



Politechnika Wrocławska

Architektura Systemów Komputerowych

Wykład 2

Dr inż. Radosław Michalski

Katedra Inteligencji Obliczeniowej, Wydział Informatyki i Zarządzania
Politechnika Wrocławska

Wersja 1.1, wiosna 2018



Źródła i licencja

Najbardziej aktualna wersja tego wykładu znajduje się tu:

<https://github.com/rmhere/lecture-comp-arch-org>

Opublikowany jest on na licencji Creative Commons Attribution NonCommercial ShareAlike license 4.0 (**CC BY-NC-SA 4.0**).



Zawartość tego wykładu

Logika Boole'a

Układy logiczne

Rodzaje bramek logicznych

Pozostałe układy



Logika Boole'a

Skrót

- ▶ działanie tylko na dwóch wartościach: prawda/fałsz (1/0)
- ▶ każdą funkcję da się przedstawić jako tablicę prawdy
- ▶ kolejność: NOT, AND, OR



Logika Boole'a

Postacie wyrażeń

Dwie postacie wyrażeń:

- ▶ alternatywna postać normalna

$$F(A, B, C) = AC + AB + BC$$

- ▶ koniunkcyjna postać normalna

$$F(A, B, C) = (A + C)(A + B)(B + C)$$



Logika Boole'a

Tożsamości boolowskie

- ▶ Komputery to układy implementujące logikę Boole'a
- ▶ Upraszczenie funkcji boolowskich zmniejsza obwód logiczny
- ▶ Środkiem do tego są tożsamości boolowskie



Logika Boole'a

Tożsamości boolowskie

► Idempotentność

$$A + A = A \qquad A \cdot A = A$$

► Przemienność

$$A + B = B + A \qquad A \cdot B = B \cdot A$$

► Łączność

$$(A + B) + C = A + (B + C) \qquad A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

► Rozdzielność

$$(A + B) \cdot (A + C) = A + B \cdot C$$
$$A \cdot (B + C) = A \cdot B + A \cdot C$$



Logika Boole'a

Tożsamości boolowskie c.d.

- ▶ Pochłanianie

$$A + A \cdot B = A \qquad A \cdot (A + B) = A$$

- ▶ Własności stałych

$$\begin{aligned} A + 0 &= A & A \cdot 0 &= 0 \\ A + 1 &= 1 & A \cdot 1 &= A \end{aligned}$$

- ▶ Własności negacji

$$A + \overline{A} = 1 \qquad A \cdot \overline{A} = 0$$

- ▶ Podwójna negacja

$$\overline{\overline{A}} = A$$



Logika Boole'a

Prawo De Morgana

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

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



Logika Boole'a

Minimalizacja funkcji boolowskich - Metoda Karnaugh

- ▶ Narysuj mapę Karnaugh dla zmiennych łączonych
- ▶ Mapa musi używać kodu Graya
- ▶ Znajdź pola sąsiednie logicznie
- ▶ Znajdź grupy wartości (od największej do najmniejszej)
- ▶ Dokonaj minimalizacji poprzez znalezienie części wspólnej



Logika Boole'a

Metoda Karnaugh'a - przykłady

- ▶ Przykład dla trzech zmiennych
- ▶ Przykład dla czterech zmiennych



Układy cyfrowe

Wprowadzenie

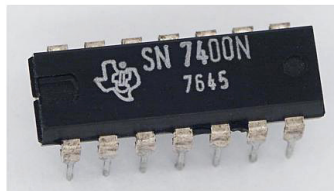
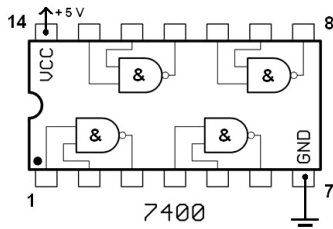
Układy cyfrowe:

- ▶ układy elektroniczne
- ▶ poziomowi napięcia przypisywana interpretacja
- ▶ typowo: dwa poziomy (wysoki i niski) - 0 i 1
- ▶ zakresy interpretacji
- ▶ separacja pomiędzy poziomami

Układy cyfrowe

Dlaczego wygodniej?

