

# Technologie informacyjne

---

Politechnika Białostocka - Wydział Elektryczny  
semestr I, studia stacjonarne I stopnia  
Rok akademicki 2017/2018

## Pracownia nr 2

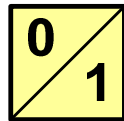
dr inż. Jarosław Forenc

## Pracownia nr 2

- Jednostki informacji
  - bit, bajt
- Kodowanie
  - znaków: ASCII, ISO 8859, Unicode
  - liczb: NKB (BCN), U2, BCD
- Liczby zmiennoprzecinkowe
  - standard IEEE 754

## Jednostki informacji - bit

- **Bit** (ang. **b**inary digit) - podstawowa jednostka informacji stosowana w informatyce i telekomunikacji
- Bit przyjmuje jedną z dwóch wartości:
  - 0 (zero)
  - 1 (jeden)
- Bit jest tożsamy z cyfrą w systemie dwójkowym
- Oznaczenia bitów:
  - „**b**” - zalecenie standardu IEEE 1541 z 2002 roku
  - „**bit**” - zalecenie standardu IEC 60027



## Jednostki informacji - bit

- Wielokrotności bitów:

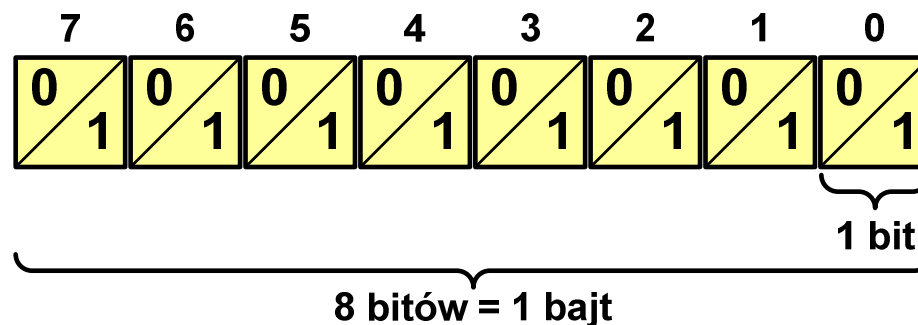
Przedrostki dziesiętne (układ SI)		
Nazwa	Symbol	Mnożnik
bit	b	---
kilobit	kb	$10^3 = 1000^1$
megabit	Mb	$10^6 = 1000^2$
gigabit	Gb	$10^9 = 1000^3$
terabit	Tb	$10^{12} = 1000^4$
petabit	Pb	$10^{15} = 1000^5$
eksabit	Eb	$10^{18} = 1000^6$
zettabit	Zb	$10^{21} = 1000^7$
jottabit	Yb	$10^{24} = 1000^8$

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** zostały wprowadzone w 1998 roku w celu odróżnienia przedrostków o mnożniku 1000 od przedrostków o mnożniku 1024

## 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 (choć nie wynika to z powyższej definicji)

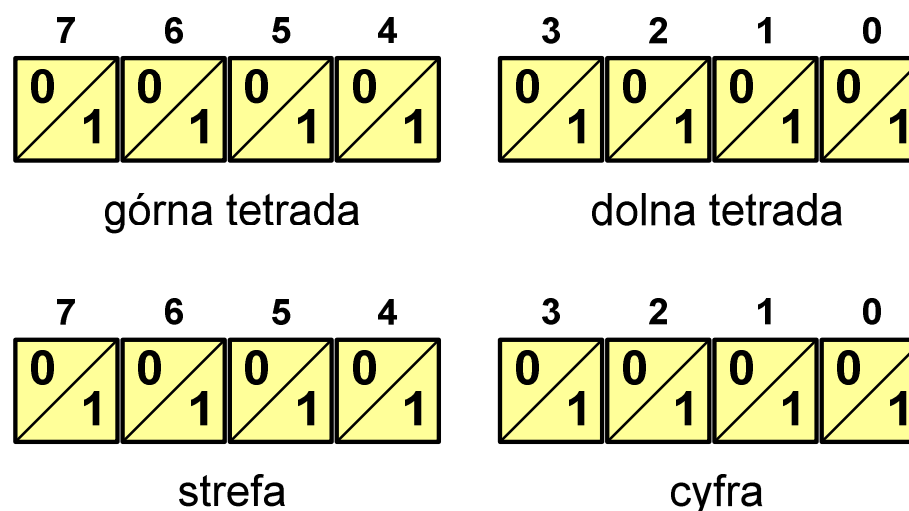


- Za pomocą jednego bajtu można zapisać **256** różnych wartości:

0000 0000 → 0  
0000 0001 → 1  
...  
1111 1110 → 254  
1111 1111 → 255

## Jednostki informacji - bajt

- 8-bitowy bajt nazywany jest także **oktetem**
- Bajt można podzielić na dwie połówki 4-bitowe nazywane **tetradami** (ang. nibbles)



- Najczęściej stosowanym skrótem dla bajtu jest wielka litera „**B**” (uwaga: **B** oznacza też **bela**, ale częściej używa się **dB** - **decybel**)

## 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	$2^{80} = 1024^8$

- Przedrostki binarne (dwójkowe) nie zostały przyjęte przez wszystkie środowiska zajmujące się informatyką

