

## PROGRAMOWALNE SYSTEMY MECHATRONIKI

### Laboratorium nr 6

#### **Podstawy programowania PLC – operacje matematyczne**

### 1. Katalog *Math Functions*

W katalogu *Integer Functions* w oprogramowaniu STEP7 v.5.4 znajdują się bloki umożliwiające wykonywanie operacji arytmetycznych na liczbach całkowitych (typ INT i DINT), są to następujące bloki (rys. 27.a):

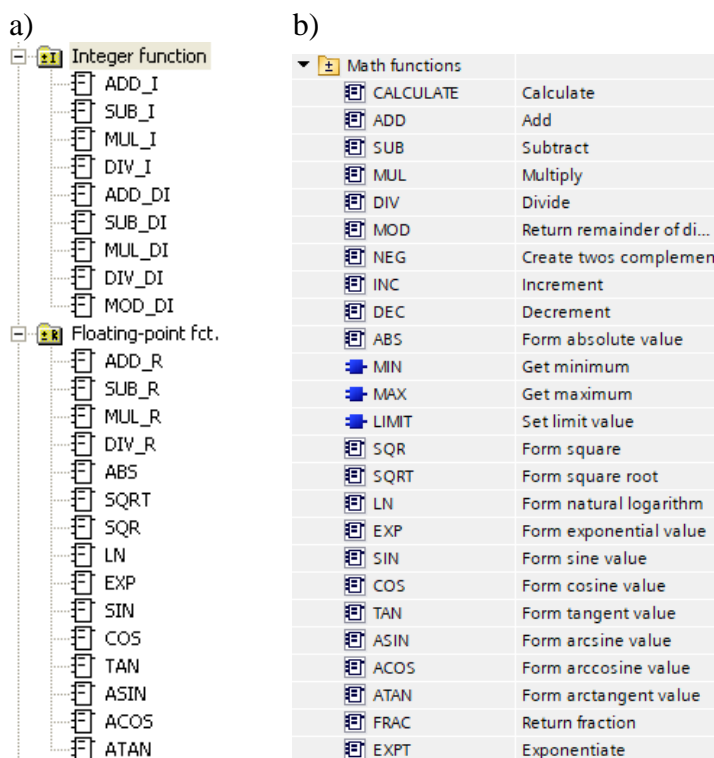
Bloki umożliwiające działania na zmiennych typu INT:

- ADD\_I – dodawanie wartości wejść,
- SUB\_I – odejmowanie,
- MUL\_I – mnożenie,
- DIV\_I – dzielenie.

Bloki umożliwiające działania na zmiennych typu DINT są analogiczne.

W katalogu *Floating-point fct.* znajdują się bloki umożliwiające wykonywanie operacji na zmiennych typu real.

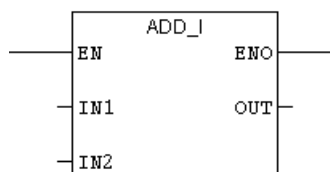
W **TIA Portal** zakres działań na liczbach został poszerzony. Bloki dostępne w katalogu *Math functions* przedstawiono na rys. 27.b).



Rys. 27. a) Katalog *Integer Functions* oraz *Floating-point fct.* w STEP7 v.5.4, b) katalog *Math functions* w TIA Portal

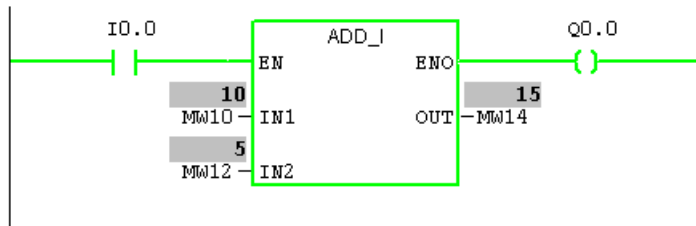
#### a) Suma liczb całkowitych (*ADD\_I*)

Blok sumowania liczb całkowitych (ang. *Add Integer*) umożliwia dodanie 16-bitowych liczb na wejściu IN1 i IN2, w chwili, gdy na wejściu EN pojawi się stan wysoki. Wynik działania jest zapisywany do adresu na wyjściu OUT.



