

Technologie informacyjne (EZ1E1003)

Politechnika Białostocka - Wydział Elektryczny
semestr I, studia niestacjonarne I stopnia
Rok akademicki 2020/2021

Pracownia nr 1

dr inż. Jarosław Forenc

Dane podstawowe

- dr inż. Jarosław Forenc
- Politechnika Białostocka, Wydział Elektryczny,
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki
ul. Wiejska 45D, 15-351 Białystok
WE-204
- e-mail: j.forenc@pb.edu.pl
- tel. (0-85) 746-93-97
- <http://jforenc.prv.pl>
 - Dydaktyka - dodatkowe materiały do pracowni
- Konsultacje
 - wtorek, 16:00-17:00, WE-204 / Teams
 - piątek, 09:00-11:00, 14:00-15:00, WE-204 / Teams
 - sobota, 16:00 -17:00, Teams (zaoczne)
 - niedziela, 14:00 - 15:30, Teams (zaoczne)

Program przedmiotu

1. Zajęcia organizacyjne. BHP na stanowisku pracy z komputerem. Licencje oprogramowania. Pozycyjne systemy liczbowe. Konwersje pomiędzy systemami liczbowymi. Jednostki informacji. Kodowanie znaków i liczb. Reprezentacja wartości liczbowych w systemach komputerowych.
2. Sprawdzian nr 1. **Przetwarzanie tekstów**. Reguły wprowadzania tekstu. Formatowanie znaków i akapitów. Zastosowanie tabulatorów. Wzory.
3. **Przetwarzanie tekstów**. Tabele. Formatowanie strony. Formatowanie nagłówka i stopki. Tworzenie dokumentu w oparciu o style. Automatyczne numerowanie rysunków i tabel. Generowanie spisu treści, rysunków i tabel.

Program przedmiotu

4. Sprawdzian nr 2. **Arkusze kalkulacyjne**. Wprowadzanie danych do arkusza. Formatowanie arkusza. Proste wzory obliczeniowe. Formuły. Adresowanie względne, bezwzględne i mieszane.
5. **Arkusze kalkulacyjne**. Operacje na macierzach i liczbach zespolonych. Funkcje logiczne. Wykresy.
6. Sprawdzian nr 3. **Grafika menedżerska i prezentacyjna**. Podstawy tworzenia prezentacji multimedialnych. Rysunki, wykresy, autokształty i pola tekstowe na slajdach. Zasady poprawnie zbudowanej prezentacji.
7. **Matlab**. Wprowadzanie poleceń, zmiennych i liczb. Operatory i wyrażenia arytmetyczne. Wprowadzanie i generowanie macierzy.

Program przedmiotu

8. **Matlab**. Operacje macierzowe i tablicowe. Rozwiązywanie układów równań. Wielomiany. Grafika dwu- i trójwymiarowa.
9. Sprawdzian nr 4. **Grafika menedżerska i prezentacyjna**. Przygotowanie własnej prezentacji.
10. **Grafika menedżerska i prezentacyjna**. Przedstawienie prezentacji. Zaliczenie zajęć.

Literatura

1. S. Gryś: „Arytmetyka komputerów w praktyce”. PWN, W-wa, 2007.
2. G. Coldwin: „Zrozumieć programowanie”. PWN, Warszawa, 2015.
3. R. Kawa, J. Lembas: „Wykłady z informatyki. Wstęp do informatyki”. PWN, Warszawa, 2017
4. A. Żarowska, W. Węglarz: „ECDL na skróty”. PWN, Warszawa, 2011.
5. M. Dziewoński: „OpenOffice 3.x PL. Oficjalny podręcznik”. Helion, Gliwice, 2009
6. https://wiki.openoffice.org/wiki/Documentation/OOo3_User_Guides/OOo3.3_User_Guide_Chapters - OpenOffice.org 3.3 User Guides
7. A. Wolański: „Edycja tekstów Praktyczny poradnik”. PWN, Warszawa, 2017.

Literatura

8. M. Stachurski, W. Treichel: „Matlab dla studentów. Ćwiczenia, zadania, rozwiązania”. Witkom, Warszawa, 2009.
9. R. Pratap: „MATLAB dla naukowców i inżynierów. Wydanie 2”. PWN, Warszawa, 2015.
10. P. Lenar: „Sekrety skutecznych prezentacji multimedialnych. Wydanie II rozszerzone”. Helion, Gliwice, 2011.

Zaliczenie

- Ogólne warunki zaliczenia przedmiotu:
 - obecność na zajęciach
 - zaliczenie w trakcie zajęć zadań przedstawionych przez prowadzącego
 - zaliczenie przygotowanych samodzielnie prac, których tematykę określa prowadzący
 - zaliczenie sprawdzianów

Podstawę do zaliczenia przedmiotu (uzyskanie punktów ECTS) stanowi stwierdzenie, że każdy z założonych efektów uczenia się został osiągnięty w co najmniej minimalnym akceptowalnym stopniu.

Efekty uczenia się

EU1	klasyfikuje i omawia z użyciem właściwej terminologii zagadnienia związane z technologiami informacyjnymi
EU2	wykorzystuje programy do przygotowania dokumentacji technicznej z elementami osadzonymi
EU3	potrafi wykonywać podstawowe obliczenia i wykorzystywać wbudowane, dostępne funkcje pakietu matematycznego
EU4	przygotowuje poprawnie wykresy i inne elementy ułatwiające interpretację wyników
EU5	potrafi zastosować funkcje matematyczne w ramach pakietu matematycznego, arkusza kalkulacyjnego
EU6	planuje, przygotowuje i omawia prezentację komputerową dotyczącą problemów technicznych

Zaliczenie

- Ocena końcowa jest wystawiana na podstawie sumy punktów za sprawdziany i przedstawienie prezentacji multimedialnej
- Sposób wyznaczania oceny końcowej:

Punkty	Ocena	Punkty	Ocena
546 - 600	5,0	366 - 425	3,5
486 - 545	4,5	306 - 365	3,0
426 - 485	4,0	0 - 305	2,0

Zaliczenie

- Terminy sprawdzianów:
 - zjazd nr 2 - sprawdzian nr 1 - zajęcia nr 1 (ocena)
 - zjazd nr 4 - sprawdzian nr 2 - przetwarzanie tekstów (ocena)
 - zjazd nr 6 - sprawdzian nr 3 - arkusze kalkulacyjne (ocena x 2)
 - zjazd nr 9 - sprawdzian nr 4 - program Matlab (ocena)
 - zjazd nr 10 - przedstawienie prezentacji (ocena)
- Sprawdziany odbędą się w trakcie zajęć na terenie Uczelni
- Za każdy sprawdzian można otrzymać od 0 do 100 pkt.

Punkty	Ocena	Punkty	Ocena
91 - 100	5,0	61 - 70	3,5
81 - 90	4,5	51 - 60	3,0
71 - 80	4,0	0 - 50	2,0

Zaliczenie (zawieszenie zajęć na Uczelni)

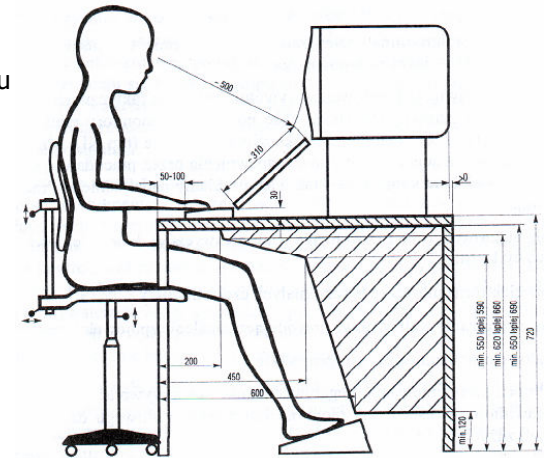
- Ogólne warunki zaliczenia przedmiotu:
 - ~~obecność na zajęciach~~
 - ~~zaliczenie w trakcie zajęć zadań przedstawionych przez prowadzącego~~
 - zaliczenie przygotowanych samodzielnie prac, których tematykę określa prowadzący
 - zaliczenie sprawdzianów
- Sprawdziany odbędą się w trakcie zajęć z wykorzystaniem technik zdalnego nauczania

Pracownia nr 1

- Bezpieczeństwo i higiena pracy przy komputerze
- Licencje oprogramowania
- Pozycyjne systemy liczbowe
- Jednostki informacji: bit, bajt
- Kodowanie
 - znaków: ASCII, ISO 8859, Unicode
 - liczb: NKB (BCN), U2, BCD
- Liczby zmiennoprzecinkowe
 - standard IEEE 754