- ▶ bramki logiczne - dwustanowe
- ▶ tranzystory uproszczyły wprowadzenie bramek logicznych
- ▶ typowa bramka logiczna to dioda oraz tranzystory
- ▶ można zastosować logikę Boole'a
- ▶ zjawisko hazardu





Układy cyfrowe

Układy kombinacyjne i sekwencyjne

- ▶ układ kombinacyjny
 - ▶ stan wyjść zależy od stanu wejść
- ▶ układ sekwencyjny
 - ▶ stan wyjść zależy od stanu wejść i stanu układu
 - ▶ prosty układ stanu: przerzutnik (ang. *flip-flop*)



Układy cyfrowe

Układy asynchroniczne i synchroniczne

- ▶ układ asynchroniczny
 - ▶ stan wyjść zmienia się możliwie szybko
 - ▶ podatne na hazard (niejednoczesna zmiana stanu)
- ▶ układ synchroniczny
 - ▶ regulowany zegarem
 - ▶ zegar może zmieniać stan wewnętrzny nawet bez zmiany wejść



Układy cyfrowe

Układy TTL

Układy TTL:

- ▶ TTL - transistor-transistor logic
- ▶ złożone z tranzystorów bipolarnych
- ▶ wprowadzone w latach 60-tych XX w.
- ▶ zasilane napięciem 5V
- ▶ logiczne "0" - 0 V ... 0.8 V
- ▶ logiczna "1" - 2,4 V... 5 V
- ▶ wiele wariantów (mały pobór mocy, większa prędkość)
- ▶ wymagają stałego zasilania (inaczej niż CMOS)



Rodzaje bramek logicznych

NOT

Funkcja logiczna: $Q = \overline{A}$

Tablica prawdy:

A	Q
0	1
1	0

Symbol:



public domain

Układ TTL: 7404



Rodzaje bramek logicznych

AND

Funkcja logiczna: $Q = A \cdot B$

Tablica prawdy:

A	B	Q
0	0	0
0	1	0
1	1	1
1	0	0

Symbol:



public domain

Układ: 7408



Rodzaje bramek logicznych

OR

Funkcja logiczna: $Q = A + B$

Tablica prawdy:

A	B	Q
0	0	0
0	1	1
1	1	1
1	0	1

Symbol:



public domain

Układ TTL: 7432



Rodzaje bramek logicznych

NAND

Funkcja logiczna: $Q = \overline{A \cdot B}$

Tablica prawdy:

A	B	Q
0	0	1
0	1	1
1	1	0
1	0	1

Symbol:



public domain

Układ TTL: 7400



Rodzaje bramek logicznych

NOR

Funkcja logiczna: $Q = \overline{A + B}$

Tablica prawdy:

A	B	Q
0	0	1
0	1	0
1	1	0
1	0	0

Symbol:



public domain

Układ TTL: 7402



Rodzaje bramek logicznych

XOR

Funkcja logiczna: $Q = A \oplus B$

Tablica prawdy:

A	B	Q
0	0	0
0	1	1
1	1	0
1	0	1

Symbol:



public domain

Układ TTL: 7486



Rodzaje bramek logicznych

XNOR

Funkcja logiczna: $Q = \overline{A \oplus B}$

Tablica prawdy:

A	B	Q
0	0	1
0	1	0
1	1	1
1	0	0

Symbol:



public domain

Układ TTL: 74266



Pozostałe układy

Przerzutniki

- ▶ posiadają swój stan
- ▶ pozwalają na wprowadzenie układu sekwencyjnego
- ▶ docelowo: pamięć
- ▶ synchroniczne/asynchroniczne

Charakterystyka przerzutnika JK: $Q_{\text{nast}} = J\bar{Q} + \bar{K}Q$

J	K	Operacja	Q_{nast}
0	0	utrzymaj stan	Q
0	1	resetuj	0
1	0	ustaw	1
1	1	przemień	\bar{Q}



Slajd końcowy

Pytania? Komentarze?

Jeśli masz pomysł jak poprawić lub wzbogacić te wykłady,
proszę zgłoś to jako issue w tym repozytorium:

<https://github.com/rmhere/lecture-comp-arch-org>