## Jednostki informacji - przykłady

- Stosujemy mnożnik używany w informatyce (1024)

$$\text{kB} \rightarrow \text{B:} \quad 1 \text{ kB} = 1 \, 024 \text{ B}$$

$$\text{MB} \rightarrow \text{B:} \quad 1 \text{ MB} = 1024 \times 1024 = 1 \, 048 \, 576 \text{ B}$$

$$\text{GB} \rightarrow \text{B:} \quad 1 \text{ GB} = 1024 \times 1024 \times 1024 = 1 \, 073 \, 741 \, 824 \text{ B}$$

$$\text{MB} \rightarrow \text{kB:} \quad 2 \text{ MB} = 2 \times 1024 = 2 \, 048 \text{ kB}$$

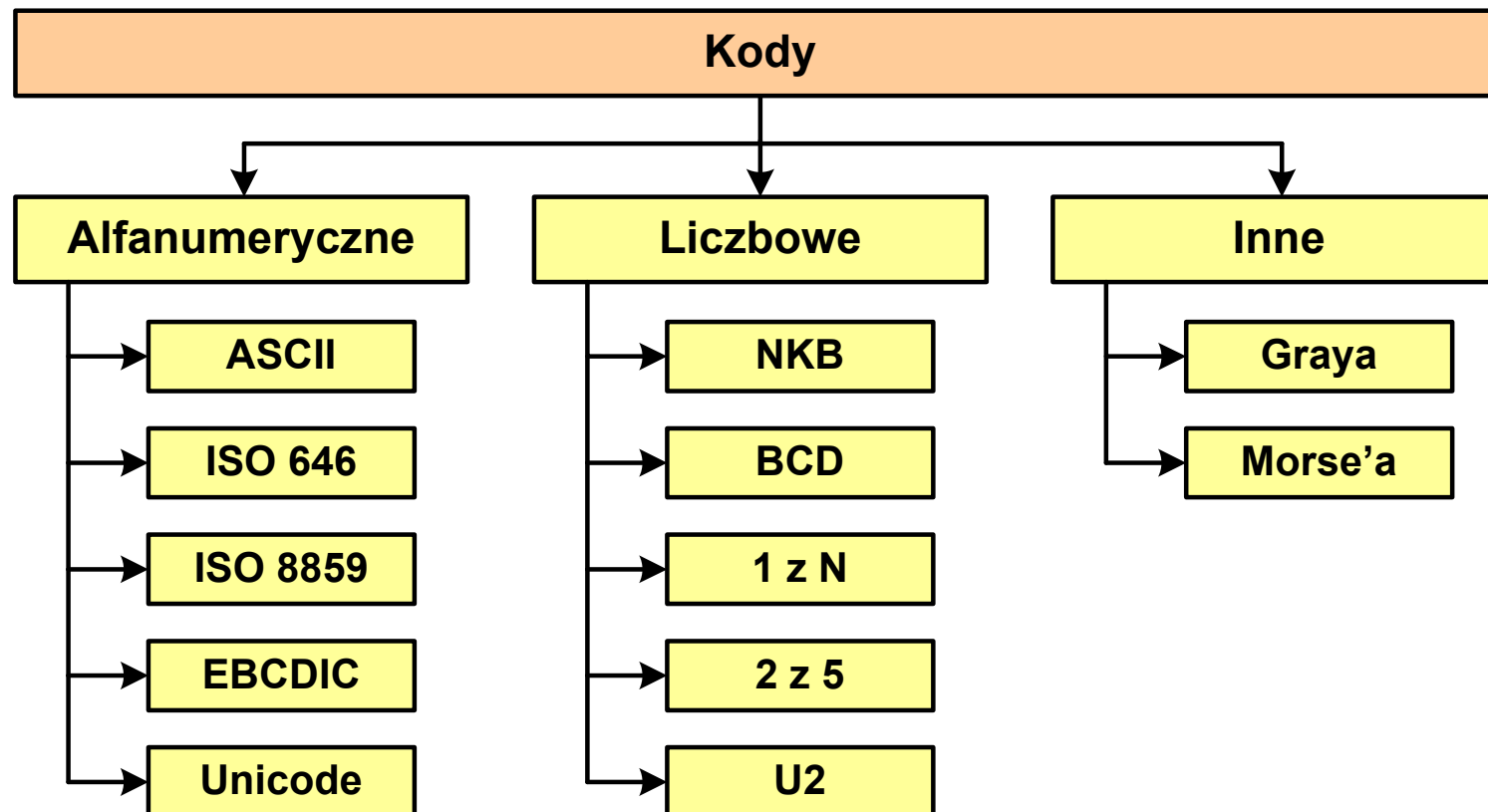
$$\text{GB} \rightarrow \text{kB:} \quad 3 \text{ GB} = 3 \times 1024 \times 1024 = 3 \, 145 \, 728 \text{ kB}$$

$$\text{TB} \rightarrow \text{kB:} \quad 4 \text{ TB} = 4 \times 1024 \times 1024 \times 1024 = 4 \, 294 \, 967 \, 296 \text{ kB}$$



# Kodowanie

- **Kodowanie** - proces przekształcania jednego rodzaju postaci informacji na inną postać



