A blue text on a black background

Description automatically generated

**Architektura systemów komputerowych**

Laboratorium nr 1

**Wstęp teoretyczny**

Układy logiczne stanowią podstawę działania systemów cyfrowych, gdzie bramki takie jak AND, OR czy NOT wykonują proste operacje na sygnałach binarnych.

|  |  |  |
| --- | --- | --- |
| **p** | **q** | **p OR q** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| **p** | **q** | **p AND q** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Dzięki odpowiednim kombinacjom tych podstawowych bramek można zrealizować bardziej złożone funkcje. Podstawowe z nich:

|  |  |
| --- | --- |
| **p** | **NOT p** |
| 0 | 1 |
| 1 | 0 |

|  |  |  |
| --- | --- | --- |
| **p** | **q** | **p XOR q** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
| **p** | **q** | **p NAND q** |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Zadanie 1.\* (Analiza przykładu)**

**Mając do dyspozycji bramki AND oraz NOT** **zbuduj bramkę OR.**

1. **𝑝 + 𝑞 - prawo podwójnej negacji**

(mówi, że podwójne zanegowanie wartości logicznej daje wynik równy wartości początkowej)

**- prawo de Morgana**

(pozwala na przekształcanie wyrażeń logicznych, zamieniając AND na OR (i odwrotnie) za pomocą negacji)

**𝑝 + 𝑞 - definicja funkcji OR**

A diagram of a circuit

Description automatically generated

Rys. 1. Schemat logiczny bramki OR

**Opis układu:**

- Odwracamy sygnały p i q (0 zamienia się na 1, a 1 na 0).

- Łączymy te odwrócone sygnały bramką AND (wynik to 1, tylko gdy oba odwrócone sygnały są 1).

- Wynik z AND znowu odwracamy, co daje wynik p OR q.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **OR** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

Tab. 1. Tabela sprawdzająca realizację bramki OR przy użyciu bramek AND i NOT.

**Zadanie 2.**

**Mając do dyspozycji bramki OR oraz NOT zbuduj bramkę AND.**

a)

b)

A diagram of a block diagram

Description automatically generated

Rys. 2. Schemat logiczny bramki AND

**Opis układu:**

Najpierw odwracamy sygnały p i q bramkami NOT. Potem łączymy je bramką OR. Na końcu odwracamy wynik OR, uzyskując p AND q.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **AND** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Tab. 2. Tabela sprawdzająca realizację bramki AND przy użyciu bramek OR i NOT.

**Zadanie 3.**

**Mając do dyspozycji bramki XOR zbuduj bramkę NOT.**

1. 

Jedno wejście bramki XOR jest zawsze ustawione na wartość 1, a drugie wejście jest sterowane sygnałem p.

* Jeśli p = 0, to p ⊕ 1 = 1, tak jak = 1
* Jeśli p = 1, to p ⊕ 1 = 0, tak jak = 0

b)

**A line of arrows and dots

Description automatically generated with medium confidence**

Rys. 3. Schemat logiczny bramki NEGACJI za pomocą XOR.

**Opis układu:**

Sygnał p jest połączony z 1 przez bramkę XOR. Dzięki temu, jeśli p to 0, wynik to 1, a jeśli p to 1, wynik to 0. W ten sposób bramka XOR z 1 działa jak negacja p.

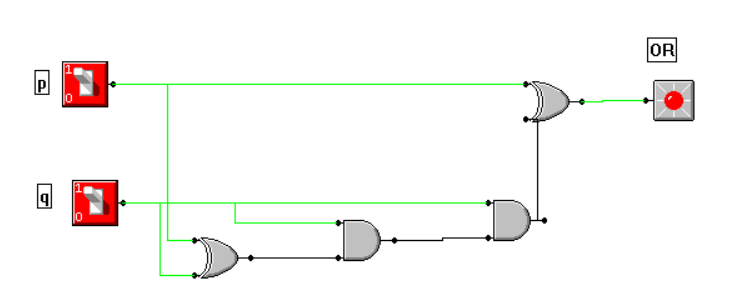
c)

|  |  |  |
| --- | --- | --- |
| **p** | **p ⊕ 1** |  |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

Tab. 3. Tabela sprawdzająca realizację bramki NEGACJI przy użyciu bramki XOR.

