CS 208

W, 17 Jan 2024

0101 as binary #

$4+1 = 5$

0101 as 4-bit two's complement

$+(4+1) = 5$

it's non-negative!

So, just use binary as usual

1100 as binary

$8+4+0+0 = 12$

1100 as 4-bit two's comp.

next page

1100 as 4-bit two's comp.

it's negative

1100 = N

complement

0011

+1

0100 = −N = 4

Strategy:

1. call 1100 = N < 0
2. compute −N (pos.)
3. convert to decimal
4. * −1

SO N = −4

0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	−1
1110	−2
1101	−3
1100	−4
1011	−5
1010	
1001	
1000	−8

$$
\begin{array}{r}
1111 \\
+\quad\ \ 1 \\
\hline
0000
\end{array}
\qquad
\begin{array}{r}
0011 \\
+\quad\ \ 1 \\
\hline
0100
\end{array}
$$

Why does "complement + 1" work?

Goal: I want $N + (-N) = 0000$

$$
\begin{array}{rl}
1100 & \leftarrow N \\
+\ 0011 & \leftarrow \text{compl. of } N \\
\hline
1111 & \\
+1 & \\
\hline
0000 &
\end{array}
$$

$-5 + 3$

$$
\begin{array}{r}
\;1011 \\
+\;0011 \\
\hline
1110
\end{array}
\leftarrow -2
$$

$5 + 5$ uh-oh

(10 is not a 4-bit representable 2's comp #)

$00011001 = 25$ as binary integer

$\begin{array}{ccccc} 16 & 8 & 4 & 2 & 1 \end{array}$

$16 + 8 + 1$

$\underline{0}0011001 = 25$ as 8-bit two's comp. int?

11111010 = 250 binary

11111010 = ? 8-bit two's comp.

neg.

compl.

0000 0101

+1

00000110 = 6

So original # is -6

$\underbrace{101}_{5}\ \underbrace{110}_{6} = ?$ in my head.

$5 \times 2^{3} + 6 = 5 \times 8 + 6 = 46$

\# bits to shift left

$10 \mid 1110$

$2 \quad\ \ 14$

$2 \times 2^{4} + 14 = 2 \times 16 + 14$
$= 46$

111×2
$7 \quad = 14$

0xFFFA   16-bit two's comp.

1111 1111 1111 1010 $\leftarrow$ −6

negative

comp

0000 0000 0000 0101

+ 1

0000 0000 0000 0110 = 6