# ASCII

## ■ ASCII - American Standard Code for Information Interchange

- 7-bitowy kod przypisujący liczby z zakresu 0-127:
  - literom (alfabet angielski)
  - cyfrom
  - znakom przestankowym
  - innym symbolom
  - poleceniom sterującym
- kody 0-31, 127 - 33 kody sterujące służące do sterowania urządzeniami typu drukarka czy terminal
- kody 32-126 - 95 kodów tworzących zbiór znaków ASCII

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	NUL	32	20	Space	64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	\	71	47	G	103	67	g
8	8	BS	40	28	(	72	48	H	104	68	h
9	9	TAB	41	29	)	73	49	I	105	69	i
10	A	LF	42	2A	*	74	4A	J	106	6A	j
11	B	VT	43	2B	+	75	4B	K	107	6B	k
12	C	FF	44	2C	,	76	4C	L	108	6C	l
13	D	CR	45	2D	-	77	4D	M	109	6D	m
14	E	SO	46	2E	.	78	4E	N	110	6E	n
15	F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[	123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D	]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

## ISO/IEC 8859

- **ISO/IEC 8859** to zestaw standardów służących do kodowania znaków za pomocą 8 bitów
- Wszystkie zestawy ISO 8859 mają znaki 0-127 takie same jak ASCII, zaś pozycjom 128-159 przypisane są dodatkowe kody sterujące, tzw. C1 (nieużywane)
  - ISO 8859-1 (Latin-1) - alfabet łaciński dla Europy zachodniej
  - **ISO 8859-2 (Latin-2)** - łaciński dla Europy środkowej i wschodniej (Polska Norma)
  - ISO 8859-3 (Latin-3) - łaciński dla Europy południowej
  - ISO 8859-4 (Latin-4) - łaciński dla Europy północnej
  - ISO 8859-5 (Cyrillic) - dla cyrylicy
  - ISO 8859-6 (Arabic) - dla alfabetu arabskiego
  - ISO 8859-7 (Greek) - dla alfabetu greckiego
  - ISO 8859-8 (Hebrew) - dla alfabetu hebrajskiego
  - ...

## ISO/IEC 8859-2

- ISO/IEC 8859-2, Latin-2 („środkowo”, „wschodnioeuropejskie”)

- przykład:

„A” - kod:  $41_{(16)} = 40_{(16)} + 1_{(16)}$

„Ę” - kod:  $CA_{(16)} = C0_{(16)} + A_{(16)}$

- dostępne języki: bośniacki, chorwacki, czeski, węgierski, polski, rumuński, serbski, serbsko-chorwacki, słowacki, słoweński, górno- i dolnołużycki
- możliwość przedstawienia znaków w języku niemieckim i angielskim
- 191 znaków łacińskiego pisma
- kodowanie zgodne z **Polską Normą**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	<i>Znaki kontrolne</i>															
10																
20	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	<i>Nie używane</i>															
90																
A0	NB SP	Ą	ˆ	Ł	ł	Ĺ	ĺ	Ś	ś	˝	ŝ	ŝ	Ť	ť	Ž	ž
B0	°	ą	ˆ	ł	ł	ĺ	ĺ	ś	ś	˝	ŝ	ŝ	ť	ť	ž	ž
C0	Ř	Á	Â	Ă	Ä	Í	Č	Ç	Č	É	Ę	Ě	Ě	Í	Î	Ď
D0	Đ	Ń	Ň	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Ů	Ü	Ý	Ť	ß
E0	ř	á	â	ă	ä	í	č	ç	č	é	ę	ě	ě	í	î	ď
F0	đ	ń	ň	ó	ô	õ	ö	÷	ř	ů	ú	ů	ü	ý	ť	·

SP - spacja

NBSP - twarda spacja

SHY - miękki dywiz (myślnik)

# Unicode



- **Unicode** - komputerowy zestaw znaków mający obejmować wszystkie pisma i inne znaki (symbole muzyczne, techniczne, wymowy) używane na świecie
- Unicode przypisuje unikalny numer każdemu znakowi, niezależny od używanej platformy, programu czy języka
- Konsorcjum: <http://www.unicode.org>
- Pierwsza wersja: **Unicode 1.0** (październik 1991)
- Ostatnia wersja: **Unicode 10.0.0** (20 czerwca 2017)
  - The Unicode Consortium. The Unicode Standard, Version 10.0.0, (Mountain View, CA: The Unicode Consortium, 2017)
  - <https://www.unicode.org/versions/Unicode10.0.0/>
  - Koduje 136.690 znaków



## Unicode - Zakresy

<u>Zakres:</u>	<u>Znaczenie:</u>
0000 - 007F	Basic Latin (to samo co w ASCII)
0080 - 00FF	Latin-1 Supplement (to samo co w ISO/IEC 8859-1)
0100 - 017F	Latin Extended-A
0180 - 024F	Latin Extended-B
0250 - 02AF	IPA Extensions
02B0 - 02FF	Spacing Modifiers Letters
...	
0370 - 03FF	Greek
0400 - 04FF	Cyrillic
...	
1D00 - 1D7F	Phonetic Extensions
1D80 - 1DBF	Phonetic Extensions Supplement
1E00 - 1EFF	Latin Extended Additional
1F00 - 1FFF	Greek Extended
...	



