= 2^9 + 2^7 + 2^6 + 2^2 \quad \Rightarrow \quad M = \underline{1011000100}_2$

Εφαρμογή 2: Μετατροπή από Δυαδικό σε Οκταδικό

- $N = b_{n-1} \times 2^{n-1} + \ldots + b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$
- Μετατροπή στο Οκταδικό – βάση = 8 = 2^3
- $N/8 = N/2^3 = b_{n-1} \times 2^{n-4} + \ldots + b_3 \times 2^0 + (b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0)/8$
- Οι όροι από αριστερά μέχρι και b_3 – που είχαν δυνάμεις του 2 με εκθέτη ≥3 – παραμένουν ακέραιοι μετά τη διαίρεση διά $8=2^3$, άρα σχηματίζουν το *Πηλίκο*
- Ο όρος $(b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0)$, δηλ. ένας τρίμπιτος αριθμός, παίρνει τιμές από 0 έως και 7, άρα είναι το *Υπόλοιπο*

⇒ Από δεξιά, ανά 3 τα bits, δίνουν από ένα οκταδικό ψηφίο

- Ομοίως, γιά Δεκαεξαδικό: από δεξιά, ανά 4 τα bits…

Παραδείγματα: Δυαδικό-Οκταδικό-Δεκαεξαδικό

Από Δυαδικό (20 bits) σε Οκταδικό						
11	010	111	001	100	110	010
3	2	7	1	4	6	2

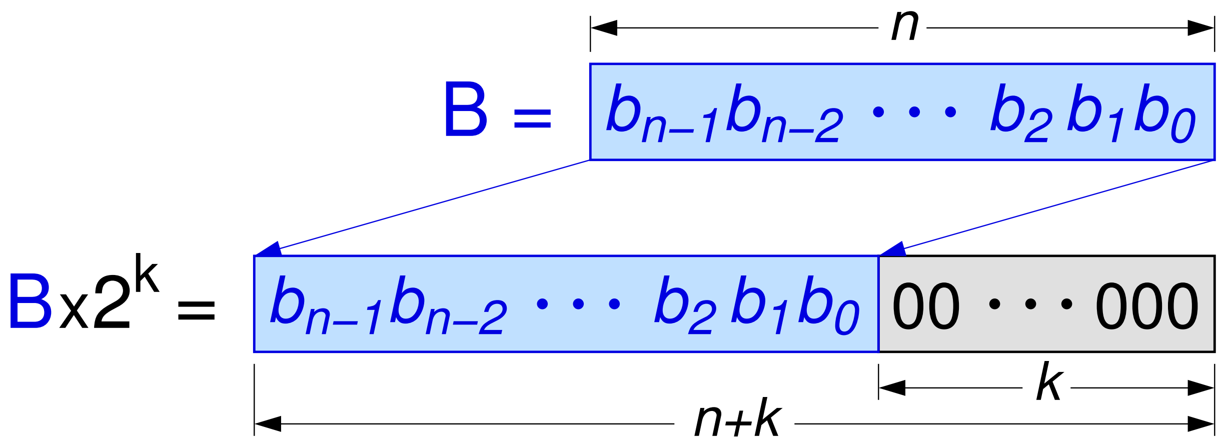
Από Οκταδ. σε Δυαδικό (16 b.)					
1	7	7	0	7	4
1	111	111	000	111	100

Από Δυαδικό (32 bits – 4 Bytes) σε Δεκαεξαδικό							
0000	1010	0110	1101	0010	1111	1011	1000
0	A	6	D	2	F	B	8

Από Δεκαεξαδικό (4 Bytes – 8 ψηφία) σε Δυαδικό							
D	E	A	D	B	E	E	F
1101	1110	1010	1101	1011	1110	1110	1111

- Από δεξιά οι ομάδες των bits, κι όπου φτάσουν

Πολλαπλασιασμός x δύναμη του 2: Αριστερή Ολίσθηση

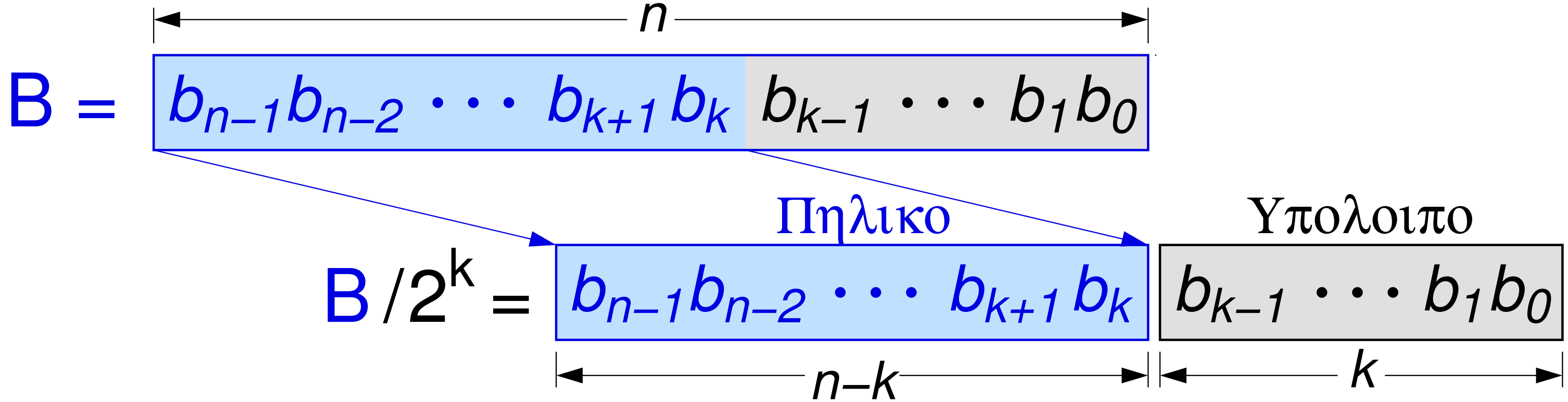


- $B = b_{n-1}\times 2^{n-1} + b_{n-2}\times 2^{n-2} + \ldots + b_2\times 2^2 + b_1\times 2^1 + b_0\times 2^0$

