

unsigned char x; // 255 to 0 | 241 | 1
char x; // 127 to -128 | 8-28-17

Range for each data type.

0111 1111

1000 0000

$(-1) \Rightarrow 1111 1111$

$-24 = -(00011000)$

11101000 8bit signed
binary
or 8bit 2's Comp.

64 \Rightarrow 01000000
+ 64 01000000
10000000 overflow

$\rightarrow (-128)!$

Overflow indicated by carry in and out of sign bit, not matching.

0111 1111 1111 1111 32767
+
01000000 0000 0000 0000 1
11000000 0000 0000 0000 0
b₁₅ b₀ -32768

$-b_{15} \cdot 2^{15} + b_{14} \cdot 2^{14} + \dots + 2^0 \cdot b_0$

Alternate Interpretation of 16-bit 2's-Complement numbers.

Logicals

a_n	b_n	$a_n \text{ and } b_n$	$a_n \text{ or } b_n$	inverse a_n	$a_n \text{ exclusive or } b_n$
a_n	b_n	$a_n \& b_n$	$a_n b_n$	$\sim a_n$	$a_n \wedge b_n$
0	0	0	0	1	0
0	1	0	1	1	1
1	0	0	1	0	1
1	1	1	1	0	0

Force High
 Force Low
 Toggle

Equivalent statements

$$I_n = I_n | \text{Mask};$$

$$I_n |= \text{Mask};$$

This is short hand for the first statement.