

Informatyka 1

Politechnika Białostocka - Wydział Elektryczny
Elektrotechnika, semestr II, studia stacjonarne I stopnia
Rok akademicki 2018/2019

Wykład nr 4 (22.03.2019)

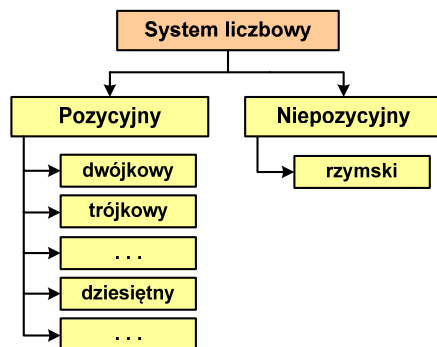
dr inż. Jarosław Forenc

Plan wykładu nr 4

- Systemy liczbowe
 - systemy pozycyjne (dziesiętny, dwójkowy, szesnastkowy)
 - systemy niepozycyjne (rzymski)
- Konwersje między systemami liczbowymi
- Język C
 - instrukcja warunkowa if
 - operatory relacyjne (porównania) i logiczne
 - wyrażenia logiczne

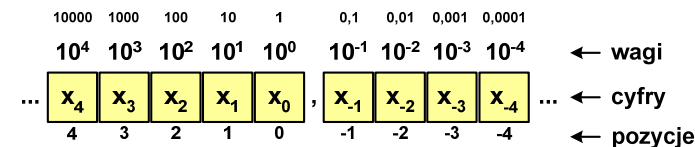
Systemy liczbowe

- **System liczbowy** - zbiór zasad umożliwiających zapis liczb za pomocą cyfr oraz wykonywanie działań na tych liczbach



- **Pozycyjny** - znaczenie cyfry jest zależne od miejsca (pozycji), które zajmuje ona w liczbie
 - system dziesiętny - liczba 111 (każda cyfra ma inne znaczenie)
- **Niepozycyjny** - znaczenie cyfry jest niezależne od miejsca położenia w liczbie
 - system rzymski - liczba III

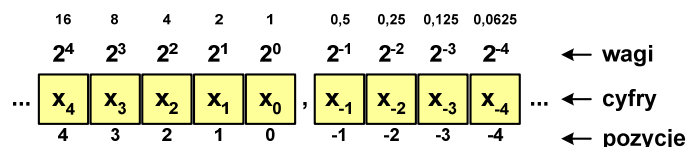
System dziesiętny (ang. decimal)



- p - podstawa systemu pozycyjnego, D - zbiór dozwolonych cyfr
- $p = 10$, $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$$\begin{aligned} 1408,25_{(10)} &= \begin{matrix} 10^3 & 10^2 & 10^1 & 10^0 & 10^{-1} & 10^{-2} \\ \boxed{1} & \boxed{4} & \boxed{0} & \boxed{8} & \boxed{2} & \boxed{5} \end{matrix} \\ &= \boxed{1 \cdot 10^3} + \boxed{4 \cdot 10^2} + \boxed{0 \cdot 10^1} + \boxed{8 \cdot 10^0} + \boxed{2 \cdot 10^{-1}} + \boxed{5 \cdot 10^{-2}} \\ &= 1000 + 400 + 0 + 8 + 0,2 + 0,05 \end{aligned}$$

System dwójkowy (ang. binary)

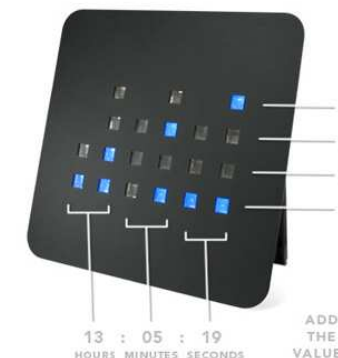


- w systemie dwójkowym: $p = 2$, $D = \{0, 1\}$

$$\begin{aligned}
 & \begin{array}{cccccccc} 2^3 & 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & 2^{-3} & \\ \hline & 1 & 1 & 0 & 1 & , & 1 & 0 & 1 \end{array} \\
 1101,101_{(2)} &= \\
 &= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\
 &= 8 + 4 + 0 + 1 + 0,5 + 0 + 0,125 \\
 &= 13,625_{(10)}
 \end{aligned}$$

System dwójkowy - zastosowania

- Powszechnie używany w informatyce, technice cyfrowej



System szesnastkowy (ang. hexadecimal)

