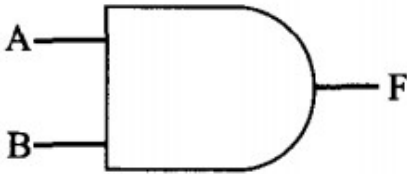

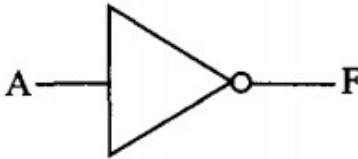
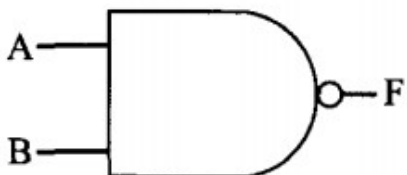
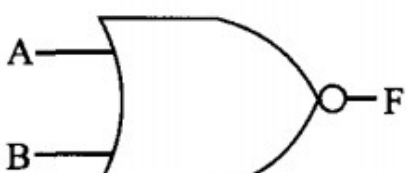


Podstawowe pojęcia:

Układ kombinacyjny – zbiór wzajemnie połączonych bramek, którego stan wyjść w dowolnej chwili jest wyłącznie funkcją stanu wejść w tej samej chwili. Po ustaleniu stanu na wejściu prawie natychmiast pojawia się sygnał na wyjściu, przy czym występują tylko opóźnienia bramkowe. Układ kombinacyjny może być zdefiniowany przez podanie:

1. tablicy prawdy
2. symbolu graficznego – schematu połączeń
3. Funkcji algebraicznej Boolea

Poniższe bramki pozwalają zaprojektować każdy układ kombinacyjny:

| Nazwa | Symbol graficzny | Funkcja algebraiczna | Tablica prawdy | | | | | | | | | | |
|-------|---|---------------------------------------|---|----|---|-----|---|-----|---|-----|---|-----|---|
| AND |  | $F = A \cdot B$ lub $F = AB$ | <table><tr><th>AB</th><th>F</th></tr><tr><td>0 0</td><td>0</td></tr><tr><td>0 1</td><td>0</td></tr><tr><td>1 0</td><td>0</td></tr><tr><td>1 1</td><td>1</td></tr></table> | AB | F | 0 0 | 0 | 0 1 | 0 | 1 0 | 0 | 1 1 | 1 |
| AB | F | | | | | | | | | | | | |
| 0 0 | 0 | | | | | | | | | | | | |
| 0 1 | 0 | | | | | | | | | | | | |
| 1 0 | 0 | | | | | | | | | | | | |
| 1 1 | 1 | | | | | | | | | | | | |
| OR |  | $F = A + B$ | <table><tr><th>AB</th><th>F</th></tr><tr><td>0 0</td><td>0</td></tr><tr><td>0 1</td><td>1</td></tr><tr><td>1 0</td><td>1</td></tr><tr><td>1 1</td><td>1</td></tr></table> | AB | F | 0 0 | 0 | 0 1 | 1 | 1 0 | 1 | 1 1 | 1 |
| AB | F | | | | | | | | | | | | |
| 0 0 | 0 | | | | | | | | | | | | |
| 0 1 | 1 | | | | | | | | | | | | |
| 1 0 | 1 | | | | | | | | | | | | |
| 1 1 | 1 | | | | | | | | | | | | |
| NOT |  | $F = \overline{A}$ lub $F = A'$ | <table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table> | A | F | 0 | 1 | 1 | 0 | | | | |
| A | F | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | |
| NAND |  | $F = (\overline{AB})$ | <table><tr><th>AB</th><th>F</th></tr><tr><td>0 0</td><td>1</td></tr><tr><td>0 1</td><td>1</td></tr><tr><td>1 0</td><td>1</td></tr><tr><td>1 1</td><td>0</td></tr></table> | AB | F | 0 0 | 1 | 0 1 | 1 | 1 0 | 1 | 1 1 | 0 |
| AB | F | | | | | | | | | | | | |
| 0 0 | 1 | | | | | | | | | | | | |
| 0 1 | 1 | | | | | | | | | | | | |
| 1 0 | 1 | | | | | | | | | | | | |
| 1 1 | 0 | | | | | | | | | | | | |
| NOR |  | $F = \overline{(A + B)}$ | <table><tr><th>AB</th><th>F</th></tr><tr><td>0 0</td><td>1</td></tr><tr><td>0 1</td><td>0</td></tr><tr><td>1 0</td><td>0</td></tr><tr><td>1 1</td><td>0</td></tr></table> | AB | F | 0 0 | 1 | 0 1 | 0 | 1 0 | 0 | 1 1 | 0 |
| AB | F | | | | | | | | | | | | |
| 0 0 | 1 | | | | | | | | | | | | |
| 0 1 | 0 | | | | | | | | | | | | |
| 1 0 | 0 | | | | | | | | | | | | |
| 1 1 | 0 | | | | | | | | | | | | |

