

## 9. Sterowniki PLC – język ST

Język *tekstu strukturalnego* ST jest odpowiednikiem języka wysokiego poziomu, zawiera podobny zestaw instrukcji jak Pascal czy C. Podstawowymi elementami języka są *wyrażenia* i *instrukcje*.

*Wyrażenie* zwraca wartość określonego typu, wartość ta wyznaczana jest na podstawie występujących w tym wyrażeniu stałych, zmiennych, operatorów oraz funkcji.

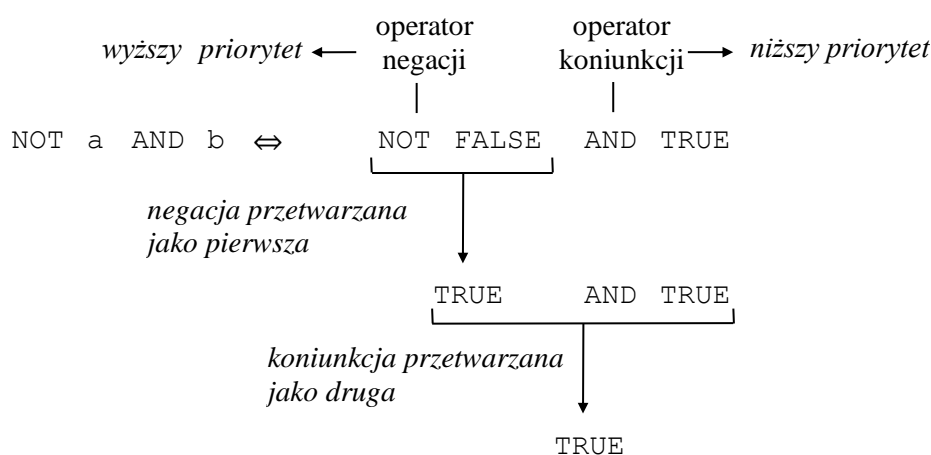
*Instrukcja* element składni języka określający wybraną operację (czynność). W językach typu Asembler każda instrukcja odpowiada jednemu rozkazowi procesora w językach wysokiego poziomu instrukcja tłumaczona jest na kilka rozkazów. Każda instrukcja w ST musi być zakończona średnikiem (;).

Język *ST* pozwala na zwięzły zapis warunków logicznych i w związku z tym będzie wykorzystywany do zapisywania *warunków przejść* w metodzie *SFC* omawianej w punkcie 10. Opis składni języka został ograniczony do operatorów i instrukcji przypisania.

### 9.1. Opis języka

#### 9.1.1. Operatory

*Operatory* łącznie z *operandami* tworzą *wyrażenia* języka. Wyznaczanie wartości wyrażenia sprowadza się do przetwarzania jego kolejnych operatorów z zachowaniem kolejności wynikającej z ich priorytetów. Wyznaczenie wartości operatora polega na wykonaniu operacji wynikającej z nazwy czy symbolu operatora, operacja ta wykonywana jest na operandach operatora, które można interpretować jako dane przetwarzane przez operator. Przykładowe wyrażenie języka *ST* wraz z diagramem pokazującym kolejność operacji wykonywanych przy obliczaniu jego wartości zostało przedstawione na rys. 9.1. W przykładzie założono, że zmienne *a* i *b* mają wartości odpowiednio: FALSE i TRUE, co oznacza, że rozważane wyrażenie NOT *a* AND *b* jest równoważne wyrażeniu NOT FALSE AND TRUE.



Rys.9.1. Obliczanie wartości wyrażenia: NOT a AND b

Z analizy rys. 9.1 wynika, że operatory oprócz priorytetów różnią się od siebie również liczbą operandów, na których działają. Operator negacji jest operatorem jednoargumentowym (działa na jednym operandzie), a operator koniunkcji jest operatorem dwuargumentowym (wymaga dwóch operandów).

