

# 19 Bramki logiczne i układy cyfrowe

## ZAGADNIENIA

- Funkcja logiczna
- Charakterystyka układów logicznych
- Rodzaje bramek logicznych
- Rodzaje układów kombinacyjnych i sekwencyjnych

**Funkcja logiczna** (tzw. boolowska) to matematyczny model opisu cyfrowego układu kombinacyjnego. Jest wyrażeniem złożonym ze zmiennych dwójkowych i określonych operacji logicznych. Dla zadanych wartości zmiennych funkcja boolowska może przyjmować wartości 0 lub 1.

Zmiennymi dwójkowymi i operacjami logicznymi zajmuje się logika binarna. Stosuje się ją do matematycznego opisu przetwarzania informacji dwójkowej. Jest szczególnie przydatna do analizy i projektowania systemów cyfrowych. Przykładowo, zachowanie układów logicznych wykonujących dwójkowe operacje arytmetyczne najwygodniej opisać za pomocą zmiennych dwójkowych i operacji logicznych, czyli funkcji boolowskich. Zmienne dwójkowe mogą przyjmować dwie różne wartości (0 lub 1). Oznacza się je literami A, B, C, x, y, z itd. Istnieją trzy podstawowe operacje logiczne: NIE (NOT), I (AND) i LUB (OR). Wspomniane operacje zapisuje się za pomocą symboli znanych z arytmetyki:

- negacja:  $\bar{A}$  (lub  $A'$ );
- iloczyn logiczny:  $A \cdot B$  (lub  $AB$ );
- suma logiczna:  $A + B$ .

## Bramki logiczne

Jest to element realizujący pewną funkcję logiczną. Argumenty funkcji i sama funkcja mogą przyjmować jedną z dwóch wartości: 0 lub 1.

### Rodzaje bramek logicznych:

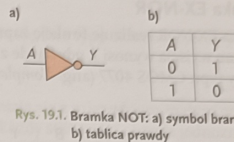
- NOT – negacja (NIE);
- AND – iloczyn logiczny lub koniunkcja (I);
- NAND (Not-AND) – negacja iloczynu logicznego (NIE I);
- OR – suma logiczna lub alternatywa (LUB);
- NOR (Not-OR) – negacja sumy logicznej (NIE LUB);
- EX-OR (EXclusive OR) – suma modulo 2 lub różnica symetryczna (ALBO);
- EX-NOR (EXclusive Not OR) – zaprzeczenie różnicy symetrycznej (NIE ALBO).

Bramki NOT, AND i OR są podstawowymi elementami logicznymi używanymi do budowy układów logicznych. Za pomocą bramek NAND i NOR oraz pary bramek AND i NOT lub OR i NOT można zbudować układ realizujący dowolną funkcję logiczną. Takie układy nazywa się funkcjonalnie pełnymi lub zupełnymi.

Poniżej przedstawiono charakterystykę bramek. Do opisu ich działania stosuje się tzw. **tablice prawdy** (tablice wartości funkcji). Zawierają one zbiór wszystkich sygnałów wejściowych i odpowiadające im sygnały wyjściowe.

## Bramka NOT

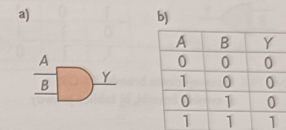
Realizuje ona operację negacji logicznej:  $Y = \bar{A}$ . W technice cyfrowej negację oznacza się jako NOT A albo  $\bar{A}$ . Bramkę NOT często nazywa się negatorem lub inwerterem, ponieważ odwraca na wyjściu dostarczony poziom sygnału logicznego (z 0 na 1 i z 1 na 0). Symbol kółka na wyjściu (lub wejściu) oznacza negację sygnału. Bramki NOT zawiera np. układ scalony typu TTL 7404.



Rys. 19.1. Bramka NOT: a) symbol bramki, b) tablica prawdy

## Bramka AND

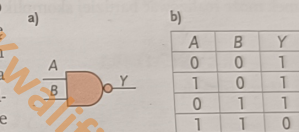
Realizuje ona operację iloczynową  $Y = A \cdot B$  (lub krócej  $Y = AB$ ). W technice cyfrowej iloczyn oznacza się jako A AND B lub AB. Bramka może mieć wiele wejść. Bramki AND zawiera np. układ scalony TTL 7408.



Rys. 19.2. Dwuwejściowa bramka AND: a) symbol bramki, b) tablica prawdy

## Bramka NAND

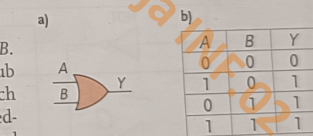
Stanowi w pewnym sensie połączenie bramek AND i NOT. Realizuje funkcję  $Y = \overline{A \cdot B}$ . Zero pojawia się na wyjściu bramki tylko wtedy, gdy na obu wejściach znajduje się jedynka. W pozostałych przypadkach na wyjściu zawsze jest stan 1. Widać więc, że stanowi odwrotność bramki AND. Wystarczy porównać tablice prawdy obu bramek. Ta bramka może mieć również wiele wejść. Bramki NAND zawiera np. układ scalony TTL 7400.



