dn23.03.2020r.

Nauczyciel: Bogusława Kocałek

Klasa: I TI Technikum Kształtowania Środowiska - Technik Informatyk

SYSTEMY OPERACYJNE

Temat: Analizowanie działania układów zbudowanych z bramek logicznych.

Wykonałam zrzuty z Podręcznika: T.Marciniuk WSiP „Urządzenia techniki komputerowej”, dostępna w Internecie. Proszę zapoznać się z treścią tego podręcznika. Ten temat omówiliśmy w klasie, mamy również sporządzoną notatkę. Proszę przygotować się z tego tematu, bo jest bardzo ważny i niektórzy muszą poprawić sprawdzian.

3 Układy cyfrowe

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 więc zakodowana za pomocą uporządkowanego ciągu cyfr. Zwykle liczba stanów wszelkich sygnałów wynosi dwa i przyjmują one wartości umowne 0 i 1. Operacje realizowane przez układy cyfrowe można opisać zgodnie z algebrą Boole'a, czyli językiem logiki matematycznej – stąd nazywa się je także **układami logicznymi**.

Układy cyfrowe są zbudowane z bramek logicznych realizujących operacje znane z algebry Boole'a: negację (NOT), iloczyn logiczny (AND), negacja iloczynu logicznego (NAND), sumę logiczną (OR), negację sumy logicznej (NOR), różnicę symetryczną (EX-OR) itp. Ze względu na stopień skomplikowania współczesnych układów cyfrowych wykonuje się je w postaci układów scalonych.

Zalety i wady układów cyfrowych

Do zalet układów scalonych zalicza się:

- bezstratne kodowanie i przesyłanie informacji – co w układach analogowych operujących na nieskończonej liczbie stanów sygnałów było niemożliwe do uzyskania;
- uproszczone zapisywanie i przechowywanie informacji cyfrowej;
- mała wrażliwość na zakłócenia elektryczne;
- możliwość tworzenia układów programowalnych, których działanie określa program komputerowy.

Wadą jest ich skomplikowanie zarówno na poziomie elektrycznym, jak i logicznym, oraz fakt, że mimo większej odporności na zakłócenia wykrywanie możliwych przekłamań stanów logicznych wymaga wprowadzenia dodatkowych zabezpieczeń.

Ze względu na sposób przetwarzania informacji rozróżnia się dwa typy układów cyfrowych: **układy kombinacyjne i układy sekwencyjne**. W układach kombinacyjnych sygnały wyjściowe zmieniają się w chwili zmian sygnałów wejściowych, czyli każdy stan wejść określa jednoznacznie stan wyjść. Natomiast w układach sekwencyjnych stan wejść nie opisuje w sposób jednoznaczny stanu wyjść, gdyż zależy on od poprzednich stanów wejść zapamiętanych w układzie.

3.1. Funkcje logiczne i ich realizacja. Algebra Boole'a

Funkcja logiczna (tzw. boolowska) jest matematycznym modelem opisu cyfrowego układu kombinacyjnego. Jest to wyrażenie złożone ze zmiennych dwójkowych oraz 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. Na przykład, układy logiczne wykonujące dwójkowe operacje arytmetyczne są układami, których zachowanie najwygodniej opisać za pomocą zmiennych dwójkowych i operacji logicznych, czyli funkcji boolowskich. Zmienne dwójkowe mogą przyjmować dwie różne wartości 0 lub 1 i są oznaczane literami A, B, C, x, y, z itd. Istnieją trzy podstawowe operacje logiczne: NIE (NOT), I (AND) oraz LUB (OR). Operacje te zapisuje się, używając znaków znanych z arytmetyki:

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

Prawa algebry Boole'a

Do podstawowych praw logicznych (twierdzeń) algebry Boole'a należą:

Tw. 1: Prawo przemienności

$A + B = B + A$ (3.1)

$AB = BA$ (3.2)

Tw. 2: Prawo łączności

$(A + B) + C = A + (B + C)$ (3.3)

$(AB)C = A(BC)$ (3.4)

Tw. 3: Prawo rozdzielności

$A(B + C) = AB + AC$ (3.5)

$A + (BC) = (A + B)(A + C)$ (3.6)

Tw. 4:

$A + A = A$ (3.7)

$AA = A$ (3.8)

Tw. 5:

$AB + A\overline{B} = A$ (3.9)

$(A + B)(A + \overline{B}) = A$ (3.10)

Tw. 6: Prawo absorpcji

$A + AB = A$ (3.11)

$A(A + B) = A$ (3.12)

Tw. 7:

$0 + A = A$ (3.13)

$0 \cdot A = 0$ (3.14)

Tw. 8:

$1 + A = 1$ (3.15)

$1 \cdot A = A$ (3.16)

Tw. 9:

$\overline{A} + A = 1$ (3.17)

$\overline{A}A = 0$ (3.18)

Tw. 10:

$A + \overline{A}B = A + B$ (3.19)

$A(\overline{A} + B) = AB$ (3.20)

Tw. 11: Prawa De Morgana

$\overline{A + B} = \overline{A}\,\overline{B}$ (3.21)

$\overline{AB} = \overline{A} + \overline{B}$ (3.22)

Poniżej przedstawiono przykład minimalizacji funkcji logicznej za pomocą praw algebry Boole'a.

Przykład

$$Y = A\overline{B} + \overline{AB} + \overline{A}\,\overline{B} = (A\overline{B} + \overline{A}\,\overline{B}) + \overline{AB} = \overline{B}(A + \overline{A}) + \overline{AB} = \overline{B} + \overline{AB} = \overline{B} + \overline{A} + \overline{B} = \overline{A} + \overline{B} = \overline{AB}$$

Ćwiczenie 3.1.

Korzystając z praw algebry Boole'a zminimalizuj poniższe funkcje:

a) $Y = A\overline{B} + \overline{AB} + \overline{A}B + B$

b) $Y = \overline{AB} + \overline{A}\,\overline{B} + B$

c) $Y = \overline{\overline{A}\,\overline{B}} + A + B$

3.2. Bramki logiczne

Bramka logiczna – element realizujący pewną prostą funkcję logiczną, której argumenty 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. Układy takie nazywa się **funkcjonalnie pełnymi** lub **zupełnymi**.
Poniżej przedstawiono charakterystykę bramek. Do opisu działania bramek stosuje się tzw. **tablice prawdy** (tablice wartości funkcji), które zawierają zbiór wszystkich sygnałów wejściowych i odpowiadające im sygnały wyjściowe.

3.2.1. Bramka NOT

Bramka NOT realizuje operację negacji logicznej: $Y = \overline{A}$. W technice cyfrowej negację oznaczamy jako *NOT A* albo $\overline{A}$. Bramka NOT często jest nazywana **negatorem** lub **inwerterem**, ponieważ odwraca dostarczony poziom sygnału logicznego na wyjściu (z 0 na 1 oraz z 1 na 0). Symbol kółka na wyjściu (lub wejściu) oznacza negację sygnału. Bramki NOT zawiera np. układ scalony typu TTL 7404.

a)

b)

A	Y
0	1
1	0

Rys. 3.1. Dwuwejściowa bramka NOT: a) symbol bramki, b) tablica prawdy

3.2.2. Bramka AND

Bramka AND realizuje operację iloczynu $Y = A \cdot B$ (lub krócej $Y = AB$). W technice cyfrowej iloczyn oznaczamy jako *A AND B* lub *AB*. Bramka ta może mieć wiele wejść. Bramki AND zawiera np. układ scalony TTL 7408.

a)

b)

A	B	Y
0	0	0
1	0	0
0	1	0
1	1	1

Rys. 3.2. Dwuwejściowa bramka AND: a) symbol bramki, b) tablica prawdy

3.2.3. Bramka NAND

Bramka ta stanowi jakby połączenie bramek AND i NOT. Realizuje funkcję $Y = \overline{AB}$. Zero pojawia się na wyjściu bramki tylko wtedy, gdy na obu wejściach jest jedynka. W pozostałych przypadkach na wyjściu zawsze jest stan 1. Widać więc, że jest ona odwrotnością bramki AND. Wystarczy porównać tablice prawdy obu bramek. Bramka ta może mieć również wiele wejść. Bramki NAND zawiera np. układ scalony TTL 7400.

a)

b)

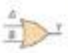
A	B	Y
0	0	1
1	0	1
0	1	1
1	1	0

Rys. 3.3. Dwuwejściowa bramka NAND: a) symbol bramki, b) tablica prawdy

3.2.4. Bramka OR

Bramka OR realizuje operację sumy logicznej $Y = A + B$. W technice cyfrowej oznaczamy jako *A OR B* lub $A + B$. Jest układem o dwóch lub większej liczbie wejść. Stan wyjścia wynosi 1, gdy przynajmniej jedno z wejść ma stan 1. Bramki OR zawiera np. układ scalony TTL 7432.

a)



b)

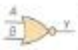
A	B	Y
0	0	0
1	0	1
0	1	1
1	1	1

Rys. 3.4. Dwuwejściowa bramka OR: a) symbol bramki, b) tablica prawdy

3.2.5. Bramka NOR

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

a)



b)

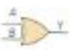
A	B	Y
0	0	1
1	0	0
0	1	0
1	1	0

Rys. 3.5. Dwuwejściowa bramka NOR: a) symbol bramki, b) tablica prawdy

3.2.6. Bramka EX-OR

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

a)



b)

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

Rys. 3.6. Dwuwejściowa bramka EX-OR: a) symbol bramki, b) tablica prawdy

3.2.7. Bramka EX-NOR

Bramka EX-NOR realizuje funkcję zaprzeczenia różnicy symetrycznej $Y = A \odot B = \overline{A \oplus B} = AB + \overline{A}\,\overline{B}$. Jest to bramka, której stan wyjścia wynosi 1, gdy każde z wejść ma stan 1 lub każde – stan 0. Bramki EX-NOR zawiera np. układ scalony typu CMOS 4077.

a)

b)

A	B	Y
0	0	1
1	0	0
0	1	0
1	1	1

Rys. 3.7. Dwuwejściowa bramka EX-NOR: a) symbol bramki, b) tablica prawdy

Pozdrawiam 1TI

Bogusława Kocałek