# Unicode

- Istnieją trzy metody kodowania (zapisu binarnego) różniące się liczbą bajtów przeznaczonych do opisanego kodu znaku

A	Ω	語	卍	UTF-32
00000041	000003A9	00008A9E	00010384	
A	Ω	語	卍	UTF-16
0041	03A9	8A9E	D800 DF84	
A	Ω	語	卍	UTF-8
41	CE A9	E8 AA 9E	F0 90 8E 84	



# Unicode

	010	011	012	013	014	015	016	017
0	Ā 0100	Ď 0110	Ġ 0120	İ 0130	ı 0140	Ō 0150	Š 0160	Ū 0170
1	ā 0101	ď 0111	ġ 0121	ı 0131	ł 0141	ó 0151	š 0161	ū 0171
2	Ǻ 0102	Ē 0112	Ɔ 0122	IJ 0132	ł 0142	Œ 0152	Ț 0162	Ț 0172
3	ǻ 0103	ē 0113	ǧ 0123	ij 0133	Ń 0143	œ 0153	ț 0163	ȕ 0173
4	Ą 0104	Ě 0114	Ĥ 0124	Ĵ 0134	ń 0144	Ŕ 0154	Ť 0164	Ŵ 0174
5	ą 0105	ě 0115	ĥ 0125	ĵ 0135	Ņ 0145	ŕ 0155	ť 0165	ŵ 0175
6	Ć 0106	Ė 0116	Ħ 0126	Ɔ 0136	ņ 0146	Ŗ 0156	Ʀ 0166	Ŷ 0176
7	ć 0107	ė 0117	ħ 0127	Ɔ 0137	Ń 0147	ŗ 0157	ƥ 0167	ŷ 0177

## European Latin

- 0100 Ā LATIN CAPITAL LETTER A WITH MACRON  
≡ 0041 A 0304 ̄
- 0101 ā LATIN SMALL LETTER A WITH MACRON  
• Latvian, Latin, ...  
≡ 0061 a 0304 ̄
- 0102 Ă LATIN CAPITAL LETTER A WITH BREVE  
≡ 0041 A 0306 ̂
- 0103 ă LATIN SMALL LETTER A WITH BREVE  
• Romanian, Vietnamese, Latin, ...  
≡ 0061 a 0306 ̂
- 0104Ą LATIN CAPITAL LETTER A WITH OGONEK  
≡ 0041 A 0328 ̇
- 0105ą LATIN SMALL LETTER A WITH OGONEK  
• Polish, Lithuanian, ...  
≡ 0061 a 0328 ̇
- 0106 Ć LATIN CAPITAL LETTER C WITH ACUTE  
≡ 0043 C 0301 ́
- 0107 ć LATIN SMALL LETTER C WITH ACUTE  
• Polish, Croatian, ...  
→ 045B ģ cyrillic small letter tshe  
≡ 0063 c 0301 ́



## Kody alfanumeryczne - przykład

- Tekst: **Sala WE-110**

S	a	l	a		W	E	-	1	1	0
---	---	---	---	--	---	---	---	---	---	---

ASCII	53	61	6c	61	20	57	45	2d	31	31	30
ISO 8859-2	53	61	6c	61	20	57	45	2d	31	31	30
Unicode	53	61	6c	61	20	57	45	2d	31	31	30

system  
szesnastkowy

ASCII	83	97	108	97	32	87	69	45	49	49	48
ISO 8859-2	83	97	108	97	32	87	69	45	49	49	48
Unicode	83	97	108	97	32	87	69	45	49	49	48

system  
dziesiętny

## Kody alfanumeryczne - przykład

- Tekst: **Żółty wąż**

Ż	ó	ł	t	y		w	ą	ż
---	---	---	---	---	--	---	---	---

ASCII	×	×	×	74	79	20	77	×	×
ISO 8859-2	AF	F3	B3	74	79	20	77	B1	BF
Unicode	17B	F3	142	74	79	20	77	105	17C

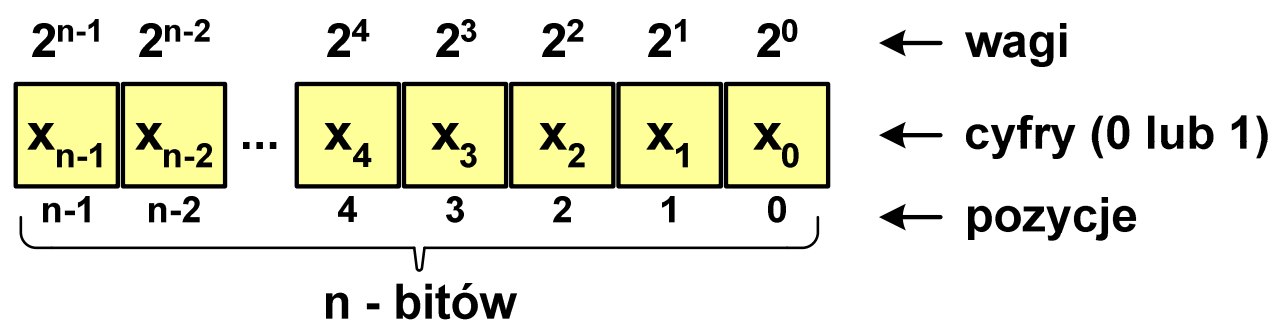
system  
szesnastkowy

ASCII	×	×	×	116	121	32	119	×	×
ISO 8859-2	175	243	179	116	121	32	119	177	191
Unicode	379	243	322	116	121	32	119	261	380

system  
dziesiętny

## NKB (BCN) - liczby całkowite bez znaku

- Zapis liczby w systemie dwójkowym:



- Używając **n-bitów** można zapisać liczbę z zakresu:

$$X_{(2)} = \langle 0, 2^n - 1 \rangle$$

8-bitów 0 ... 255

16-bitów 0 ... 65 535

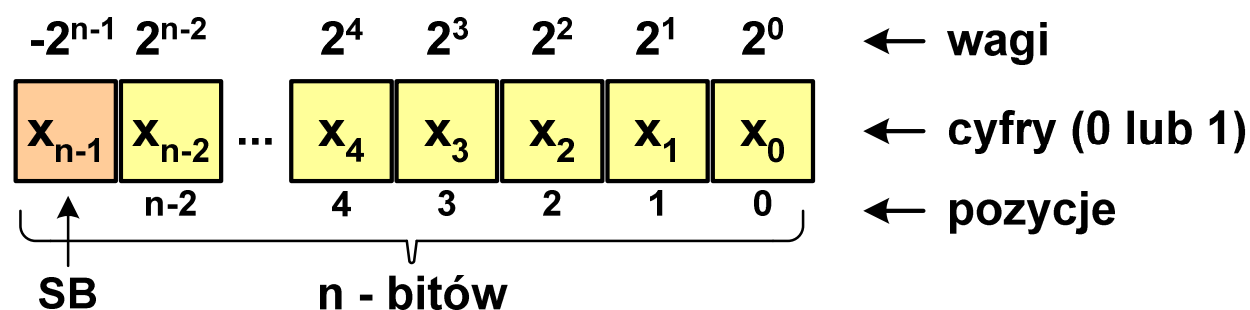
32-bity 0 ... 4 294 967 295

64-bity 0 ... 18 446 744 073 709 551 615

18 trylionów 446 miliardów 744 biliony 73 miliardy 709 milionów 551 tysięcy 615

## U2 - liczby całkowite ze znakiem

- ZU2, uzupełnień do dwóch, two's complement



- Najstarszy bit jest bitem znaku liczby: 0 - dodatnia, 1 - ujemna
- Wartość liczby:

$$X_{(10)} = x_0 \cdot 2^0 + x_1 \cdot 2^1 + x_2 \cdot 2^2 + \dots + x_{n-2} \cdot 2^{n-2} + x_{n-1} \cdot (-2^{n-1})$$

## U2 - zamiana na liczbę przeciwną

- **Krok 1:** inwersja (negacja) wszystkich bitów modułu liczby, tj. zamiana 0 na 1 i 1 na 0
- **Krok 2:** zwiększenie wyniku o 1

$$-75_{(10)} = ?_{(U2)}$$

$75_{(10)}$ 

0	1	0	0	1	0	1	1
---	---	---	---	---	---	---	---

negacja

1	0	1	1	0	1	0	0
---	---	---	---	---	---	---	---

+1

1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

 $(U2)$

$$-75_{(10)} = 10110101_{(U2)}$$

$$10110101_{(U2)} = ?_{(10)}$$

1	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---

negacja

0	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---

+1

0	1	0	0	1	0	1	1
---	---	---	---	---	---	---	---

 $75_{(10)}$

$$10110101_{(U2)} = -75_{(10)}$$

## BCD

- **B**inary-**C**oded **D**ecimal - dziesiętny zakodowany dwójkowo
- **BCD** - sposób zapisu liczb polegający na zakodowaniu kolejnych cyfr liczby dziesiętnej w 4-bitowym systemie dwójkowym (NKB)
- Istnieje kilka wariantów kodu BCD

NKB	BCD 8421	Excess-3	BCD 2421
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0010
3	0011	0110	0011
4	0100	0111	0100
5	0101	1000	1011
6	0110	1001	1100
7	0111	1010	1101
8	1000	1011	1110
9	1001	1100	1111

$$168_{(10)} = ?_{(BCD)}$$

$$\underbrace{0001}_1 \quad \underbrace{0110}_6 \quad \underbrace{1000}_8$$

$$168_{(10)} = 000101101000_{(BCD)}$$

$$1001 | 0101 | 0011_{(BCD)} = ?_{(10)}$$

$$\underbrace{1001}_9 \quad \underbrace{0101}_5 \quad \underbrace{0011}_3$$

$$100101010011_{(BCD)} = 953_{(10)}$$

## Zapis zmiennoprzecinkowy liczby rzeczywistej

- Postać zmiennoprzecinkowa umożliwia zapis bardzo dużych lub bardzo małych liczb w prostszej i wygodniejszej formie

- $12\ 000\ 000\ 000\ 000 = 1,2 \cdot 10^{13}$
- $-30\ 000\ 000\ 000\ 000\ 000\ 000 = -3,0 \cdot 10^{19}$
- $0,000\ 000\ 000\ 001 = 1,0 \cdot 10^{-12}$

