Representation of numbers (continued)

Two's complement representation

-128	64	32	16	8	4	2	1
1	0	1	0	1	0	0	1

The most significant bit has a negative value. So to the number above represents -8710 (-128 + 32 + 8 + 1).

Negative numbers can be determined by first writing its positive, flipping each bit and adding 1.

	-128	64	32	16	8	4	2	1
Positive 87_{10}	0	1	0	1	0	1	1	1
Flip the bits	1	0	1	0	1	0	0	0
Add 1								1
	1	0	1	0	1	0	0	1

Overflow

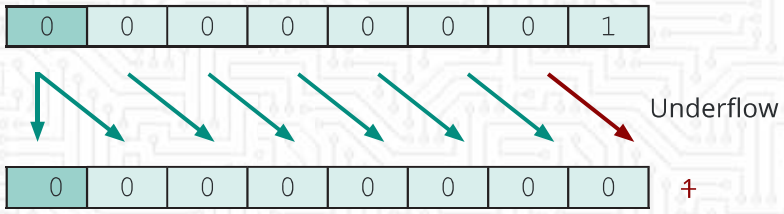
```
  11011001
+ 01010010
----------
 ↯00101011
```

Overflow

When the number is too large to be stored by the register, the left-most bit cannot be stored and therefore cannot be counted.

Underflow

An arithmetic shift right (see right) on the following number (1_{10}) would be divided by half and so should be 0.5_{10}.

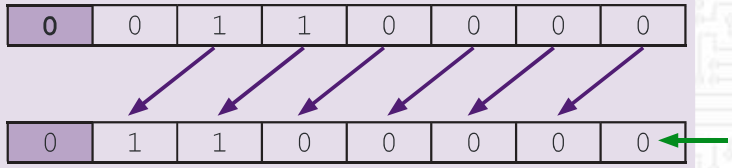


When the number is too small to be stored by the register, the right-most bit cannot be stored and therefore cannot be counted.

Arithmetic shift functions

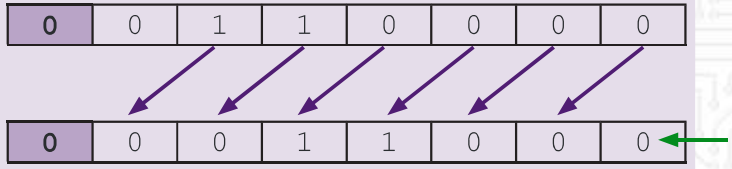
Moving the bits to either the left of the right, doubles (×2) or halves (÷2) the value with each.

Shift left



48 becomes 96
and so one shift left doubles the value

Shift right



48 becomes 24
and so one shift right halves the value