Wrap-around Semantics

- Cause of unbelievably expensive bugs, e.g. the Ariane 5 Flight 501.

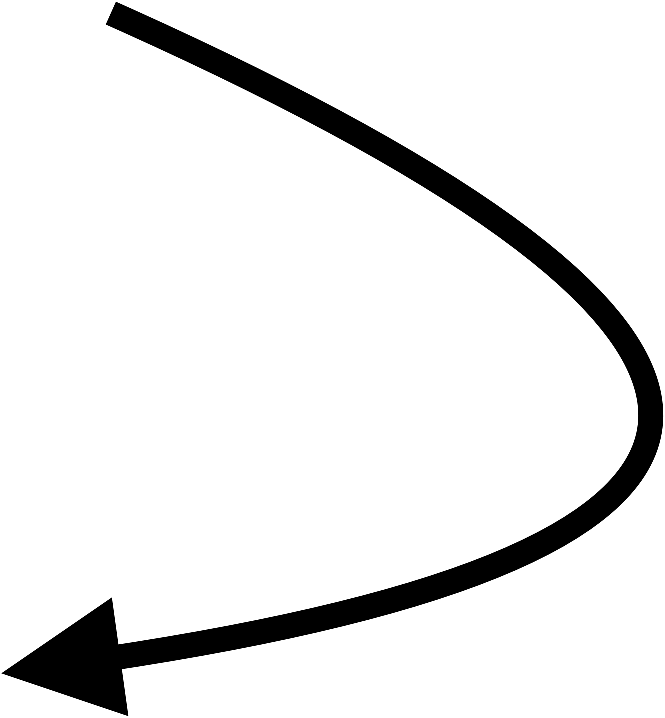
- 2^{64} - 1 is the largest value that can be represented by 64 bits. In selfie this value is denoted `UINT64_MAX`.

```
UINT64_MAX + 1 = 0
UINT64_MAX + 2 = 1
UINT64_MAX + 3 = 2
        ...
```

1 000000000…000000000

111111111…111111111

$2^{64} - 1$



Adding `1` to **`UINT64_MAX`** leads to a wrap-around, where only the 64 LSB are considered.

$$2^{64} - 1 + 1$$

64 bit

Arithmetic Instructions

`sltu`	`$rd`	`$rs1`	`$rs2`

- Set `$rd` to 1 if `$rs1 < $rs2`.

- This is the only instruction needed to implement <, >, <=, >=, == and `!=`.

- How this is done:

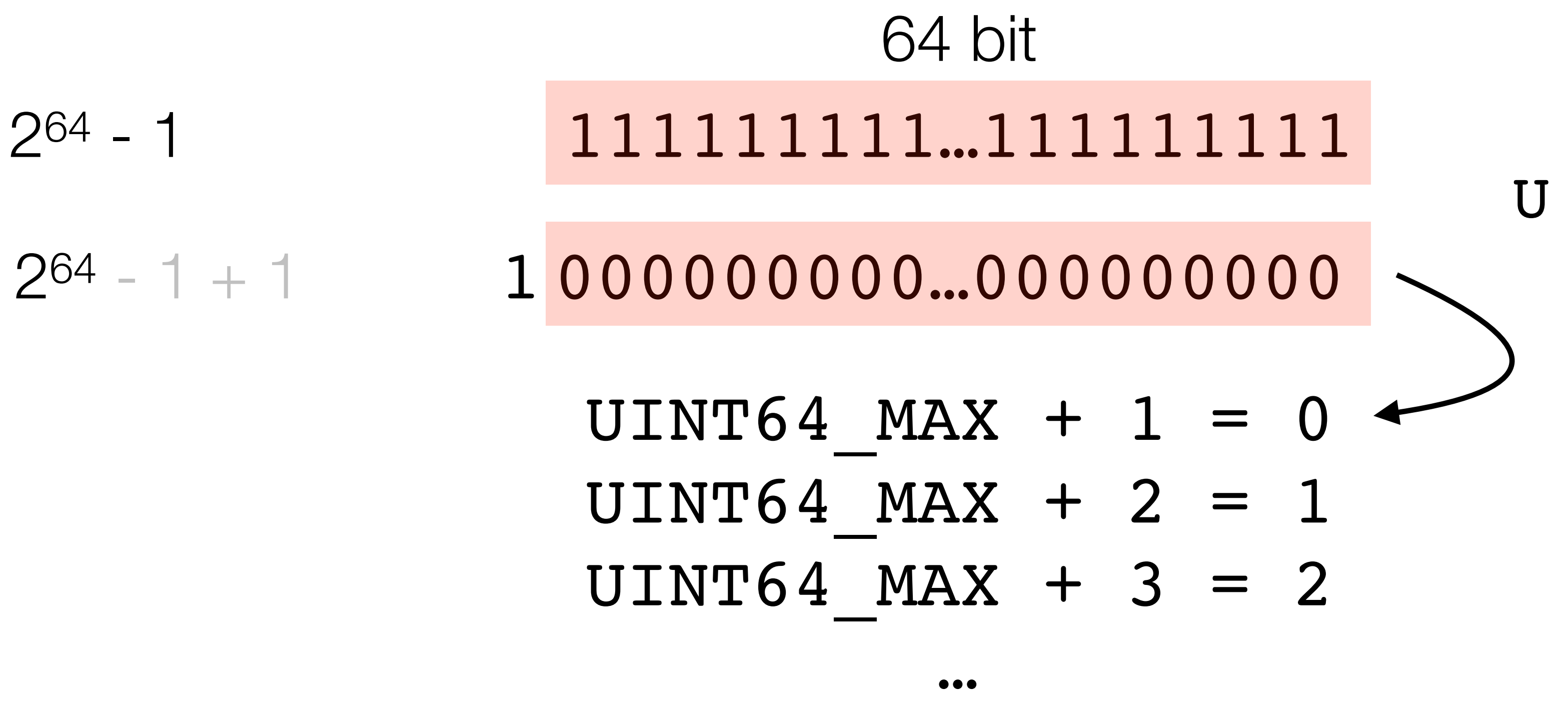
    - == is implemented using `b - a < 1`.

In unsigned arithmetic only 0 satisfies this condition.

Wrap-around Semantics

- Cause of unbelievably expensive bugs, e.g. the Ariane 5 Flight 501.

- $2^{64} - 1$ is the largest value that can be represented by 64 bits. In selfie this value is denoted `UINT64_MAX`.



64 bit

$2^{64} - 1$ 111111111…111111111

$2^{64} - 1 + 1$ 1 000000000…000000000

```
UINT64_MAX + 1 = 0
UINT64_MAX + 2 = 1
UINT64_MAX + 3 = 2
…
```

Adding `1` to `UINT64_MAX` leads to a wrap-around, where only the 64 LSB are considered.