Rys. 23. Blok *ADD\_I*

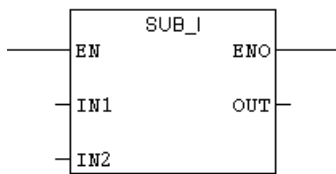
Przykład zastosowania bloku **ADD\_I** przedstawiono na rys. 24.



Rys. 24. Przykład zastosowania bloku **ADD\_I**

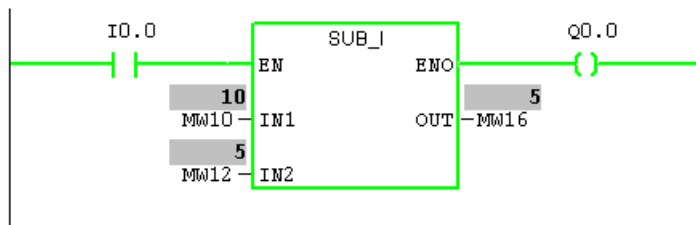
b) Różnica liczb całkowitych (**SUB\_I**)

Blok różnicy liczb całkowitych (ang. *Subtract Integer*) umożliwia odejmowanie 16-bitowych liczb, od liczby na wejściu IN1 jest odejmowana liczba na wejściu IN2, w chwili, gdy na wejściu EN pojawi się stan wysoki. Wynik działania jest zapisywany do adresu na wyjściu OUT.



Rys. 25. Blok **SUB\_I**

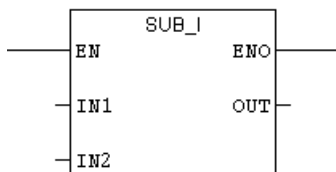
Przykład zastosowania bloku **SUB\_I** przedstawiono na rys. 26.



Rys. 26. Przykład zastosowania bloku **SUB\_I**

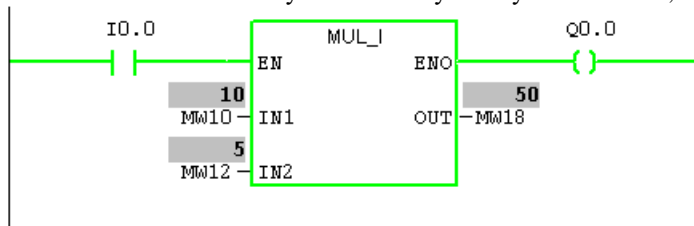
c) Iloczyn liczb całkowitych (**MUL\_I**)

Blok iloczynu liczb całkowitych (ang. *Multiply Integer*) umożliwia mnożenie 16-bitowych liczb na wejściu IN1 i IN2, w chwili, gdy na wejściu EN pojawi się stan wysoki. Wynik działania jest zapisywany do adresu na wyjściu OUT.



Rys. 27. Blok **MUL\_I**

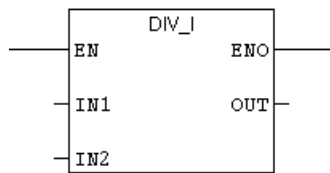
Przykład zastosowania bloku **MUL\_I** przedstawiono na rys. 28.



Rys. 28. Przykład zastosowania bloku *MUL\_I*

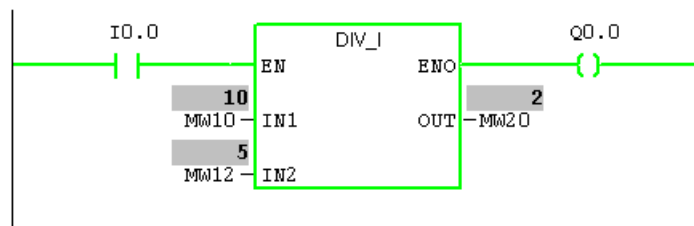
d) Dzielenie liczb całkowitych (*DIV\_I*)

Blok dzielenia liczb całkowitych (ang. *Divide Integer*) umożliwia podzielenie 16-bitowej liczby na wejściu IN1 przez 16-bitową liczbę na wejściu IN2, w chwili, gdy na wejściu EN pojawi się stan wysoki. Wynik działania jest zapisywany do adresu na wyjściu OUT.



