Arithmetic Operations

Here we will be covering the basic math

- Addition
- Subtraction
- Division
- Multiplication

Signed Numbers: a number that can be both positive or negative

Addition

Instructions for adding are quite simple, we will use the `add` pseudo instruction. Usage: add destination, source

```
; First we will prepare the registers
mov r8, 10
mov r9, 10
add r9, r8 ; the solution is stored int the r9 register
```

Subtraction

Subtracting follows the same approach.

Usage: sub destination, source

```
; prep the registers
mov r8, 10
mov r9, 3
sub r8, r9
```

Multiplication

This instruction is a bit trickier because we need to prepare some predefined general purpose registers to complete this operation. We can also specify whether we want to perform signed and unsigned division. The `mul` instruction handles unsigned multiplication and `imul` handles signed multiplication.

The following registers will be in use (and their lower tier versions can be used as well):

- Operand Size: rax
- Other Operand: rdx
- Lower part of result: rax

Syntax: rax * rdx

Example code

```
mov rax, 3
mov rdx, 4
mul rdx ; result is 12
```

Division/Modulo

When we divide, we also need to keep an extra register in mind and that will be our remainder value. The `div` instruction allows us to perform two instructions at once. Using `idiv` will accommplish the same goal but for unsigned integers.

Registers in use:

- rax
- rcx

Syntax: rax:rcx

Example code

```
mov rax, 10 ; dividend
mov rcx, 5 ; divisor
div rcx
mov rdi, rax ; quotient is stored in rax
```

External Links

- x86 Assembly/Arihmetic
- NASM Arithmetic Tutorial