- System heksadecymalny
- $p = 16$, $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- Powszechnie używany w informatyce - jeden bajt można zapisać za pomocą tylko dwóch cyfr szesnastkowych

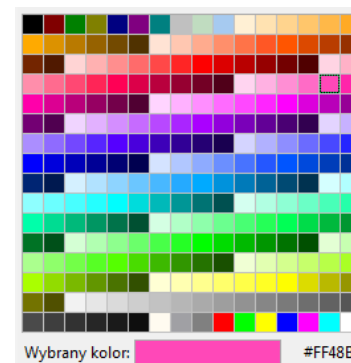
$$3A5D_{(16)} = 3 \cdot 16^3 + 10 \cdot 16^2 + 5 \cdot 16^1 + 13 \cdot 16^0 = 14941_{(10)}$$

- Sposoby zapisu liczb w systemie szesnastkowym:

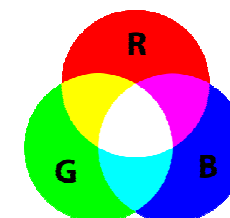
3A5Dh	0x3A5D	#3A5D
$3A5D_{(16)}$	$3A5D_{16}$	$3A5D_{hex}$
$(3A5D)_{hex}$	$(3A5D)_{16}$	$\$3A5D$

System szesnastkowy - zastosowania

- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Każda barwa przyjmuje wartość z zakresu: $0..255_{(10)}$, $00..FF_{(16)}$



#FF48B8



System szesnastkowy - zastosowania

- Zapis 24-bitowego koloru RGB (Red-Green-Blue), 16 mln kolorów
- Kolory w dokumentach HTML:

```
<BODY bgcolor="#336699" text="#000000" link="#FFFF00"  
vlink="#33FFFF" alink="#FF0000">
```

ARCHIWUM

ENGLISH

Studia stacjonarne:

Poniedziałek:
12:15 - 14:00 [Informatyka 1 - wykład](#), sem. 2 ED, WE-Aula II

Wtorek:
08:30 - 10:00 [Informatyka 1 - prac.](#), sem. 2 ED, gr. PS3, WE-110
12:15 - 13:45 [Informatyka 1 - prac.](#), sem. 2 ED, gr. PS2, WE-110
14:15 - 15:45 [Informatyka 1 - prac.](#), sem. 2 ED, gr. PS1, WE-110

Przykład systemu niepozycyjnego - system rzymski

- W systemie rzymskim posługujemy się siedmioma znakami:
I - 1 V - 5 X - 10 L - 50 C - 100 D - 500 M - 1000
- Za pomocą dostępnych symboli można określić liczby od 1 do 3999
- System **addytywny** - wartość liczby określa się na podstawie sumy wartości cyfr, np.
 - II (1 + 1 = 2), XXX (10 + 10 + 10 = 30)
 - CLX (100 + 50 + 10 = 160), MMXII (1000 + 1000 + 10 + 1 + 1 = 2012)
- Wyjątkiem od powyższej zasady są liczby do opisu których używa się odejmowania, np.
 - IV (5 - 1 = 4), IX (10 - 1 = 9), XL (50 - 10 = 40), XC (100 - 10 = 90)
- Stosowany w łacińskiej części Europy do końca Średniowiecza
- Niewygodny w prowadzeniu nawet prostych działań arytmetycznych, brak ułamków

System szesnastkowy - zastosowania

- 48-bitowy adres fizyczny urządzenia (MAC - Media Access Control)

88:AD:D2:09:41:3B

producent numer egzemplarza

- <http://hwaddress.com>

HWAddress Home Countries Companies 88:AD:D2 Search

OUI	MAC range	Company
88-AD-D2	88-AD-D2-00-00-00 - 88-AD-D2-FF-FF-FF	Samsung Electronics Co.,Ltd

Przykład systemu niepozycyjnego - system rzymski

- Zasady tworzenia liczb:
 - zestawiamy odpowiednie znaki od oznaczającego liczbę największą do oznaczającego liczbę najmniejszą
 $XVI = 10(X) + 5(V) + 1(I) = 16$
 - jeżeli składnik liczby, którą piszemy, jest wielokrotnością liczby nominalnej, wtedy zapisywany jest z użyciem kilku następujących po sobie znaków
 $CCC = 100(C) + 100(C) + 100(C) = 300$
 - dodatkowo należy zachować zasadę nie pisania czterech tych samych znaków po sobie, lecz napisać jeden znak wraz ze znakiem oznaczającym wartość większą o jeden rząd liczbowy
 $CD = 500(D) - 100(C) = 400$