**Zadanie 4.**

**Mając do dyspozycji bramki XOR oraz AND zbuduj bramkę OR.**

1. i b) 

Rys. 4.1. Schemat logiczny bramki OR za pomocą XOR i AND.

Rysunek ten jest równoznaczny z wyrażeniem , które spełnia działanie bramki OR.

Da się jednak zbudować bardziej elegancki i uproszczony schemat równoznaczny z wyrażeniem, które jest tożsame z p + q (OR).

A diagram of a circuit

Description automatically generated

Rys. 4.2. Schemat logiczny bramki OR za pomocą XOR i AND.

**Opis układu:**

- Sygnały p i q najpierw przechodzą przez **bramkę XOR,** co wykrywa, czy sygnały są różne (daje 1, gdy jedno z wejść jest 1, a drugie 0).

- Sygnały **p i q są też łączone bramką AND,** która wykrywa, czy oba sygnały są 1.

- **Wyniki z XOR i AND są łączone bramką XOR,** co daje 1, gdy którykolwiek z sygnałów p lub q jest 1 – co jest równoważne operacji OR.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** | **(p ⊕ q) ⊕(p \* q)** | **OR** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

Tab. 4. Tabela sprawdzająca realizację bramki OR przy użyciu bramek XOR i AND.

**Zadanie 5.**

**Mając do dyspozycji bramki XOR oraz OR zbuduj bramkę AND.**

a)

b)

**A diagram of a circuit

Description automatically generated**

Rys. 5. Schemat logiczny bramki AND za pomocą XOR i OR.

**Opis układu:**

**-Bramka OR** sprawdza, czy p lub q jest 1.

**-Bramka XOR** sprawdza, czy p i q są różne.

-**Końcowa bramka XOR** łączy wyniki z OR i XOR – jeśli p i q są takie same (1 i 1), wynik to 1, co odpowiada działaniu bramki AND.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** | **(p ⊕ q) ⊕(p +q)** | **AND** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Tab. 5. Tabela sprawdzająca realizację bramki AND przy użyciu bramek XOR i OR.

**Zadanie 6.**

**Mając do dyspozycji bramki AND, OR oraz NOT zbuduj bramkę XOR.**

a)

b)

A diagram of a block diagram

Description automatically generated

Rys. 6. Schemat logiczny bramki XOR za pomocą AND, NOT i OR**.**

**Opis układu:**

-**Bramki NOT** – Odwracają wartości p i q, zamieniając 1 na 0 i 0 na 1.

**-Bramki AND** – Sprawdzają dwa przypadki:

p AND ¬q – czyli 1, gdy p jest 1, a q jest 0

q AND ¬p – czyli 1, gdy q jest 1, a p jest 0

-Bramka OR – Łączy wyniki z obu bramek AND. Daje 1 tylko wtedy, gdy jedno z wejść  
 (p lub q) jest 1, a drugie 0, co jest równoznaczne z działaniem bramki XOR.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **XOR** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

Tab. 6. Tabela sprawdzająca realizację bramki XOR przy użyciu bramek

AND, NOT i OR.

**Zadanie 7.**

**Zbuduj bramkę NAND korzystając z bramek AND oraz NOT.**

a)

b)

A diagram of a diagram

Description automatically generated

Rys. 7. Schemat logiczny bramki NAND za pomocą AND i NOT.

**Opis układu:**

-**Bramka AND** – Sprawdza, czy oba sygnały p i q są 1. Jeśli tak, wynik to 1; w przeciwnym razie 0.

-**Bramka NOT** – Odwraca wynik bramki AND. Oznacza to, że gdy p i q są oba 1, wynik końcowy to 0, a we wszystkich innych przypadkach – 1, dokładnie tak jak bramka NAND.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **NAND** |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

Tab. 7. Tabela sprawdzająca realizację bramki NAND przy użyciu bramek

AND I NOT.

**Zadanie 8.**

**Zbuduj bramkę NOR korzystając z bramek OR oraz NOT**

a)

b)

A diagram of a diagram

Description automatically generated

Rys. 8. Schemat logiczny bramki NOR za pomocą OR i **NOT.**

**Opis układu:**

-**Bramka OR** – Sprawdza, czy p lub q są równe 1. Jeśli tak, wynik to 1.

