

Systemy Wbudowane i Techniki Cyfrowe

Adam Szmigielski

aszmigie@pjwstk.edu.pl

Laboratorium robotyki s09

materiały: *ftp(public) : //aszmigie/SWB*

Wprowadzenie, funkcje boolowskie i bramki logiczne - wykład 1

Semestralny plan wykładu

1. Wprowadzenie, funkcje boolowskie i bramki logiczne,
2. Minimalizacja funkcji boolowskich,
3. Multipleksery, demultipleksery, enkodery,
4. Układy sekwencyjne - wiadomości podstawowe,
5. Projektowanie synchronicznych układów sekwencyjnych,
6. Arytmetyka binarna,
7. Mikroprocesory i mikrokontrolery,
8. Systemy wbudowane - wprowadzenie,
9. Przerwania, poling, timery,
10. Komunikacja mikrokontrolera z urządzeniami zewnętrznymi,
11. Współpraca urządzeń zewnętrznych z mikrokontrolerem,

12. Systemy wbudowane w układach sterowania,
13. Akwizycja danych,
14. Maszyna wirtualna,
15. Systemy operacyjne a systemy wbudowane.

Semestralny plan ćwiczeń

1. Zajęcia wstępne - *Zapoznanie się z symulatorem układów logicznych, proste układy logiczne,*
2. Minimalizacja funkcji boolowskich - *Realizacja funkcji boolowskich z wieloma zmiennymi metodami Map Karnough,*
3. Multipleksery, demultipleksery, enkodery - *Realizacja funkcji wielu zmiennych na multiplexerach, budowa enkoderów.*
4. Układy sekwencyjne - *Przerzutniki - budowa rejestrów, liczników,*
5. Projektowanie synchronicznych układów sekwencyjnych - *Synteza układów,*
6. Arytmetyka binarna - *Budowa ALU, interpretacja wyników,*
7. Mikroprocesory i mikrokontrolery - **AVR Studio** *Instrukcje arytmetyczne na rejestrach, język assembler,*

8. Systemy wbudowane - *Środowisko AVR Studio języki assembler, C, basic,*
9. Przerwania, poling, timery - *Realizacja przebiegów czasowych z wykorzystaniem timerów, obsługa przycisków,*
10. Komunikacja mikrokontrolera z urządzeniami zewnętrznymi - *Obsługa portu szeregowego z wykorzystaniem języków assembler, C i basic,*
11. Współpraca urządzeń zewnętrznych z mikrokontrolerem - *Sterowanie serwami modelarskimi, wyzwalanie zdarzeń przerwaniami, modulacja PWM,*
12. Systemy wbudowane w układach sterowania - *Przetwornik AC, magistrale, automatyzacja zbierania danych pomiarowych,*
13. Akwizycja danych - *System akwizycji i wizualizacji danych z wykorzystaniem .Net,*
14. Maszyna virtualna - **FreeRTOS, nVM** *Cross-kompilacja,*
15. Maszyna virtualna - **FreeRTOS, nVM** *zastosowanie maszyny virtualnej.*

Warunki zaliczenia przedmiotu SWB

- W czasie semestru można zdobyć 100 punktów (100%),
- Jest jedna ocena z przedmiotu na którą składają się:
 - Ocena z wykładu 30% - 2 kolokwia po 15 punktów,
 - Ocena z ćwiczeń 70% - 14 ocenianych ćwiczeń po 5 punktów każde,

Szczegółowa punktacja

ocena	liczba punktów
2	0- 50
3	50,5 - 60
3,5	60,5 - 70
4	70,5 - 80
4,5	80,5 - 90
5	90,5 - 100

TEC i SWB informacja dla ITN

- Osoby które w części lub całości nie zaliczyły przedmiotu TEC (egzamin lub laboratoria) muszą w ramach transferu **w całości zaliczyć** przedmiot SWB.
- Dokładniejsze informacje zawiera *ZARZĄDZENIE DZIEKANA z dnia 18 czerwca 2008 r.* dot. transferu przedmiotu TEC na SWB.

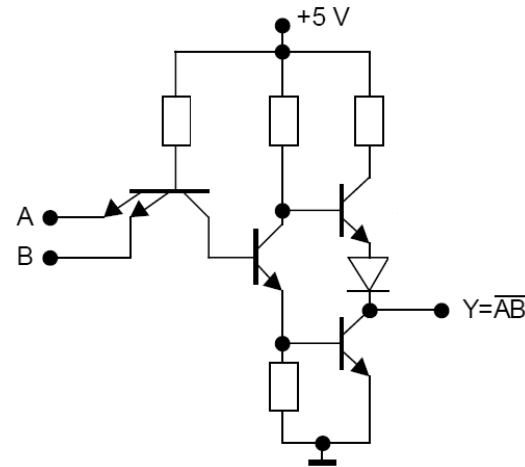
Ćwiczenia laboratoryjne z przedmiotu SWB

- *Ćwiczenia laboratoryjne* przeprowadzane są w zwykłych pracowniach komputerowych,
- *Ćwiczenia 1 - 6* wykonywane będą na symulatorze *logisim-win-2.1.6.exe* lub nowszej wersji.
- *Ćwiczenia 7 - 15* wykonywane będą na systemie wykorzystującym układ AVR ATmega32 i komputerach PC,
- *Ćwiczenie 1*, wprowadzające, nie będzie oceniane.
- Na *ocenę z ćwiczenia* wpływa:
 - *stopień przygotowania do ćwiczenia* - krótki sprawdzian wejściowy lub inny, wskazany przez prowadzącego, sposób weryfikacji,
 - *sposób wykonania ćwiczenia*,
 - *końcowy efekt* - o ile prowadzący ćwiczenia zażyczy, potwierdzony odpowiednią dokumentacją.

Sygnaly analogowe i cyfrowe

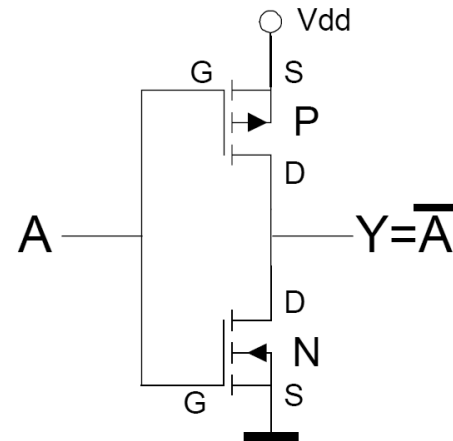
- *Sygnal analogowy*: - sygnał, który może przyjmować dowolną wartość z ciągłego przedziału
- *Sygnal cyfrowy*: - sygnał, którego dziedzina i zbiór wartości są dyskretne.

Sygnal cyfrowy - TTL



- Układy TTL zbudowane są z tranzystorów bipolarnych i zasila się je napięciem stałym 5 V.
- Gdy potencjał ma wartość od $0V \div 0,8V$ (w odniesieniu do masy) sygnał TTL jest niski - **logiczne "0"**.
- Przy wartości potencjału między $2V \div 5V$ jest stan wysoki - **logiczna "1"**.
- Gdy wartość napięcia jest z przedziału $0,8V \div 2V$ - sygnał jest nieokreślony.

Sygnal cyfrowy - CMOS



- Układy CMOS zbudowane są z się z tranzystorów MOS o przeciwnym typie przewodnictwa i połączonych w taki sposób, że w ustalonym stanie logicznym przewodzi tylko jeden z nich,
- Układy CMOS są relatywnie proste i tanie w produkcji, umożliwiając uzyskanie bardzo dużych gęstości upakowania,
- Układy cyfrowe wykonane w technologii CMOS mogą być zasilanie napięciem 3 ÷ 18V,

- Praktycznie nie pobierają mocy statycznie, tylko przy zmianie stanu logicznego,
- Poziomy logiczne są zbliżone do napięć zasilających (masa - logiczne "0", zasilanie "1"). Czasami stosuje się klasyfikacje procentową - "0" - odpowiadają napięcia z zakresu 0 – 30%, "1" - 70 – 100%.