Rys. 19.3. Dwuwejściowa bramka NAND: a) symbol bramki, b) tablica prawdy

## Bramka OR

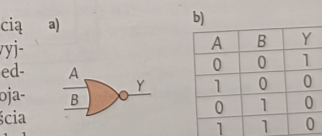
Realizuje ona operację sumy logicznej  $Y = A + B$ . W technice cyfrowej sumę oznacza się jako A OR B lub  $A + B$ . Ta bramka jest układem o co najmniej dwóch wejściach. Stan wyjścia wynosi 1, gdy przynajmniej jedno z wejść ma stan 1. Bramki OR zawiera np. układ scalony TTL 7432.



Rys. 19.4. Dwuwejściowa bramka OR: a) symbol bramki, b) tablica prawdy

## Bramka NOR

Funkcjonalnie ta bramka jest dokładną odwrotnością bramki OR. Realizuje funkcję  $Y = \overline{A + B}$ . Zero na wyjściu pojawia się zawsze wtedy, gdy przynajmniej na jednym z wejść znajduje się jedynka logiczna. Stan 1 pojawia się na wyjściu tylko wtedy, gdy wszystkie wejścia są ustawione w stan 0. Bramki NOR zawiera np. układ scalony TTL 7402.



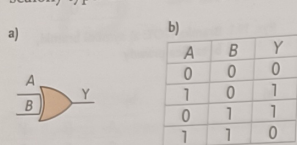
Rys. 19.5. Dwuwejściowa bramka NOR: a) symbol bramki, b) tablica prawdy

## Bramka EX-OR

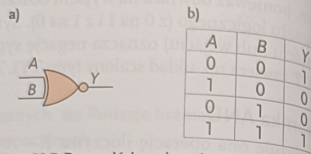
Realizuje ona funkcję różnicy symetrycznej  $Y = A \oplus B = \overline{A}B + A\overline{B}$ . Stan na jej wyjściu wynosi 1, gdy stany na wejściach są różne (0 i 1 lub 1 i 0). Bramki EX-OR zawiera np. układ scalony TTL 7486.

## Bramka EX-NOR

Bramka EX-NOR realizuje funkcję zaprzeczenia różnicy symetrycznej  $Y = A \odot B = \overline{A \oplus B} = \overline{A \oplus B}$ . Stan wyjścia wynosi 1, gdy każde z wejść ma stan 1 lub 0. Bramki EX-NOR zawiera np. układ scalony typu CMOS 4077 (ang. *Complementary Metal-Oxide Semiconductor*).



Rys. 19.6. Dwuwejściowa bramka EX-OR:  
a) symbol bramki, b) tablica prawdy



Rys. 19.7. Dwuwejściowa bramka EX-NOR:  
a) symbol bramki, b) tablica prawdy

## Realizacja funkcji logicznych na bramkach

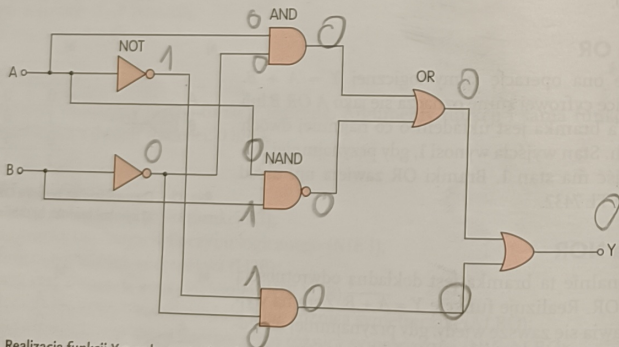
Wiemy już, że każda bramka wykonuje jakąś małą, prostą funkcję logiczną. Grupa kilku bramek może realizować bardziej skomplikowane funkcje.

### PRZYKŁAD 19.1

Mamy funkcję:

$$Y = \overline{A} \overline{B} + \overline{A} B + A \overline{B}$$

Przykładową realizację tej funkcji przedstawiono na rys. 19.8.



Rys. 19.8. Realizacja funkcji Y z wykorzystaniem bramek logicznych

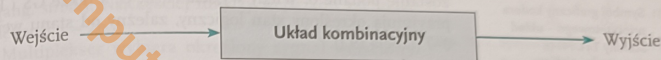
W praktyce do realizacji funkcji logicznych wykorzystuje się układy logiczne, zwane cyfrowymi. **Układy cyfrowe** są rodzajem układów elektronicznych, w których sygnały napięciowe przyjmują tylko określoną liczbę stanów z przypisanymi im wartościami liczbowymi. Informacja wewnątrz urządzeń cyfrowych jest zakodowana za pomocą uporządkowanego ciągu cyfr. Zwykle liczba stanów wszelkich sygnałów wynosi 2 i przyjmują one wartości umowne 0 i 1. Układy cyfrowe są zbudowane z bramek logicznych, realizujących operacje znane z algebry Boole'a.