-**Bramka NOT** – Odwraca wynik bramki OR. Dzięki temu, jeśli p lub q są 1, wynik końcowy to 0, a tylko gdy oba są 0, wynik to 1.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **NOR** |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 |

Tab. 8. Tabela sprawdzająca realizację bramki NOR przy użyciu bramek

OR i NOT.

**Zadanie 9.**

**Mając do dyspozycji bramki NAND zbuduj bramkę NOT.**

a)

b)

A computer graphics of a red box and a green arrow

Description automatically generated

Rys. 9. Schemat logiczny bramki NOT za pomocą NAND.

**Opis układu:**

- **Bramka AND** – Sygnał p jest podłączony do obu wejść bramki AND, co sprawia, że wynik zależy tylko od wartości p.

- **Bramka NOT** – Wynik z bramki AND jest odwracany, co daje przeciwną wartość do p.

**Dlaczego to działa?**

Podając sygnał p dwukrotnie na bramkę AND, wynik zawsze odpowiada wartości p. Następnie odwracamy ten wynik za pomocą NOT, co tworzy funkcję negacji – wynik jest 1, gdy p jest 0, i 0, gdy p jest 1.

c)

|  |  |  |
| --- | --- | --- |
| **p** |  | **NOT** |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

Tab. 9. Tabela sprawdzająca realizację bramki NOT przy użyciu bramek

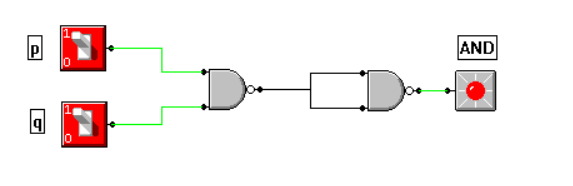
NAND.

**Zadanie 10.**

**Mając do dyspozycji bramki NAND zbuduj bramkę AND**

a)

b)



Rys.10. Schemat logiczny bramki AND za pomocą NAND.

**Opis układu:**

**- Bramka NAND** – Łączy sygnały p i q, dając wynik przeciwny do AND. Wynik będzie 0 tylko wtedy, gdy p i q są oba 1, w przeciwnym razie 1.

**- Podwójna negacja** – Wynik z pierwszej bramki NAND trafia na kolejną bramkę NAND, która również działa jako negacja. Dzięki temu podwójna negacja przywraca oryginalną logikę AND.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **AND** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Tab. 10. Tabela sprawdzająca realizację bramki AND przy użyciu bramek

NAND.

**Zadanie 11.**

**Mając do dyspozycji bramki NAND zbuduj bramkę OR.**

a)

b)

A computer generated diagram of a block diagram

Description automatically generated with medium confidence

Rys.11. Schemat logiczny bramki OR za pomocą NAND.

**Opis układu:**

**-Pierwsze bramki NAND** – Sygnał p jest podawany na bramkę NAND z dwoma wejściami p.  
To powoduje, że wynik to NOT p. To samo dzieje się dla q, więc druga bramka NAND daje NOT q.

-**Końcowa bramka NAND** – Wyniki NOT p i NOT q są połączone w trzeciej bramce NAND, co daje NOT (NOT p AND NOT q). Zgodnie z prawami logiki, jest to równoważne p OR q.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **OR** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

Tab. 11. Tabela sprawdzająca realizację bramki OR przy użyciu bramek

NAND.

**Zadanie 12.**

**Mając do dyspozycji bramki NOR zbuduj bramkę NOT.**

a) 

b)

A black and white drawing of a wire

Description automatically generated

Rys.12. Schemat logiczny bramki NOT za pomocą NOR.

**Opis układu:**

**- Bramka NOR –** Sygnał p jest podłączony do obu wejść bramki NOR.

- Bramka NOR zwraca 1 tylko wtedy, gdy oba wejścia wynoszą 0. Ponieważ oba wejścia są p, bramka NOR zwraca 1 tylko wtedy, gdy p jest 0. Zwraca także o gdy oba wejścia są 1.

c)

|  |  |  |
| --- | --- | --- |
| **p** |  | **NOT** |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

Tab. 12. Tabela sprawdzająca realizację bramki NOR przy użyciu bramek NOT

.

