# AC500 V3 PLC – Mathematical operators
ADD, SUB, MUL, DIV, GT, … Operators and Examples

The Mathematical operators, also known as arithmetic operators, are used for simple mathematical calculations. The operators ADD, SUB, MUL and DIV represent the basic mathematical operations +, -, * & /. The operators GT, GE, EQ, NE, LE & LT are comparing two values and outputting a Boolean signal. Here the basic mathematical operations are >, $\geq$, =, $\neq$, $\leq$ & <.

- The ADD / MUL operator calculates the Sum / Product of the inputs. These operators can have more than two inputs.
- The SUB/ DIV operator calculates the Difference / Quotient. Here only two inputs are possible.
- The operator GT (greater than) checks if the first input is greater than the second input. The same way also the other operators GE (greater equal), EQ (equal), NE (not equal), LE (lower equal), LT (lower than) are working.
- All operators work with all integer and real data types. Also mixed inputs are possible.

**Examples:**
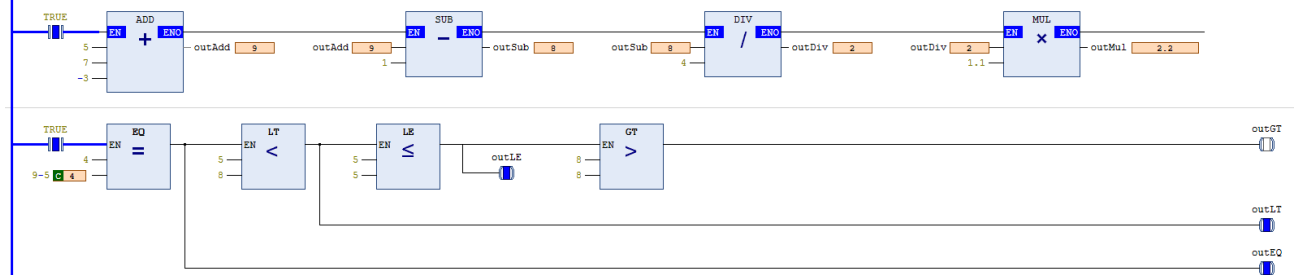


The screenshot above shows a simple logic in ladder diagram. The top row shows add, subtract, divide & multiply. The bottom row shows as example equal, lower than, lower equal and greater than.

Starting on top left. The ADD block has 5, 7 and -3 as inputs. The sum is 9. The SUB block subtracts 1 from 9, resulting in 8. The DIV block divides 8 by 4, resulting in 2. The MUL block multiplies the 2 with 1.1, resulting in 2.2.

In the bottom row it is checked if 4 is equal to 9-4 , which is the case. Also 5 is lower than 8. The third block checks that 5 is lower or equal to 5, which is true. The last condition check whether 8 is greater than 8 which is not the case, as they are equal.

When working with structured text the operators are replaced by the symbols as shown in the table below.

| LD, CFC, FBD | ADD | SUB | MUL | DIV | GT | GE | EQ | NE | LE | LT |
|---|---|---|---|---|---|---|---|---|---|---|
| **ST, SFC** | + | - | * | / | >= | > | = | <> | <= | < |

The same logic as shown and described above, can be found in structured text below.

```
outAdd  9      := 5 + 7 - 3;
outSub  8      := outAdd  9    - 1;
outDiv  2      := outSub  8   / 4;
outMul    2.2    := outDiv  2    * 1.1;

OutEQ TRUE  := 4 = 9-5;
OutLT TRUE  := 5 < 8;
OutLE TRUE  := 5 <= 5;
OutGT FALSE  := 8 > 8; RETURN
```

```
outAdd := 5 + 7 - 3;
outSub := outAdd - 1;
outDiv := outSub / 4;
outMul := outDiv * 1.1;

OutEQ := 4 = 9-5;
OutLT := 5 < 8;
OutLE := 5 <= 5;
OutGT := 8 > 8;
```