

# Informatyka 1 (EZ1E2008)

Politechnika Białostocka - Wydział Elektryczny  
Elektrotechnika, semestr II, studia niestacjonarne I stopnia  
Rok akademicki 2021/2022

## Wykład nr 3 (18.03.2022)

dr inż. Jarosław Forenc

## Plan wykładu nr 3

- Język C
  - instrukcja if, operatory relacyjne i logiczne, wyrażenia logiczne
  - operator warunkowy, instrukcja switch
- Informatyka i informacja, informacja analogowa i cyfrowa
- Systemy liczbowe
  - liczby i cyfry
  - systemy pozycyjne i niepozycyjne (rzymski)
  - konwersje między systemami liczbowymi
- Jednostki informacji cyfrowej
  - bit, bajt słowo

## Przykład: 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 <b>0</b> , a argument równy zero na wartość <b>1</b>
&&	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) (!!!)

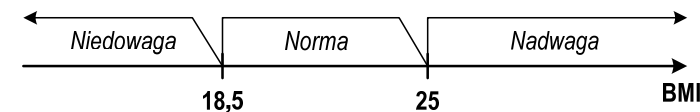
## Przykład: obliczanie BMI (Body Mass Index)

- 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



## Przykład: obliczanie BMI (Body Mass Index)

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

```
int main(void)
```

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

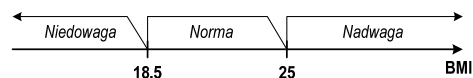
```
    printf("Podaj mase [kg]: "); scanf("%lf",&masa);  
    printf("Podaj wzrost [m]: "); scanf("%lf",&wzrost);  
    bmi = masa / (wzrost*wzrost);  
    printf("bmi: %.2f\n",bmi);
```

```
    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
```



## Przykład: obliczanie BMI (Body Mass Index)

- 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");
```

## Język C - Operator warunkowy

- Operator warunkowy składa się z dwóch symboli i trzech operandów

```
wyrażenie1 ? wyrażenie2 : wyrażenie3
```

- Najczęściej zastępuje proste instrukcje if-else

```
float akcyza, cena, pojemnosc;
```

```
if (pojemnosc <= 2000)  
    akcyza = cena*0.031; /* 3.1% */  
else  
    akcyza = cena*0.186; /* 18.6% */
```

```
akcyza = pojemnosc <= 2000 ? cena*0.031 : cena*0.186 ;
```

## Język C - Operator warunkowy

```
if (x < 0)  
    y = -x;  
else  
    y = x;
```

```
y = x < 0 ? -x : x;
```

- obliczenie modułu liczby x

```
if (a > b)  
    max = a;  
else  
    max = b;
```

```
max = a > b ? a : b;
```

- wyznaczenie max z dwóch liczb

- Operator warunkowy ma bardzo niski priorytet
- Niższy priorytet mają tylko operatory przypisania (=, +=, -=,...) i operator przecinkowy (,)

## Przykład: operator warunkowy

- Studenci chcą dojechać z akademika do sklepu - ile taksówek powinni zamówić? (Jedna taksówka może przewieźć 4 osoby.)

```
#include <stdio.h>

int main(void)
{
    int st, taxi;

    printf("Podaj liczbę studentów: ");
    scanf("%d", &st);

    taxi = st / 4 + (st % 4 ? 1 : 0);

    printf("Liczba taxi: %d\n", taxi);

    return 0;
}
```

Podaj liczbę studentów: 23  
Liczba taxi: 6

## Przykład: sprawdzenie parzystości liczby

```
#include <stdio.h>

int main(void)
{
    int x;

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

    if (x%2==0)
        printf("Liczba parzysta\n");
    else
        printf("Liczba nieparzysta\n");

    printf("Liczba %s\n", x%2==0 ? "parzysta":"nieparzysta");

    return 0;
}
```

Podaj x: -3  
Liczba nieparzysta  
Liczba nieparzysta

## Język C - Instrukcja switch

- Instrukcja wyboru wielowariantowego **switch**