**Zadanie 13.**

**Mając do dyspozycji bramki NOR zbuduj bramkę OR**

a) 

b)

A diagram of a diagram

Description automatically generated

Rys.13. Schemat logiczny bramki NOR za pomocą OR.

**Opis układu:**

**-Pierwsza bramka NOR** – Sygnał p jest podany na pierwszą bramkę NOR, która ma oba swoje wejścia podłączone do p. Wynik tej bramki to NOT p. To samo dzieje się dla q – druga bramka NOR, mająca oba wejścia podłączone do q, daje wynik NOT q.

**-Końcowa bramka NOR** – Wyniki NOT p i NOT q z pierwszych dwóch bramek NOR są połączone w trzeciej bramce NOR. Ta bramka NOR daje wynik NOT (NOT p AND NOT q), co jest równoważne p OR q.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **OR** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

Tab. 13. Tabela sprawdzająca realizację bramki NOR przy użyciu bramek

OR.

**Zadanie 14.**

**Mając do dyspozycji bramki NOR zbuduj bramkę AND**

a) 

b)

A diagram of a connection

Description automatically generated with medium confidence

Rys.14. Schemat logiczny bramki NOR za pomocą AND.

**Opis układu:**

Dzięki pierwszym dwóm bramkom NOR uzyskujemy negacje p i q. Trzecia bramka NOR przekształca te negacje w sposób, który daje wynik taki sam, jak bramka AND – wynik jest 1 tylko wtedy, gdy zarówno p, jak i q wynoszą 1.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **AND** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |

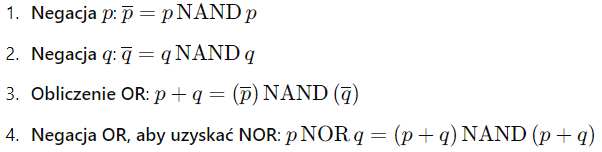
Tab. 14. Tabela sprawdzająca realizację bramki NOR przy użyciu bramek

AND.

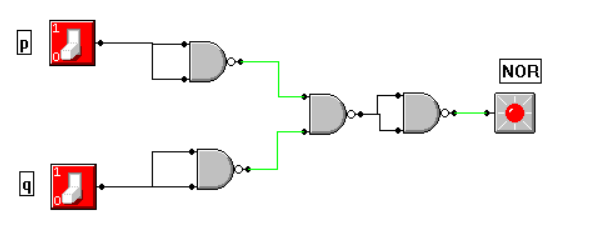
**Zadanie 15.**

**Mając do dyspozycji bramki NAND zbuduj bramkę NOR**

a)  = A



b)



Rys.15. Schemat logiczny bramki NAND za pomocą NOR.

**Opis układu:**

**-Pierwsze dwie bramki NAND** – p jest podany na pierwszą bramkę NAND z dwoma wejściami p, co daje NOT p. Podobnie, q jest podany na drugą bramkę NAND z dwoma wejściami q, co daje NOT q.

-**Trzecia bramka NAND (OR)** – Wyniki NOT p i NOT q trafiają na trzecią bramkę NAND, co daje p OR q.

-**Czwarta bramka NAND (NOR)** – Wynik p OR q jest podany na czwartą bramkę NAND z dwoma wejściami, co daje NOT (p OR q).

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **NOR** |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 |

Tab. 15. Tabela sprawdzająca realizację bramki NAND przy użyciu bramek

NOR.

**Zadanie 16.**

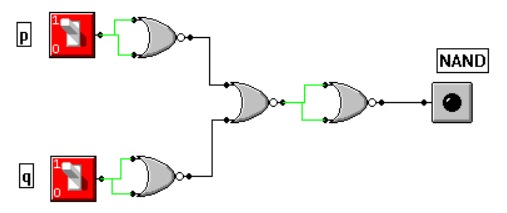
**Mając do dyspozycji bramki NOR zbuduj bramkę NAND**

a)  = B

**A white background with black text

Description automatically generated**

b)



Rys.16. Schemat logiczny bramki NAND za pomocą NOR.

**Opis układu:**

-**Pierwsze bramki NOR** – negacja p i q

Sygnał p jest podany na bramkę NOR, która ma oba swoje wejścia podłączone do p. Wynik tej bramki to NOT p – działa jak negacja p. Sygnał q jest podany na drugą bramkę NOR z dwoma wejściami q, co daje NOT q – działa jak negacja q.