## Układy cyfrowe kombinacyjne i sekwencyjne

Ze względu na sposób przetwarzania informacji rozróżnia się dwa typy układów cyfrowych:

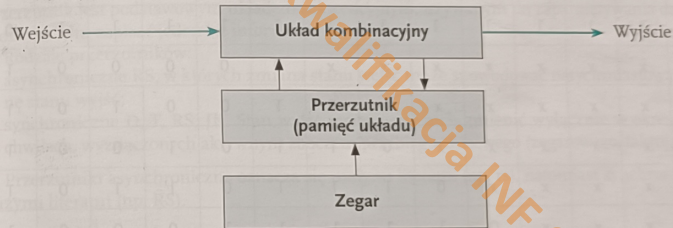
- układy kombinacyjne;
- układy sekwencyjne.

W układach kombinacyjnych sygnały wyjściowe zmieniają się z mierzalnym małym opóźnieniem w stosunku do zmian sygnałów wejściowych, czyli każdy stan wejść określa jednoznacznie stan wyjść (rys. 19.9).



Rys. 19.9. Schemat blokowy układu kombinacyjnego

Natomiast w układach sekwencyjnych stan wejść nie opisuje w sposób jednoznaczny stanu wyjść. Stan wyjść zależy bowiem od poprzednich stanów wejść, zapamiętanych w układzie. Układy sekwencyjne nazywa się też układami kombinacyjnymi z pamięcią (rys. 19.10). Układy sekwencyjne dzieli się na synchroniczne i asynchroniczne.



Rys. 19.10. Schemat blokowy układu sekwencyjnego

## Kodery

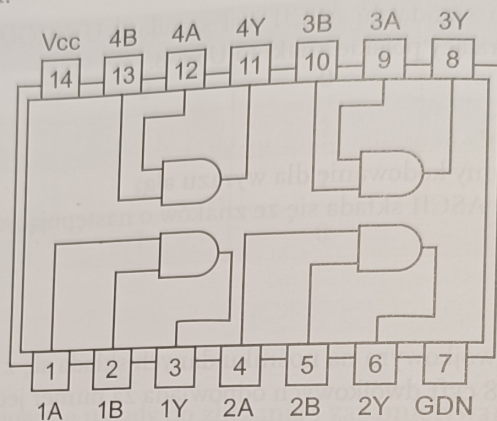
Niektóre układy cyfrowe zajmują się kodowaniem, czyli przypisywaniem różnym informacjom pewnych symboli. Kod jest zestawem symboli przypisanych danej informacji. Opisany wcześniej system dwójkowy jest naturalnym kodem dwójkowym.

Jak już wspomniano, układy cyfrowe współpracują ze sobą przy użyciu systemu dwójkowego. Ostatecznie jednak urządzenia komunikują się z człowiekiem, co wymusza często przedstawianie danych w systemie dziesiętnym. Ponieważ naturalny kod dwójkowy jest technicznie trudny do konwersji na system dziesiętny, stosuje się kody dwójkowo-dziesiętne, zwane również kodami BCD (ang. *Binary Coded Decimal*). Każdej cyfrze dziesiętnej kody przypisują na stałe określoną liczbę dwójkową. Oznacza to, że każdą cyfrę koduje się oddzielnie.

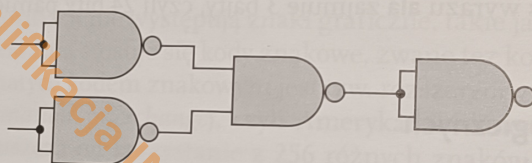
Koder jest cyfrowym układem kombinacyjnym mającym  $n$  wejść oraz  $k$  wyjść. Służy do przetworzenia kodu 1 z  $n$  (np. 1 z 10) w określony dwójkowy kod wyjściowy. Ponieważ istnieje możliwość jednoczesnej aktywacji więcej niż jednego wejścia, koder uznaje zawsze informację z najstarszego wejścia. Jednocześnie ignoruje informacje na pozostałych wejściach – stąd nazywa koder priorytetowy. Wartość na wyjściu koda odpowiada wybranemu numerowi wejścia, przedstawionemu w kodzie dwójkowym.

## SPRAWDŹ SWOJĄ UMIEJĘTNOŚCI

1. Sprawdź, jakie bramki logiczne wykorzystano w budowie układu przedstawionego poniższym rysunku.



2. Przeanalizuj układ bramek pokazanych na rysunku. Zamiennikiem której bramki może być ten układ?



3. Skorzystaj z aplikacji CEDAR Logic Simulator lub innego symulatora i narysuj układ bramek widoczny na rysunku, a następnie określ stan na wyjściu układu.