Bezpieczeństwo i higiena pracy przy komputerze

- Praca przy komputerze powoduje następujące dolegliwości:
 - bóle szyi i kręgosłupa
 - bóle głowy
 - zmęczenie i łzawienie oczu
 - skurcze mięśni nóg
 - mrowienie, drętwienie, niedowład kończyn
 - łokieć tenisisty
 - zespół cieśni nadgarstka (spowodowany uciskiem nerwu pośrodkowego w kanale nadgarstka)
 - zespół suchego oka



Prawidłowo zorganizowane stanowisko do pracy z komputerem

Bezpieczeństwo i higiena pracy przy komputerze

- Czynniki wpływające na prawidłową i bezpieczną pracę z komputerem:
 - odpowiednie meble z krzesłem o regulowanej wysokości oraz kącie nachylenia oparcia; stosownie duży blat, pozwalający zachować właściwą odległość monitora od oczu (40-75 cm)
 - odpowiednie oświetlenie, dające niezbyt duże różnice jasności ekranu monitora i pomieszczenia oraz eliminujące odbicia
 - ustawienie komputera z wygodnym dostępem do klawiatury i myszy
 - ustawienie monitora tyłem do ściany i tyłem do światła dziennego
 - odpowiedni mikroklimat: temperatura 21-22 stopnie Celsjusza, wilgotność powietrza 50-60%, lekki ruch powietrza
 - dokonywanie przerw w trakcie pracy przy komputerze oraz ćwiczeń relaksacyjnych
 - regularne czyszczenie ekranu i klawiatury

Licencje oprogramowania

- **Licencja na oprogramowanie** - umowa na korzystanie z utworu jakim jest aplikacja komputerowa, zawierana pomiędzy podmiotem, któremu przysługują majątkowe prawa autorskie do utworu, a osobą, która zamierza z danej aplikacji korzystać
- Co zawiera licencja?
 - sposób wykorzystania oprogramowania czyli tzw. pola eksploatacji (komercyjne, prywatne)
 - ograniczenie co do liczby komputerów, na których można zainstalować oprogramowanie
 - klauzulę o wyłączonej odpowiedzialności producenta z tytułu używania oprogramowania przez licencjodawcę (producent nie odpowiada za skutki błędów w programach)

Ważniejsze typy licencji oprogramowania

- **OEM (Original Equipment Manufacturer)**
 - oprogramowanie sprzedawane razem ze sprzętem komputerowym, na którym jest zainstalowane
 - nie może być przeniesione na inny komputer ani odsprzedane
 - po wymianie części w komputerze licencja może być unieważniona
- **Freeware**
 - umożliwia bezpłatne korzystanie z oprogramowania bez ujawnienia kodu źródłowego
 - osoby trzecie nie mogą czerpać korzyści finansowych z jego dystrybucji
 - może być objęte prawami autorskimi lub nie

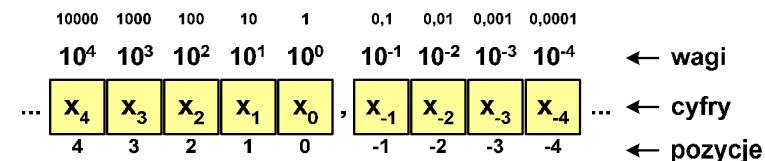
Ważniejsze typy licencji oprogramowania

- **GNU General Public Licence**
 - oprogramowanie bezpłatne, które można uruchamiać w dowolnych celach, analizować jak działa i dostosowywać do swoich potrzeb, udoskonalać i publicznie rozpowszechniać udoskonaloną wersję
- **Donationware**
 - oprogramowanie, z którego można bezpłatnie korzystać, modyfikować je, kopiować i dystrybuować pod warunkiem, że zapłaci się autorowi symboliczną kwotę
- **Adware**
 - oprogramowanie rozpowszechniane za darmo, ale zawierające funkcje wyświetlającą reklamy (zazwyczaj banery)
 - wydawca oprogramowania zarabia na tych reklamach, zwykle jest też możliwość nabycia wersji programu bez reklam za opłatą

Ważniejsze typy licencji oprogramowania

- **Shareware**
 - oprogramowanie rozpowszechniane bez opłat, ale z ograniczeniami
 - po okresie próbnym (określony czas lub liczba uruchomień) mającym zachęcić do jego nabycia należy wykupić pełną licencję na program lub usunąć go
 - czasami w okresie próbnym pewne funkcje programu są zablokowane i stają się dostępne dopiero po wykupieniu licencji
- **Trial**
 - w pełni funkcjonalne oprogramowanie, którego działa przez z góry określony czas (najczęściej od 7 do 30 dni)

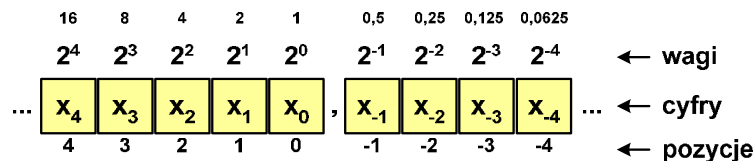
Pozycyjne systemy liczbowe: dziesiętny



- Podstawa: $p = 10$, dozwolone cyfry: $D = \{0,1,2,3,4,5,6,7,8,9\}$
- Przykład: $1408,25_{(10)}$

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

Pozycyjne systemy liczbowe: dwójkowy



■ W systemie dwójkowym: $p = 2$, $D = \{0,1\}$

■ Przykład: $1101,101_{(2)}$

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

Pozycyjne systemy liczbowe: konwersje

■ Algorytm Hornera

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

$$\begin{array}{l}
 626_{(10)} = ?_{(2)} \\
 626/2 = 313 \text{ reszta } 0 \\
 313/2 = 156 \text{ reszta } 1 \\
 156/2 = 78 \text{ reszta } 0 \\
 78/2 = 39 \text{ reszta } 0 \\
 39/2 = 19 \text{ reszta } 1 \\
 19/2 = 9 \text{ reszta } 1 \\
 9/2 = 4 \text{ reszta } 1 \\
 4/2 = 2 \text{ reszta } 0 \\
 2/2 = 1 \text{ reszta } 0 \\
 1/2 = 0 \text{ reszta } 1
 \end{array}$$

kolejność odczytywania cyfr liczby w systemie dwójkowym

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

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

Pozycyjne systemy liczbowe: konwersje

■ Algorytm Hornera

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

$$\begin{array}{l}
 626_{(10)} = ?_{(7)} \\
 626/7 = 89 \text{ reszta } 3 \\
 89/7 = 12 \text{ reszta } 5 \\
 12/7 = 1 \text{ reszta } 5 \\
 1/7 = 0 \text{ reszta } 1
 \end{array}$$

$$626_{(10)} = 1553_{(7)}$$

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

$$\begin{array}{l}
 626_{(10)} = ?_{(14)} \\
 626/14 = 44 \text{ reszta } 10 \rightarrow A \\
 44/14 = 3 \text{ reszta } 2 \\
 3/14 = 0 \text{ reszta } 3
 \end{array}$$

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

Pozycyjne systemy liczbowe: konwersje

- Konwersje pomiędzy systemem dwójkowym (2) i czwórkowym (4)

$$110110011_{(2)} = ?_{(4)} \qquad 12303_{(4)} = ?_{(2)}$$

$$\begin{array}{cccccc} \underline{01} & | & \underline{10} & | & \underline{11} & | & \underline{00} & | & \underline{11} \\ \underline{1} & & \underline{2} & & \underline{3} & & \underline{0} & & \underline{3} \end{array} \qquad \begin{array}{ccccc} \overset{1}{01} & \overset{2}{10} & \overset{3}{11} & \overset{0}{00} & \overset{3}{11} \\ \underline{01} & \underline{10} & \underline{11} & \underline{00} & \underline{11} \end{array}$$

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



- Konwersje pomiędzy systemem dwójkowym (2) i ósemkowym (8)

$$10110011_{(2)} = ?_{(8)} \qquad 263_{(8)} = ?_{(2)}$$

$$\begin{array}{ccc} \underline{010} & | & \underline{110} & | & \underline{011} \\ \underline{2} & & \underline{6} & & \underline{3} \end{array} \qquad \begin{array}{ccc} \overset{2}{010} & \overset{6}{110} & \overset{3}{011} \\ \underline{010} & \underline{110} & \underline{011} \end{array}$$

