

ADD

Сложение переменных типов: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL и LREAL.

Две переменных типа TIME можно складывать (напр. $t\#45s + t\#50s = t\#1m35s$). Результат имеет тип TIME.

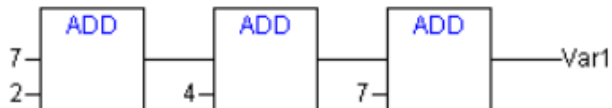
Пример IL:

```
LD 7
ADD 2,4,7
ST Var1
```

Пример ST:

```
var1 := 7+2+4+7;
```

Пример FBD:



MUL

Перемножение значений переменных типов: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL и LREAL.

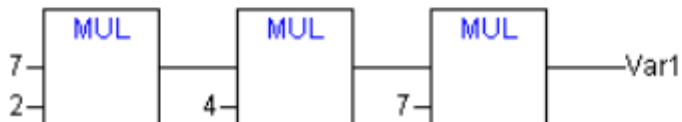
Пример IL:

```
LD 7
MUL 2,4,7
ST Var1
```

Пример ST:

```
var1 := 7*2*4*7;
```

Пример FBD:



SUB

Вычитание значений переменных типов: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL и LREAL.

Переменной TIME можно присвоить результат вычитания двух других переменных типа TIME. Отрицательное время не определено.

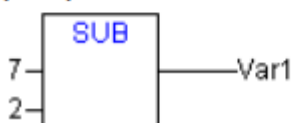
Пример IL:

```
LD 7
SUB 2
ST Var1
```

Пример ST:

```
var1 := 7-2;
```

Пример FBD:



DIV

Деление значений переменных типов: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL и LREAL.

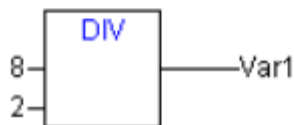
Пример IL:

```
LD 8
DIV 2
ST Var1 (* Result is 4 *)
```

Пример ST:

```
var1 := 8/2;
```

Пример FBD:



Примечание: Определив в своем проекте функции с именами **CheckDivByte**, **CheckDivWord**, **CheckDivDWord** и **CheckDivReal** вы сможете контролировать делитель и обрабатывать, например, деление на 0.

Внимание: Результат деления на 0 может отличаться на разных целевых платформах

Рассмотрим простейший пример применения CheckDivReal.

Пример функции CheckDivReal:

```
FUNCTION CheckDivReal : REAL
VAR_INPUT
    divisor:REAL;
END_VAR
```

```
IF divisor = 0 THEN
    CheckDivReal:=1;
ELSE
    CheckDivReal:=divisor;
END_IF;
```

Оператор DIV использует выход функции CheckDivReal как делитель. В следующей программе этот прием предотвращает деление на 0, делитель (d) заменяется с 0 на 1. В итоге получается результат 799.

```
PROGRAM PLC_PRG
VAR
    erg:REAL;
    v1:REAL:=799;
    d:REAL;
END_VAR
```

```
erg:= v1 / d;
```

Внимание: CheckDiv-функции, содержащиеся в библиотеке Check.Lib представляют собой примеры их реализации. Прежде чем использовать эту библиотеку, убедитесь, что она работает так, как нужно в вашем случае, либо создайте собственные функции непосредственно в вашем проекте.

MOD

Остаток от деления значений переменных типов: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT. Результат всегда целое число.

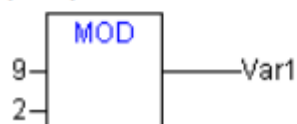
Пример IL:

```
LD 9
MOD 2
ST Var1 (* Результат 1 *)
```

Пример ST:

```
var1 := 9 MOD 2;
```

Пример FBD:

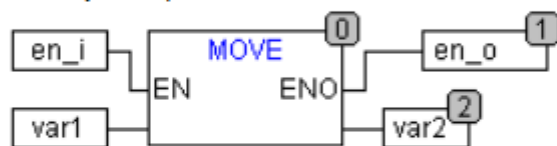


MOVE

Присвоение значения одной переменной другой соответствующего типа. В графических редакторах CFC и LD существует возможность управлять разрешением работы блока (разрешать или запрещать операцию) с помощью входов EN/ENO. В FBD этого делать нельзя.

Пример применения EN/ENO в CFC:

Только если значение *en_i* равно TRUE, значение переменной *var1* будет присвоено *var2*.



Пример IL:

```
LD ivar1
MOVE ivar2
ST ivar2 (* Результат: ivar2 принимает значение
ivar1 *)
```

(! Аналогичный результат дает:

```
LD ivar1
ST ivar2 )
```

Пример ST:

```
ivar2 := MOVE(ivar1);
(! Аналогичный результат дает: ivar2 := ivar1; )
```

Операторы выборки

SEL

Бинарный выбор.

```
OUT := SEL(G, IN0, IN1) означает:  
OUT := IN0 if G=FALSE;  
OUT := IN1 if G=TRUE.
```

IN0, IN1 и OUT могут быть любого типа, G должно быть типа BOOL. Бинарный выбор возвращает одно из двух: IN0, если G ЛОЖЬ, или IN1, если G ИСТИНА.

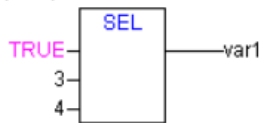
Пример IL:

```
LD TRUE  
SEL 3,4 (* IN0 = 3, IN1 = 4 *)  
ST Var1 (* Результат = 3 *)  
LD FALSE  
SEL 3,4  
ST Var1 (* Результат is 3 *)
```

Пример ST:

```
Var1:=SEL(TRUE,3,4); (* Результат is 4 *)
```

Пример FBD:



Внимание: Выражение, стоящее перед IN0 или IN1, может не вычисляться, если соответствующий вход не выбран, что определяется значением G.