- Zapis liczby zmiennoprzecinkowej ma postać

$$L = (-1)^S \cdot M \cdot B^E$$

gdzie:

- S** - znak liczby (ang. sign), przyjmuje wartość 0 lub 1
- M** - mantysa (ang. mantissa), liczba ułamkowa
- B** - podstawa systemu liczbowego (ang. base)
- E** - wykładnik (ang. exponent), cecha, liczba całkowita

## Postać znormalizowana zapisu liczby

- Tę samą liczbę można zapisać w różnych sposób

$$243 \cdot 10^1 = 24,3 \cdot 10^2 = 2,43 \cdot 10^3 = 0,243 \cdot 10^4$$

- W postaci znormalizowanej mantysa spełnia nierówność:

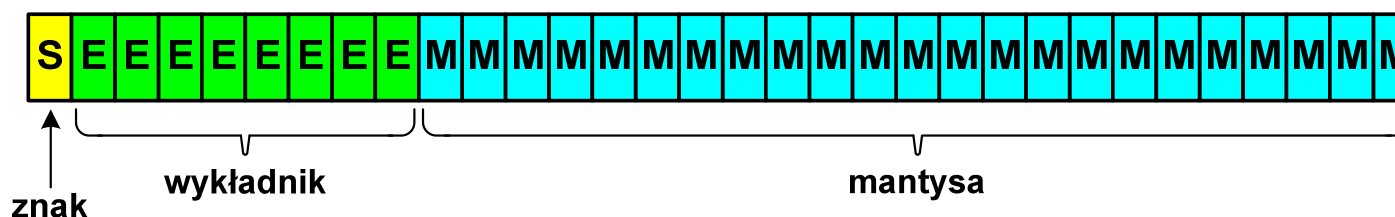
$$B > |M| \geq 1$$

- $2,43 \cdot 10^3$  - to jest postać znormalizowana, gdyż:  $10 > |2,43| \geq 1$
- $0,243 \cdot 10^4$  - to nie jest postać znormalizowana
- $24,3 \cdot 10^2$  - to nie jest postać znormalizowana



## Liczby zmiennoprzecinkowe w systemie binarnym

- Liczba bitów przeznaczonych na mantysę i wykładnik jest ograniczona



- W systemie binarnym podstawa systemu jest stała:  $B = 2$

$$L = (-1)^S \cdot M \cdot 2^E$$

- Wykładnik jest zapisywany jako wartość przesunięta o pewną stałą (ang. **biased exponent**) - zapis z przesuniętym wykładnikiem

$$L = (-1)^S \cdot M \cdot 2^{E-\text{BIAS}}$$

- Wartości przesunięcia: **127** (format 32-bit.), **1023** (format 64-bit.)

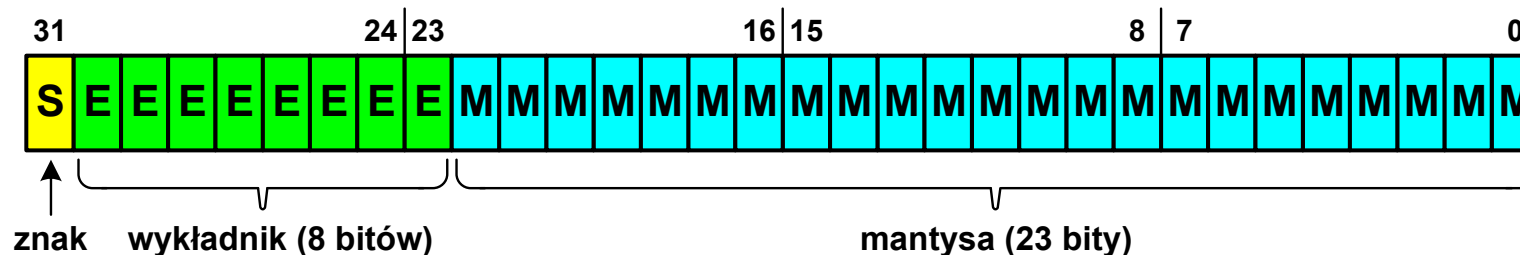
## Standard IEEE 754

- Standard opracowany w celu ujednoczenia operacji na liczbach zmiennoprzecinkowych na różnych platformach sprzętowych
- IEEE Std. 754-2008** - IEEE Standard for Floating-Point Arithmetic

Precyzja	Długość słowa [bity]	Znak [bity]	Wykładnik		Mantysa	
			Długość [bity]	Zakres	Długość [bity]	Cyfry znaczące
Pojedyncza (Single Precision, binary32)	32	1	8	$2^{\pm 127} \approx 10^{\pm 38}$	23	7
Pojedyncza rozszerzona (Single Extended)	$\geq 43$	1	$\geq 11$	$\geq 2^{\pm 1023} \approx 10^{\pm 308}$	$\geq 31$	$\geq 10$
Podwójna (Double Precision, binary64)	64	1	11	$2^{\pm 1023} \approx 10^{\pm 308}$	52	16
Podwójna rozszerzona (Double Extended)	$\geq 79$	1	$\geq 15$	$\geq 2^{\pm 16383} \approx 10^{\pm 4932}$	$\geq 63$	$\geq 19$