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

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

Διαίρεση διά δύναμη του 2: Δεξιά Ολίσθηση



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

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

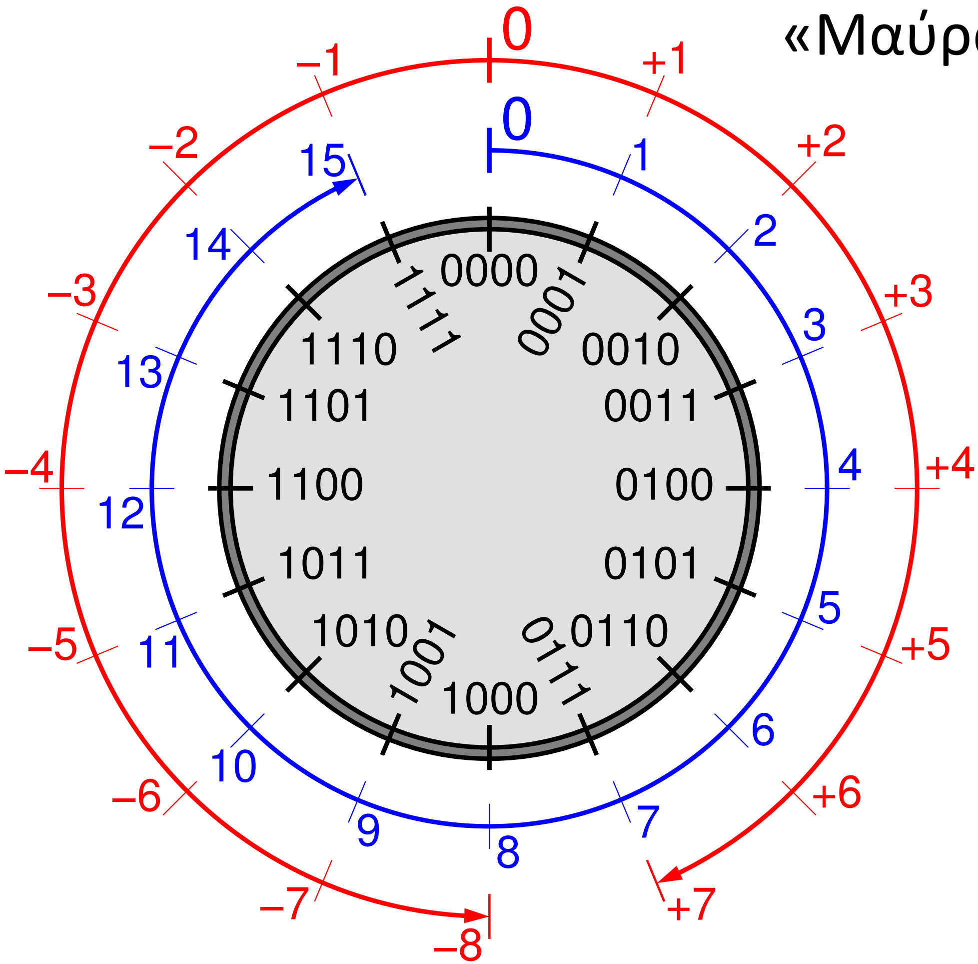
- «Δεξιά Ολίσθηση» (“Right Shift”) κατά *k* bits

Εφαρμογή Πηλίκου-Υπολ: Μεγάλη Μνήμη από μικρές

- Διαμοιρασμός στοιχείων σε Ομάδες των *Δ* (διαιρέτης – εδώ *Δ*=5)
- Το στοιχείο *N* σε ποιάν Ομάδα και σε ποιά Θέση βρίσκεται;
- Ομάδα = *Πηλίκο* (*N/Δ*)
- Θέση εντός ομάδας = *Υπόλοιπο* (*N/Δ*)

Εφαρμογή:
Μεγάλη Μνήμη
κατασκευασμένη από
πολλές μικρότερες

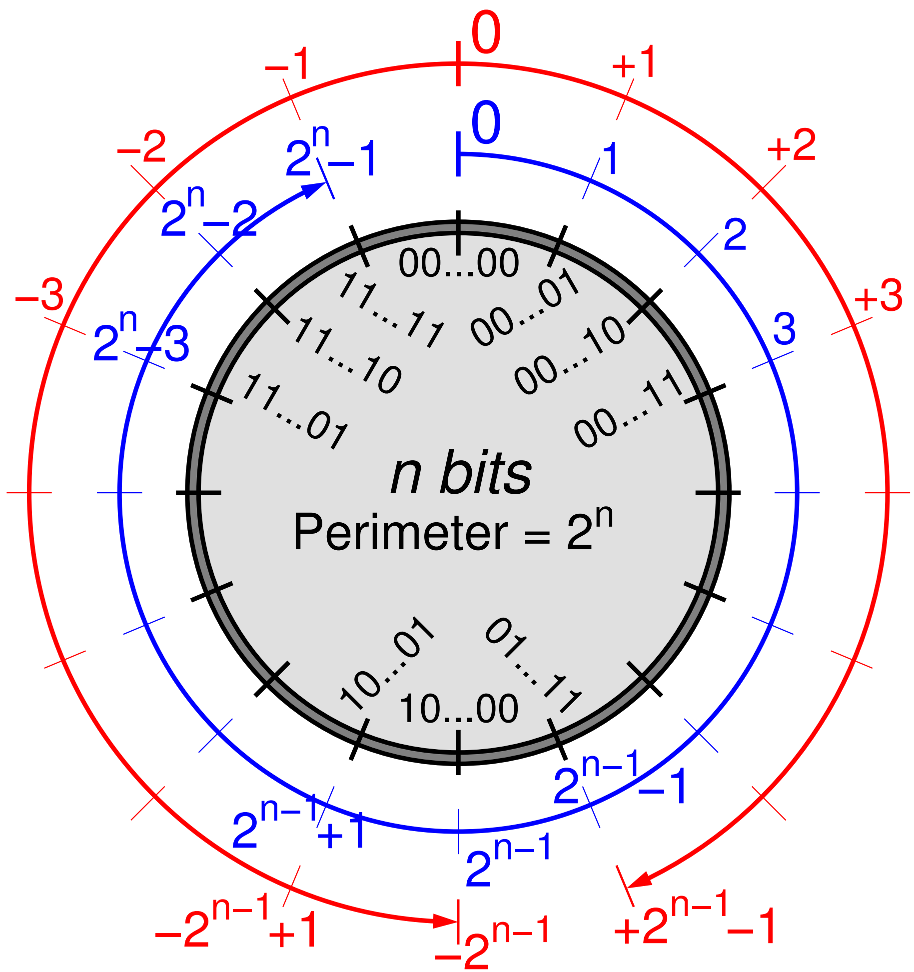
Προσημασμένοι Ακέραιοι σε Συμπλήρωμα ως προς 2



«Μαύρα» bits: ερμηνεία με δύο κώδικες:

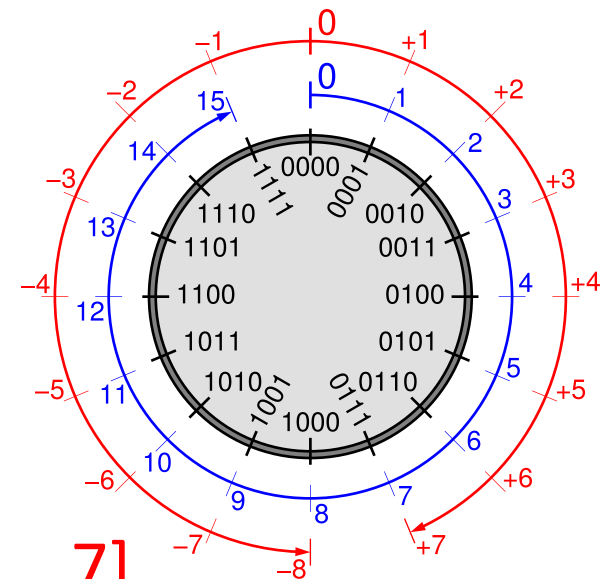
- Unsigned (μη προσημα-σμένοι): θετικοί & μηδέν
- Signed (προσημασμένοι):
- Μισοί θετικοί & μηδέν
  - MS bit == 0
- Μισοί αρνητικοί
  - MS bit == 1
- Μικροί θετικοί: ίδιοι signed
- Μεγάλοι θετικοί: διαφέρουν

Ποιοί Προσημασμένοι Ακέραιοι παρίστανται με *n* bits



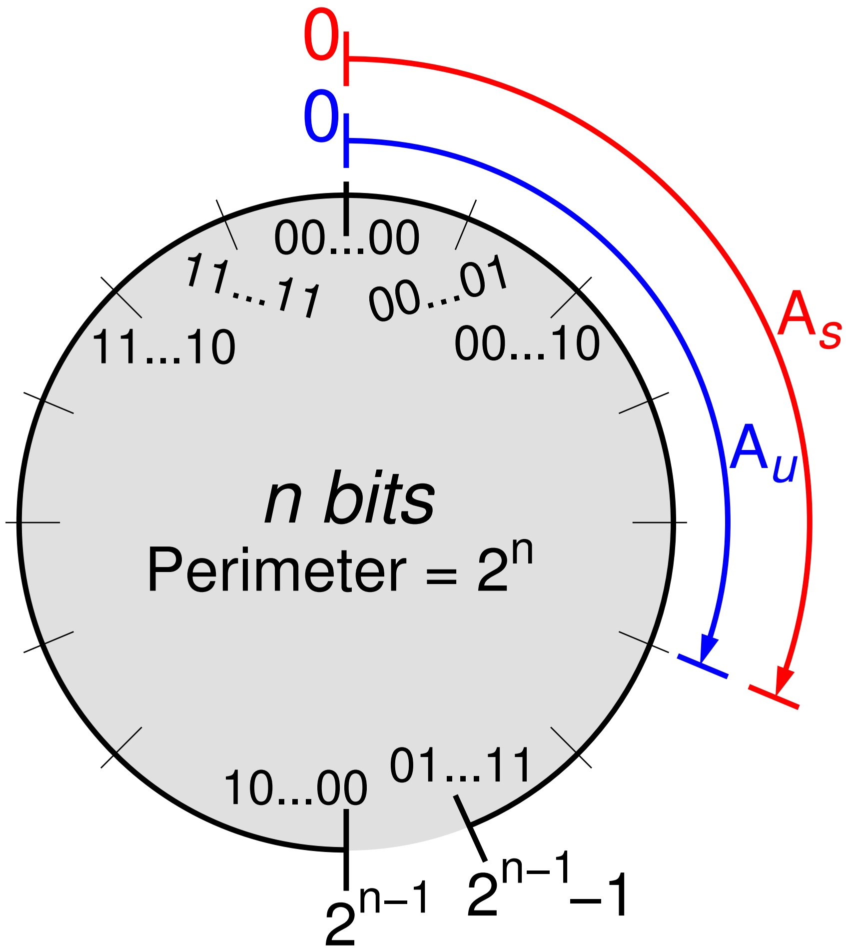
- n bits → 2^n το πλήθος ακερ.
- $0 \le$ unsigned $\le 2^n$-1
- $-2^{n-1} \le$ signed $\le +2^{n-1}$-1
- Αρνητικοί: μισοί απ' όλους = μισοί από τους $2^n \Rightarrow 2^{n-1}$
- Θετικοί & μηδέν: οι άλλοι μισοί από $2^n \Rightarrow 2^{n-1} \Rightarrow$ από το 0 έως και το 2^{n-1}-1

Πόσα bits χρειάζονται γιά δοθέντα αριθμό;



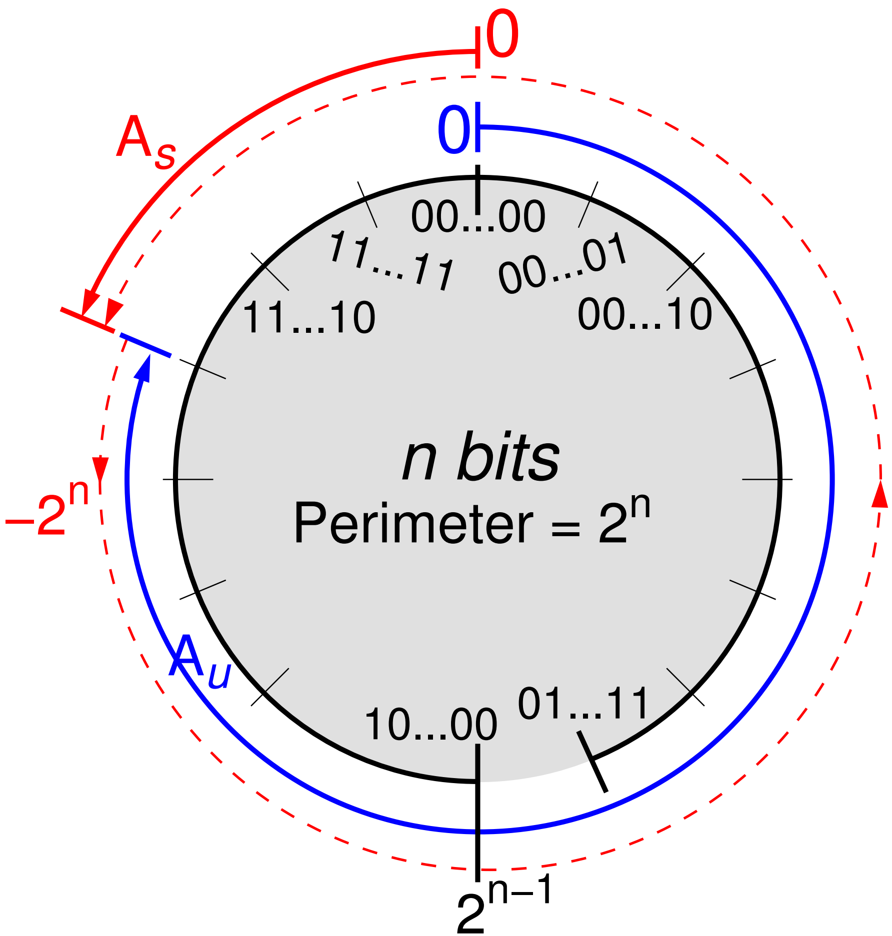
- Ο *unsigned* 12 πόσα bits χρειάζεται;
- 4 bits ⇒ 2^4=16 αριθμοί: [0,… 15]
- Ο *unsigned* 12 ∈ [0,… 15] και γράφεται: 1100
- Ο *signed* 12 πόσα bits χρειάζεται;
- 4 bits ⇒ 2^4=16 αριθμοί: [-8,… -1, 0, 1,… 7]
- 5 bits ⇒ 2^5=32 αριθμοί: [-16,… -1, 0, 1,… +15]
- Ο *signed* 12 ∈ [-16,… +15] και γράφεται: 01100
- Ο *signed* 1100 ποιός είναι; MSbit==1 άρα αρνητικός: -4