```
switch (wyrażenie)
{
    case wyrażenie Stałe: instrukcje;
    case wyrażenie Stałe: instrukcje;
    case wyrażenie Stałe: instrukcje;
    ...
    default: instrukcje;
}
```

- **wyrażenie Stałe** - wartość typu całkowitego, znana podczas kompilacji
  - stała liczbowa, np. 3, 5, 9
  - znak w apostrofach, np. 'a', 'z', '+'
  - stała zdefiniowana przez **const** lub **#define**

## Język C - Instrukcja switch

- Program wyświetlający słownie liczbę z zakresu 1..5 wprowadzoną z klawiatury

```
#include <stdio.h>

int main(void)
{
    int liczba;

    printf("Podaj liczbę (1..5): ");
    scanf("%d", &liczba);
}
```

## Język C - Instrukcja switch

```
switch (liczba)
{
    case 1: printf("Liczba: jeden\n");
            break;
    case 2: printf("Liczba: dwa\n");
            break;
    case 3: printf("Liczba: trzy\n");
            break;
    case 4: printf("Liczba: cztery\n");
            break;
    case 5: printf("Liczba: piec\n");
            break;
    default: printf("Inna liczba\n");
}
```

Podaj liczbe: 2  
Liczba: dwa

Podaj liczbe: 0  
Inna liczba

## Język C - Instrukcja switch

```
switch (liczba)
{
    case 1:
    case 3:
    case 5: printf("Liczba nieparzysta\n");
            break;
    case 2:
    case 4: printf("Liczba parzysta\n");
            break;
    default: printf("Inna liczba\n");
}
```

Podaj liczbe: 2  
Liczba parzysta

- Te same instrukcje mogą być wykonane dla kilku etykiet `case`

## Język C - Instrukcja switch

```
switch (liczba)
{
    case 1: case 3: case 5:
            printf("Liczba nieparzysta\n");
            break;
    case 2: case 4:
            printf("Liczba parzysta\n");
            break;
    default: printf("Inna liczba\n");
}
```

Podaj liczbe: 2  
Liczba parzysta

- Etykiety `case` mogą być pisane w jednym wierszu

## Język C - Instrukcja switch

```
switch (liczba%2)
{
    case 1: case -1:
            printf("Liczba nieparzysta\n");
            break;
    case 0:
            printf("Liczba parzysta\n");
}
```

Podaj liczbe: 2  
Liczba parzysta

- Część domyślna (`default`) może być pominięta

## Język C - Instrukcja switch (bez break)