Biorąc pod uwagę typ operandów i typ wartości wynikowej zwykle wyróżnia się grupę operatorów logicznych, relacyjnych i arytmetycznych (tab. 9.1-9.3). Zgodnie z normą funkcję operatorów języka pełnią również:

- nawiasy ( ), które pozwalają na zmianę kolejności wykonywania operacji,
- funkcje (zarówno standardowe, jak i zdefiniowane przez użytkownika).

Tab. 9.1. Operatory logiczne

<i>Symbol</i>	<i>Opis</i>	<i>Przykład</i>
NOT	negacja logiczna	NOT TRUE wartość: FALSE
AND lub &	koniunkcja	TRUE AND FALSE wartość: FALSE
XOR	alternatywa wykluczająca	TRUE XOR TRUE wartość: FALSE
OR	alternatywa	TRUE OR FALSE wartość: TRUE

Tab. 9.2. Operatory relacyjne

<i>Symbol</i>	<i>Opis</i>	<i>Przykład</i>
>	większy	3 > 2 wartość: TRUE
>=	większy bądź równy	3 >= 2 wartość: TRUE
<	mniejszy	3 < 2 wartość: FALSE
<=	mniejszy bądź równy	3 <= 2 wartość: FALSE
=	równy	t#25h = t#1d1h wartość: TRUE
<>	różny	2#100 <> 4 wartość: FALSE

Tab. 9.3. Operatory arytmetyczne języka ST

<i>Symbol</i>	<i>Opis</i>	<i>Przykład</i>
**	potęgowanie	3**2 wartość: 9
-	negacja arytmetyczna	-3 wartość: -3
*	mnożenie	3*2 wartość: 6
/	dzielenie	3/2 wartość: 1.5
MOD	reszta z dzielenia	3 MOD 2 wartość: 1
+	dodawanie	3+2 wartość: 5
-	odejmowanie	3-2 wartość: 1

Priorytety operatorów pokazuje tab. 9.4. Operatory umieszczone w tej samej komórce tabeli mają taki sam priorytet i są przetwarzane w kolejności występowania w wyrażeniu, zaczynając od strony lewej do prawej.

Tab. 9.4. Priorytety operatorów w języku ST

Priorytet										
najwyższy $\xrightarrow{\hspace{10em}}$ najniższy										
( )	funkcja	**	- NOT	* / MOD	+ -	> >= < <=	= <>	AND &	XOR	OR

### 9.1.2. Instrukcje

Instrukcje języka pozwalają na wykonanie określonych czynności (działań). W ST zdefiniowane są instrukcje znane z innych języków programowania takie jak instrukcja warunkowa (IF), instrukcja wyboru (CASE) czy instrukcje pętli (FOR, REPEAT, WHILE). Opis instrukcji zostanie ograniczony tylko do podstawowej instrukcji każdego języka programowania, tzn. *instrukcji przypisania*. Instrukcja ta zastępuje aktualną wartość wybranej zmiennej nową wartością wyznaczoną w trakcie interpretacji wyrażenia, np.:

$$z := (b \text{ AND } c) \text{ OR } (\text{NOT } a);$$

przypisuje zmiennej z wartość wynikającą z wartości wyrażenia logicznego  $b \cdot c + \bar{a}$ . Nawiasy w powyższej instrukcji nie są niezbędne, zostały zastosowane w celu zwiększenia czytelności wyrażenia.

## 9.2. Zadania

Zapisz w języku ST warunek logiczny prawdziwy, jeśli:

- zmienna x1 ma wartość FALSE,
- zmienna x1 ma wartość TRUE a zmienna x2 ma wartość FALSE,
- zmienna x1 ma wartość TRUE lub zmienna x2 ma wartość FALSE,
- zmienna x1 ma wartość TRUE a zmienna x2 ma wartość FALSE lub zmienna x1 ma wartość FALSE a zmienna x2 ma wartość TRUE.