Μετατροπή *Unsigned* ↔ *Signed* γιά Θετικούς



- Δοθέντων *n* bits που ερμηνεύ-ονται ως ο *unsigned* A_u με MS bit == 0, δηλ.: $0 \le A_u \le 2^{n-1}-1$
- Τότε αυτά ερμηνεύονται ως ο signed A_s όπου: $A_s = A_u$
- Αντιστρόφως, δοθέντος επιθυ-μητού signed $A_s \ge 0$, αυτός ανα-παρίσταται όπως ο unsigned A_u με *n* bits όπου $A_u = A_s$ και *n* τέτοιο ώστε: $A_s \le 2^{n-1}-1$

Μετατροπή *Unsigned* ↔ *Signed* γιά Αρνητικούς



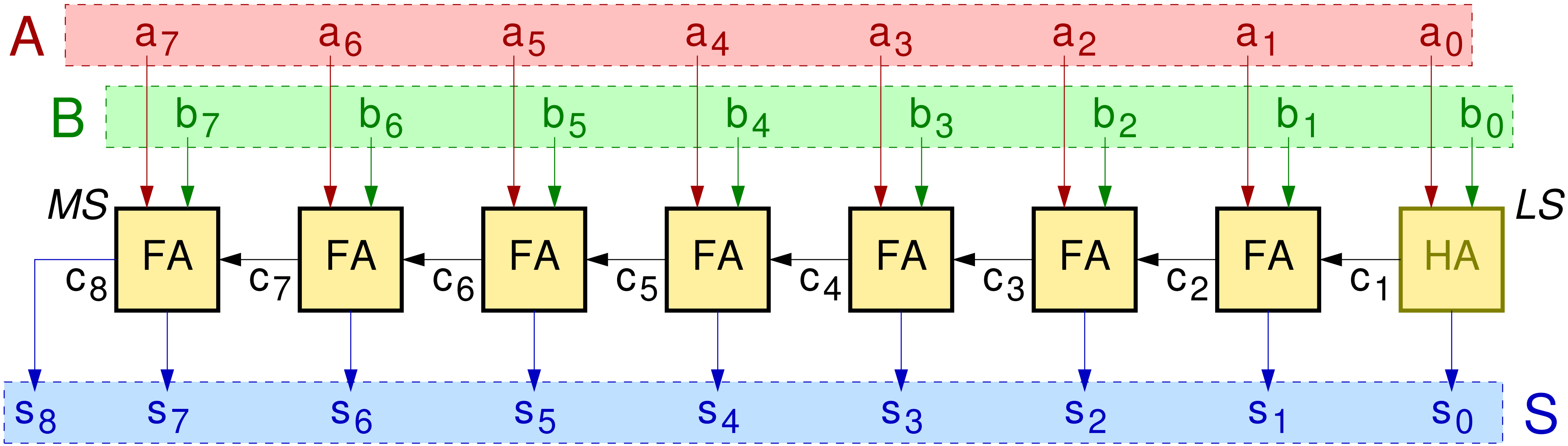
- Δοθέντων *n* bits που ερμηνεύονται ως ο *unsigned* A_u με MS bit == 1, δηλ: $2^{n-1} \leq A_u \leq 2^n-1$
- Τότε αυτά ερμηνεύονται ως ο signed A_s όπου: $A_s = A_u-2^n < 0$
- Αντιστρόφως, δοθέντος επιθυμητού signed $A_s<0$, αυτός αναπαρίσταται όπως ο unsigned A_u με *n* bits όπου: $A_u = A_s+2^n$ και *n* τέτοιο ώστε: $-2^{n-1} \leq A_s$

Παραδείγματα

- Να γραφτεί σε 2's complement ο signed **-40**
  - Χρειάζεται 7 bits: $2^7=128$ οι αριθμοί από -64 έως και +63
  - $A_s = -40$ αρνητικός, άρα ίδιος με $A_u = A_s + 2^7 = -40 + 128$

  $\Rightarrow$ $A_u = 128-40 = 88 = 64 + 24 = 64+16+8 \longrightarrow$ 1011000
  - Επαλήθευση ($2^η$ μέθοδος): $1011000 = -2^6+2^4+2^3 = -64+16+8 = -40$
- Ο προσημασμένος **11010** ποιός είναι στο δεκαδικό;
  - Αρχίζει με 1, άρα αρνητικός. Έχει 5 bits άρα «κύκλος» Π = 32
  - $A_u = 11010 = 2^4+2^3+2^1 = 16+8+2 = 26$
  - Επειδή αρνητικός, 5 bits: $A_s = A_u - 32 = 26-32 = -6$
  - Επαλήθευση ($2^η$ μέθ.): $-6 = -16+10 = -16+8+2 = -2^4+2^3+2^1 = 11010$

Πολύμπιτος Αθροιστής: Αλυσίδα Κρατουμένων



- Άθροισμα δύο 8-μπιτων ακεραίων: S = A+B
- Το άθροισμα χρειάζεται 9 bits γιά να χωρά πάντα
- Αλυσίδα από Full-Adders – ο πρώτος Half-Adder
- Κρατούμενα από Least Significant (LS) προς Most Signif. (MS)

Πίνακας Αληθείας Κρατουμένου και Αθροίσματος

a_i	b_i	Σ_{10}	c_{i+1}	s_i
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	2	1	0

↑ Ημιαθροιστής (Half Adder – HA)