```
switch (liczba)
{
    case 1: printf("Liczba: jeden\n");
    case 2: printf("Liczba: dwa\n");
    case 3: printf("Liczba: trzy\n");
    case 4: printf("Liczba: cztery\n");
    case 5: printf("Liczba: piec\n");
    default: printf("Inna liczba\n");
}
```

Podaj liczbe: 2  
Liczba: dwa  
Liczba: trzy  
Liczba: cztery  
Liczba: piec  
Inna liczba

- Pominięcie instrukcji **break** spowoduje wykonanie wszystkich instrukcji występujących po danym **case** (do końca **switch**)

## Informacja

- **Informatyka** (ang. computer science)
  - dziedzina nauki i techniki zajmująca się gromadzeniem, przetwarzaniem i wykorzystywaniem **informacji**
- **Informacja** - wielkość abstrakcyjna, która może być:
  - przechowywana w pewnych obiektach
  - przesyłana pomiędzy pewnymi obiektami
  - przetwarzana w pewnych obiektach
  - stosowana do sterowania pewnymi obiektami
- **Dane** - surowe fakty i liczby
- **Przetwarzanie danych** - logicznie powiązany zespół czynności pozwalających na uzyskanie z danych niezbędnych informacji



## Informatyka

- **Informatyka** (ang. computer science)
  - dziedzina nauki i techniki zajmująca się gromadzeniem, przetwarzaniem i wykorzystywaniem **informacji**
  - w języku polskim termin informatyka zaproponował w październiku 1968 r. prof. Romuald Marczyński na konferencji poświęconej „maszynom matematycznym”
  - wzorem nazwy były francuskie **informatique** i niemieckie **Informatik**
- **Informatykę** można rozpatrywać jako:
  - samodzielną dyscyplinę naukową
  - narzędzie wykorzystywane przez inne nauki
  - gałąź techniki
  - przemysł wytwarzający sprzęt (hardware) i oprogramowanie (software)

## Informacja

- Co oznaczają poniższe dane?

0001010100001110001010000010000



00010101	00000111	00010100	00010000
----------	----------	----------	----------

Kod binarny?



0001	0101	0000	0111	0001	0100	0001	0000
------	------	------	------	------	------	------	------

A może BCD?



1	5	0	7	1	4	1	0
---	---	---	---	---	---	---	---

Liczba: **15 071 410** ?



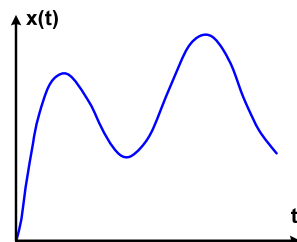
15 lipca 1410 roku

Data !!!

## Informacja analogowa i cyfrowa

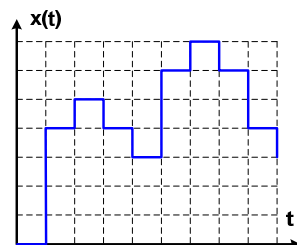
### ■ Sygnał analogowy

- może przyjmować dowolną wartość z ciągłego przedziału (nieskończonego lub ograniczonego zakresem zmienności)
- wartości mogą zostać określone w każdej chwili czasu dzięki funkcji matematycznej opisującej dany sygnał



### ■ Sygnał cyfrowy

- dziedzina i zbiór wartości są dyskretne
- sygnał ciągły, który może zmieniać swoją wartość tylko w określonych chwilach czasu i może przyjmować tylko określone wartości



## Liczby i cyfry

### ■ Liczba - pojęcie abstrakcyjne, abstrakcyjny wynik obliczeń, wartość

- umożliwia wyrażenie wyniku liczenia przedmiotów oraz mierzenia wielkości

