J.Nawrocki, M. Antczak, G. Palik, A.Widelska

Ćwiczenie nr 3

Sprzęt

Wprowadzenie

1. Podstawowe funkcje logiczne:

a) negacja (NOT):

$f(a) = \overline{a}$

a	0	1
nie a	1	0

b) suma (alternatywa, OR):

f(a,b) = a v b

a	0	0	1	1
b	0	1	0	1
avb	0	1	1	1

c) iloczyn (koniunkcja, AND):

f(a,b) = a ^ b = ab

a	0	1	0	1
b	0	0	1	1
a^b	0	0	0	1

d) implikacja (wynikania):

a => b

a	0	1	0	1
b	0	0	1	1
a=>b	1	0	1	1

e) równoważność:

a <=> b

a	0	1	0	1
b	0	0	1	1
a<=>b	1	0	0	1

f) EX-OR (różnica symetryczna, suma modulo 2):

$f(a,b) = a \oplus b = a\overline{b} \vee \overline{a}b$

a	0	0	1	1
b	0	1	0	1
a+b	0	1	1	0

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

2. Wprowadzenie do algebry Boole'a:

TWIERDZENIA:

a) o podwójnej negacji:

$\overline{\overline{a}} = a$

b) o dopełnieniu:

$a \vee \overline{a} = 1$

$a \wedge \overline{a} = 0$

c) o alternatywie:

a v 0 = a
a v 1 = 1

d) o koniunkcji:

a ^ 0 = 0
a ^ 1 = a

e) o przemienności:

a v b = b v a
a ^ b = b ^ a

f) o łączności:

a v (b v c) = (a v b) v c
a ^ (b ^ c) = (a ^ b) ^ c

g) o rozdzielności:

a v (b ^ c) = (a v b) ^ (a v c)
a ^ (b v c) = (a ^ b) v (a ^ c)

h) o idempotentności:

a v a = a
a ^ a = a

i) o absorpcji:

a v (a ^ b) = a
a ^ (a v b) = a

a	0	1	0	1
b	0	0	1	1
a(avb)	0	1	0	1

a(a v b) = a

j) De Morgana:

$\overline{a \vee b} = \overline{a} \wedge \overline{b}$

$\overline{a \wedge b} = \overline{a} \vee \overline{b}$

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

Przykład zastosowania:

$L= \overline{w}\ v\ \overline{z}\ v\ (x\ v\ wz)(y\ v\ z) = \overline{w}\ v\ \overline{z}\ v\ xy\ v\ xz\ v\ wzy\ vwzz = \overline{w}\ v\ \overline{z}\ v\ xy\ v\ xz\ vwzy\ v\ wz =$

$= \overline{w}\ v\ \overline{z}\ v\ xy\ v\ xz\ v\ wz\overbrace{(y\ v\ 1)}^{1} = \overline{w}\ v\ \overline{z}\ v\ xy\ v\ xz\ v\ wz = \overline{w}\ v\ \overline{z}\ v\ xy\ v\ xz\ v\ z = 1$

$\{a\ v\ (b \wedge c) = (a\ v\ b)(a\ v\ c)\}$

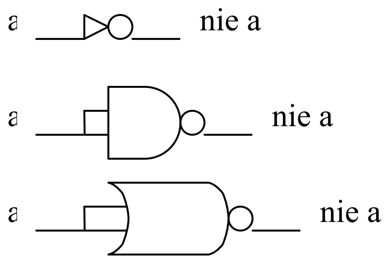
$\{a\ v\ (b \wedge \overline{a}) = (a\ v\ b)(a\ v\ \overline{a}) = a\ v\ b\}$

$\{\overline{w}\ v\ (zw) = \overline{w}\ v\ z\}$

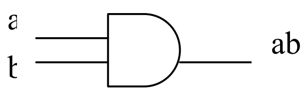
3. Podstawowe układy i funktory logiczne:

1) Funktory:

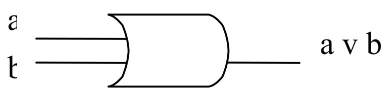
a) negacja NOT:



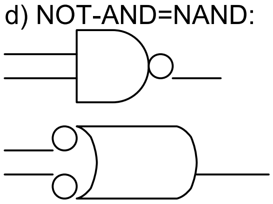
b) iloczyn AND:



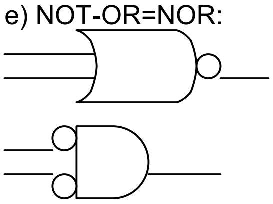
c) suma OR:



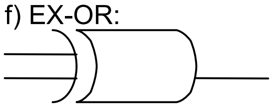
d) NOT-AND=NAND:



e) NOT-OR=NOR:



f) EX-OR:



2) Układy:

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

a) kombinacyjne – każda kombinacja sygnału wejściowego określa jednoznacznie kombinację sygnałów wyjściowych,
b) sekwencyjne – kombinacja sygnału wejściowego jest nazywana stanem wejść układu, natomiast kombinacja sygnałów wyjściowych jest nazywana stanem wyjść układu.

4. Postać kanoniczna funkcji - twierdzenie o rozkładzie:
Dowolną funkcję przełączającą można rozłożyć na dwa składniki.

$$f(x_1,x_2,...,x_n) = x_1 f(1,x_2,...,x_n) \text{ v } \overline{x}_1 f(0,x_2,...,x_n)$$

5. Tablice (tabele, siatki):
a) prawdy – stanowi pierwszy zapis rozwiązywanego problemu konstrukcyjnego, jest daną wejściową dla tablicy Karnaugh,
b) Karnaugh – stanowi daną wejściową dla określania postaci funkcji, jej wiersze i kolumny określają zmienne, a ich przecięcia – wyniki operacji na tych zmiennych.

Zasady zakreślania grup:
- liczba pól elementarnych łączonych za sobą musi być potęgą liczby 2,
- łączone pola muszą być polami sąsiadującymi ze sobą, tzn. oddzielone od siebie linią prostą lub poprzeczną lub krawędzią tablicy,
- połączone pola muszą mieć kształt symetryczny względem swych osi (prostokąty lub kwadraty).

6. Przerzutniki:
Przerzutniki należą do grupy układów sekwencyjnych. Ogólny schemat przerzutnika:

```
                                 ____________
wejścia informacyjne ----->|              Q|---→ wyjścia
                     ----->|               |
                           |               |
                           |              _|
wejście zegarowe    ----->|CLK           Q|---→
                           |_R_S___________|
                             ^  ^
                             |  |
                           wejścia programujące (przygotowujące)
```

Przerzutniki są to układy, które na wyjściu mogą przyjmować chwilowo lub na stałe tylko 2 stany (przerzutniki cyfrowe - stan 0 lub 1). Ze względu na sposób działania przerzutniki dzielimy na:
- bistabilne (posiadają 2 stany stabilne)
- monostabilne (posiadają1 stan stabilny)
- astabilne (nie posiadają stanu stabilnego)

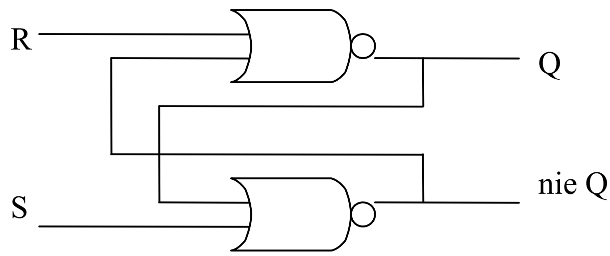
Przerzutniki cyfrowe posiadają następujące wejścia:
- informacyjne – 0 lub 1 przypisywana na Q
- programujące – asynchroniczne, pozwalają ustawić wyjście Q w stan 1 lub 0
- zegarowe – służą do synchronizacji pracy przerzutnika

Istnieją dwa sposoby wyzwalania przerzutnika:
- stanem (0 lub 1)
- zboczem (narastającym lub opadającym)

Przerzutniki cyfrowe dzielimy na:
- asynchroniczne – nie posiadają wejścia zegarowego
- synchroniczne – posiadają wejście zegarowe

Przerzutnik typu RS (asynchroniczny):

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.



