Example: 2's complement using 4 bits.

Binary	Value	
0000	0	positive half.
0001	1	
0010	2	
0011	3	
0100	4	
0101	5	
0110	6	
0111	7	
1000	-8	negative half.
1001	-7	
1010	-6	
1011	-5	
1100	-4	
1101	-3	
1110	-2	
1111	-1	

check:

$$\begin{array}{r} 4 \\ +\ (-4) \\ \hline 0 \end{array} \qquad \begin{array}{r} 0100 \\ +\ 1100 \\ \hline 0000 \end{array}$$

Converting +ve to -ve (and vice versa)

- Given a number, k, the negative, $-k$, can be found by complementing all bits and then add 1.

e.g. $7 = 0111$ $\quad 1000 + 1 = 1001$

- shortest method: starting from the right most bits. keep all of the bits up to and including the first 1 and then complement the rest.

e.g. $(-20)_{10} = (?)_2$

$20 = 010100 \Rightarrow -20 = 101100$

check:

$$\begin{array}{r} 010100 \\ +\ 101100 \\ \hline 000000 \end{array}$$

e.g. Find $-(3A7)_{16}$

$(3A7)_{16} = 0011\,1010\,0111$

$-(3A7) = 1100\,0101\,1001 = (C59)_{16}$

Multiplication. (by the base)

- in decimal., to multiply by 10, you obtain a zero on the right.

  eg. $34 \times 10 = 340$.

- in binary. to multiply by 2, you obtain a zero on the right.

  eg. $(0111)_2 \times (10)_2 = 01110$

  eg. $(074)_{16} \times 8 = ?$

  Note: to perform $k \times 2^n$, we "shift k to the left by n bits."

  $(074)_{16} = 0000\,0111\,0100 \xrightarrow{\times 8} = 0011\,1010\,0000$

  $= (340)_{16}$

Division. (by the base)

- in decimal, to divide by 10, you drop the rightmost digit.

  eg. $347 \div 10 = 34$.

- in binary. to divide by 2. you drop the right most bit

  eg. $(01110)_2 \div (10)_2 = 0111$

  eg. $(3A7)_{16} \div 8 = ?$

  $0011\,1010\,0111 \rightarrow 0000\,0111\,0100 = (074)_{16}$

  Note: To perform $k \div 2^n$, we "shift k to the right by n bits"

Bit significance. (for an n-bit number)

$2^{n-1}\ 2^{n-2}\ 2^{n-3} \ldots\ldots\ 2^2\ 2^1\ 2^0$

most-significant bit (msb) — least significant bit (lsb)

In 2's complement, the msb is called the sign bit, it's 0 for all positive value, and 1 for all negative values

Representing Text

Letters & punctuation (also known as characters) are represented using ASCII code (American Standard Code for Information Interchange):

character	ASCII (hex)
A	41
B	42
C	43
D	44
⋮	⋮
Z	5A
a	61
b	62
c	63
d	64
⋮	⋮
z	7A
0	30
1	31
2	32
3	33
⋮	⋮
9	39
space	20
!	21
⋮	⋮

- Each character can be stored in a computer as a byte.

eg. Hi there !

= 48 69 20 74 68 65 72 65 21

H i ␣ t h e r e !

- A sequence of characters is called a strains
- A computer file that contains only ASCII codes is called a text file. Such file can be read / written by an text Editor.