### ■ Cyfra - umowny znak (symbol) stosowany do zapisu liczby

- liczba znaków służących do zapisu jest zależna od **systemu liczbowego** i przyjętego sposobu zapisu
- system dziesiętny - 10 znaków
- system szesnastkowy - 16 znaków
- system rzymski - 7 znaków

### ■ Cyfry rzymskie

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

## Informacja analogowa i cyfrowa

### ■ Zalety sygnałów cyfrowych:

- odporne na zakłócenia
- powtarzalne (np. kopia filmu na DVD i VHS)
- możliwość przesyłania na duże odległości
- możliwość szyfrowania sygnału (kryptografia)
- niższe koszty przetwarzania

### ■ Wady sygnałów cyfrowych:

- ograniczenie częstotliwości próbkowania (sygnał analogowy zamieniony na cyfrowy i ponownie na analogowy nie jest już tym samym sygnałem)

## Liczby i cyfry

### ■ Cyfry arabskie (pochodzą z Indii)

- arabskie, standardowe europejskie

1	2	3	4	5	6	7	8	9	0
---	---	---	---	---	---	---	---	---	---

- indyjsko-arabskie

१	२	३	४	५	६	७	८	९	०
1	2	3	4	5	6	7	8	9	0

- wschodnio-indyjsko-arabskie

١	٢	٣	٤	٥	٦	٧	٨	٩	٠
1	2	3	4	5	6	7	8	9	0

### ■ W niektórych systemach jako cyfry stosowane są litery, np.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



## Liczby i cyfry

- Inne przykłady zapisu cyfr i liczb:

cyfry etruskie

I	Λ	X	XX	ΛXX	↑	*	(C)	⊙	(Φ)
1	5	10	20	25	50	100	1000		

cyfry w pisowni chińskiej

jeden	一	sześć	六
dwa	二	siedem	七
trzy	三	osiem	八
cztery	四	dziewięć	九
pięć	五	dziesięć	十
zero	另		

[ucznieschinskiego.com](http://ucznieschinskiego.com)

cyfry grecko-jońskie

α	β	γ	δ	ε	ς	ζ	η	θ
1	2	3	4	5	6	7	8	9
ι	κ	λ	μ	ν	ξ	ο	π	ρ
10	20	30	40	50	60	70	80	90
σ	τ	υ	φ	χ	ψ	ω	Ͱ	ͱ
100	200	300	400	500	600	700	800	900
Ͳ	ͳ	ʹ	͵	Ͷ	ͷ	͸	͹	ͺ
1000	2000	3000	4000	5000	6000	7000	8000	9000

ͻͼͽͿ

## Liczby i cyfry

- Inne przykłady zapisu cyfr i liczb:

liczby w piśmie klinowym (Babilończycy)

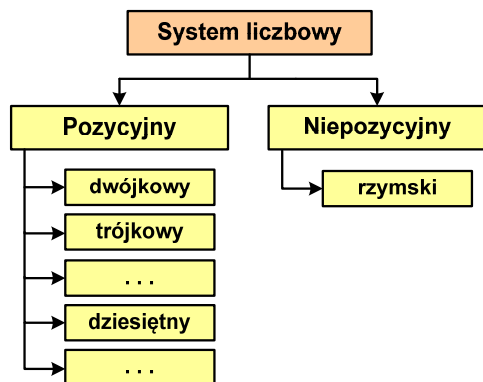
1	2	3	4	5	6	7	8	9
10	20	30	40	100	200	1000		

system prekolumbijski

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19

## 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

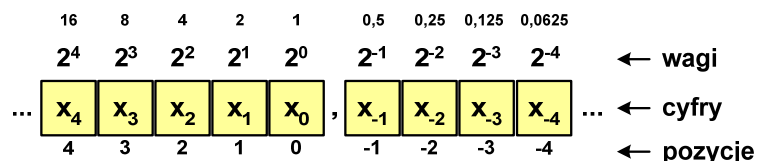
## System dziesiętny (ang. decimal)

10000	1000	100	10	1	0,1	0,01	0,001	0,0001		
$10^4$	$10^3$	$10^2$	$10^1$	$10^0$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	← wagi	
...	$x_4$	$x_3$	$x_2$	$x_1$	$x_0$	$x_{-1}$	$x_{-2}$	$x_{-3}$	$x_{-4}$	← cyfry
	4	3	2	1	0	-1	-2	-3	-4	← pozycje

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

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

## System dwójkowy (ang. binary)



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

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

## System 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

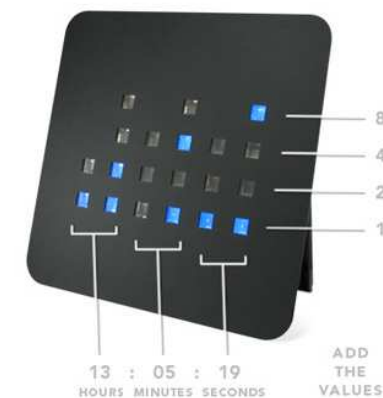
$$3A5D_{(16)} = 3 \cdot 16^3 + 10 \cdot 16^2 + 5 \cdot 16^1 + 13 \cdot 16^0 = 14941_{(10)}$$