RYSUNEK A.1. Podstawowe bramki logiczne

Dowolna funkcja Boole'a może być zrealizowana za pomocą wyłącznie bramek pochodzących z takiego zbioru. Następujące zbiory są funkcjonalnie pełne:

- AND, OR, NOT
- AND, NOT
- OR, NOT
- NAND
- NOR

Prawa algebry Boole'a

Do podstawowych praw algebry Boole'a należą następujące twierdzenia

1. Prawo przemienności sumy: $A + B = B + A$
2. Prawo przemienności iloczynu: $AB = BA$
3. Prawo łączności sumy: $(A + B) + C = A + (B + C)$
4. Prawo łączności iloczynu: $(AB)C = A(BC)$
5. Prawo rozdzielności iloczynu względem sumy: $A(B + C) = AB + AC$
6. Prawo rozdzielności sumy względem iloczynu: $A + (BC) = (A + B)(A + C)$
7. $A + A = A$
8. $AA = A$
9. $AB + A\sim B = A$
10. $(A + B)(A + \sim B) = A$
11. $A + AB = A$
12. $A(A + B) = A$
13. $0 + A = A$
14. $0 \cdot A = 0$
15. $1 + A = 1$
16. $1 \cdot A = A$
17. $\sim A + A = 1$
18. $\sim A \cdot A = 0$
19. $A + \sim AB = A + B$
20. $A(\sim A + B) = AB$
21. Prawo podwójnej negacji: $\sim\sim A = A$
22. Prawo De Morgana: $\sim(A + B) = \sim A \cdot \sim B$ – negacja sumy jest równa iloczynowi negacji
23. Prawo De Morgana: $\sim(A \cdot B) = \sim A + \sim B$ – negacja iloczynu jest równa sumie negacji

Realizacja funkcji Boole'a

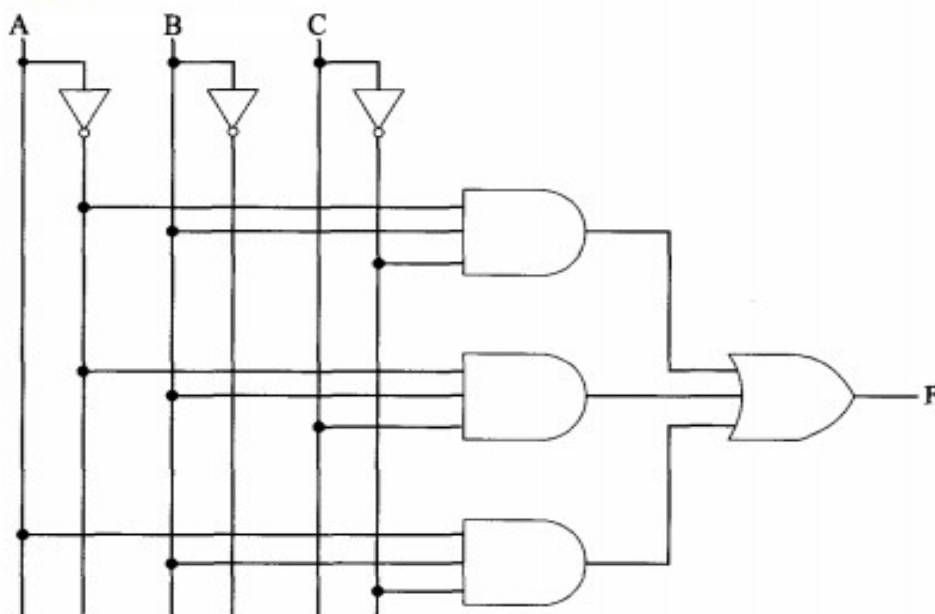
Dowolna funkcja Boole'a może być zrealizowana w postaci elektronicznej jako sieć bramek. Dla danej funkcji istnieje pewna liczba rozwiązań alternatywnych. Rozważmy funkcję Boole'a reprezentowaną przez tablicę prawdy w tabeli A.3. Możemy wyrazić tę funkcję po prostu przez wyszczególnienie kombinacji wartości A, B i C, które powodują, że F jest równe 1:

$$F = \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC \quad (\text{A-1})$$