## Standard IEEE 754 - liczby 32-bitowe

- Liczba pojedynczej precyzji przechowywana jest na 32 bitach:



- **Bit znaku:** 0 - liczba dodatnia, 1 - liczba ujemna
- **Wykładnik** zapisywany jest na z nadmiarem o wartości 127 i przyjmuje wartości od -127 do 128
- **Mantysa** w większości przypadków jest znormalizowana
- Mantysa zawiera się w przedziale 1 i 2, jej pierwszy bit jest zawsze równy 1 i nie jest zapamiętywany
- Bit ten jest automatycznie uwzględniany podczas wykonywania obliczeń

## Standard IEEE 754 - liczby 32-bitowe

### ■ Przykład:

- obliczmy wartość dziesiętną liczby zmiennoprzecinkowej

$$01000010110010000000000000000000_{(IEEE754)} = ?_{(10)}$$

- dzielimy liczbę na części

$$\underbrace{0}_{S\text{-bit znaku}} \quad \underbrace{10000101}_{E\text{-wykładnik}} \quad \underbrace{10010000000000000000000000000000}_{M\text{-mantysa (tylko część ułamkowa)}}$$

- określamy **znak liczby**

$$S = 0 \quad \text{– liczba dodatnia}$$

- obliczamy **wykładnik** pamiętając, że w reprezentacji 32-bitowej nadmiar wynosi 127

$$E = 10000101_{(2)} = 128 + 4 + 1 = 133 - \underbrace{127}_{\text{nadmiar}} = 6_{(10)}$$

## Standard IEEE 754 - liczby 32-bitowe

### ■ Przykład (cd.):

- wyznaczamy **mantysę** dopisując na początku 1 (1 - część całkowita) i stawiając przecinek

$$\begin{aligned} M &= 1,100100000000000000000000 = \\ &= 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-4} = 1 + 0,5 + 0,0625 = 1,5625_{(10)} \end{aligned}$$

- wartość dziesiętną liczby zmiennoprzecinkowej obliczamy według wzoru:

$$L = (-1)^S \cdot M \cdot 2^E$$

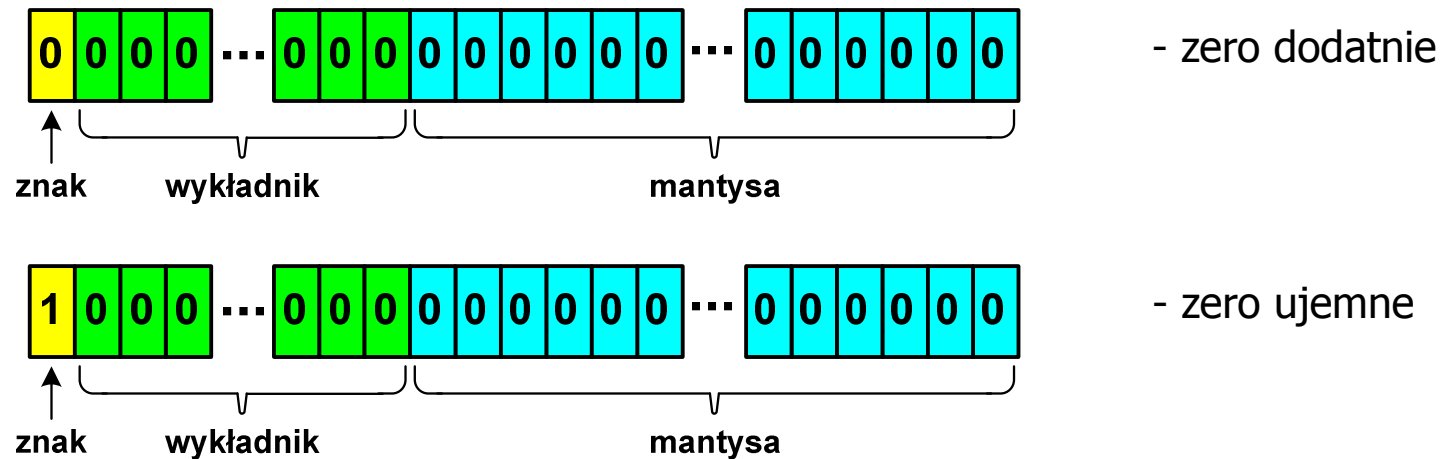
- podstawiając otrzymujemy:

$$S = 0, \quad E = 6_{(10)}, \quad M = 1,5625_{(10)}$$

$$L = (-1)^0 \cdot 1,5625 \cdot 2^6 = 100_{(10)}$$

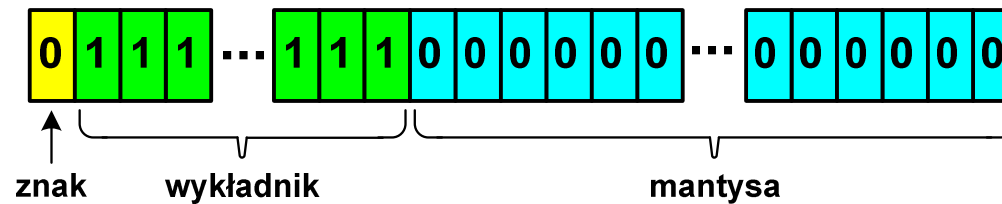
$$01000010110010000000000000000000_{(IEEE754)} = 100_{(10)}$$