-**Trzecia bramka NOR** – operacja OR

Wyniki NOT p i NOT q z pierwszych dwóch bramek trafiają na trzecią bramkę NOR. Ta bramka NOR realizuje operację NOT (NOT p OR NOT q), co jest równoważne p OR q. W efekcie, ta bramka daje wynik OR dla p i q.

**-Czwarta bramka NOR** – końcowa negacja OR (NOR)

Wynik p OR q z trzeciej bramki NOR jest podany na czwartą bramkę NOR z wejściami połączonymi z tym samym sygnałem. Ta końcowa bramka NOR odwraca wynik, tworząc NOT (p OR q), co jest równoważne bramce NOR.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **NAND** |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

Tab. 16. Tabela sprawdzająca realizację bramki NAND przy użyciu bramek

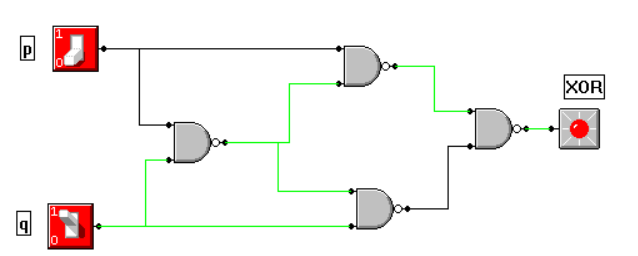
NOR.

**Zadanie 17.**

**Mając do dyspozycji bramki NAND zbuduj bramkę XOR.**

a)  = C

b)



Rys.17. Schemat logiczny bramki XOR za pomocą NAND.

**Opis układu:**

-**Negacja p i q** – Pierwsze bramki NAND tworzą NOT p i NOT q przez podanie p i q na bramki NAND z ich samymi sobą.

-**Częściowe wyniki AND** – Kolejne bramki NAND generują:

- p AND NOT q

- q AND NOT p

-**Końcowa bramka NAND (XOR)** – Ostatnia bramka NAND łączy wyniki p AND NOT q i q AND NOT p, co daje p XOR q.

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **XOR** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

Tab. 17. Tabela sprawdzająca realizację bramki XOR przy użyciu bramek

NAND.

**Zadanie 18.**

**Mając do dyspozycji bramki NOR zbuduj bramkę XOR**

a) = D

b)

A diagram of a block diagram

Description automatically generated

Rys.18. Schemat logiczny bramki NOR za pomocą XOR.

**Opis układu:**

-**Pierwsza bramka NOR**:

Pierwsza bramka NOR ma jedno wejście podłączone do p i jedno do q.

Wynik tej bramki NOR to NOT (p OR q).

**-Druga bramka NOR:**

Druga bramka NOR ma jedno wejście podłączone do wyniku z pierwszej bramki, a drugie wejście podłączone bezpośrednio do p.

Wynik tej bramki NOR to NOT (p OR NOT (p OR q)).

**-Trzecia bramka NOR:**

Trzecia bramka NOR ma jedno wejście podłączone do wyniku z pierwszej bramki, a drugie wejście do q.

Wynik tej bramki NOR to NOT (q OR NOT (p OR q)).

**-Czwarta bramka NOR:**

Wyniki z drugiej i trzeciej bramki NOR są podłączone do czwartej bramki NOR.

Wynik tej bramki NOR to NOT ((NOT (p OR NOT (p OR q))) OR (NOT (q OR NOT (p OR q))))

c)

|  |  |  |  |
| --- | --- | --- | --- |
| **p** | **q** |  | **XOR** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

Tab. 18. Tabela sprawdzająca realizację bramki NOR przy użyciu bramek

XOR.

**Wnioski**

* Bramki NAND i NOR są uniwersalne – można zbudować wszystkie inne (AND, OR, NOT, XOR) bramki tylko za ich pomocą.
* Podwójna negacja pozwala przywrócić wartość początkową, co jest przydatne przy przekształcaniu układów logicznych.
* Prawo de Morgana pozwala zamienić AND na OR (i odwrotnie) przy użyciu negacji.
* Dzięki stosowaniu bramek NAND i NOR można uprościć skomplikowane układy, redukując liczbę potrzebnych bramek.