R	0	0	1	1
S	0	1	0	1
Q	1/0	1/1	1	z
nie Q	0/1	0/0	1	z

z – zabroniony

Przerzutnik typu D i T (synchroniczny):

```
      _______
-----|D,T   Q|-----
     |       |
---o|>CLK   _|
     |_____ Q|-----
```

Typ D:

D	Q_t	Q_{t+1}
0	0	0
0	1	0
1	0	1
1	1	1

D	Q_{t+1}
0	0
1	1

Typ T:

T	Q_t	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

T	Q_{t+1}
0	Q_t
1	nie Q_t

Przerzutnik typu JK (synchroniczny):

```
      _______
-----|J     Q|-----
     |       |
---o|>CLK   _|
     |      Q|-----
-----|K______|
```

J	K	Q_t	Q_{t+1}

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

0	0	0	0
1	0	1	1
0	1	0	0
1	1	1	0
1	0	0	1
0	0	1	1
1	1	0	1
0	1	1	0

J	K	Q_{t+1}
0	0	Q_t
0	1	0
1	0	1
1	1	nie Q_t

7. Rejestry:

Rejestrem nazywamy układ zbudowany z przerzutników służący do przechowywania informacji. Liczba bitów informacji, jaka może być przechowywana w rejestrze jest nazywana długością rejestru i odpowiada zawsze liczbie przerzutników, z których jest zbudowany rejestr.

```
                        zapis równoległy
                          |   |   |   |
                         _v__v__v__v_
zapis szeregowy --->|__|__|__|__|---> odczyt szeregowy
                          |   |   |   |
                          V   V   V   V
                        odczyt równoległy
```

W zależności od sposobu wprowadzania i wyprowadzania informacji do i z rejestru dzielimy je na:
- szeregowo – szeregowe SISO
- szeregowo – równoległe SIPO
- równoległo – szeregowe PISO
- równoległo – równoległe PIPO

Rejestry szeregowe czasami nazywamy przesuwnymi, jednokierunkowymi lub rewersyjnymi (możliwości przesuwania w obie strony).

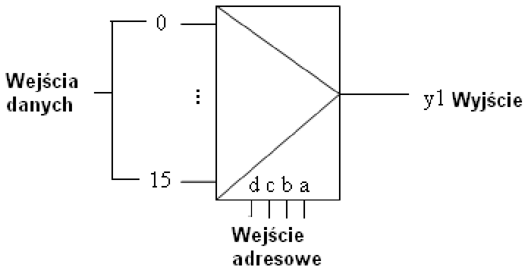
Podstawowym zastosowaniem rejestru jest przechowywanie informacji. W tym celu stosuje się przede wszystkim rejestry równoległe. Rejestr przesuwny, rewersyjny, może również służyć jako prosty układ mnożący lub dzielący przez 2 (2^n). Przesunięcie o jedną pozycję w lewo odpowiada mnożeniu przez 2, natomiast w prawo – dzieleniu przez 2.

8. Multipleksery / demultipleksery:

Multiplekser, zwany selektorem danych, służy do wyboru jednego z kilku sygnałów wejściowych i przekazania go na wyjście układu. Charakteryzuje się wieloma wejściami danych, dokładnie jednym wyjściem oraz wejściami adresowymi.
Demultiplekser umożliwia przesyłanie do jednego z wyjść układu sygnału doprowadzonego do jego wejścia. Charakteryzuje się dokładnie jednym wejściem danych, wieloma wyjściami oraz wejściami adresowymi.
Dla obu układów liczba wejść adresowych musi być na tyle wystarczająca, aby możliwe było zaadresowanie wszystkich wejść / wyjść układu.

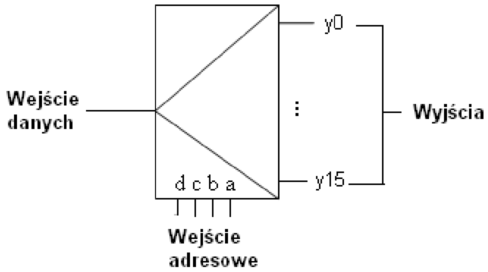
(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

Multiplekser



Ideą multipleksera jest przesłanie na wyjście (każdy multiplekser posiada tylko jedno wyjście) danej wartości z wejścia danych jednoznacznie identyfikowanego przez zapisany binarnie adres ustawiany na wejściu adresowym.
Rozmiar wejścia adresowego jest równy liczbie bitów potrzebnych w celu jednoznacznego odwzorowanie adresu każdego wejścia danych (np.: dla 16 wejść danych <0...15> - adres musi się składać z 4 bitów "abcd" = 0011 = 3).
Adres konkretnego wejścia danych jest ustawiany na wejściu adresowym w kolejności od najmniej znaczącego bitu.
Przykładowo:
Załóżmy, że mamy multiplekser z 16 wejściami (0..15) oraz wejście adresowe ma rozmiar 4 bitów. Chcemy przesłać na wyjście zawartość wejścia danych o adresie równym 3. Wtedy naszą trójkę zapisujemy binarnie na 4 bitach: 3 = 0011 (abcd)
Postać wejścia adresowego: (dcba) 1100