Rys. 29. Blok *DIV\_I*

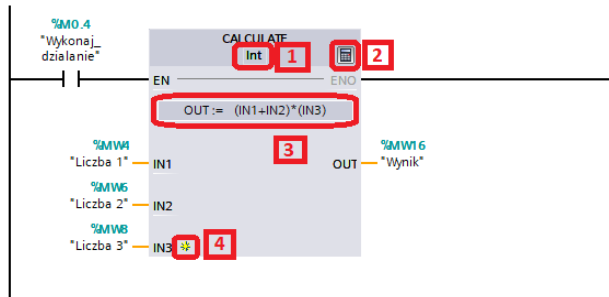
Przykład zastosowania bloku *DIV\_I* przedstawiono na rys. 30.



Rys. 30. Przykład zastosowania bloku *DIV\_I*

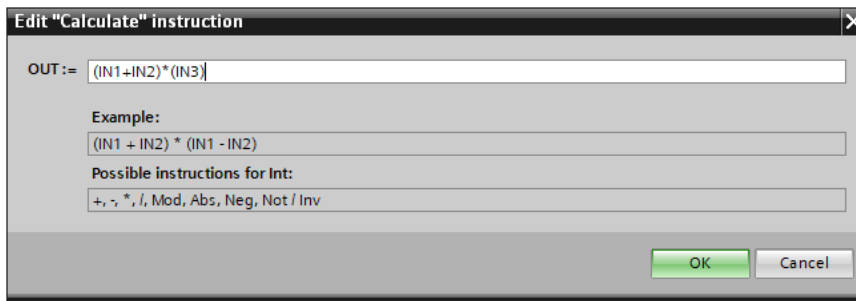
e) Blok *Calculate*

Blok *Calculate* zastępuje działanie innych bloków dyskretnych wykonujących pojedyncze operacje na liczbach. Posiada on wejście EN i wyjście ENO sygnału RLO, wejścia IN1,...,INn (*n* – liczba wejść) liczb na który ma zostać wykonane działanie oraz wyjście OUT, na którym generowany jest wynik wynikający z obliczenia formuły dla danych wartości wejść. Umożliwia on wybór typu zmiennej (1), rys. 31, w której zostanie wygenerowany wynik. Kliknięcie jednokrotne w symbol kalkulatora (2) lub dwukrotne w obszar wpisywania formuły (3) powoduje otwarcie okna edycji formuły rys. 32. Kliknięcie gwiazdki w lewym dolnym rogu bloku *Calculate* (4) powoduje zwiększenie o jedno liczby wejść do bloku.



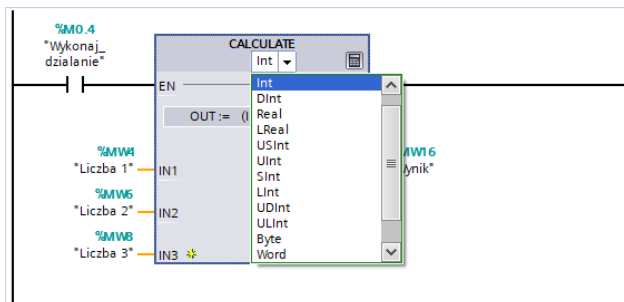
Rys. 31. Blok *Calculate*

Okno edycji formuły (rys. 32) umożliwia wpisanie formuły umożliwiającej wyznaczenie wyjścia OUT na podstawie stanu wejść  $IN_n$ . W zależności od typu wybranej zmiennej różny może być zestaw instrukcji możliwych do wykonania na wartościach wejść, wyszczególniony w dolnej części okna. Po zakończeniu edycji formuły wprowadzone zmiany akceptujemy przyciskiem OK.