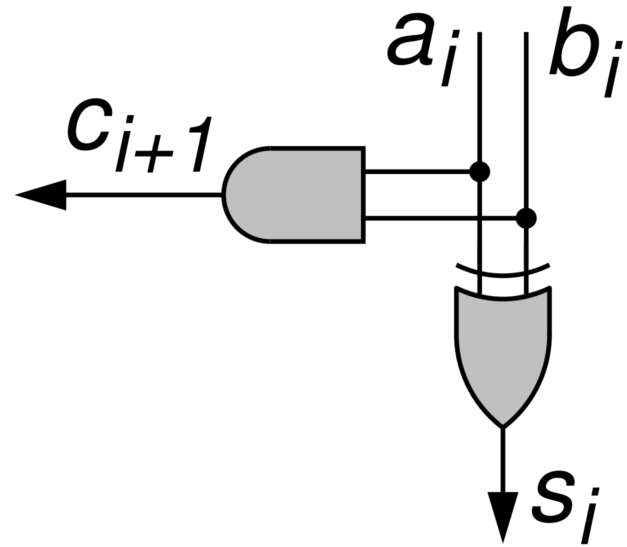
Πλήρης Αθροιστής (Full Adder – FA) →

a_i	b_i	c_i	Σ_{10}	c_{i+1}	s_i
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	2	1	0
1	0	0	1	0	1
1	0	1	2	1	0
1	1	0	2	1	0
1	1	1	3	1	1

Ποτέ δεν γενιέται πάνω από 1 bit κρατουμένου

Ημιαθροιστής: Άθροισμα δύο bits ίδιας Σημαντικότητας

a_i	b_i	Σ_{10}	c_{i+1}	s_i
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	2	1	0



- Είσοδοι a_i και b_i πρέπει ίδιας «σημαντικότητας» (significance), δηλαδή συντελεστές της ίδιας δύναμης του 2
- Έξοδος αθροίσματος, s_i: ίδια σημαντικότητα με εισόδους
- Έξοδος κρατουμένου, c_{i+1}: σημαντικ. κατά 1 μεγαλύτερη

