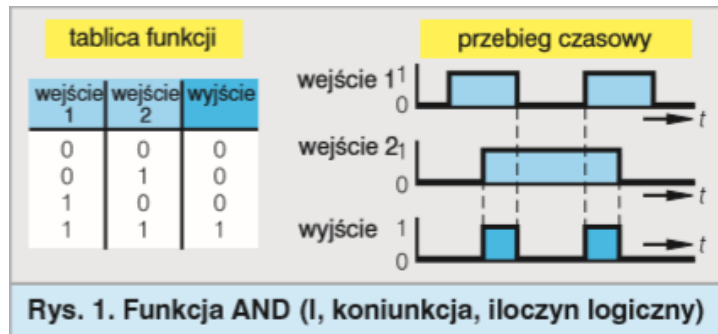


Podstawowe funkcje logiczne.

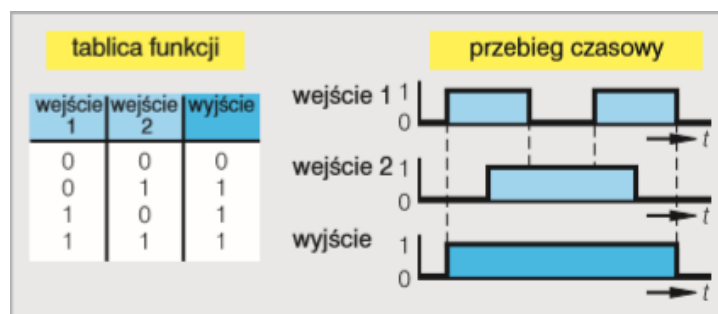
Podstawowymi funkcjami logicznymi są:

- **AND** – koniunkcja, iloczyn logiczny, I,
- **OR** – alternatywa, suma logiczna, LUB,
- **NOT** – negacja, zaprzeczenie, NIE.



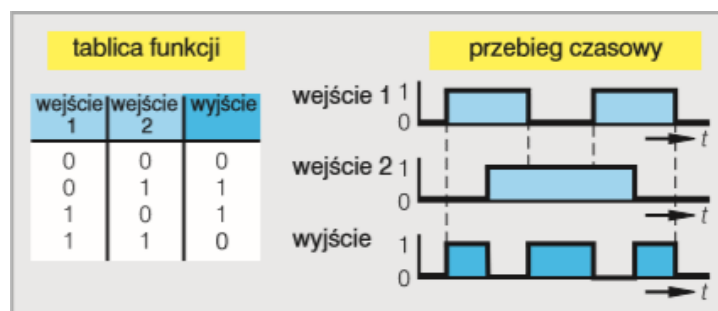
Na rys. 1 zdefiniowano dwuargumentową funkcję logiczną **AND**, podając jej tablicę wartości (tablicę funkcji) oraz wykres czasowy. Przyjmuje ona stan logiczny 1, jeżeli wszystkie jej wejścia są również w stanie 1.

Funkcja logiczna **OR** (alternatywa) przyjmuje wartość logiczną 1, jeżeli co najmniej jedno z jej wejść będzie miało również wartość 1 (rys. 2).



Rys.2 Funkcja OR (LUB, alternatywa, suma logiczna)

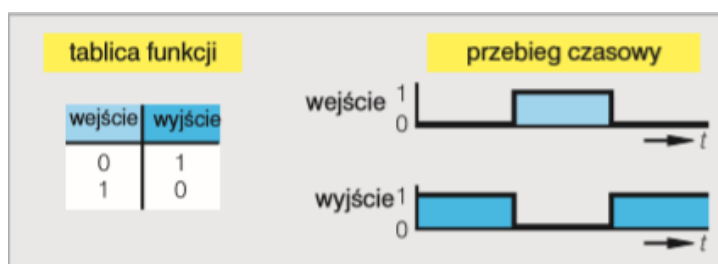
Wyróżnia się jeszcze funkcję alternatywy zwaną wykluczającą (wyłączającą) **XOR** (ang. *eXclusive OR*), która przyjmuje wartość logiczną 1, jeżeli tylko i wyłącznie jedno z jej wejść będzie miało również wartość 1 (rys. 3).



