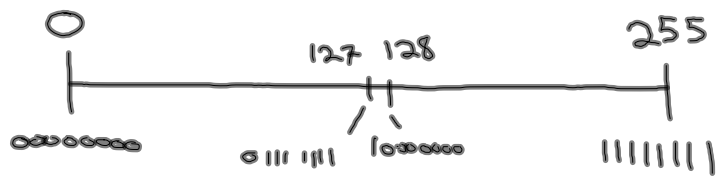


8 bits



sign-magnitude (use 1 bit to rep. sign)
1: -, 0: +



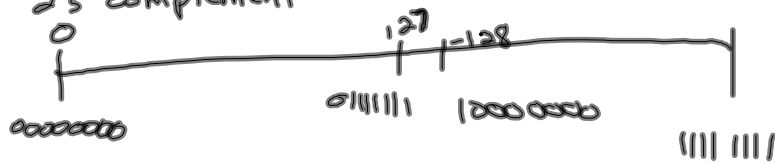
excess 128



excess 127



2's complement



$$00010101 \quad 21$$

$$\begin{array}{r} 11101010 \\ +1 \\ \hline 11101011 \end{array} \quad \begin{array}{l} \text{comp.} \\ \text{add 1} \\ -21 \end{array}$$

$$\begin{array}{r} 10000000 \\ 01111111 \text{ comp.} \\ +1 \\ \hline 10000000 \end{array} \quad \begin{array}{l} - \\ \\ \\ 128 \end{array}$$

$$\begin{array}{r} 11111111 \\ 00000000 \text{ comp.} \\ +1 \\ \hline 00000001 \leftarrow 1 \end{array}$$

10001100	2's	sig.
	-116	-12

2's comp

01110011	
+ 1	
<hr/>	
01110100	116

sign-mag

-0001100	-12
<u> </u>	
12	

exc. 128

10001100	140
	<hr/>
	-128
	<hr/>
	12

Sign mag.

10110001

8 bit

10000000 00110001

16 bit

2's comp

00000001

8 bit

00000000 00000001

16 bit

2's comp

00000101

5

1111010
+1

8 bit

1111011

00000000 00000101

5 16 bit

11111111 1111010
+1

-5

11111111 1111011

sign-extension

int
long
double
float
char
bool
short

int
 $-2^{31} \dots 2^{31} - 1$

unsigned int
 $0 \dots 2^{32} - 1$

logic / bits

0: false
1: true

A && B

comp.

A	$\neg A, \sim A, \bar{A}$
0	1
1	0

and

A & B

$A \wedge B,$

AB, A.B

A	B	
0	0	0
0	1	0
1	0	0
1	1	1

or

A	B	$A \vee B, A + B$ $A B, A B$
0	0	0
0	1	1
1	0	1
1	1	1

Java String s

if (s != null && s.length() > 10)

↑ execute 1 if it is false

if (s != null & s.length() > 10)

↑ execute both

C++/Java

&
| } }

bitwise operators

A	0	1	0	1	1	0	0	1	
B	1	0	1	1	0	1	1	1	
A & B	0	0	0	1	0	0	0	1	
A B	1	1	1	1	1	1	1	1	