Πλήρης Αθροιστής: Άθρ. τριών bits ίδιας Σημαντικότητας

c_{i+1}

a_i \ b_ic_i	00	01	11	10
0	0	0	1	0
1	0	1	1	1

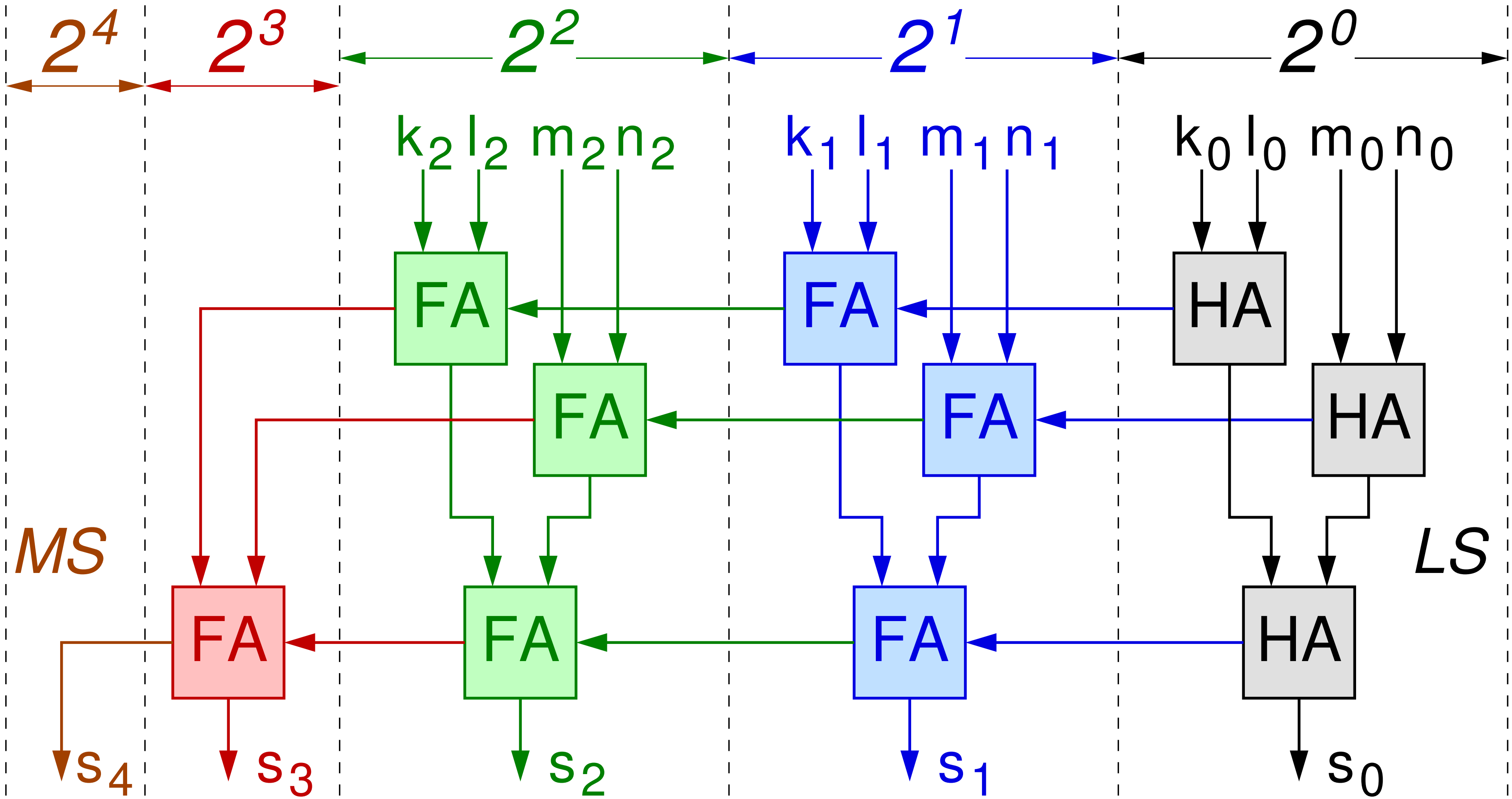
s_i

a_i \ b_ic_i	00	01	11	10
0	0	1	0	1
1	1	0	1	0

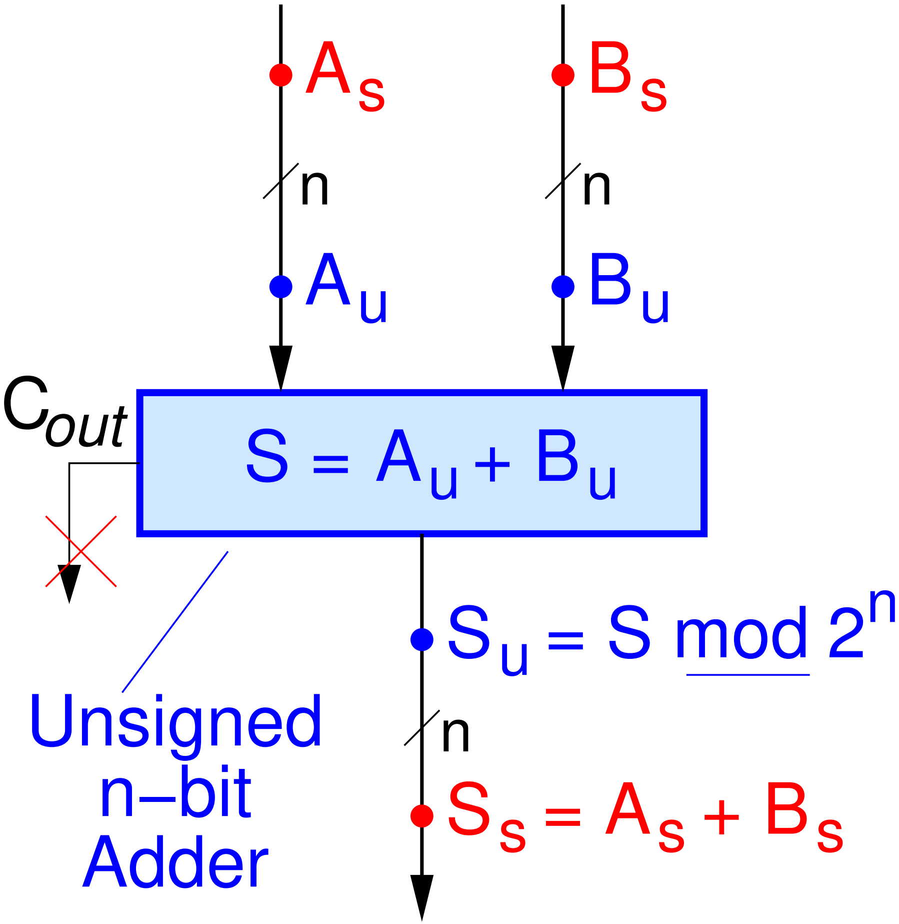
(note: *Odd Parity* function)

a_i	b_i	c_i	Σ_{10}	c_{i+1}	s_i
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	2	1	0
1	0	0	1	0	1
1	0	1	2	1	0
1	1	0	2	1	0
1	1	1	3	1	1

Αθροιστής Δένδρου: οι λεπτομέρειες



Πρόσθεση Προσημασμένων Συμπληρώματος του 2



- Με τον ίδιο απαράλλακτο αθροιστή απρόσημων!
- *Πρέπει* το Carry-out να *αγνοηθεί*
- *Εάν* το αλγεβρικό άθροισμα των προσημασμένων χωρά σε *n* bits, τότε & μόνο τότε αυτό θα είναι πάντα σωστό
- Απόδειξη: 4 περιπτώσεις

Παραδείγματα

- (-40)+(+8) = (-32)
  - 7 bits, από διαφάνεια 8
  - (-40) = 1011000
  - (+8) = 0001000
  - 1100000 = -2^6+2^5 = -64+32 = -32

$$\begin{array}{rr} & 1011000 \\ + & 0001000 \\ \hline \cancel{(C_{out}=0)} & 1100000 \end{array}$$

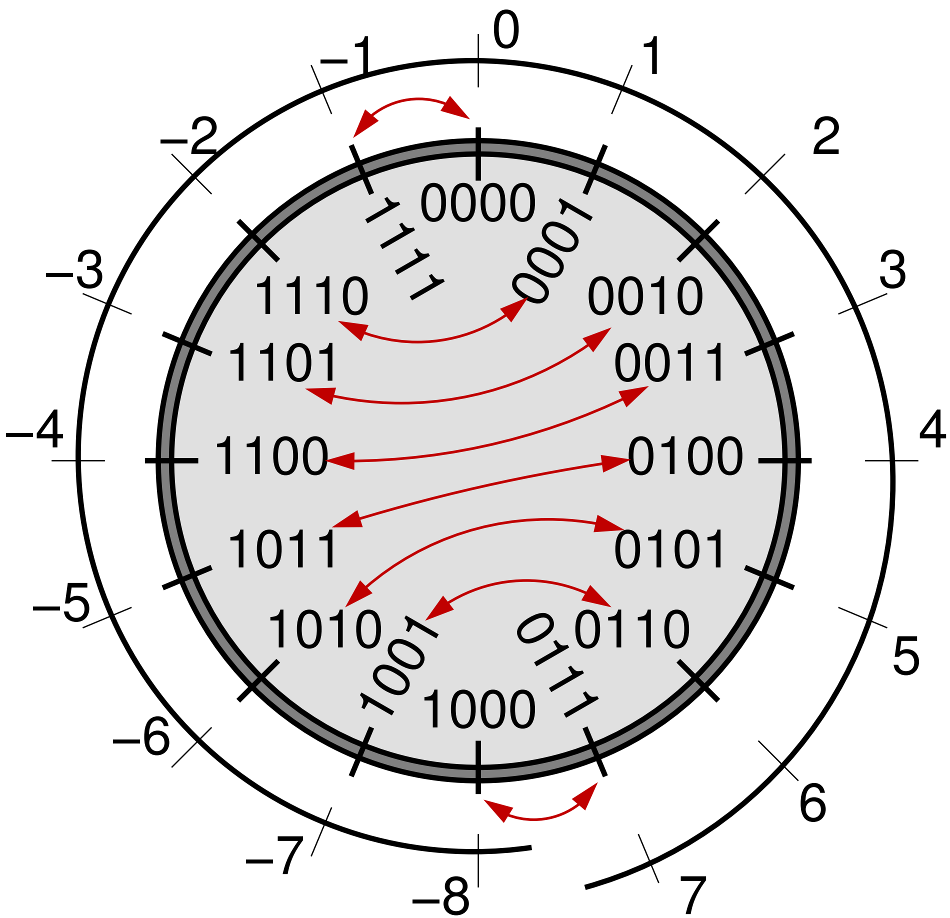
- (-40)+(+47) = (+7)
  - (+47) = 32+8+4+2+1 = 0101111

$$\begin{array}{rr} & 1011000 \\ + & 0101111 \\ \hline \cancel{(C_{out}=1)} & 0000111 \end{array}$$

- (-40)+(-13) = (-53)
  - B_s = (-13) = 115 - 128 = $B_u - 2^7$
  - B_u = 115 = 64+32+16+2+1 = 1110011
  - 1001011 = $-2^6+2^3+2^1+2^0$= -64+11 = -53

$$\begin{array}{rr} & 1011000 \\ + & 1110011 \\ \hline \cancel{(C_{out}=1)} & 1001011 \end{array}$$