Rys. 3. Funkcja XOR (ALBO, alternatywa wykluczająca)

Przy programowaniu funkcji logicznych, których argumentami są sygnały binarne pochodzące z sensorów, należy uwzględnić następujące informacje:

- sensory wyposażone są na ogół w dwa wyjścia, jedno stanowiące negację drugiego. Oznacza to, że wystąpienie stanu, który ma być sygnalizowany przez sensor, objawia się pojawieniem sygnału 1 na jednym z wyjść (**wyjście powtarzające**), natomiast sygnału 0 – na drugim (**wyjście zanegowane**),
- jeżeli sygnał z sensora skojarzony zostanie zestykiem normalnie otwartym (**NO**, ang. *Normally-Open*), to gdy sensor wygeneruje sygnał 1, zestyk ten zostanie zamknięty, natomiast jeżeli wygeneruje sygnał 0, zestyk pozostanie otwarty. W analogicznych sytuacjach zestyk normalnie zamknięty (**NC**, *Normally-Closed*) zachowa się dokładnie przeciwnie. Zatem zestyk NC stanowi negację zestyku NO (funkcja NOT, rys. 4).



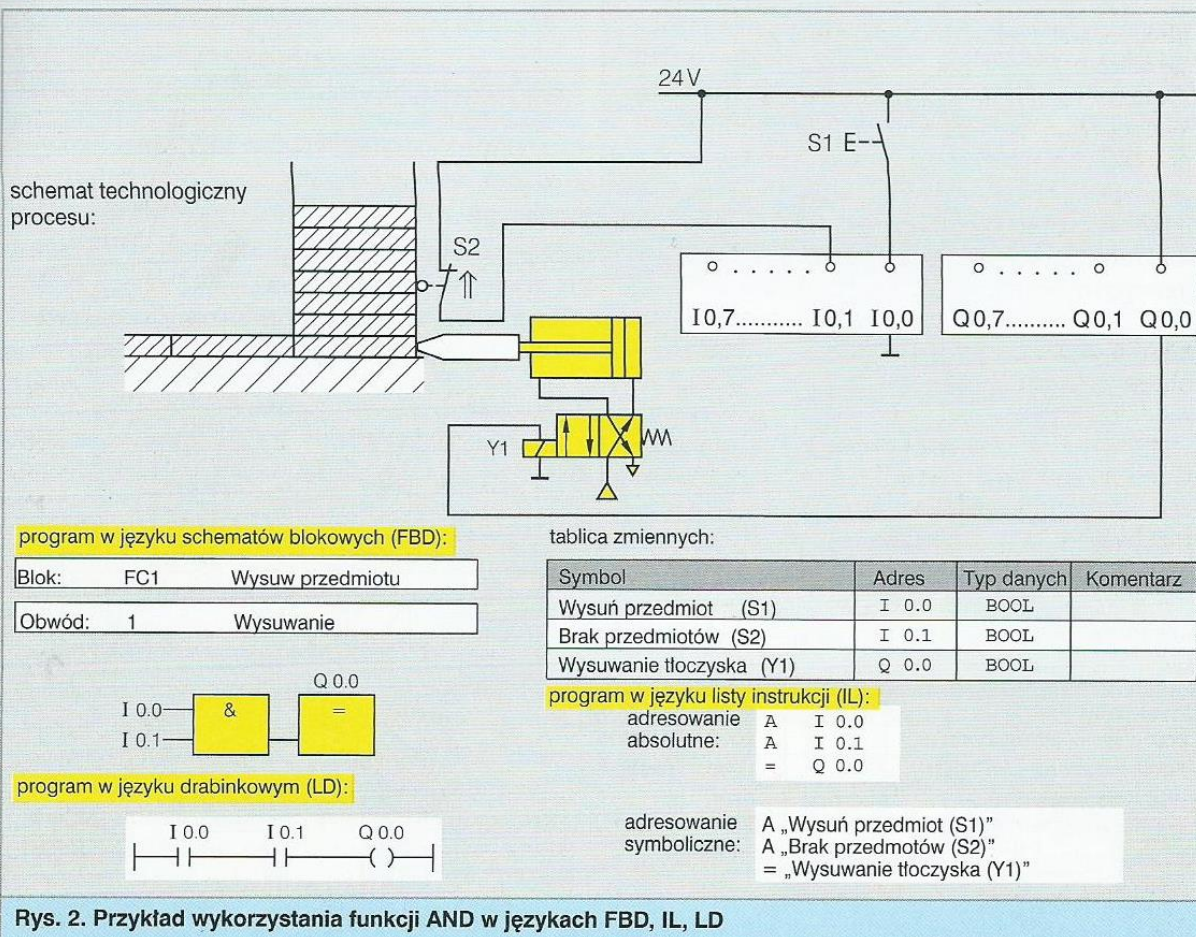
Rys. 4. Funkcja NOT (NIE, negacja, zaprzeczenie).

Wymienione powyżej funkcje AND, OR i NOT stanowią **system funkcjonalnie pełny**, gdyż z ich pomocą można zbudować każdy układ logiczny (kombinacyjny lub sekwencyjny – z pamięcią). Jest to system nieminimalny, bowiem podobny cel można osiągnąć, stosując wyłącznie funkcje **NAND** (ang. *Not AND*), na bazie których można zrealizować wszystkie funkcje podstawowe (AND, OR, NOT). Tak, więc funkcja NAND stanowi **system funkcjonalnie pełny minimalny**. Podobnie jest w przypadku funkcji **NOR** (ang. *Not OR*).

