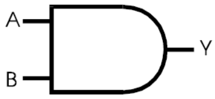
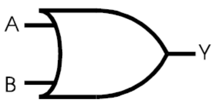
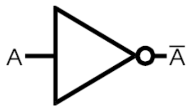
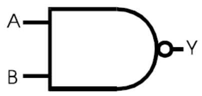
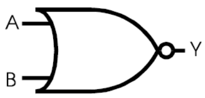
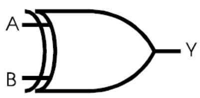
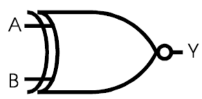
Bramki logiczne

Bramki logiczne - ang. *gates* (nazywane także *funktorami logicznymi*) są najprostszymi układami cyfrowymi realizującymi elementarne funkcje logiczne. Służą one do budowy układów logicznych o większej złożoności. Podstawowe bramki logiczne, ich nazwy, symbole graficzne, opis algebraiczny oraz tablice prawdy przedstawiono w tabeli poniżej.

Tablica 3.1 Podstawowe bramki logiczne

Funkcja logiczna	Symbol logiczny	Wyrażenie algebraiczne	Tabela prawdy		
			Wejście		Wyjście
			A	B	Y
AND		$A \cdot B = Y$	0 0 1 1	0 1 0 1	0 0 0 1
OR		$A + B = Y$	0 0 1 1	0 1 0 1	0 1 1 1
NOT		$\bar{A} = Y$	0 1		1 0
NAND		$\overline{A \cdot B} = Y$	0 0 1 1	0 1 0 1	1 1 1 0
NOR		$\overline{A + B} = Y$	0 0 1 1	0 1 0 1	1 0 0 0
XOR		$A \oplus B = Y$	0 0 1 1	0 1 0 1	0 1 1 0
XNOR		$\overline{A \oplus B} = Y$	0 0 1 1	0 1 0 1	1 0 0 1

Bramki **AND**, **OR**, **NOT** tworzą tzw. ***funkcjonalnie pełny*** zestaw elementów. Oznacza to, że można z nich zbudować dowolnie złożony układ logiczny.
Za pomocą wyłącznie bramek **NAND** albo wyłącznie bramek **NOR** można także zrealizować dowolnie złożoną funkcję, w tym również funkcje podstawowe **AND**, **OR**, **NOT**. Z tego powodu mówimy, że bramki **NAND** i **NOR** oddzielnie tworzą tzw. ***minimalny*** zestaw ***funkcjonalnie pełny***. W pierwszej chwili może się wydawać, że stosowanie funktorów **NAND** lub **NOR** do realizacji prostych funkcji **iloczynu**, **sumy** i **negacji** jest niepotrzebne i tylko komplikuje postawione zadanie. W praktyce okazuje się, że zalety stosowania jednoelementowego zbioru do realizacji dowolnej funkcji logicznej są bardzo duże. Przeglądając katalogi producentów układów cyfrowych można z łatwością zauważyć, że najszerszą ofertę stanowią bramki **NAND**, gdyż są one najchętniej stosowane przez użytkowników (można powiedzieć, że bramka **NAND** jest bramką uniwersalną).
Przykład:
Wykaż, że A·(A+B)=A.
Rozwiązanie:

A	B	A+B	A·(A+B)
1	1	1	1
0	1	1	0
1	0	1	1
0	0	0	0

Przykłady:
Wykaż, że:

Ćwiczenie 1:
$$a + bc = (a + b) \cdot (a + c)$$

Ćwiczenie 2:
$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

Ćwiczenie 3:
$$a + b = \overline{\overline{(a + b)}} = \overline{(\bar{a} \cdot \bar{b})}$$

Ćwiczenie 4:
$$\overline{\overline{(a + b)}} \cdot \overline{\overline{(\bar{a} + b)}} = \overline{\overline{\overline{\bar{a}\bar{b}} \cdot \overline{a\bar{b}}}}$$

Ćwiczenie 5:

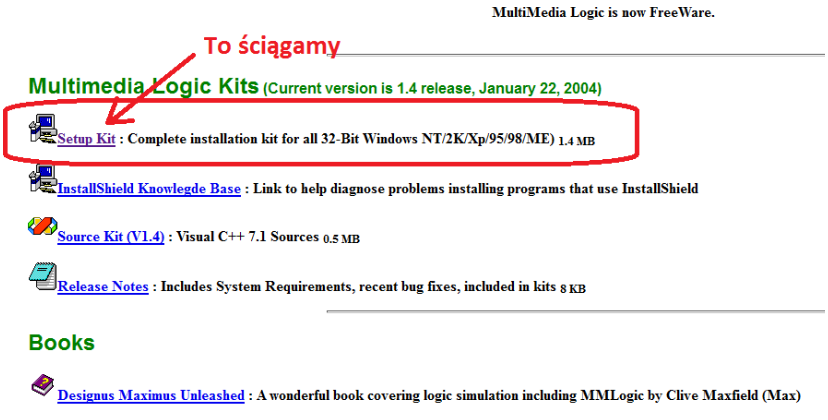
Mając do dyspozycji bramki AND i NOT zbuduj bramkę OR.

Ćwiczenie 6:
Mając do dyspozycji bramki OR i NOT zbuduj bramkę AND.

Ćwiczenie 7:
Mając do dyspozycji bramki OR, AND oraz NOT zbuduj bramkę XOR.

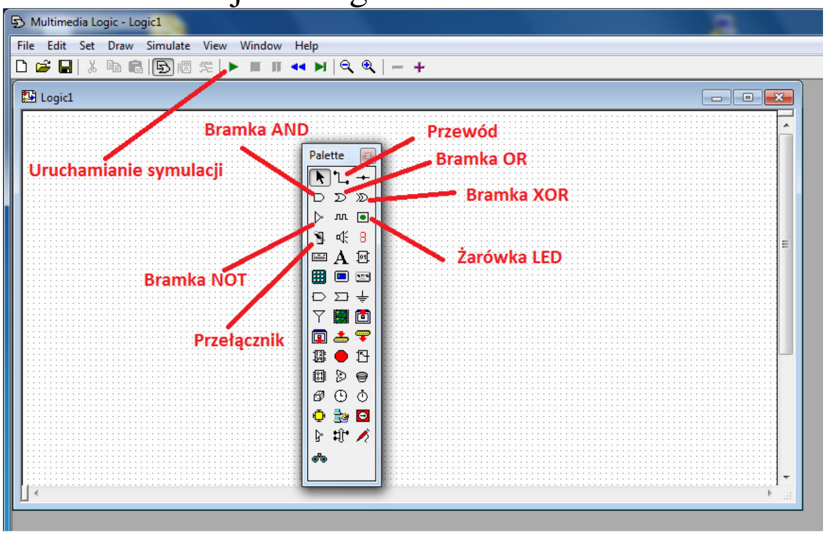
Program:

1. Ściągamy program ze strony: http://www.softronix.com/logic.html



2. Instalujemy program.

Krótka instrukcja obsługi:



Aby zbudować model należy przeciągnąć na obszar roboczy potrzebne elementy (bramki), połączyć je przewodami, umieścić włącznik, ewentualnie inne potrzebne elementy, a następnie uruchomić symulację układu.

Realizacja układu $A \cdot (A+B)=A$: