

Systemy Wbudowane i Techniki Cyfrowe

Adam Szmigielski

aszmigie@pjwstk.edu.pl

Laboratorium robotyki s03

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. Kombinacyjne bloki funkcjonalne,
4. Układy sekwencyjne - wiadomości podstawowe,
5. Projektowanie synchronicznych układów sekwencyjnych,
6. Arytmetyka binarna,
7. Pamięci półprzewodnikowe,
8. Mikroprocesory μP i mikrokontrolery μC ,
9. Mikrokontroler AVR ATmega32,
10. Programowanie mikrokontrolerów,
11. Peryferia mikrokontrolerów,

12. Obsługa zdarzeń - przerwania, zapytywanie,
13. Komunikacja mikrokontrolera z urządzeniami zewnętrznymi,
14. Systemy wbudowane, zastosowanie μC w układach sterownia,
15. Systemy operacyjne czasu rzeczywistego. Przykłady systemów wbudowanych.

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. Kombinacyjne bloki funkcjonalne - *Realizacja funkcji wielu zmiennych,*
4. Układy sekwencyjne - *Przerzutniki - budowa rejestrów, liczników,*
5. Projektowanie synchronicznych układów sekwencyjnych - *Synteza układów,*
6. Arytmetyka binarna - *Budowa ALU, interpretacja wyników,*
7. Pamięci półprzewodnikowe - *Pamięci ROM i RAM, dekodery adresów,*
8. Mikroprocesory i mikrokontrolery - **Bascom-AVR** *Zapoznanie się ze środowiskiem - kompilator symulator, terminal,*
9. Mikrokontroler AVR ATmega32 - *Operacje na rejestrach, pamięci, zmiennych, łącze szeregowo (Bascom-AVR - symulator lub rzeczywiste*

urządzenie.)

10. Programowanie mikrokontrolerów *Wstawki języka assembler, licznik resetów,*
11. Peryferia mikrokontrolerów - *Przetwornik AC, odczyt i zapis z portu, wyświetlacz LCD,*
12. Obsługa zdarzeń - przerwania, poling, - *Realizacja przebiegów czasowych z wykorzystaniem timerów, obsługa przycisków,*
13. Komunikacja mikrokontrolera z urządzeniami zewnętrznymi - *Współpraca μC z komputerem PC,*
14. Systemy wbudowane w układach sterowania - *Interakcja wejścia i wyjścia μC - np. obserwacja stanu czujników z reakcją na ich stan,*
15. RTOS - *Kontynuacja poprzedniego zadania wzbogacona o komunikację i obliczenia na komputerze PC.*

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

Dla osób, które kończą przedmiot SWB egzaminem przewidziana jest możliwość zwolnienia z egzaminu w oparciu o wyniki z kolokwiów i ćwiczeń.

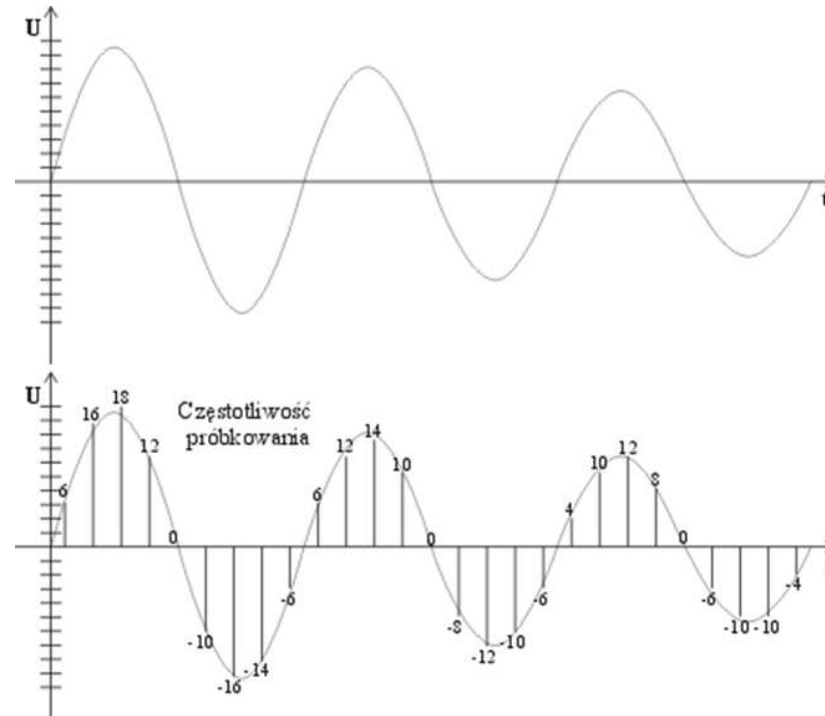
TEC i SWB informacja dla ITN

- 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

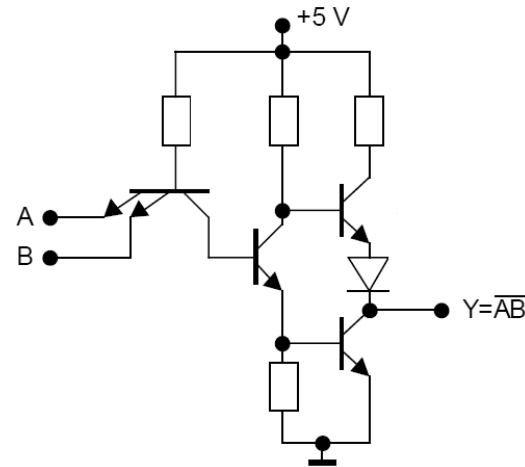
- *Ćwiczenia laboratoryjne* przeprowadzane są w zwykłych pracowniach komputerowych,
- *Ćwiczenia 1 - 7* wykonywane będą na symulatorze *logisim-win-2.3.2* lub nowszej wersji.
- *Ćwiczenia 8 - 15* wykonywane będą z wykorzystaniem oprogramowania *BASCOM-AVR*, mikrokontrolera *AVR ATmega32* i komputera 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* - dokumentacją z przeprowadzonego ćwiczenia.

Sygnaly analogowe i cyfrowe



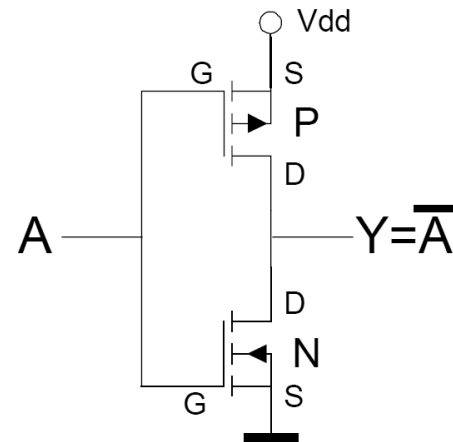
- *Sygnal analogowy* - sygnał, który może przyjmować dowolną wartość z ciągłego przedziału,
- *Sygnal dyskretny* - sygnał powstały poprzez próbkowanie sygnału ciągłego,
- *Reprezentacja binarna* - odczyt próbek zapisany binarnie.

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%.

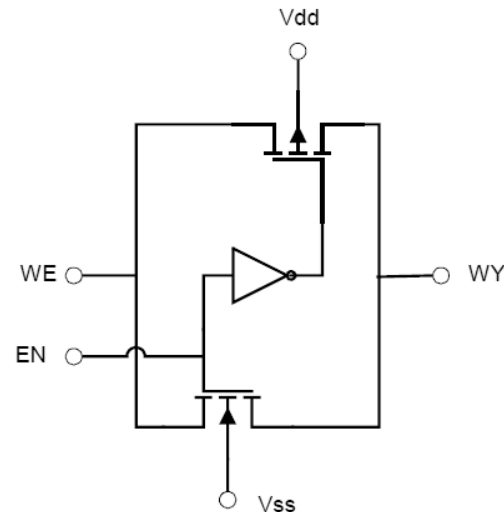
Układy niskonapięciowe (Low Voltage)

- 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 zbiorów (część wspólna, suma, dopełnienie), np. dwuelementowa algebra wartości logicznych $\{0, 1\}$ z działaniami koniunkcji \wedge , alternatywy \vee i negacji \neg .
- Istnieją inne tradycje oznaczeń w teorii algebr Boole'a:
 - część wspólna \cap , suma \cup i dopełnienie \sim
 - koniunkcja \wedge , alternatywa \vee i negacja \neg
 - 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	$a + \bar{a} = 1$	$a \cdot \bar{a} = 0$

Istotne dla techniki cyfrowej prawa algebry Boole'a

- prawa de Morgana:

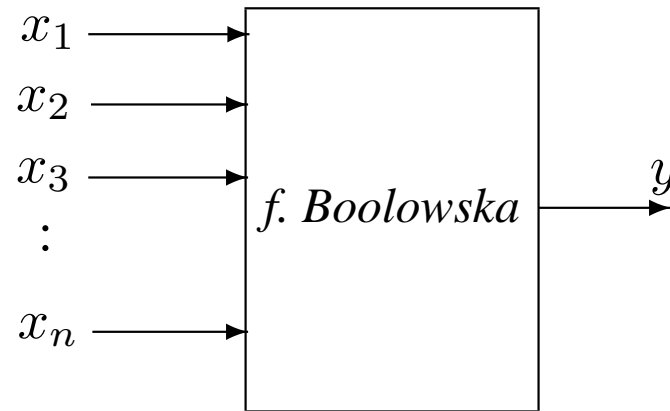
$$\overline{a + b} = \bar{a} \cdot \bar{b}$$

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

- 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 $f(a) = \neg a$)

a	f(a)
0	1
1	0

- Funkcja dwóch zmiennych (np. funkcja *mod2*: $f(a, b) = a \otimes b$)

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

Zbiory zer i jedynek w postaci binarnej i dziesiętnej

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

$$f^1 = \begin{bmatrix} 01 \\ 10 \end{bmatrix} \text{ - zbiór jedynek w postaci binarnej}$$

$$f^0 = \begin{bmatrix} 00 \\ 11 \end{bmatrix} \text{ - zbiór zer w postaci binarnej}$$

$$f^1 = \{1, 2\} \text{ -zbiór jedynek w postaci dziesiętnej}$$

$$f^0 = \{0, 3\} \text{ -zbiór zer w postaci dziesiętnej}$$

Sumacyjna postać kanoniczna

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

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: $f(a, b) = (\bar{a} \wedge b) \vee (a \wedge \bar{b})$.

Zapis dziesiętny: $f(a, b) = \sum(1, 2)$

Iloczynowa postać kanoniczna

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

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: $f(a, b) = (a \vee b) \wedge (\bar{a} \vee \bar{b})^a$.

Zapis dziesiętny $f(a, b) = \prod(0, 3)$


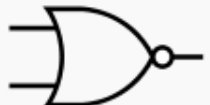
^anależy pamiętać o zanegowaniu zmiennych, tj.

Nawiasowi $(a \vee b)$ odpowiada sytuacja, gdy $a = 0$ i $b = 0$.

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																			

Zadania na ćwiczenia

1. Powtórzenie wiadomości z logiki klasycznej, zdania logiki klasycznej, alternatywa, koniunkcja, negacja, implikacja, prawa logiki klasycznej: deMorgana, podwójnego przeczenia inne, tautologie, tabele prawdy,
2. Zapoznanie się ze programem *Logisim*, realizacja za pomocą bramek logicznych prostych funkcji logicznych.

Ćwiczenie to nie jest punktowane.

Literatura

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