Przykład wykorzystania funkcji AND w programie sterującym:

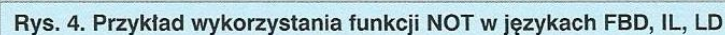
Siłownik dwustronnego działania powinien wysunąć jeden przedmiot z komory magazynu opadowego po naciśnięciu przycisku łącznika S1. Jeżeli liczba przedmiotów w magazynie osiągnie stan rezerwy, magazyn jest traktowany jako pusty (**rys. 2**) i proces wysuwania kolejnych przedmiotów powinien zostać zatrzymany. Poziom rezerwy sygnalizuje wyłącznik krańcowy S2.

Rozwiązanie zadania zostało przedstawione na rys. 2 w postaci programów napisanych w języku listy instrukcji (IL), języku drabinkowym (LD) oraz w języku bloków funkcyjnych (FBD).



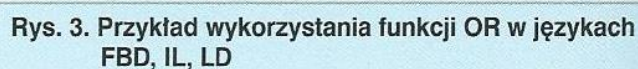
Rys. 2. Przykład wykorzystania funkcji AND w językach FBD, IL, LD

W algorytmie sterowania magazynem opadowym, rozważanym na poprzedniej stronie, należy zapewnić prawidłową współpracę z transporterem taśmowym, tak aby uniemożliwić wysuwanie kolejnego przedmiotu, gdy poprzedni znajduje się jeszcze na taśmie. Obecność przedmiotu na taśmie wykrywa sensor stykowy (NO) B3 (**rys. 4**).



Wysuwanie tłoczyska siłownika dwustronnego działania następuje w trakcie wciskania dowolnego z przycisków: ręcznego lub nożnego. Po zwolnieniu przycisku tłoczysko siłownika powinno się wycofać do pozycji startowej.

Symbol	Adres	Typ danych	Komentarz
Przycisk nożny (S1)	I 0.0	BOOL	
Przycisk ręczny (S2)	I 0.1	BOOL	
Wysuwanie tłoczyska (Y1)	Q 0.0	BOOL	



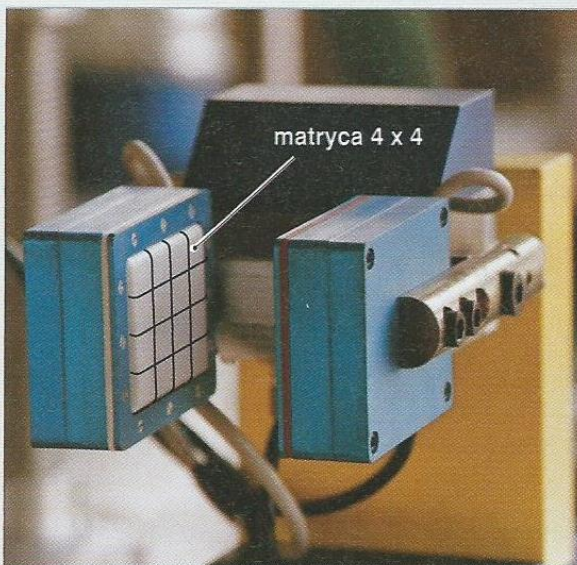
Przykład kombinacyjnego układu sterowania chwytakiem robota:

Chwytnik robota jest wyposażony w szesnaście sensorów dotykowych ułożonych w matrycy 4x4 (rys. 2). Uchwycenie detalu sygnalizowane jest przez jeden z szesnastu sensorów – w idealnym przypadku powinien to być jeden z czterech środkowych sensorów (rys. 3). Uchwycenie nieprawidłowe, sygnalizowane przez sensory skrajne, powinno być powtórzone. Robot powinien zainicjować zmianę położenia chwytaka za pomocą sygnału $W = 1$, przesuwając go o jedno pole sensora, odpowiednio: na lewo ($L = 1$), na prawo ($R = 1$), w górę ($O = 1$) lub w dół ($U = 1$).

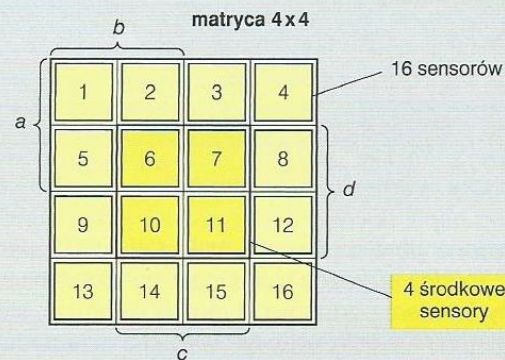
Sensory zostały tak zakodowane, że dotknięcie dwóch górnych poziomych rzędów wywołuje sygnał $a = 1$, dwóch rzędów pionowych z lewej strony – sygnał $b = 1$, dwóch rzędów pionowych środkowych – sygnał $c = 1$ oraz dwóch rzędów poziomych środkowych – sygnał $d = 1$ (rys. 3).