Przykład systemu niepozycyjnego - system rzymski

- Zasady odczytu liczb:
 - cyfry jednakowe są dodawane
 $MMM = 1000(M) + 1000(M) + 1000(M) = 3000$
 - cyfry mniejsze stojące przed większymi są odejmowane od nich
 $CDXCIV = 500(D) - 100(C) + 100(C) - 10(X) + 5(V) - 1(I) = 494$
 - cyfry mniejsze stojące za większymi są do nich dodawane
 $MDCLX = 1000(M) + 500(D) + 100(C) + 50(L) + 10(X) = 1660$

Konwersja na system dziesiętny

- $p = 4, D = \{0, 1, 2, 3\}$

$$21302_{(4)} = ?_{(10)}$$

$$21302_{(4)} = 2 \cdot 4^4 + 1 \cdot 4^3 + 3 \cdot 4^2 + 0 \cdot 4^1 + 2 \cdot 4^0 = 2 \cdot 256 + 64 + 48 + 0 + 2 = 626_{(10)}$$

- $p = 17, D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G\}$

$$AC24_{(17)} = ?_{(10)}$$

$$AC24_{(17)} = 4 \cdot 17^3 + 2 \cdot 17^2 + 12 \cdot 17^1 + 10 \cdot 17^0 = 4 \cdot 4913 + 2 \cdot 289 + 12 \cdot 17 + 10 = 20166 + 578 + 204 + 10 = 20958_{(10)}$$

Konwersja na system dziesiętny (schemat Hornera)

- $p = 4, D = \{0, 1, 2, 3\}$

$$21302_{(4)} = w_{(10)} \quad x_4 x_3 x_2 x_1 x_0 = w_{(10)}$$

$$w_{(10)} = 0$$

$$w_{(10)} = x_4 + w_{(10)} \cdot p = 2 + 0 \cdot 4 = 2$$

$$w_{(10)} = x_3 + w_{(10)} \cdot p = 1 + 2 \cdot 4 = 9$$

$$w_{(10)} = x_2 + w_{(10)} \cdot p = 3 + 9 \cdot 4 = 39$$

$$w_{(10)} = x_1 + w_{(10)} \cdot p = 0 + 39 \cdot 4 = 156$$

$$w_{(10)} = x_0 + w_{(10)} \cdot p = 2 + 156 \cdot 4 = 626_{(10)}$$

Konwersja z systemu dziesiętnego na dowolny

- zamiana liczby z systemu $p = 10$ na system $p = 2$

$$626_{(10)} = ?_{(2)}$$

$626/2 = 313$	$reszta\ 0$	↑ kolejność odczytywania cyfr liczby w systemie dwójkowym
$313/2 = 156$	$reszta\ 1$	
$156/2 = 78$	$reszta\ 0$	
$78/2 = 39$	$reszta\ 0$	
$39/2 = 19$	$reszta\ 1$	
$19/2 = 9$	$reszta\ 1$	
$9/2 = 4$	$reszta\ 1$	
$4/2 = 2$	$reszta\ 0$	
$2/2 = 1$	$reszta\ 0$	
$1/2 = 0$	$reszta\ 1$	

kończymy, gdy liczba dziesiętna ma wartość 0

Konwersja z systemu dziesiętnego na dowolny

- zamiana liczby z systemu $p = 10$ na system $p = 7$

$$626_{(10)} = ?_{(7)} \qquad 626_{(10)} = 1553_{(7)}$$

$626/7 = 89$	$reszta$	3	↑
$89/7 = 12$	$reszta$	5	
$12/7 = 1$	$reszta$	5	
$1/7 = 0$	$reszta$	1	

- zamiana liczby z systemu $p = 10$ na system $p = 14$

$$626_{(10)} = ?_{(14)} \qquad 626_{(10)} = 32A_{(14)}$$

$626/14 = 44$	$reszta$	$10 \rightarrow A$	↑
$44/14 = 3$	$reszta$	2	
$3/14 = 0$	$reszta$	3	

Szybkie konwersje: $2 \rightarrow 4, 8, 16$ $4, 8, 16 \rightarrow 2$

$2 \rightarrow 4$

$$110110011_{(2)} = ?_{(4)}$$

01	$ $	10	$ $	11	$ $	00	$ $	11
$\underset{1}{}$		$\underset{2}{}$		$\underset{3}{}$		$\underset{0}{}$		$\underset{3}{}$

$$110110011_{(2)} = 12303_{(4)}$$

$2 \rightarrow 8$

$$10110011_{(2)} = ?_{(8)}$$

010	$ $	110	$ $	011
$\underset{2}{}$		$\underset{6}{}$		$\underset{3}{}$

$$10110011_{(2)} = 263_{(8)}$$

$2 \rightarrow 16$

$$1011010_{(2)} = ?_{(16)}$$

0101	$ $	1010
$\underset{5}{}$		$\underset{A}{}$

$$1011010_{(2)} = 5A_{(16)}$$

$4 \rightarrow 2$

$$12303_{(4)} = ?_{(2)}$$

1	2	3	0	3
$\underset{1}{}$	$\underset{2}{}$	$\underset{3}{}$	$\underset{0}{}$	$\underset{3}{}$

$$12303_{(4)} = 110110011_{(2)}$$

$8 \rightarrow 2$

$$263_{(8)} = ?_{(2)}$$

2	6	3
$\underset{2}{}$	$\underset{6}{}$	$\underset{3}{}$

$$263_{(8)} = 10110011_{(2)}$$

$16 \rightarrow 2$

$$5A_{(16)} = ?_{(2)}$$

5	A
$\underset{5}{}$	$\underset{A}{}$

$$5A_{(16)} = 1011010_{(2)}$$

Język C - Pierwiastek kwadratowy

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    float x, y;

    printf("Podaj liczbe: ");
    scanf("%f", &x);

    y = sqrt(x);

    printf("Pierwiastek liczby: %f\n", y);

    return 0;
}
```

Podaj liczbe: 15
Pierwiastek liczby: 3.872983

Podaj liczbe: -15
Pierwiastek liczby: -1.#IND00

Język C - Pierwiastek kwadratowy

```
#include <stdio.h>
#include <math.h>

int main(void)
{
    float x, y;

    printf("Podaj liczbe: ");
    scanf("%f", &x);

    if (x>=0)
    {
        y = sqrt(x);
        printf("Pierwiastek liczby: %f\n", y);
    }
    else
        printf("Blad! Liczba ujemna\n");

    return 0;
}
```

Podaj liczbe: 15
Pierwiastek liczby: 3.872983

Podaj liczbe: -15
Blad! Liczba ujemna

Język C - instrukcja warunkowa if

```
if (wyrażenie)  
    instrukcja1
```

- jeśli **wyrażenie** jest prawdziwe, to wykonywana jest **instrukcja1**
- gdy **wyrażenie** jest fałszywe, to **instrukcja1** nie jest wykonywana

```
if (wyrażenie)  
    instrukcja1  
else  
    instrukcja2
```

- jeśli **wyrażenie** jest prawdziwe, to wykonywana jest **instrukcja1**, zaś **instrukcja2** nie jest wykonywana
- gdy **wyrażenie** jest fałszywe, to wykonywana jest **instrukcja2**, zaś **instrukcja1** nie jest wykonywana

■ Wyrażenie w nawiasach:

- **prawdziwe** - gdy jego wartość jest różna od zera
- **fałszywe** - gdy jego wartość jest równa zero

Język C - instrukcja warunkowa if

```
if (wyrażenie)  
    instrukcja
```

■ Instrukcja:

- **prosta** - jedna instrukcja zakończona średnikiem
- **złożona** - jedna lub kilka instrukcji objętych nawiasami klamrowymi

```
if (x>0)  
    printf("inst1");
```

```
if (x>0)  
{  
    printf("inst1");  
    printf("inst2");  
    ...  
}
```

Język C - instrukcja warunkowa if

```
if (wyr)  
    instr;
```

```
if (wyr)  
    instr;  
else  
    instr;
```

```
if (wyr)  
{  
    instr;  
    instr;  
}  
else  
    instr;
```

```
if (wyr)  
{  
    instr;  
}  
else  
{  
    instr;  
}
```

```
if (wyr)  
{  
    instr;  
    instr;  
}
```

```
if (wyr)  
{  
    instr;  
    instr;  
}  
else  
{  
    instr;  
    instr;  
}
```

```
if (wyr)  
    instr;  
else  
{  
    instr;  
    instr;  
}
```

Język C - Operatory relacyjne (porównania)

Operator	Przykład	Znaczenie
>	a > b	a większe od b
<	a < b	a mniejsze od b
>=	a >= b	a większe lub równe b
<=	a <= b	a mniejsze lub równe b
==	a == b	a równe b
!=	a != b	a nierówne b (a różne od b)

■ Wynik porównania jest wartością typu **int** i jest równy:

- **1** - gdy warunek jest prawdziwy
- **0** - gdy warunek jest fałszywy (nie jest prawdziwy)

Język C - Operatory logiczne

Operator	Znaczenie	Opis
!	NOT, nie	jednoargumentowy operator negacji logicznej - zmienia argument różny od zera na wartość 0, a argument równy zero na wartość 1
&&	AND, i	dwuargumentowy operator koniunkcji, iloczyn logiczny
	OR, lub	dwuargumentowy operator alternatywy, suma logiczna

- Wynikiem zastosowania operatorów logicznych && i || jest wartość typu int równa 1 (prawda) lub 0 (fałsz)