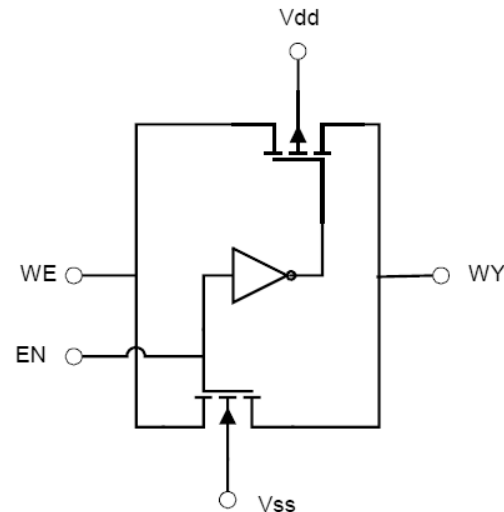
Układy niskonapięciowe (Low Voltage)

- Obecnie istnieje wyraźna tendencja do obniżania napięcia zasilania,
- Produkowane są serie układów cyfrowych CMOS przystosowane do zasilania napięciem 3,3V, 2,5V czy nawet 1,8V,

Trzeci stan logiczny i bramki typu "open collector"

- Oprócz logicznego "0" i logicznej "1" istnieje trzeci stan logiczny - **stan wysokiej impedancji** (ang. high impedance),
- Gdy punkt układu nie jest połączony galwanicznie z układem cyfrowym znajduje się on w **w stanie wysokiej impedancji**,
- Aby punkt obwodu będący w stanie wysokiej impedancji mógł być traktowany jako logiczne "0" albo "1" należy poprzez rezystor połączyć go odpowiednio do masy lub zasilania. Rezystory tego typu noszą nazwę **rezystorów podciągających** (ang. pull up resistor),
- Budowane są bramki logiczne, których wyjście pozostawać może w stanie wysokiej impedancji.

Bramki transmisyjne



- Oprócz standardowych bramek w technologii CMOS produkowane są *bramki transmisyjne*, które można traktować jako klucz analogowy,
- Bramka ta składa się z dwóch komplementarnych tranzystorów połączonych równolegle oraz inwertera, zapewniającego sterowanie bramek w przeciwfazie.
- W tej technologii są wykonane mulypleksery i demulypleksery, które mogą przełączać również sygnały analogowe - patrz dokumentacje 4051.

Algebry Boole'a

- Algebry Boole'a to rodzina wszystkich podzbiorów ustalonego zbioru wraz działaniami na zbiorach jako operacjami algebry oraz dwuelementowa algebra wartości logicznych $\{0, 1\}$ z działaniami koniunkcji \wedge , alternatywy \vee i negacji \neg .
- Istnieją inne tradycje oznaczeń w teorii algebr Boole'a:
 - koniunkcja \wedge , alternatywa \vee i negacja \neg
 - koniunkcji \cap , alternatywy \cup i negacji \sim
 - koniunkcji \cdot , alternatywy $+$ i negacji $-$

Własności algebry Boole'a

łączność	$(ab)c = a(bc)$	$(a + b) + c = a + (b + c)$
przemienność	$ab = ba$	$a + b = b + a$
rozdzielność	$a + (bc) = (a + b) \cdot (a + c)$	$a \cdot (b + c) = (ab) + (ac)$
absorpcja	$a(a + b) = a$	$a + (ab) = a$
pochłanianie	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$

Podstawowe prawa algebry Boole'a

- prawa de Morgana:

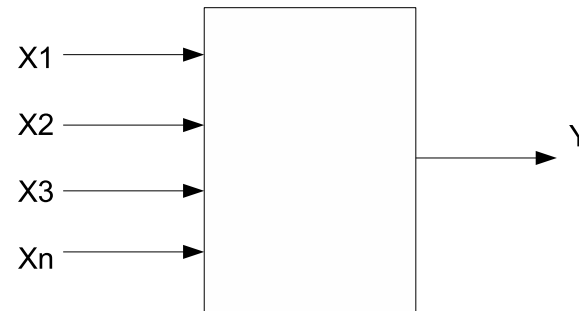
$$\overline{a + b} = \bar{a} \cdot \bar{b}$$

$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

- prawo sklejania:

$$a\bar{b} + ab = a$$

Funkcja Boolowska



- *Funkcją boolowską n argumentową nazywamy odwzorowanie $f : B^n \rightarrow B$, gdzie $B = \{0, 1\}$ jest zbiorem wartości funkcji.*
- *Funkcja boolowska jest matematycznym modelem układu kombinacyjnego.*

Opis funkcji Boolowskiej - tabele prawdy

- funkcja jednej zmiennej (np. negacja $f(x) = \neg x$)

x	f(x)
0	1
1	0

- Funkcja dwóch zmiennych (np. koniunkcja $f(a, b) = a \wedge b$)

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

Zbiory zer i jedynek w postaci binarnej i dziesiętnej

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

$f^1 = [11]$ - zbiór jedynek w postaci binarnej

$f^0 = \begin{bmatrix} 00 \\ 01 \\ 10 \end{bmatrix}$ - zbiór zer w postaci binarnej

$f^1 = \{3\}$ -zbiór jedynek w postaci dziesiętnej

$f^0 = \{0, 1, 2\}$ -zbiór zer w postaci dziesiętnej

Sumacyjna postać kanoniczna

a	b	$f(a, b)$
0	0	0
0	1	0
1	0	0
1	1	1

Postać sumacyjna: funkcja f jest sumą iloczynów

$$f = \dots (\dots \wedge \dots \wedge \dots) \vee (\dots \wedge \dots \wedge \dots) \vee (\dots \wedge \dots \wedge \dots) \dots$$

Wyrażenie w nawiasie (iloczyn) odpowiada jednej jedynce.

W tym konkretnym przypadku: $f = a \wedge b$.

Zapis dziesiętny: $f(a, b) = \sum\{3\}$

Iloczynowa postać kanoniczna

a	b	$f(a, b)$
0	0	0
0	1	0
1	0	0
1	1	1

Postać sumacyjna: funkcja f jest iloczynem sum


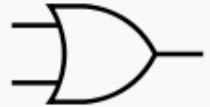

$$f = \dots (\dots \vee \dots \vee \dots) \wedge (\dots \vee \dots \vee \dots) \wedge (\dots \vee \dots \vee \dots) \dots$$

Wyrażenie w nawiasie (suma) odpowiada jednemu zeru.


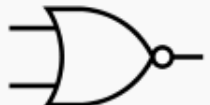
W tym konkretnym przypadku: $f = (a \wedge b) \vee (a \wedge \bar{b}) \vee (\bar{a} \wedge b)$.

Zapis dziesiętny $f(a, b) = \prod\{0, 1, 2\}$

Bramki logiczne

AND		$A \cdot B$	<table border="1"><thead><tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr><tr><th>A</th><th>B</th><th>A AND B</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	INPUT		OUTPUT	A	B	A AND B	0	0	0	0	1	0	1	0	0	1	1	1
INPUT		OUTPUT																			
A	B	A AND B																			
0	0	0																			
0	1	0																			
1	0	0																			
1	1	1																			
OR		$A + B$	<table border="1"><thead><tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr><tr><th>A</th><th>B</th><th>A OR B</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	INPUT		OUTPUT	A	B	A OR B	0	0	0	0	1	1	1	0	1	1	1	1
INPUT		OUTPUT																			
A	B	A OR B																			
0	0	0																			
0	1	1																			
1	0	1																			
1	1	1																			
NOT		\bar{A}	<table border="1"><thead><tr><th>INPUT</th><th>OUTPUT</th></tr><tr><th>A</th><th>NOT A</th></tr></thead><tbody><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></tbody></table>	INPUT	OUTPUT	A	NOT A	0	1	1	0										
INPUT	OUTPUT																				
A	NOT A																				
0	1																				
1	0																				