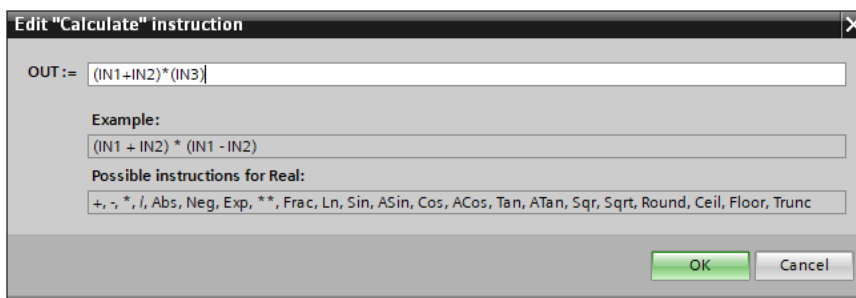
Rys. 32. Widok okna edycji formuły polecenia *Calculate*

Instrukcja *Calculate* umożliwia zapisanie wyniku działania w zmiennej o typie wymienionym na rozwijanej liście, pokazanej na rys. 33.



Rys. 33. Lista typów zmiennych wyniku operacji *Calculate*

Wybór typu zmiennej wyniku np. Real powoduje wyświetlenie szerszej liczby możliwych operacji na danych, niż w przypadku zmiennej Int, rys. 34.

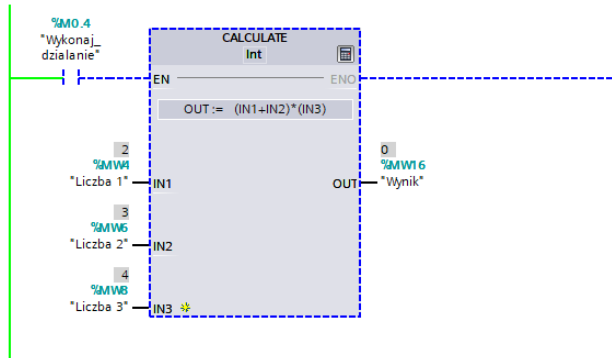


## Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

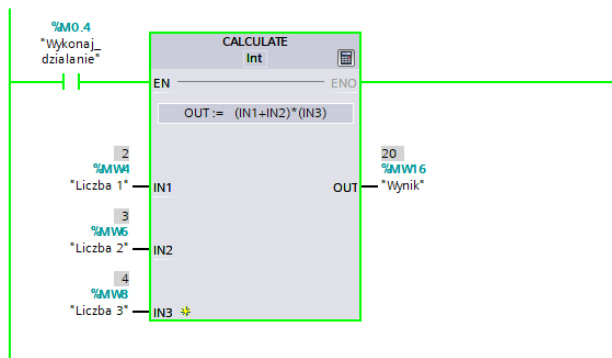
Rys. 34. Okno edycji formuły w przypadku wyboru zmiennej Real do reprezentacji wyniku

Wykonanie bloku *Calculate* następuje, gdy na wejściu EN jest stan wysoki. W przeciwnym razie blok nie jest wykonywany (rys.35).



Rys. 35. Blok *Calculate* nie jest wykonywany

Pojawienie się stanu wysokiego na wejściu EN powoduje wykonanie operacji, rys. 36.



Rys. 36. Blok *Calculate* jest wykonywany

### 2. Zadania do wykonania:

1. Napisać program realizujący kalkulator, wyznaczający wynik w reakcji na stan wysoki na wejściu **I0.[nr\_zespołu-1]**. Kalkulator oblicza wynik działania podanego w tabeli 1 na 3 cyfrach wprowadzanych do systemu z zastosowaniem zadajnika BCD. Wynik prezentowany jest na wyświetlaczu BCD. Program zrealizuj w dwóch wersjach: a) z zastosowaniem dyskretnych bloków realizacji poszczególnych działań, b) z zastosowaniem bloku *Calculate*.

Tabela 1. Działania dla poszczególnych zespołów

Nr zespołu	częstotliwość [Hz]
1	$(a+b)*c$
2	$(b+c)/a$
3	$(c+a)*b$
4	$(a+b)/c$
5	$a*b+c$
6	$(b+c)*a$
7	$(c+a)/b$
8	$a*c+b$

### 3. Sprawozdanie powinno zawierać:

1. Wstęp teoretyczny.
2. Opis realizowanych zadań.

## **Katedra Mechaniki Stosowanej i Robotyki**

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

3. Listingi programów z komentarzem dotyczącym funkcji poszczególnych linii kodu.
4. Opis działania programów z ilustracją graficzną na podstawie działania PLC.
5. Wnioski.