- Sposoby zapisu liczb w systemie szesnastkowym:

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

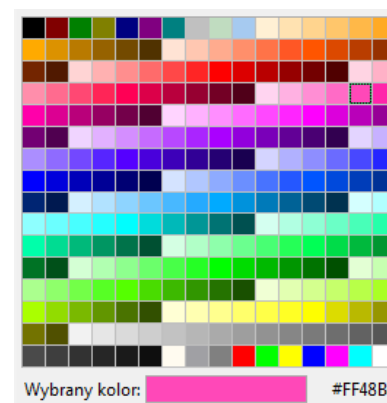
## System dwójkowy - zastosowania

- Powszechnie używany w informatyce, technice cyfrowej

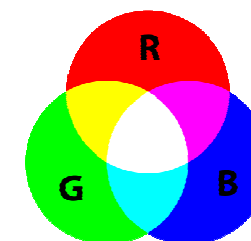


## System szesnastkowy - zastosowania

- 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>

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 (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 na system dziesiętny

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

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

$$\begin{matrix} 4^4 & 4^3 & 4^2 & 4^1 & 4^0 \\ 2 & 1 & 3 & 0 & 2 \end{matrix}$$

$$21302_{(4)} = 2 \cdot 4^0 + 0 \cdot 4^1 + 3 \cdot 4^2 + 1 \cdot 4^3 + 2 \cdot 4^4$$

$$21302_{(4)} = 2 \cdot 1 + 0 \cdot 4 + 3 \cdot 16 + 1 \cdot 64 + 2 \cdot 256$$

$$21302_{(4)} = 2 + 0 + 48 + 64 + 512 = 626_{(10)}$$

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

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

$$\begin{matrix} 17^3 & 17^2 & 17^1 & 17^0 \\ A & C & 2 & 4 \end{matrix}$$

$$AC24_{(17)} = 4 \cdot 17^0 + 2 \cdot 17^1 + 12 \cdot 17^2 + 10 \cdot 17^3$$

$$AC24_{(17)} = 4 \cdot 1 + 2 \cdot 17 + 12 \cdot 289 + 10 \cdot 4913$$

$$AC24_{(17)} = 4 + 34 + 3468 + 49130 = 52636_{(10)}$$

## Konwersja z systemu dziesiętnego na dowolny

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

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

$$626_{(10)} = 1001110010_{(2)}$$

$$626/2 = 313 \text{ reszta } 0$$

$$313/2 = 156 \text{ reszta } 1$$

$$156/2 = 78 \text{ reszta } 0$$

$$78/2 = 39 \text{ reszta } 0$$

$$39/2 = 19 \text{ reszta } 1$$

$$19/2 = 9 \text{ reszta } 1$$

$$9/2 = 4 \text{ reszta } 1$$

$$4/2 = 2 \text{ reszta } 0$$

$$2/2 = 1 \text{ reszta } 0$$

$$1/2 = 0 \text{ reszta } 1$$

kolejność odczytywania  
cyfr liczby w systemie  
dwójkowym

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	↑	→ A
44/14 = 3	reszta	2		
3/14 = 0	reszta	3		

## Jednostki informacji - bit

- Bit** (ang. **binary digit**) - podstawowa jednostka informacji stosowana w informatyce i telekomunikacji
- Określa najmniejszą ilość informacji potrzebną do stwierdzenia, który z dwóch możliwych stanów przyjął układ
- Bit przyjmuje jedną z dwóch wartości:
  - 0 (zero)
  - 1 (jeden)
- Bit jest tożsamy z cyfrą w systemie dwójkowym
- Oznaczenia bitów:
  - standard IEEE 1541 (2002) - mała litera „b”
  - standard IEC 60027 - „bit”

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

$2 \rightarrow 4$

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

01	10	11	00	11
1	2	3	0	3

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

$2 \rightarrow 8$

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