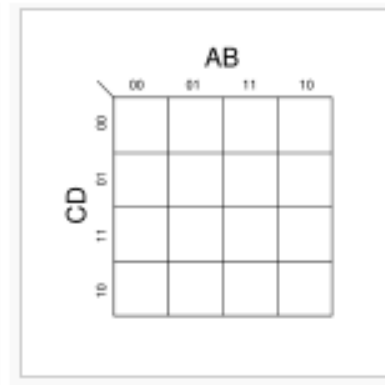
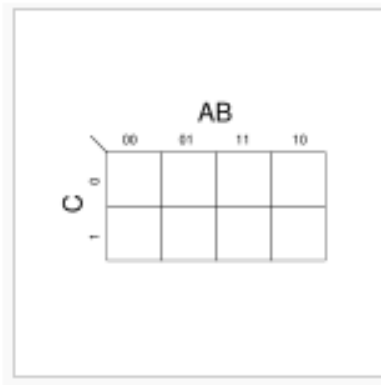
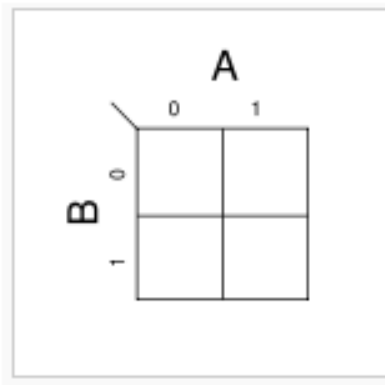
Popularne bramki logiczne

NAND		$\overline{A \cdot B}$	<table border="1"><thead><tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr><tr><th>A</th><th>B</th><th>A NAND B</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></tbody></table>	INPUT		OUTPUT	A	B	A NAND B	0	0	1	0	1	1	1	0	1	1	1	0
INPUT		OUTPUT																			
A	B	A NAND B																			
0	0	1																			
0	1	1																			
1	0	1																			
1	1	0																			
NOR		$\overline{A + B}$	<table border="1"><thead><tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr><tr><th>A</th><th>B</th><th>A NOR B</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></tbody></table>	INPUT		OUTPUT	A	B	A NOR B	0	0	1	0	1	0	1	0	0	1	1	0
INPUT		OUTPUT																			
A	B	A NOR B																			
0	0	1																			
0	1	0																			
1	0	0																			
1	1	0																			

Kod Graya

Kod Graya, zwany również kodem refleksyjnym, jest dwójkowym kodem bezwagowym niepozycyjnym, który charakteryzuje się tym, że dwa kolejne słowa kodowe różnią się tylko stanem jednego bitu. Jest również kodem cyklicznym, bowiem ostatni i pierwszy wyraz tego kodu także spełniają w/w zasadę.

Mapy Karnaugh



inne metody minimalizacji

Literatura

1. Wilkinson B.: *Układy Cyfrowe*, Warszawa, WKiŁ 2001
2. Skorupski A.: *Układy Cyfrowe*, Warszawa, WKiŁ 2001
3. Doliński Jarosław.: *Mikrokontrolery AVR w praktyce*, btc ISBN: 83-910067-6-X, Warszawa 2003, 2004
4. Andrzej Pawluczuk : *Sztuka programowania mikrokontrolerów AVR - przykłady*, Wydawnictwo BTC, ISBN: 978-83-60233-21-4
5. Baranowski Rafał.: *Mikrokontrolery AVR ATmega w praktyce*, btc ISBN: 83-60233-02-0