- Negating 2's complement numbers
  - Complement each bit and then add 1.
- Value of an N-bit 2's complement number $b_{n-1}b_{n-2}\ldots b_2b_1b_0$

$$-b_{n-1}2^{n-1} + \sum_{k=0}^{n-2} b_k 2^k$$

- Sign extension of 2's complement numbers
  - Replicate the most significant bit (MSB) to make numbers longer
  - For example, going from 4-bit to 8-bit numbers:
    - 0101 (+5) should become 0000 0101 (+5).
    - But 1100 (-4) should become 1111 1100 (-4).
- Bit-wise shifting: (see back)
- Subtraction: implement by negating the 2nd input and then adding.
- Overflow occurs when:
  - If you add two *positive* numbers and get a *negative* result.
  - If you add two *negative* numbers and get a *positive* result.

---

0100 $= +4_{10}$ (a positive number in 4-bit two's complement)
$=$ (invert all the bits)
$= -4_{10}$ (and add one)

If 01101 is the 5-bit representation for 13, what is the 2's complement representation for -13?

| | |
|---|---|
| We have the unsigned 8-bit word: | $b_7b_6b_5b_4b_3b_2b_1b_0$ |
| And we want the 8-bit word: | $0\,0\,0\,0\,0\,b_5b_4b_3$ |

| | |
|---|---|
| We have 2 unsigned 8-bit words: | $a_7a_6a_5a_4a_3a_2a_1a_0$ |
| | $b_7b_6b_5b_4b_3b_2b_1b_0$ |
| And we want the 8-bit word: | $a_7b_6a_5b_4a_3b_2a_1b_0$ |

| | |
|---|---|
| We have 2 unsigned 8-bit words: | $a_7a_6a_5a_4a_3a_2a_1a_0$ |
| | $b_7b_6b_5b_4b_3b_2b_1b_0$ |
| And we want the 8-bit word: | $a_3a_2a_1a_0b_3b_2b_1b_0$ |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 0 | 1 | 1 | 11 |
| + | 1 | 1 | 1 | 0 | 0 | + (-4) |

| | | | | |
|---|---|---|---|---|
| | 1 | 1 | 0 | 1 |
| - | 1 | 0 | 1 | 0 |