Demultiplekser



Ideą demultipleksera jest przesłanie na konkretnie zaadresowane wyjście (np.: y3), identyfikowane jednoznacznie przez binarnie zapisany adres ustawiany na wejściu adresowym, danej wartości z wejścia danych (każdy demultiplekser posiada tylko jedno wejście danych).
Rozmiar wejścia adresowego oraz sposób ustawiania konretnego adresu jest identyczny jak w przypadku multipleksera.
Przykładowo:
Załóżmy, że mamy demultiplekser z 16 wyjściami (y0..y15), wejście adresowe ma rozmiar 4 bitów. Chcemy przesłać wartość z wejścia danych na wyjście y5. Wtedy adres dziesiętny 5 zapisujemy binarnie na 4 bitach: 5 = 0101 (abcd)
Postać wejścia adresowego: (dcba) 1010

Wyjaśnienia

1. Tworzenie tablicy Karnaugh.
   - pierwszą połowę zmiennych przypisujemy do wierszy,
   - drugą połowę zmiennych przypisujemy do kolumn,
   - liczba zmiennych przypisanych do wierszy i kolumn powinna różnić się maksymalnie o 1,
   - indeksy tablicy (wiersze i kolumny) numerowane są przy pomocy binarnego kodu Graya.

   Binarny kod Graya:
   - stworzenie odbicia lustrzanego do istniejących (początkowych) słów kodowych (przepisanie w odwrotnej kolejności),
   - dopisanie do początkowych słów kodowych bitu o wartości **0**, a do dopisanych słów kodowych bitu o wartości **1**.

   Przykłady:

kod 1-bitowy	odbicie lustrzane	dopisanie **0** i **1**
0	0	**0**0
1	1	**0**1
	1	**1**1
	0	**1**0

kod 2-bitowy	odbicie lustrzane	dopisanie **0** i **1**
00	00	**0**00
01	01	**0**01
11	11	**0**11
10	10	**0**10
	10	**1**10
	11	**1**11
	01	**1**01
	00	**1**00

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

Przykładowa tablica Karnaugh dla 5 zmiennych:

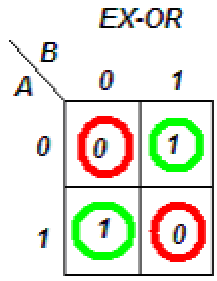
abc\de	00	01	11	10
000	0	0	1	0
001	1	0	0	0
011	1	0	0	0
010	0	1	0	0
110	0	0	1	1
111	0	0	1	1
101	1	1	1	1
100	0	1	1	0

2. Wyznaczanie funkcji na podstawie tablicy Karnaugh.
EX-OR (różnica symetryczna, suma modulo 2):

$$f(a,b) = a \oplus b = a\bar{b} \vee \bar{a}b$$

a	0	0	1	1
b	0	1	0	1
a+b	0	1	1	0

Na podstawie powyższej tablicy prawdy tworzymy odpowiadającą jej tablicę Karnaugh, której postać znajduje się poniżej.



Spróbujmy w tym momencie wyznaczyć funkcję na podstawie powyższego schematu. Zakreślamy odpowiednio kółkami maksymalną ilość zer lub jedynek. Najpierw przeprowadzimy rozumowanie dla jedynek, a następnie dla zer.

Dla jedynek:

Założyliśmy, że jeżeli wartość wejściowa np.: dla A (będąca w naszym przypadku etykietą kolumny) reprezentuje 1, to tworząc funkcję wyjściową przepisujemy A. W przeciwnym razie przepisujemy $\bar{A}$. **W przypadku jedynek naszą funkcje tworzymy poprzez sumę wszystkich iloczynów zmiennych, dla których wyjście w tablicy Karnaugh jest równe 1.** Mamy dwie jedynki w zielonych kółeczkach, a więc mamy w funkcji sumę dwóch iloczynów. W takim razie nasza funkcja wyjściowa wygląda następująco:

$$f_{a,b(1)} = A\bar{B} + \bar{A}B$$

Dla zer:

Założyliśmy, że jeżeli wartość wejściowa np.: dla A (będąca w naszym przypadku etykietą kolumny) reprezentuje 0 to tworząc funkcję wyjściową przepisujemy A. W przeciwnym razie przepisujemy $\bar{A}$. **W przypadku zer naszą funkcje tworzymy poprzez iloczyn wszystkich sum zmiennych, dla których wyjście w tablicy Karnaugh jest równe 0.** Mamy dwa zera w czerwonych kółeczkach, a więc mamy w funkcji iloczyn dwóch sum. W takim razie nasza funkcja wyjściowa wygląda następująco:

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

$$f_{a,b(0)} = (\bar{B} + \bar{A})*(A + B) = \bar{B}A + \underset{\bar{0}}{\bar{B}B} + \underset{\bar{0}}{\bar{A}A} + \bar{A}B = \bar{B}A + \bar{A}B$$

Wniosek!!!
Wyznaczone funkcje za pomocą zer i jedynek muszą być sobie równe, a więc $\mathbf{f_{a,b(1)} = f_{a,b(0)}}$

Wniosek!!!
Powyższe zależności dotyczące sposobu wyznaczania funkcji biorące pod uwagę zera lub jedynki wynikają z **praw de Morgana**, które zostały zdefiniowane powyżej.

3. **Dyskryminator** jest to układ, który ma na wejściu liczbę 4-bitową (4 wejścia) i jedno wyjście. Wartość na wyjściu jest równa 1, jeżeli na wejściu są liczby z zakresu od 9-12. W przeciwnym razie na wyjściu dyskryminatora jest liczba 0.

4. **„Dwójka licząca”** to układ, w którym na wyjściu, przy kolejnych taktach zegara, pojawiają się naprzemiennie zero oraz jeden. Przykładowe kolejne wartości na wyjściu Q = {0, 1, 0, 1, 0, 1, ...}.

Zadania

Zad. 1. Udowodnij, że:

$(x\bar{y} \vee y)\,(x \vee \bar{y}) = x$

Zad. 2. Wylicz następujące wyrażenia:

a) $a \vee (\bar{a}b)$

b)**(*)** $ab \vee \bar{a}b\bar{c} \vee bc$

c)**(*)** $(a \vee \bar{b} \vee ab)(a \vee \bar{b})\,\bar{a}b$

Zad. 3. Przedstaw daną funkcję za pomocą funktorów:

$\overline{(a \vee \text{nie } b)}\,(\bar{a} \vee \bar{b})$

Zad. 4. Mając daną tablicę prawdy dla funkcji logicznej OR skonstruuj tablicę Karnaugh i optymalną funkcję, którą ona reprezentuje (zarówno dla 1 jak i 0).

Zad. 5. Mając daną tablicę Karnaugh skonstruuj optymalną funkcję, którą ona reprezentuje (dla 1).

a)

ab\cd	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	0	0
10	0	0	1	0

b)**(*)**

ab\cd	00	01	11	10
00	1	1	0	0
01	0	1	0	1
11	0	0	0	1
10	1	1	0	0

Zad. 6. Zaprojektuj układ, który będzie dawał 1 na wyjściu dla liczby parzystej dla wartości wejścia od 0 do 7. Następnie zmodyfikuj go tak, aby składał się jedynie z funktorów NAND **(*)** oraz z funktorów NOR **(*)** (3 wersje schematów).

(zakładamy, że 0 jest nieparzyste)

Zad. 7.(*) Zaprojektować sumator dwóch liczb dwubitowych.

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.

Zad. 8.(*) Zaprojektować układ dyskryminatora liczb 4 bitowych. Układ wytwarza liczby z przedziału od 7 do 10.

Zad. 9.(*) Przedstawić przerzutnik typu $\overline{R}\ \overline{S}$ korzystając jedynie z dwóch funktorów oraz zapisać dla niego tablicę prawdy (UWAGA!!! Stan dla wejść 00 jest stanem zabronionym).

Zad. 10. Dokonać konwersji przerzutnika typu JK do przerzutnika typu T oraz D**(*)**.

Zad. 11. Zbuduj dwójkę liczącą korzystając z przerzutnika typu D.

(*) gwiazdką oznaczone są zadania, które nie są realizowane na ćwiczeniach i są przeznaczone do wykonania jako zadania domowe.