Συμπλήρωμα ως προς 1 (bitwise NOT) & ιδιότητες

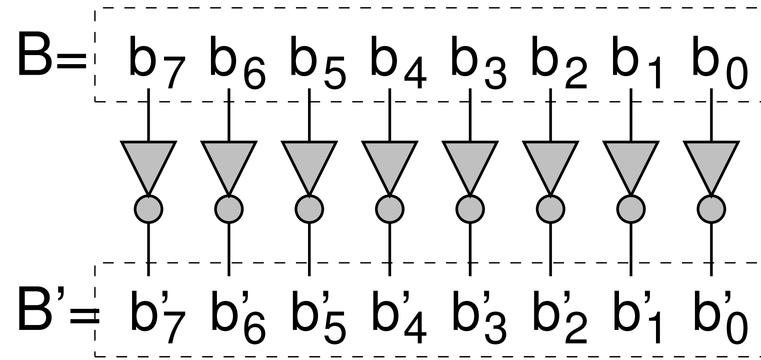


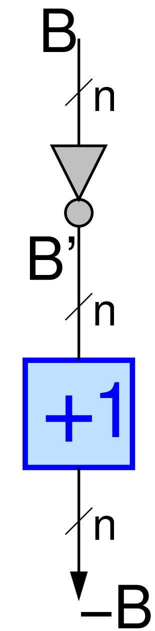
$$
\begin{array}{rl}
B = & 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1 \\
B' = & 1\ 1\ 0\ 1\ 0\ 0\ 1\ 0 \quad + \\
\hline
B+B' = & 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \ = \text{"}-1\text{"}
\end{array}
$$

- «Συμπλήρωμα ως προς 1» (1's complement) = τα ΟΧΙ των bits, καθένα
- Αθροιζόμενο με τον αρχικό αριθμό δίνει πάντα -1

Αντίθετος Αριθμού = Bitwise NOT συν 1

$$
\begin{array}{rl}
B = & 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1 \\
B' = & 1\ 1\ 0\ 1\ 0\ 0\ 1\ 0 \quad + \\
\hline
B+B' = & 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \quad = \text{"}-1\text{"}
\end{array}
$$





- $B + B' = -1 \Rightarrow (B + B') + 1 = 0$

$\Rightarrow B + (B' + 1) = 0$

– όταν (B' + 1) δεν προκαλεί υπερχείλιση (βλ. επόμενη διαφ.)

$\Rightarrow$ Ο ***αντίθετος*** του B είναι ο: $-B = B' + 1$

Παραδείγματα

- Γράψτε τον (-40) σαν signed 2's cmpl.
  - 7 bits, από διαφάνεια 8
  - (+40) = 32+8 = 0101000
  - bitwise NOT: 1010111

  ⇒Αντίθετος -(+40) = -40 = 1011000 (ίδιο με πριν)

$$\begin{array}{r} 1010111 \\ 1 \;+ \\ \hline \cancel{(C_{out}=0)}\ 1011000 \end{array}$$

Αντίθετος του
(-7) = 1001
NOT = 0110

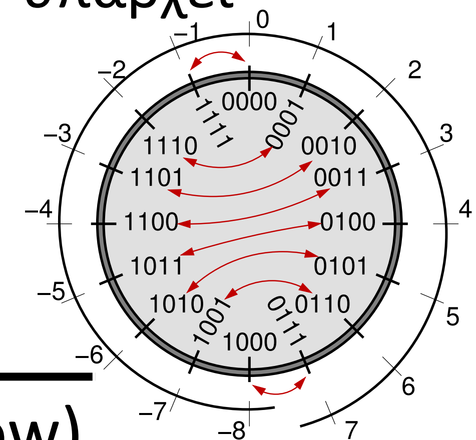
$$\begin{array}{r} 0110 \\ 1 \;+ \\ \hline 0111 = +7 \end{array}$$

Αντίθετος του
(+7) = 0111
NOT = 1000

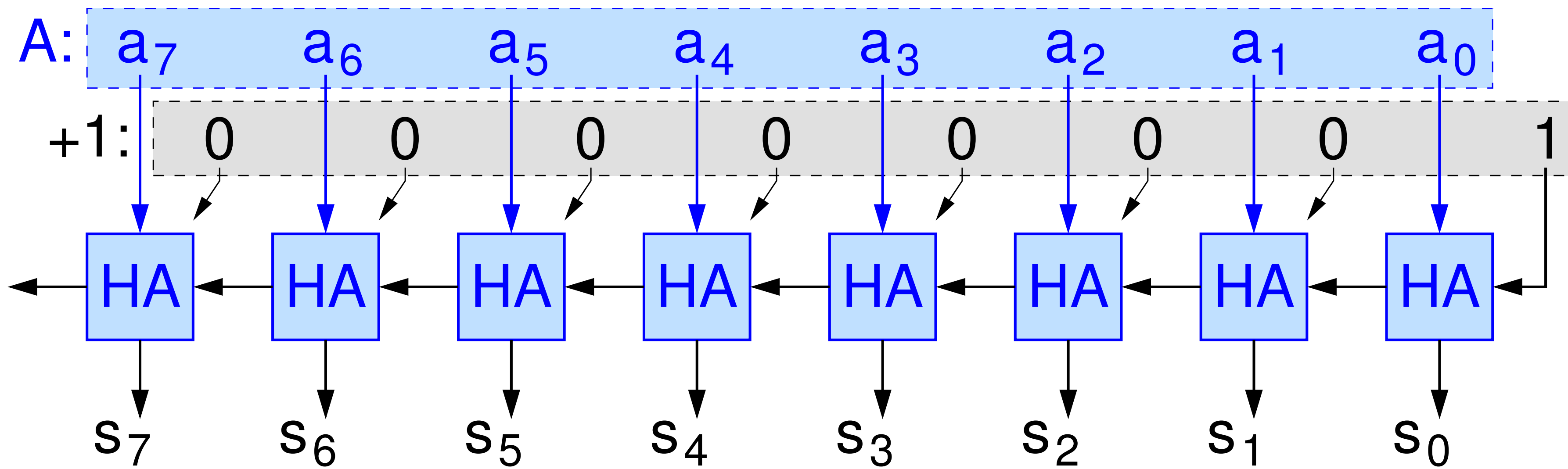
$$\begin{array}{r} 1000 \\ 1 \;+ \\ \hline 1001 = -7 \end{array}$$

Αντίθετος του -8 δεν υπάρχει
(-8) = 1000
NOT = 0111

$$\begin{array}{r} 0111 \\ 1 \;+ \\ \hline 1000 = -8 \text{ (overflow)} \end{array}$$