TABELA A.3. Funkcja Boole'a trzech zmiennych

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Istnieją trzy kombinacje wartości wejściowych, które powodują, że F jest równe 1, i jeśli wystąpi którakolwiek z tych kombinacji, to wynik jest równy 1. Ta forma wyrażenia z oczywistych powodów nazywa się **sumą iloczynów**. Na rysunku A.4 jest pokazana bezpośrednia realizacja za pomocą bramek AND, OR i NOT.



RYSUNEK A.4. Realizacja tabeli A.3 za pomocą sumy iloczynów

Na ogół jest możliwe wyprowadzenie prostszego wyrażenia Boole'a na podstawie tablicy prawdy. Przy prostszym wyrażeniu Boole'a do realizacji funkcji będzie potrzebnych mniej bramek. W celu uproszczenia mogą być stosowane następujące metody:

- upraszczanie algebraiczne
- mapy Karnaugh
- tablice Quine'a-McKluskeya.

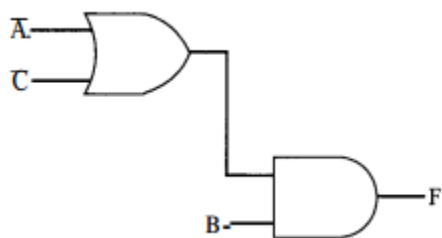
Upraszczenie algebraiczne polega na stosowaniu tożsamości – praw algebry Boole’a zaprezentowanych wcześniej. Upraszczając je możemy otrzymać następujące postacie:

$$f = \tilde{A}B + B\tilde{C}$$

lub

$$f = B(\tilde{A} + \tilde{C})$$

Realizacja układowa to:

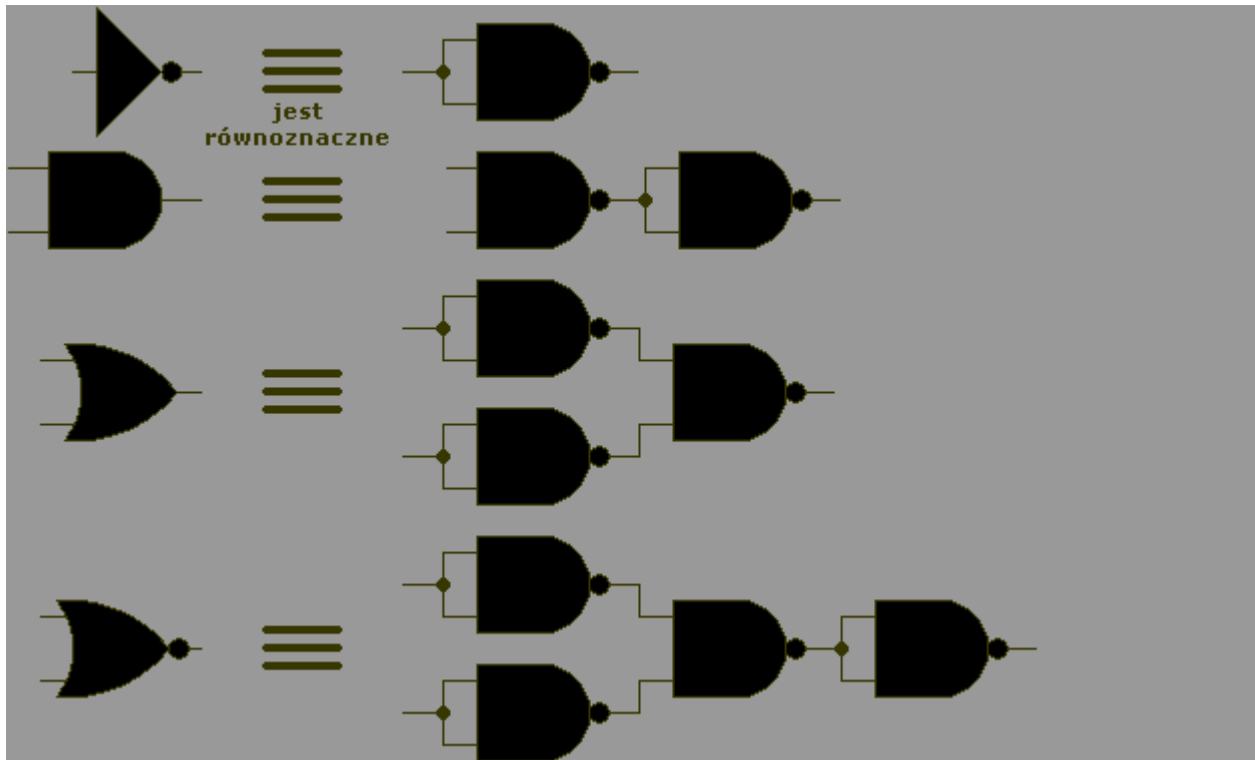


Mapa Karnaugh jest wygodną metodą upraszczania funkcji Boole'a o niewielkiej liczbie zmiennych (4+6). Mapa jest tablicą złożoną z 2^n kwadratów reprezentujących możliwe kombinacje wartości n zmiennych binarnych.

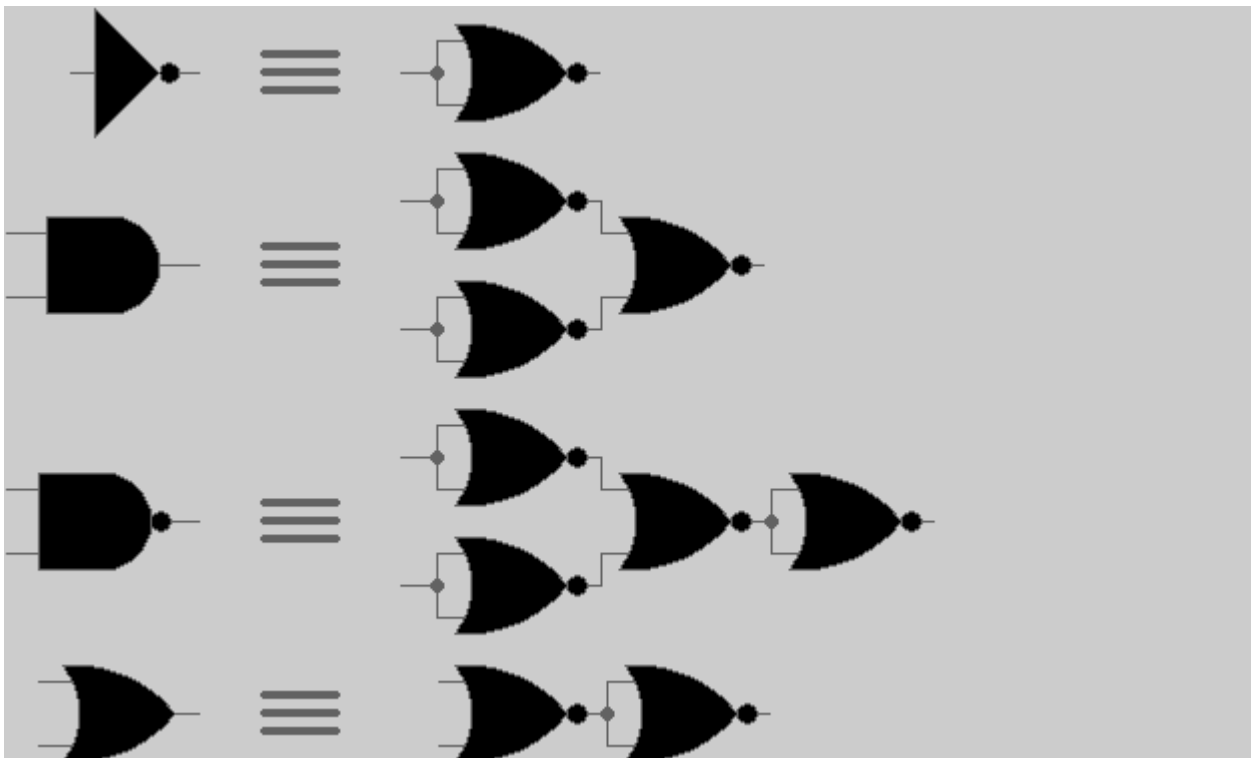
Projektowanie układu cyfrowego realizującego zadaną funkcję:

(Źródło: <http://technical.elektronowolt.net/?page=realizacja>)