010	110	011
2	6	3

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

$2 \rightarrow 16$

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

0101	1010
5	A

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

$4 \rightarrow 2$

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

01	10	11	00	11
1	2	3	0	3

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

$8 \rightarrow 2$

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

010	110	011
2	6	3

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

$16 \rightarrow 2$

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

0101	1010
5	A

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

## Jednostki informacji - bit

- Wielokrotności bitów:

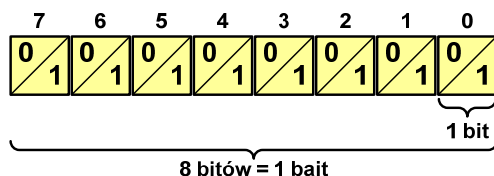
Przedrostki dziesiętne (układ SI)		
Nazwa	Symbol	Mnożnik
bit	b	---
kilobit	kb	$10^3 = 1000^1$
megabit	Mb	$10^6 = 1000^2$
gigabit	Gb	$10^9 = 1000^3$
terabit	Tb	$10^{12} = 1000^4$
petabit	Pb	$10^{15} = 1000^5$
eksabit	Eb	$10^{18} = 1000^6$
zettabit	Zb	$10^{21} = 1000^7$
jottabit	Yb	$10^{24} = 1000^8$

Przedrostki binarne (IEC 60027-2)		
Nazwa	Symbol	Mnożnik
bit	b	---
kibibit	Kib	$2^{10} = 1024^1$
mebibit	Mib	$2^{20} = 1024^2$
gibibit	Gib	$2^{30} = 1024^3$
tebibit	Tib	$2^{40} = 1024^4$
pebibit	Pib	$2^{50} = 1024^5$
eksbibit	Eib	$2^{60} = 1024^6$
zebibit	Zib	$2^{70} = 1024^7$
jobibit	Yib	$2^{80} = 1024^8$

- Przedrostki binarne** - wprowadzone w 1999 roku w celu odróżnienia przedrostków o mnożniku 1000 ( $10^3$ ) od przedrostków o mnożniku 1024 ( $2^{10}$ )

## Jednostki informacji - bajt

- **Bajt** (ang. byte) - najmniejsza adresowalna jednostka informacji pamięci komputerowej składająca się z bitów
- W praktyce przyjmuje się, że jeden bajt to 8 bitów

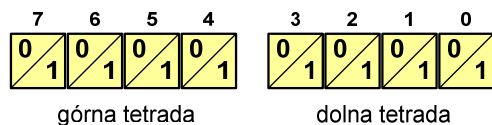


- Za pomocą jednego bajtu można zapisać  $2^8 = 256$  różnych wartości:

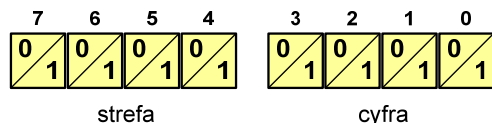
0000 0000	→	0	...	...
0000 0001	→	1	1111 1101	→ 253
0000 0010	→	2	1111 1110	→ 254
...	...	...	1111 1111	→ 255

## Jednostki informacji - tetrada

- Bajt 8-bitowy można podzielić na dwie połówki 4-bitowe nazywane **tetradami** (ang. nibbles)
- Rozróżniamy bardziej znaczącą (górną) i mniej znaczącą (dolną) tetradę



- Spotyka się też określenie **strefa** i **cyfra**



## Jednostki informacji - bajt

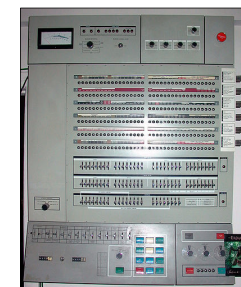
- Wielokrotności bajtów:

Przedrostki dziesiętne (układ SI)		
Nazwa	Symbol	Mnożnik
bajt	B	---
kilobajt	kB	$10^3 = 1000^1$
megabajt	MB	$10^6 = 1000^2$
gigabajt	GB	$10^9 = 1000^3$
terabajt	TB	$10^{12} = 1000^4$
petabajt	PB	$10^{15} = 1000^5$
eksabajt	EB	$10^{18} = 1000^6$
zettabajt	ZB	$10^{21} = 1000^7$
jottabajt	YB	$10^{24} = 1000^8$

Przedrostki binarne (IEC 60027-2)		
Nazwa	Symbol	Mnożnik
bajt	B	---
kibibajt	KiB	$2^{10} = 1024^1$