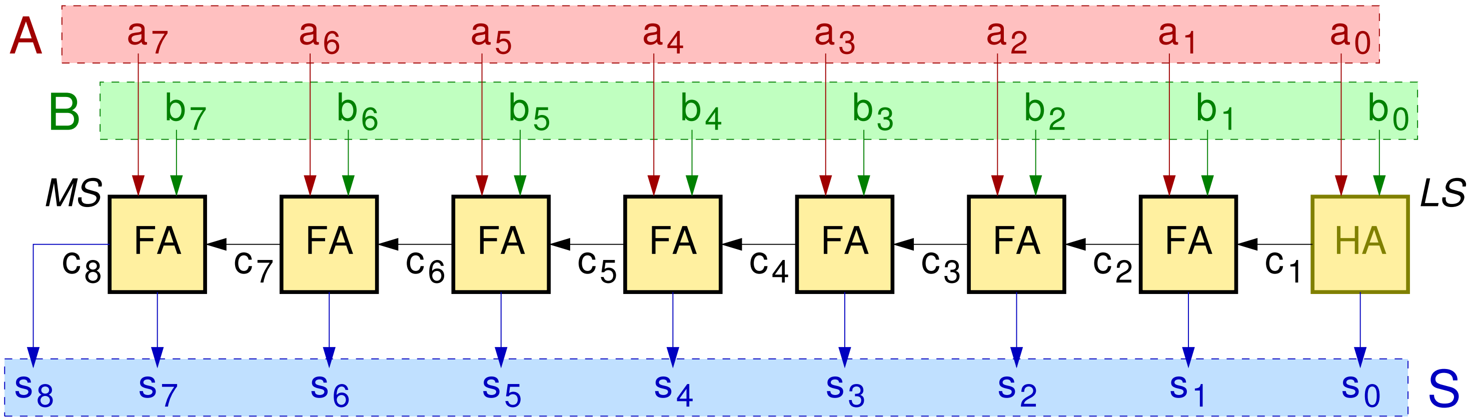


Αυξητής (Incrementor) (+1) μέσω Ημιαθροιστών μόνο

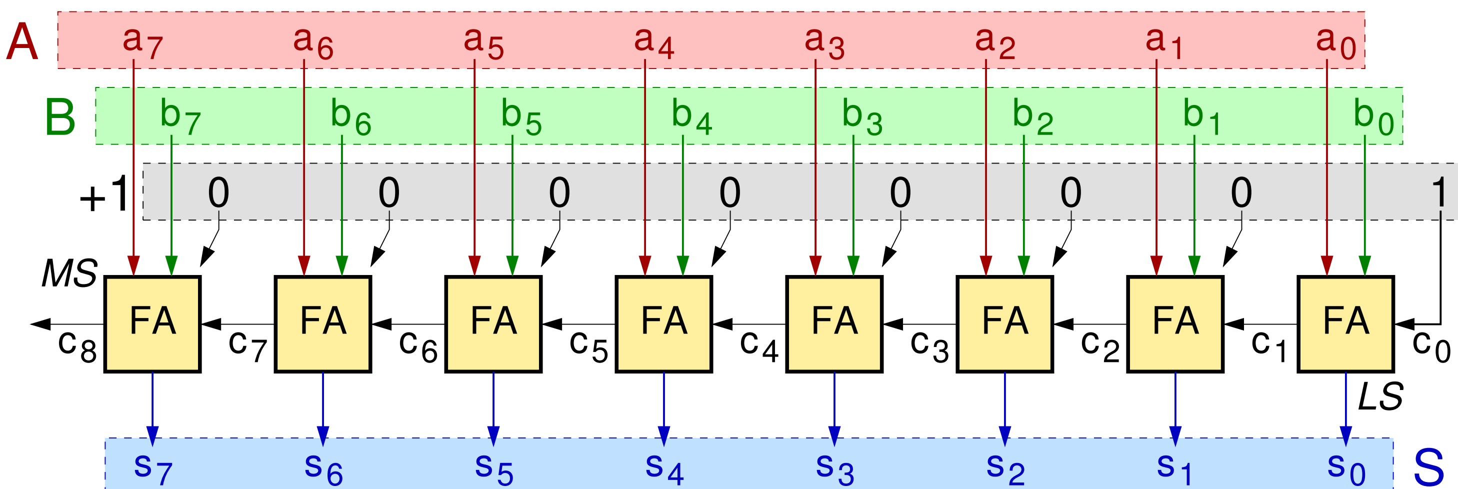


- Τα μηδενικά περιττό να προστεθούν
- Μόνον το LS bit έχει δεύτερο προσθετέο ≠ 0
  - εκείνο ακριβώς το bit που *δεν* έχει κρατούμενο εισόδου
- Σε όλα τα bits αρκεί Ημιαθροιστής (Half Adder – HA)

A+B+1: αρκεί ένας μόνον Αθροιστής (με FA's παντού)



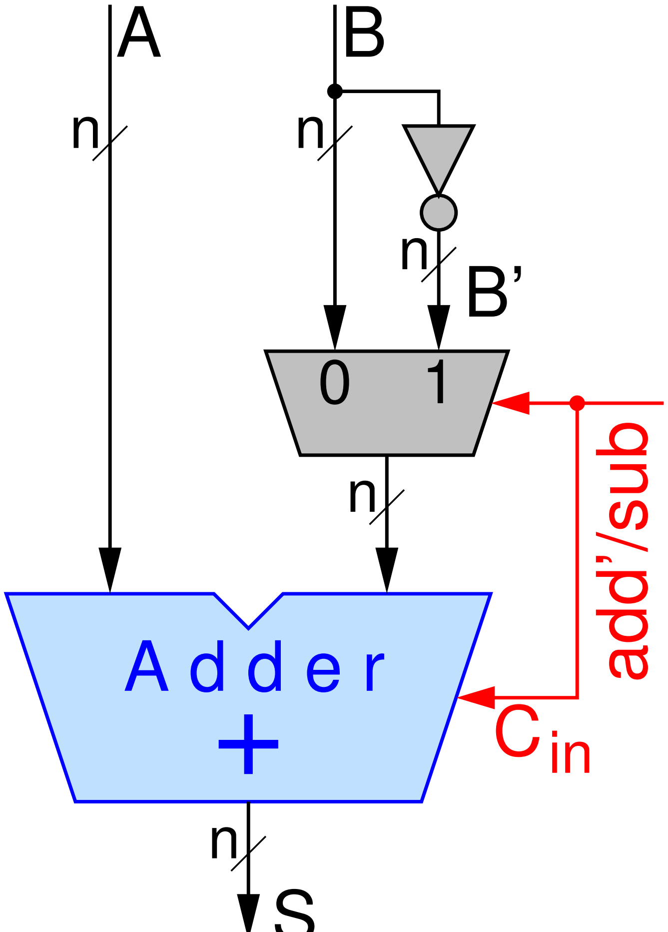
Κλασικός Αθροιστής δύο αριθμών, A+B



A+B+1:

αρκεί απλώς ο δεξιός HA να γίνει FA

Κύκλωμα Προσθαφαιρέτη



Όταν add'/sub = 0, τότε:

- Δεύτερη είσοδος αθροιστή = B
- $C_{in} = 0$

$\Rightarrow S = A + B$

Όταν add'/sub = 1, τότε:

- Δεύτερη είσοδος αθροιστή = B'
- $C_{in} = 1$

$\Rightarrow S = A + B' + 1 = A + (-B) = A-B$