Division

Pieter P

NEON doesn't have any integer division instructions, because they are expensive to implement in hardware. Luckily, you can often replace divisions by other instructions, like bit shifts and multiplications.

Dividing by powers of two

Unsigned integers

To divide an unsigned integer by 2^k, you can just use a bit shift by k bits to the right.

```
uint16_t div_by_16_a(uint16_t x) {
    return x / 16;
}

uint16_t div_by_16_b(uint16_t x) {
    return x >> 4; // 16 = 2⁴
}
```

Of course, any half-decent compiler will produce the same instructions for both of the functions above, so it's much better to write `x / 16`, because it clearly shows your intent.

To write the same division for NEON, you can use the `vshr_n_u16` intrinsic:

```
#include <arm_neon.h>

uint16x4_t div_by_16(uint16x4_t x) {
    return vshr_n_u16(x, 4); // 16 = 2⁴
}

uint16x8_t div_by_16(uint16x8_t x) {
    return vshrq_n_u16(x, 4); // 16 = 2⁴
}
```

The name is derived from vector shift right, `n` indicates a fixed number of bits is used, and `u16` is the type of elements in the vector (16-bit unsigned integers in this example).
There are two versions, the one without the `q` suffix operates on double-word vector registers (2×32 bits), and the one with the `q` suffix operates on quad-word vector registers (4×32 bits).