Binary Addition

Stick with 8 bits and throw anything else away. Sometimes the results are wrong (and sometimes right even when throwing away bits).

Simple example: -1 plus -1

start on the right

$$
\begin{array}{rcccccccc}
 & & & & & & & 1 & \\
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
+ & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
 & & & & & & & & 0
\end{array}
$$

carry

"sum" bit

$1+1=2$ $=10$ in binary

carry ↑ ↑ "sum" bit

next bit

$$
\begin{array}{rcccccccc}
 & & & & & & 1 & 1 & \\
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
+ & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
 & & & & & & & 1 & 0
\end{array}
$$

carry

next sum

$1+1+1=3$ $=11$ in binary

carry ↑ ↑ "sum" bit

continue

$$
\begin{array}{rccccccccc}
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\
 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
+ & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}
$$

this last bit is discarded / ignored

Result

If Unsigned		If signed
255 ←	1 1 1 1 1 1 1 1	→ -1
255 ←	+ 1 1 1 1 1 1 1 1	→ -1
254 ←	1 1 1 1 1 1 1 0	→ -2 ← correct

incorrect (should be 256 more than this)

Binary Subtraction

Example

As with base 10 start on the right and subtract digit by digit, borrowing when needed.

First 3 bits

```
  10010011
- 01101010
  --------
       001
```

No borrowing required so far

Next two bits

borrow here

this is now 2 = 10 in binary

```
     0 2
  10010011
- 01101010
  --------
     01001
```

2 − 1 = 1

0 − 0 = 0

Next bit requires going two bits more

now 2

```
  0 1 2 0
  10010011
- 01101010
  --------
  00101001
```

2 − 1 = 1

these don't require a borrow

Previous step repeated

one step

now 2 with the borrowed 1

```
  01 2 0
  10010011
- 01101010
  --------
  00101001
```

With one or more zeros to the left you may find it easier to think of the zero(s) and the next one as a binary number. E.g. 10. Subtracting 1 from that gives 1 here so replace 10 with 01.

Another example of this

Another Example

$$
\begin{array}{r}
10000100 \\
-\ 01101101 \\
\hline
\end{array}
$$

Do these together

first borrow

$$
\begin{array}{r}
01\ {}^{1} \\
10000\cancel{1}\cancel{0}\,0 \\
-\ 01101101 \\
\hline
111
\end{array}
$$

2 − 1 = 1

borrow will be needed

Do together

second borrow

$$
\begin{array}{r}
01111\ {}^{1}0\ 1\ {}^{1} \\
\cancel{100000}\cancel{1}\cancel{0}\,0 \\
-\ 01101101 \\
\hline
00010111
\end{array}
$$

2 − 1 = 1

If needed, assume there is a 9th bit that is 1, but stick with an 8-bit result.

Example: zero minus negative one

imaginary 9th-bit

Now 2 with the borrow

$$
\begin{array}{r}
0\ 1111111\ {}^{1} \\
\cancel{1}\,\cancel{0000000}\,0 \\
-\ 11111111 \\
\hline
00000001
\end{array}
$$

ignore 9th bit anyway in result

2 − 1 = 1

Note: the 9th bit position has a place value of $2^9 = 256$ so when results are wrong they are off by 256