MAX

Функция максимум возвращает наибольшее из двух значений

```
OUT := MAX(IN0, IN1)
```

IN0, IN1 и OUT могут быть любого типа.

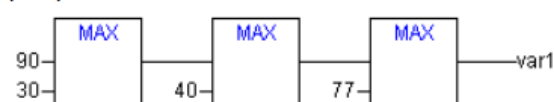
Пример IL:

```
LD 90  
MAX 30  
MAX 40  
MAX 77  
ST Var1 (* Результат 90 *)
```

Пример ST:

```
Var1:=MAX(30,40); (* Результат 40 *)  
Var1:=MAX(40,MAX(90,30)); (* Результат 90 *)
```

Пример FBD:



MIN

Функция минимум возвращает наименьшее из двух значений.

```
OUT := MIN(IN0, IN1)
```

IN0, IN1 и OUT могут быть любого типа.

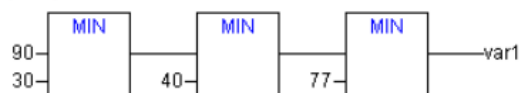
Пример IL:

```
LD 90  
MIN 30  
MIN 40  
MIN 77  
ST Var1 (* Результат 30 *)
```

Пример ST:

```
Var1:=MIN(90,30); (* Результат 30 *)  
Var1:=MIN(MIN(90,30),40); (* Результат 30 *)
```

Пример FBD:



LIMIT

Ограничитель

```
OUT := LIMIT(Min, IN, Max) означает:  
OUT := MIN (MAX (IN, Min), Max)
```

Max задает верхнюю и Min нижнюю границы ограничителя. Если IN больше верхнего или меньше нижнего пределов, результат 'обрезается' соответственно до Max или Min.

IN и OUT могут быть любого типа.

Пример IL:

```
LD 90  
LIMIT 30,80  
ST Var1 (* Результат 80 *)
```

Пример ST:

```
Var1:=LIMIT(30,90,80); (* Результат 80 *)
```

MUX

Мультиплексор. Возвращает K-е значение из входных переменных.

```
OUT := MUX(K, IN0, ..., INn) означает:  
OUT := INK.
```

IN0, ..., INn и OUT могут быть любого типа. Переменная K должна быть BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT или UDINT.

Пример IL:

```
LD 0  
MUX 30,40,50,60,70,80  
ST Var1 (* Результат 30 *)
```

Пример ST:

```
Var1:=MUX(0,30,40,50,60,70,80); (* Результат 30 *)
```

Внимание: В результате оптимизации выражение, стоящее перед входом, может не вычисляться, если соответствующий вход не выбран. В режиме эмуляции все выражения вычисляются.

Операторы сравнения

GT

Больше

Двоичный оператор возвращает TRUE, если значение первого параметра больше второго.

Операнды могут быть типов BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME и STRING.

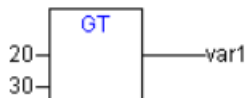
Пример IL:

```
LD 20
GT 30
ST Var1 (* Результат ЛОЖЬ *)
```

Пример ST:

```
VAR1 := 20 > 30 > 40 > 50 > 60 > 70;
```

Пример FBD:



LT

Меньше

Двоичный оператор возвращает TRUE, если значение первого параметра меньше второго.

Операнды могут быть типов BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME и STRING.

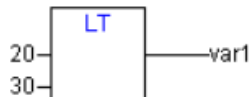
Пример IL:

```
LD 20
LT 30
ST Var1 (* Результат ИСТИНА *)
```

Пример ST:

```
VAR1 := 20 < 30;
```

Пример FBD:



LE

Меньше или равно

Двоичный оператор возвращает TRUE, если значение первого параметра меньше или равно второму.

Операнды могут быть типов BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME и STRING.

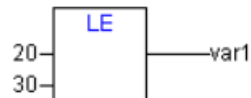
Пример IL:

```
LD 20
LE 30
ST Var1 (*Результат - ИСТИНА*)
```

Пример ST:

```
VAR1 := 20 <= 30;
```

Пример FBD:



GE

Больше или равно

Двоичный оператор возвращает TRUE, если значение первого параметра больше или равно второму.

Операнды могут быть типов BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME и STRING.

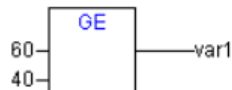
Пример IL:

```
LD 60
GE 40
ST Var1 (*Результат - ИСТИНА*)
```

Пример ST:

```
VAR1 := 60 >= 40;
```

Пример FBD:



EQ

Равно

Двоичный оператор возвращает TRUE, если значение первого параметра равно второму.

Операнды могут быть типов BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME и STRING.

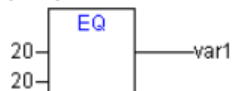
Пример IL:

```
LD 40
EQ 40
ST Var1 (*Результат - ИСТИНА*)
```

Пример ST:

```
VAR1 := 40 = 40;
```

Пример FBD:



NE

Не равно

Двоичный оператор возвращает TRUE, если значение первого параметра не равно второму.

Операнды могут быть типов BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME и STRING.

Пример IL:

```
LD 40
NE 40
ST Var1 (*Результат - FALSE*)
```

Пример ST:

```
VAR1 := 40 <> 40;
```

Пример FBD:

