

Podstawy informatyki (EKS1C1007)

Politechnika Białostocka - Wydział Elektryczny
Ekoenergetyka, semestr I, studia stacjonarne I stopnia
Rok akademicki 2021/2022

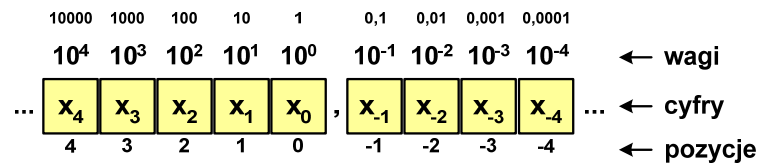
Pracownia nr 1 (04/05.10.2021)

dr inż. Jarosław Forenc

Pracownia nr 1

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

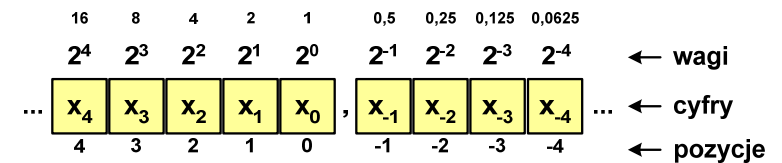
Pozycyjne systemy liczbowe: dziesiętny



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

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

Pozycyjne systemy liczbowe: dwójkowy



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

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

Pozycyjne systemy liczbowe

X ₍₂₎	X ₍₃₎	X ₍₄₎	X ₍₅₎	X ₍₈₎	X ₍₁₀₎	X ₍₁₂₎	X ₍₁₆₎
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
10	2	2	2	2	2	2	2
11	10	3	3	3	3	3	3
100	11	10	4	4	4	4	4
101	12	11	10	5	5	5	5
110	20	12	11	6	6	6	6
111	21	13	12	7	7	7	7
1000	22	20	13	10	8	8	8
1001	100	21	14	11	9	9	9
1010	101	22	20	12	10	A	A
1011	102	23	21	13	11	B	B
1100	110	30	22	14	12	10	C
1101	111	31	23	15	13	11	D
1110	112	32	24	16	14	12	E
1111	120	33	30	17	15	13	F

Konwersja na system dziesiętny

□ $p = 4, D = \{0, 1, 2, 3\}$

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

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

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

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

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

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

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

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

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

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

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

Konwersja z systemu dziesiętnego na dowolny

■ Algorytm Hornera

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

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

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

$$\begin{array}{l} 626/2 = 313 \text{ reszta } 0 \\ 313/2 = 156 \text{ reszta } 1 \\ 156/2 = 78 \text{ reszta } 0 \\ 78/2 = 39 \text{ reszta } 0 \\ 39/2 = 19 \text{ reszta } 1 \\ 19/2 = 9 \text{ reszta } 1 \\ 9/2 = 4 \text{ reszta } 1 \\ 4/2 = 2 \text{ reszta } 0 \\ 2/2 = 1 \text{ reszta } 0 \\ 1/2 = 0 \text{ reszta } 1 \end{array}$$

kolejność odczytywania
cyfr liczby w systemie
dwójkowym

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

$2 \rightarrow 4$

$$\begin{array}{c} 110110011_{(2)} = ?_{(4)} \\ \underline{01} \mid \underline{10} \mid \underline{11} \mid \underline{00} \mid \underline{11} \\ \quad 1 \quad \quad 2 \quad \quad 3 \quad \quad 0 \quad \quad 3 \\ 110110011_{(2)} = 12303_{(4)} \end{array}$$

$2 \rightarrow 8$

$$\begin{array}{c} 10110011_{(2)} = ?_{(8)} \\ \underline{010} \mid \underline{110} \mid \underline{011} \\ \quad 2 \quad \quad 6 \quad \quad 3 \\ 10110011_{(2)} = 263_{(8)} \end{array}$$

$2 \rightarrow 16$

$$\begin{array}{c} 1011010_{(2)} = ?_{(16)} \\ \underline{0101} \mid \underline{1010} \\ \quad 5 \quad \quad A \\ 1011010_{(2)} = 5A_{(16)} \end{array}$$

$4 \rightarrow 2$

$$\begin{array}{c} 12303_{(4)} = ?_{(2)} \\ \underline{01} \quad \underline{10} \quad \underline{11} \quad \underline{00} \quad \underline{11} \\ \quad 1 \quad \quad 2 \quad \quad 3 \quad \quad 0 \quad \quad 3 \\ 12303_{(4)} = 110110011_{(2)} \end{array}$$



$8 \rightarrow 2$

$$\begin{array}{c} 263_{(8)} = ?_{(2)} \\ \underline{010} \quad \underline{110} \quad \underline{011} \\ \quad 2 \quad \quad 6 \quad \quad 3 \\ 263_{(8)} = 10110011_{(2)} \end{array}$$

$16 \rightarrow 2$

$$\begin{array}{c} 5A_{(16)} = ?_{(2)} \\ \underline{0101} \quad \underline{1010} \\ \quad 5 \quad \quad A \\ 5A_{(16)} = 1011010_{(2)} \end{array}$$

Jednostki informacji - bit

- **Bit** (ang. **binary 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
- Wielokrotności bitów (zgodnie z układem SI):
 - kilobit - kb (10^3), megabit - Mb (10^6), gigabit - Gb (10^9)
 - terabit - Tb (10^{12}), petabit - Pb (10^{15}), eksabit - Eb (10^{18})

Jednostki informacji - bajt

- Wielokrotności bajtów:

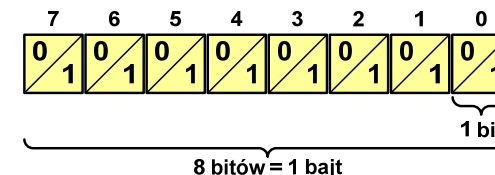
Przedrostki dziesiętne (układ SI)			Przedrostki binarne (IEC 60027-2)		
Nazwa	Symbol	Mnożnik	Nazwa	Symbol	Mnożnik
bajt	B	---	bajt	B	---
kilobajt	kB	$10^3 = 1000^1$	kibibajt	KiB	$2^{10} = 1024^1$
megabajt	MB	$10^6 = 1000^2$	mebibajt	MiB	$2^{20} = 1024^2$
gigabajt	GB	$10^9 = 1000^3$	gibibajt	GiB	$2^{30} = 1024^3$
terabajt	TB	$10^{12} = 1000^4$	tebibajt	TiB	$2^{40} = 1024^4$
petabajt	PB	$10^{15} = 1000^5$	pebibajt	PiB	$2^{50} = 1024^5$
eksabajt	EB	$10^{18} = 1000^6$	eksbibajt	EiB	$2^{60} = 1024^6$
zettabajt	ZB	$10^{21} = 1000^7$	zebibajt	ZiB	$2^{70} = 1024^7$
jottabajt	YB	$10^{24} = 1000^8$	jobibajt	YiB	$2^{80} = 1024^8$

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

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)



- 8-bitowy bajt nazywany jest także **oktetem**
- Za pomocą jednego bajtu można zapisać **256** różnych wartości:

0000 0000	→	0
0000 0001	→	1	1111 1110	→ 254
...	1111 1111	→ 255
- Najczęściej stosowanym skrótem dla bajtu jest wielka litera „B” (uwaga: B oznacza też **bela**, ale częściej używa się **dB** - **decybel**)

Jednostki informacji - przykłady

- Stosujemy mnożnik używany w informatyce (**1024**)

kB → B: 1 kB = 1 024 B

MB → B: 1 MB = 1024 × 1024 = 1 048 576 B

GB → B: 1 GB = 1024 × 1024 × 1024 = 1 073 741 824 B

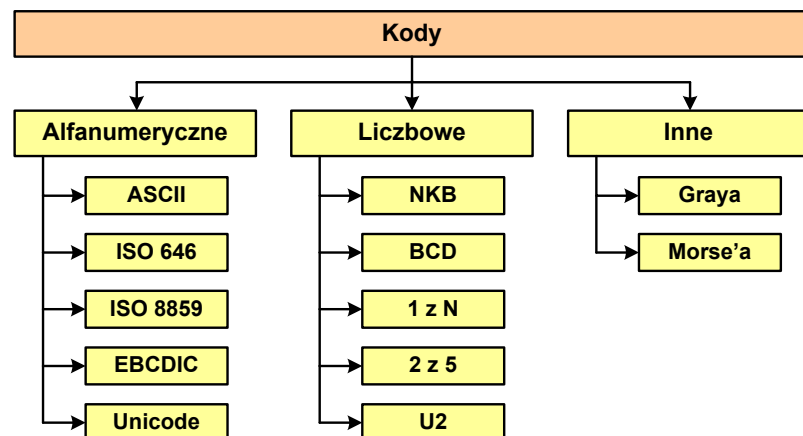
MB → kB: 2 MB = 2 × 1024 = 2 048 kB

GB → kB: 3 GB = 3 × 1024 × 1024 = 3 145 728 kB

TB → kB: 4 TB = 4 × 1024 × 1024 × 1024 = 4 294 967 296 kB

Kodowanie

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



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
 - 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
 - ...

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 8859-2

- ISO/IEC 8859-2, Latin-2 („środkowo”, „wschodnioeuropejskie”)
- przykład:
 - „A” - kod: $41_{(16)} = 40_{(16)} + 1_{(16)}$
 - „Ę” - kod: $C_{A(16)} = C0_{(16)} + A_{(16)}$
- 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

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	Znaki kontrolne															
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	Nie używane															
90																
A0	NB SP	À	Á	Â	Ã	Ä	Å	Ā	Ą	Ć	Č	Ĉ	Ď	Ě	Š	Ž
B0	°	à	á	â	ã	ä	å	ā	ą	ć	č	ĉ	ď	ě	š	ž
C0	Ř	Ā	Ă	Ä	Å	Ā	Ą	Ć	Č	Ĉ	Ď	Ě	Š	Ž	Í	Ď
D0	Đ	Ñ	Ń	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Û	Ü	Ý	Ť	Š
E0	đ	ñ	ń	ó	ô	õ	ö	×	ř	ů	ú	û	ü	ý	ť	š
F0	đ	ñ	ń	ó	ô	õ	ö	÷	ř	ů	ú	û	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 14.0.0** (wrzesień 2021)
 - The Unicode Consortium. The Unicode Standard, Version 14.0.0, (Mountain View, CA: The Unicode Consortium, 2021)
 - <http://www.unicode.org/versions/Unicode13.0.0/>
 - Koduje 144.697 znaków

Unicode - Zakresy



Zakres:	Znaczenie:
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 opisanie kodu znaku

A	Ω	語	𐄌	UTF-32
00000041	000003A9	00008A9E	00010384	

A	Ω	語	𐄌	UTF-16
0041	03A9	8A9E	D800 DF84	

A	Ω	語	𐄌	UTF-8
41	CE A9	E8 AA 9E	F0 90 8E 84	

źródło: The Unicode Consortium. The Unicode Standard, Version 5.2.0

Unicode



	010	011	012	013	014	015	016	017
0	Ā	Đ	Ġ	Ĭ	Ĳ	Ō	Š	Ū
	0100	0110	0120	0130	0140	0150	0160	0170
1	ā	đ	ġ	ĭ	ĳ	ō	š	ū
	0101	0111	0121	0131	0141	0151	0161	0171
2	Ă	Ē	Ģ	Ĳ	Ł	Œ	Ţ	Ū
	0102	0112	0122	0132	0142	0152	0162	0172
3	ă	ē	ģ	ĳ	ł	œ	ț	ū
	0103	0113	0123	0133	0143	0153	0163	0173
4	Ą	Ĕ	Ĥ	Ĵ	ń	Ŗ	Ŧ	Ű
	0104	0114	0124	0134	0144	0154	0164	0174
5	ą	ė	ĥ	ĵ	ņ	ŗ	ŧ	ű
	0105	0115	0125	0135	0145	0155	0165	0175
6	Ć	Ė	Ħ	Ķ	ņ	Ŗ	Ŧ	Ŷ
	0106	0116	0126	0136	0146	0156	0166	0176
7	ć	ė	ħ	ķ	ņ	ŗ	ŧ	ŷ
	0107	0117	0127	0137	0147	0157	0167	0177

European Latin

0100	Ā	LATIN CAPITAL LETTER A WITH MACRON	≡ 0041 A 0304 5
0101	ā	LATIN SMALL LETTER A WITH MACRON	• Latvian, Latin, ... ≡ 0061 a 0304 5
0102	Ă	LATIN CAPITAL LETTER A WITH BREVE	≡ 0041 A 0306 8
0103	ă	LATIN SMALL LETTER A WITH BREVE	• Romanian, Vietnamese, Latin, ... ≡ 0061 a 0306 8
0104	Ą	LATIN CAPITAL LETTER A WITH OGONEK	≡ 0041 A 0328 9
0105	ą	LATIN SMALL LETTER A WITH OGONEK	• Polish, Lithuanian, ... ≡ 0061 a 0328 9
0106	Ć	LATIN CAPITAL LETTER C WITH ACUTE	≡ 0043 C 0301 6
0107	ć	LATIN SMALL LETTER C WITH ACUTE	• Polish, Croatian, ... → 045B ħ cyrillic small letter tsh ≡ 0063 c 0301 6

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	system szesnastkowy
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	
ASCII	83	97	108	97	32	87	69	45	49	49	48	system dziesiętny
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	

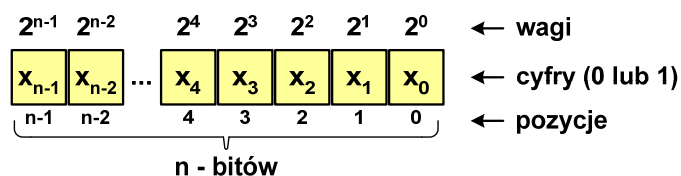
Kody alfanumeryczne - przykład

- Tekst: **Żółty wąż**

	Ż	ó	ł	t	y		w	a	ż	
ASCII	×	×	×	74	79	20	77	×	×	system szesnastkowy
ISO 8859-2	Af	F3	B3	74	79	20	77	B1	Bf	
Unicode	17B	F3	142	74	79	20	77	105	17C	
ASCII	×	×	×	116	121	32	119	×	×	system dziesiętny
ISO 8859-2	175	243	179	116	121	32	119	177	191	
Unicode	379	243	322	116	121	32	119	261	380	

Liczby całkowite bez znaku

- Zapis liczby w systemie dwójkowym (NKB - Naturalny Kod Binarny):



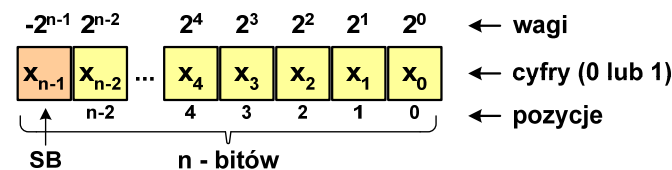
- Używając **n-bitów** można zapisać liczbę z zakresu:

$X_{(2)} = \langle 0, 2^n - 1 \rangle$	8-bitów	0 ... 255
	16-bitów	0 ... 65 535
	32-bitów	0 ... 4 294 967 295
	64-bitów	0 ... 18 446 744 073 709 551 615

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

Liczby całkowite ze znakiem (U2)

- **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 - liczby całkowite ze znakiem

Zamiana liczby dziesiętnej na kod U2:

- liczba dodatnia

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

- zamieniamy liczbę na NKB

$$75_{(10)} = 1001011_{(NKB)}$$

- Dodajemy bit znaku: 0

$$75_{(10)} = 01001011_{(U2)}$$

- liczba ujemna

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

- zamieniamy **moduł** liczby na U2

$$|-75_{(10)}| = 75_{(10)} = 01001011_{(U2)}$$

- negujemy wszystkie bity i dodajemy 1

$$\begin{array}{r} 01001011 \\ \text{negacja: } 10110100 \\ +1: \quad \quad 1 \\ \hline -75_{(10)} = 10110101_{(U2)} \end{array}$$

Zapis zmiennoprzecinkowy liczby rzeczywistej

- Postać zmiennoprzecinkowa zapisu liczby: $1,2 \cdot 10^{13} \quad -4,53 \cdot 10^{-12}$

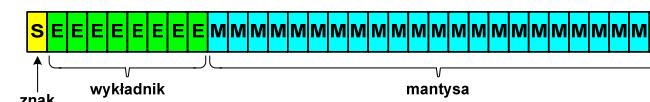
- Elementy zapisu:

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

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

- W systemie binarym



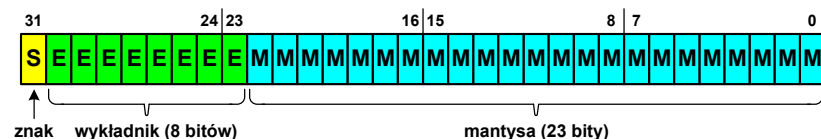
- Podstawa systemu jest stała: $B = 2$

- BIAS** - przesunięcie wykładnika:
127 (format 32-bit.), 1023 (format 64-bit.)

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

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 ($2 > |M| \geq 1$), jej pierwszy bit jest zawsze równy 1 i nie jest zapamiętywany
- Bit ten jest automatycznie uwzględniany podczas wykonywania obliczeń

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

Standard IEEE 754 - liczby 32-bitowe

■ Przykład:

- obliczmy wartość dziesiętną liczby zmiennoprzecinkowej

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

- dzielimy liczbę na części

$$\begin{array}{c} 0 \quad 1000101 \quad 1001000000000000000000 \\ \text{S-bit znaku} \quad \text{E-wykładnik} \quad \text{M-mantysa (tylko część ułamkowa)} \end{array}$$

- określamy **znak liczby**

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

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

$$E = 1000101_{(2)} = 128 + 4 + 1 = 133 - \underset{\text{nadmiar}}{127} = 6_{(10)}$$

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:

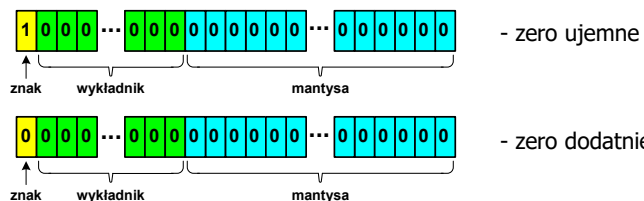
$$S = 0, \quad E = 6_{(10)}, \quad M = 1,5625_{(10)}$$

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

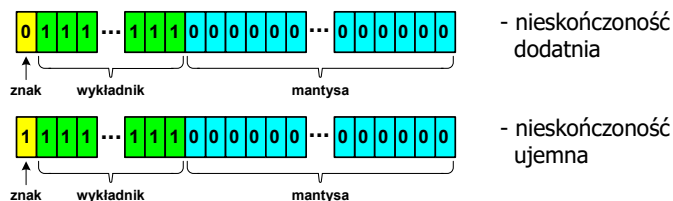
$$01000010110010000000000000000000_{(IEEE754)} = 100_{(10)}$$

Standard IEEE 754 - wartości specjalne

■ Zero:



■ Nieskończoność:



■ Liczba zdenormalizowana, nieliczby (QNaN, SNaN)