Zadania:

- zbudować tablicę zawierającą wszystkie możliwe kombinacje wartości sygnałów a , b , c , d i następnie przyporządkować im poszczególne sensory. Przykładowo, sensor 7 należy do obszarów a , c i d ($a=c=d=1$), natomiast nie należy do obszaru b ($b=0$),
- uzupełnić tablicę z p. a) wartościami sygnałów W , R , L , O , U , korygującymi położenie chwytaka. Przykładowo, w przypadku trafienia na sensor 1 chwytak należy uruchomić ($W = 1$) oraz przesunąć w lewo ($L = 1$) i do góry ($O = 1$), ponadto należy zablokować ruch w prawo ($R = 0$) i w dół ($U = 0$),



Rys. 2. Chwytnik robota z sensorami dotykowymi w matrycy 4x4



Rys. 3. Ułożenie pól sensorów dotykowych

- c) podać normalną postać koniunkcyjną funkcji określającej wartość sygnału W i następnie uprościć ją za pomocą **tablicy Karnaugh**¹,
d) podać normalne postacie alternatywne funkcji określających wartość sygnałów R , L , O i U i następnie uprościć je za pomocą tablicy Karnaugh. Przykładowo, przesuwanie w lewo ($L=1$) powinno nastąpić, jeżeli zainicjowany sensor należy do obszaru b (powtórzenie b) i jednocześnie nie należy do obszaru a (negacja a),
e) napisać program w języku listy instrukcji (IL) dla sterownika PLC sterującego procesem chwywania za pomocą sygnałów W , R , L , O i U .

Rozwiązanie:

Ad a) i b) – **Tab. 1**

Ad c) – **Tab. 2 i Tab. 3**

Ad d) – **Tab. 2 i Tab. 4**

Ad e) – **Tab. 2**

Tab. 1. Rozwiązanie zadania a) i b)

a	b	c	d	sensor nr.	W	R	L	O	U
0	0	0	0	16	1	1	0	0	1
0	0	0	1	12	1	1	0	0	0
0	0	1	0	15	1	0	0	0	1
0	0	1	1	11	0	0	0	0	0
0	1	0	0	13	1	0	1	0	1
0	1	0	1	9	1	0	1	0	0
0	1	1	0	14	1	0	0	0	1
0	1	1	1	10	0	0	0	0	0
1	0	0	0	4	1	1	0	1	0
1	0	0	1	8	1	1	0	0	0
1	0	1	0	3	1	0	0	1	0
1	0	1	1	7	0	0	0	0	0
1	1	0	0	1	1	0	1	1	0
1	1	0	1	5	1	0	1	0	0
1	1	1	0	2	1	0	0	1	0
1	1	1	1	6	0	0	0	0	0

Tab. 3. Tablica Karnaugh dla sygnału W

ab	cd			
	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	1	1	0	1
10	1	1	0	1

$$W = (a+b+\bar{c}+\bar{d})(a+\bar{b}+\bar{c}+\bar{d})(\bar{a}+\bar{b}+\bar{c}+\bar{d})(\bar{a}+\bar{b}+\bar{c}+\bar{d}) = \bar{c} + \bar{d}$$

$$W = \bar{c} \vee \bar{d}$$

$$W = \overline{\bar{c} + \bar{d}} = \overline{\bar{c} \bar{d}} ; \text{ z użyciem funkcji NAND}$$

Tab. 4. Tablica Karnaugh dla sygnału R

ab	cd			
	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	0	0	0	0
10	1	1	0	0

$$R = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd = \bar{b} \bar{c}$$

$$R = \bar{b} \wedge \bar{c}$$

$$R = \overline{\bar{\bar{b} \bar{c}}} = \overline{\bar{b} + \bar{c}} ; \text{ z użyciem funkcji NOR}$$

Tab. 2. Rozwiązanie zadania c), d) i e)

tablica funkcji	IL
$W = \bar{c} \vee \bar{d}$	ON c ON d = W
$\bar{W} = c \wedge d$	A c A d = \bar{W}
$R = \bar{b} \wedge \bar{c}$	AN b AN c = R
$L = b \wedge \bar{c}$	A b AN c = L
$O = a \wedge \bar{d}$	A a AN d = O
$U = \bar{a} \wedge \bar{d}$	AN a AN d = U
	xxx