```
if (x>5 && x<8)
```

```
if (x<=5 || x>8)
```

Język C - Wyrażenia logiczne

- Wyrażenia logiczne mogą zawierać:

- operatory relacyjne
- operatory logiczne
- operatory arytmetyczne
- operatory przypisania
- zmienne
- stałe
- wywołania funkcji
- ...

Operator	Typ operatora
!	logiczny
* / %	arytmetyczne
+ -	arytmetyczne
> < >= <=	relacyjne
== !=	relacyjne
&&	logiczny
	logiczny
=	przypisania

- Kolejność operacji wynika z **priorytetu operatorów**

Język C - Wyrażenia logiczne

```
int x = 0, y = 1, z = 2;
```

```
if (x == 0)
```

wynik: 1 (prawda)

```
if (x = 0)
```

wynik: 0 (fałsz) (!!!)

```
if (x != 0)
```

wynik: 0 (fałsz)

```
if (x =! 0)
```

wynik: 1 (prawda) (!!!)

```
if (z > x + y)
```

wynik: 1 (prawda)

```
if (z > (x + y))
```

Język C - Wyrażenia logiczne

```
int x = 0, y = 1, z = 2;
```

```
if (x>2 && x<5)
```

wynik: 0 (fałsz)

```
if ((x>2) && (x<5))
```

- Wyrażenia logiczne obliczane są od strony lewej do prawej
- Proces obliczeń kończy się, gdy wiadomo, jaki będzie wynik całego wyrażenia

```
if (2 < x < 5)
```

wynik: 1 (prawda) (!!!)

Język C - Wyrażenia logiczne

- W przypadku sprawdzania czy wartość wyrażenia jest równa lub różna od zera można zastosować skrócony zapis
- Zamiast:

```
if ( x == 0 )
```

```
if ( x != 0 )
```

można napisać:

```
if ( !x )
```

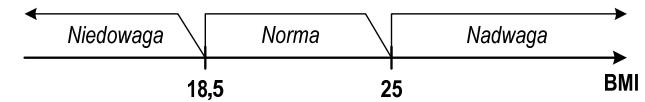
```
if ( x )
```

Język C - BMI

- BMI** - współczynnik powstały przez podzielenie **masy** ciała podanej w kilogramach przez **kwadrat wzrostu** podanego w metrach

$$BMI = \frac{masa}{wzrost^2}$$

- Dla osób dorosłych:
 - BMI < 18,5 - wskazuje na niedowagę
 - BMI ≥ 18,5 i BMI < 25 - wskazuje na prawidłową masę ciała
 - BMI ≥ 25 - wskazuje na nadwagę



Język C - BMI

```
#include <stdio.h>
```

```
int main(void)
```

```
{  
    double masa, wzrost, bmi;
```

```
    printf("Podaj mase [kg]: "); scanf("%lf",&masa);  
    printf("Podaj wzrost [m]: "); scanf("%lf",&wzrost);  
    bmi = masa / (wzrost*wzrost);  
    printf("bmi: %.2f\n",bmi);
```

```
    if (bmi<18.5)  
        printf("Niedowaga\n");  
    if (bmi>=18.5 && bmi<25)  
        printf("Norma\n");  
    if (bmi>=25)  
        printf("Nadwaga\n");
```

```
    return 0;
```

```
}
```

```
Podaj mase [kg]: 84  
Podaj wzrost [m]: 1.85  
bmi: 24.54  
Norma
```

Język C - BMI

- Zamiast trzech instrukcji if:

```
if (bmi<18.5)  
    printf("Niedowaga\n");  
if (bmi>=18.5 && bmi<25)  
    printf("Norma\n");  
if (bmi>=25)  
    printf("Nadwaga\n");
```

można zastosować tylko dwie:

```
if (bmi<18.5)  
    printf("Niedowaga\n");  
else  
    if (bmi<25)  
        printf("Norma\n");  
    else  
        printf("Nadwaga\n");
```


Koniec wykładu nr 4

Dziękuję za uwagę!