## Standard IEEE 754 - zero

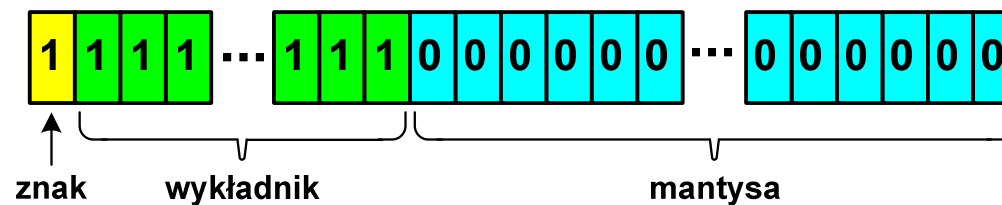


- bit znaku może przyjmować dowolną wartość
- przy porównaniach zero dodatnie i ujemne są traktowane jako równe sobie

## Standard IEEE 754 - nieskończoność



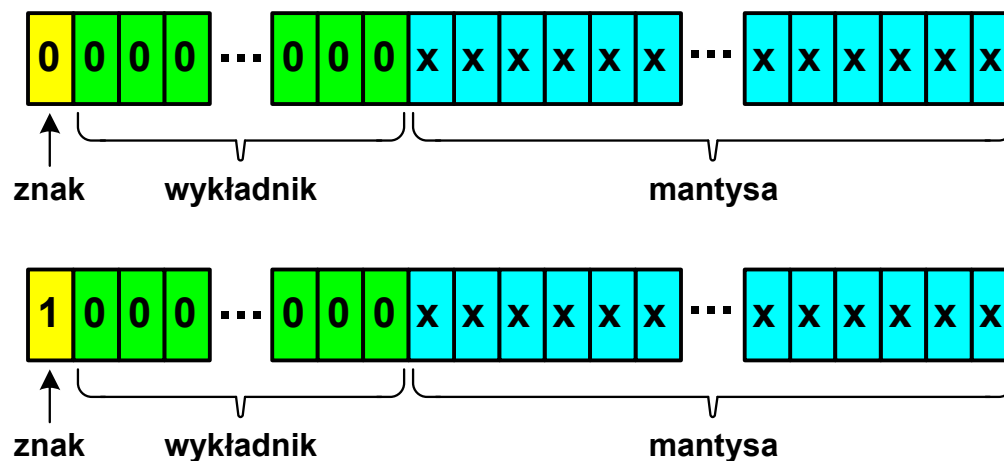
- nieskończoność dodatnia



- nieskończoność ujemna

- bit znaku określa czy mamy nieskończoność dodatnią czy ujemną
- nieskończoność występuje w przypadku wystąpienia **nadmiaru** (przepełnienia) oraz przy dzieleniu przez zero

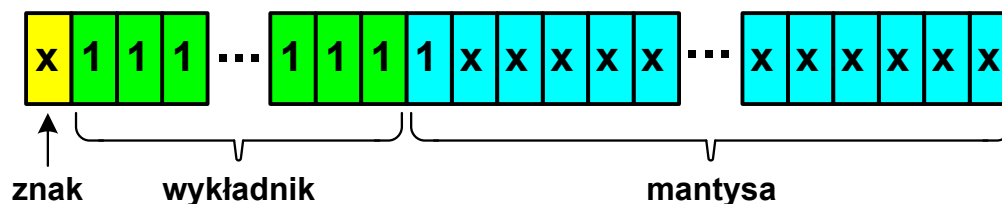
## Standard IEEE 754 - liczba zdenormalizowana



- pojawia się, gdy występuje **niedomiar** (ang. **underflow**), ale wynik operacji można jeszcze zapisać denormalizując mantysę
- wtedy mantysa nie posiada domyślnej części całkowitej równej 1, tzn. reprezentuje liczbę o postaci **0,xxx...xxx**, a nie **1,xxx...xxx**

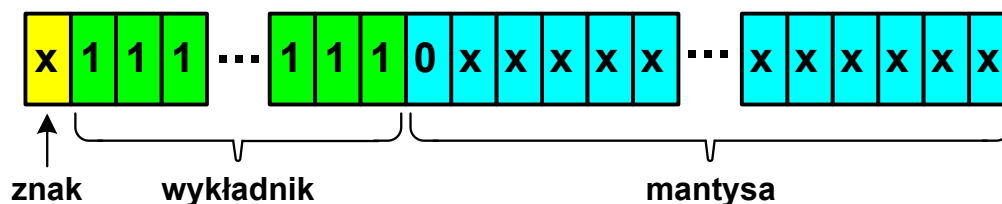


## Standard IEEE 754 - nieliczby



QNaN (Quiet NaN)  
ciche nieliczby

- ciche nieliczby „przechodzą” przez działania arytmetyczne (ich wystąpienie nie powoduje przerwania wykonywania programu)
- najczęściej oznaczają wartość niezdefiniowaną



SNaN (Signaling NaN)  
głośne nieliczby

- powodują powstanie wyjątków w operacjach arytmetycznych i przerwanie wykonywania programu
- najczęściej oznaczają wartość niedozwoloną