Zamiennik: NAND

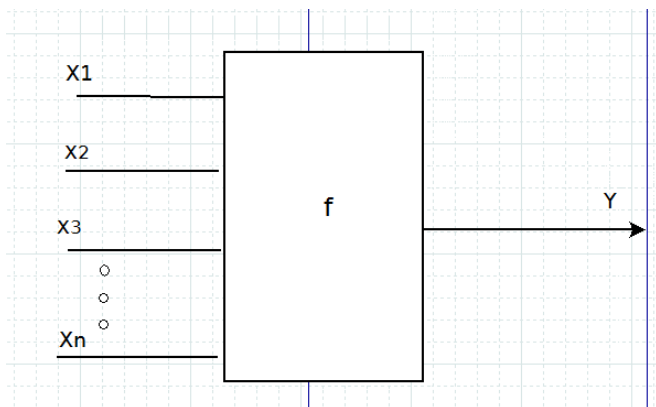


Zamiennik: NOR



Tablica prawdy

Tablica prawdy dla funkcji opisanej wyrażeniem lub słownie jest tablicą pokazującą, jakie wartości funkcji odpowiadają wejściowym sygnałom.



Przykład: Przedstawić tablicę prawdy dla funkcji:

$$f(a, b, c) = a\tilde{b} + c$$

Wypisujemy po lewej stronie sygnały wejściowe i wartości, które mogą przyjmować:

| Wartosc dwójkowo | a | b | c | f |
|------------------|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 |

Zadania z bramek logicznych i projektowania układów:

Upraszczanie wyrażeń:

Zadanie 1.

Mając dane wyrażenia opisujące połączenia sygnałów, uprość je i narysuj realizację układową tych wyrażeń:

$$(a + \tilde{b})[abc + \tilde{b}(a + c)] + abc\tilde{c}(a + \tilde{a}b) =$$

$$ab + \tilde{a}b\tilde{c} + bc =$$

$$(ab + c + d)(\tilde{c} + d)(\tilde{c} + d + e) =$$

Zadanie 2.

Mając podane wyrażenia narysuj ich realizację układową tak aby wykorzystać do tego tylko bramki NAND lub NOR (można skorzystać z uproszczeń jeśli będzie to możliwe):

$$f = (\tilde{a} + b) + (\tilde{c} + a) + (\tilde{a} + \tilde{c})$$

$$f = \tilde{a}b + \tilde{a}b\tilde{c} + ac$$

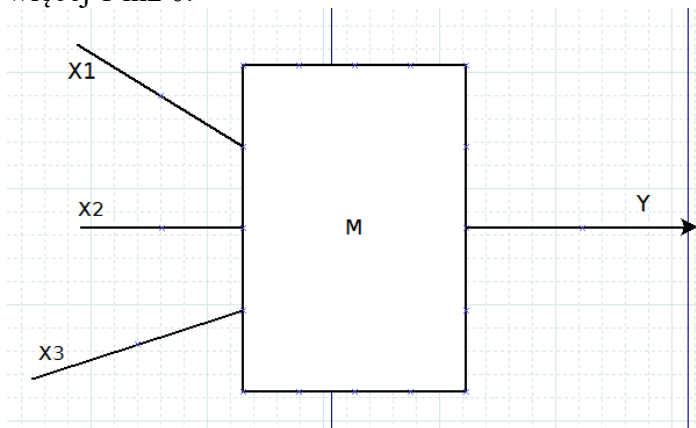
$$f = ab + \tilde{a}c + bc$$

Projektowanie:

Zadanie 1:

Znaleźć funkcję f realizowaną przez układ z rysunku 1.

Element M na rysunku jest elementem większościowym – daje na wyjściu 1 gdy na we x_1, x_2, x_3 jest więcej 1 niż 0.



$$y = \begin{cases} 0 & \text{gdy} & k_0 > k_1 \\ 1 & \text{gdy} & k_1 > k_0 \end{cases}$$

Zaprojektować minimalny układ NAND realizujący tą funkcję.

Zadanie 2.

Zaprojektować układ o trzech wejściach c, b, a z elementów NOT, AND, OR, który ma sygnał 1 na wyjściu jeżeli na wejściu pojawi się liczba dwójkowa cba nieparzysta lub podzielna przez 3.

Sygnał a odpowiada najmłodszemu bitowi w tej liczbie. W każdej kombinacji wejściowej przynajmniej jeden z sygnałów c, b, a jest różny od 0.