mebibajt	MiB	$2^{20} = 1024^2$
gibibajt	GiB	$2^{30} = 1024^3$
tebibajt	TiB	$2^{40} = 1024^4$
pebibajt	PiB	$2^{50} = 1024^5$
eksbibajt	EiB	$2^{60} = 1024^6$
zebibajt	ZiB	$2^{70} = 1024^7$
jobibajt	YiB	$2^{80} = 1024^8$

## Jednostki informacji - bajt

- W pierwszych komputerach bajt mógł mieć inną liczbę bitów: 4, 6, 7, 9, 12
- 8-bitowy bajt:
  - koniec 1956 r. - pierwsze zastosowanie
  - 1964 r. - uznanie za standard (IBM System/360)
- Inna nazwa 8-bitowego bajtu - **oktet**
- Najczęściej stosowanym skrótem dla bajtu jest wielka litera „B”
  - „B” używane jest także do oznaczania **bela** - jednostki miary wielkości ilorazowych
  - zamiast bela częściej używa się jednostki podwielokrotnej - **decybela (dB)** więc nie ma problemu z rozróżnieniem obu jednostek



## Jednostki informacji - bajt

- Przedrostki binarne (dwójkowe) nie zostały przyjęte przez wszystkie środowiska zajmujące się informatyką
- Producenci nośników pamięci korzystają z przedrostków dziesiętnych

Prefiks	Nazwa	System SI	System binarny	Różnica
k	kilo	$10^3 = 1000$	$2^{10} = 1024$	2,40%
M	mega	$10^6 = 1\,000\,000$	$2^{20} = 1\,048\,576$	4,86%
G	giga	$10^9 = 1\,000\,000\,000$	$2^{30} = 1\,073\,741\,824$	7,37%
T	tera	$10^{12} = 1\,000\,000\,000\,000$	$2^{40} = 1\,099\,511\,627\,776$	9,95%

- Z ulotki „Dysk Desktop HDD - zestawienie danych”, Seagate:
  - w przypadku oznaczania pojemności dysków, jeden gigabajt (oznaczany także jako „GB”) jest równy jednemu miliardowi bajtów, a jeden terabajt (oznaczany także jako „TB”) jest równy jednemu bilionowi bajtów

## Słowo maszynowe (słowo)

- **Słowo maszynowe** (słowo - ang. word) - jednostka danych używana przez określony komputer (określoną architekturę)
- Słowo składa się odgórnie określonej liczby bitów, nazywanej **długością** lub **szerokością słowa** (najczęściej jest to potęga 2, np. 8, 16, 32, 64 bity)
- Zazwyczaj wielkość słowa określa:
  - rozmiar rejestrów procesora
  - rozmiar szyny danych i szyny adresowej
- Architektury:
  - 8-bitowa: Intel 8080, Z80, Motorola 6800, Intel 8051
  - 16-bitowa: Intel 8086, Intel 80286
  - 32-bitowa: Intel od 80386 do i7, AMD od 5x86 do Athlona, ARM
  - 64-bitowa: Intel Itanium, Pentium 4/EM64T, Core 2, Core i7  
AMD Opteron, Athlon 64, Athlon II

## Jednostki informacji - bajt

- Seagate ST1000DM003 (1 TB)
- Drive specification:
  - formatted capacity: 1000 GB (1 TB)
  - guaranteed sectors: 1,953,525,168
  - bytes per sector: 4096  
(4K physical emulated at 512-byte sectors)
- Pojemność dysku:
  - $1.953.525.168 \times 512 = 1.000.204.886.016$  bajtów
  - $1.000.204.886.016 / (1024) = 976.762.584$  kB
  - $1.000.204.886.016 / (1024 \times 1024) = 953.870$  MB
  - $1.000.204.886.016 / (1024 \times 1024 \times 1024) = 931,5$  GB



## Koniec wykładu nr 3

Dziękuję za uwagę!