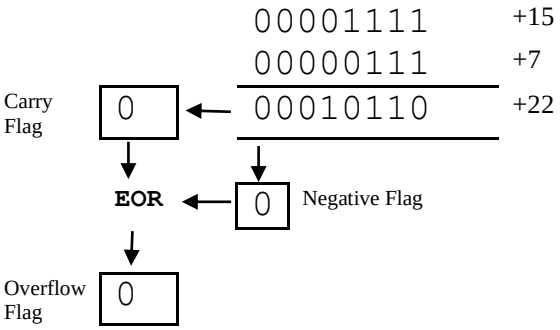
Detecting Overflow

Most processors have a carry flag and an overflow flag. It is easy to get them confused, because they do similar jobs. The carry flag and the negative flag have an exclusive OR performed upon them. The result is copied to the overflow flag.

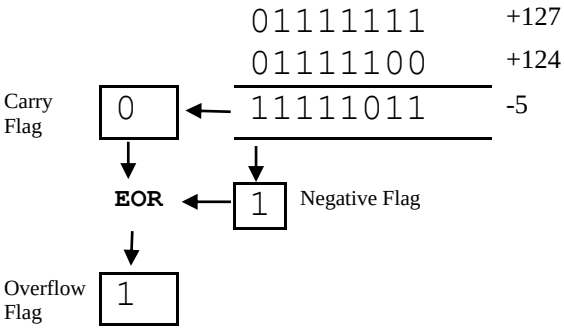
Add 2 small positive numbers

Here we shall add the numbers +15 and +7. The result, +22, will fit easily into 8 bits.



Add 2 large positive numbers

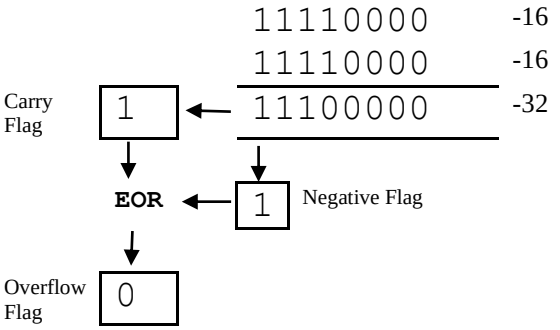
Here we shall add the numbers +127 and +124. The result, +251, will not fit into 8 bits (if two's complement is being used), which means that the result that is left in the accumulator is not to be trusted.



There has been a carry from bit 6 into bit 7 (the leftmost bit) but no carry from bit 7 into the carry flag. This means that we are adding two positives and the result is too large.

Add 2 small negative numbers

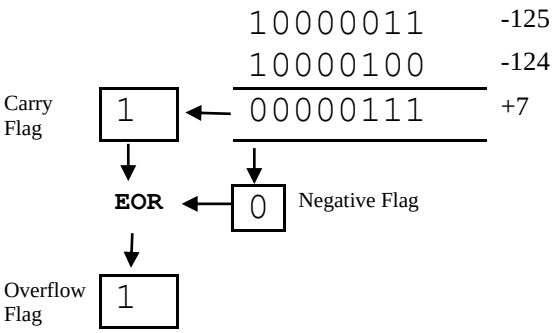
Here we shall add the numbers -16 and -16. The result, -32, will fit easily into 8 bits.



When you add the two most significant bits, with the carry from the previous column, you get 1, carry 1, so the carry is flag is set to indicate that a 1 has 'fallen off the end'. However, the most significant bit of the result is a 1, which indicates that it is a negative number – the negative flag will therefore be set. The fact that they are both 1 indicates that the result can be trusted. It doesn't matter that there was a carry. This is why the two's complement system of storing negative numbers is used.

Add 2 large negative numbers

Here we shall add the numbers -125 and -124. The result, -249, will not fit into 8 bits (if two's complement is being used), which means that the result that is left in the accumulator is not to be trusted.



There has been a carry from bit 7 into the carry flag, but the result that is left in the accumulator is positive. That must mean that we are adding two negative numbers, but getting a positive result. Overflow must have occurred.