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

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})

Pozycyjne systemy liczbowe: konwersje

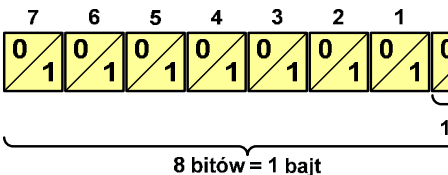
- Konwersje pomiędzy systemem dwójkowym (2) i szesnastkowym (16)

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

$$\begin{array}{cc} \underline{0101} & | & \underline{1010} \\ \underline{5} & & \underline{A} \end{array} \qquad \begin{array}{cc} \overset{5}{0101} & \overset{A}{1010} \\ \underline{0101} & \underline{1010} \end{array}$$

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

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 - bajt

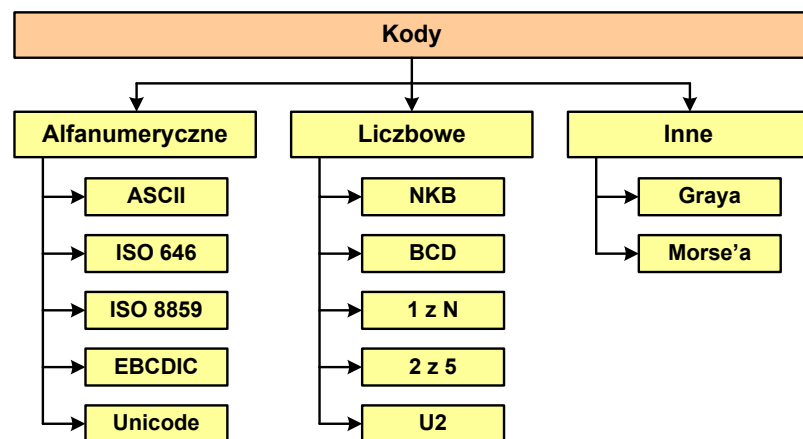
- Wielokrotności bajtów:

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

Kodowanie

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



Jednostki informacji - przykłady

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

KB → B: $1 \text{ KB} = 1024 \text{ B}$

MB → B: $1 \text{ MB} = 1024 \times 1024 = 1048576 \text{ B}$

GB → B: $1 \text{ GB} = 1024 \times 1024 \times 1024 = 1073741824 \text{ B}$

MB → kB: $2 \text{ MB} = 2 \times 1024 = 2048 \text{ kB}$

GB → kB: $3 \text{ GB} = 3 \times 1024 \times 1024 = 3145728 \text{ kB}$

TB → kB: $4 \text{ TB} = 4 \times 1024 \times 1024 \times 1024 = 4294967296 \text{ kB}$

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
 - 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: $C4_{(16)} = C0_{(16)} + A_{(16)}$
- 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	Nieużywane															
90																
A0	NBSP	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î
B0	°	ä	å	ä	å	ä	å	ä	å	ä	å	ä	å	ä	å	ä
C0	Ā	Ă	Ȧ	Ȧ	Ĺ	Ć	Č	Ď	Ě	Ě	Ě	Ě	Ě	Ě	Ě	Ě
D0	Đ	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń	Ń
E0	Š	š	š	š	š	š	š	š	š	š	š	š	š	š	š	š
F0	đ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ	ñ

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 13.0.0** (marzec 2020)
 - The Unicode Consortium. The Unicode Standard, Version 13.0.0, (Mountain View, CA: The Unicode Consortium, 2020)
 - <http://www.unicode.org/versions/Unicode13.0.0/>
 - Koduje 143.859 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
00000041	000003A9	00008A9E	00010384

UTF-32

A	Ω	語	𐄌
0041	03A9	8A9E	D800 DF84

UTF-16

A	Ω	語	𐄌
41	CE A9	E8 AA 9E	F0 90 8E 84

UTF-8

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

Unicode



	010	011	012	013	014	015	016	017
0	Ā	Đ	Ġ	Ī	Ĳ	Ō	Š	Ū
1	ā	đ	ġ	ī	ĳ	ō	š	ū
2	Ă	Ĕ	Ģ	Ĳ	Ĳ	Ţ	Ũ	
3	ă	ĕ	ģ	ĳ	ĳ	ţ	ũ	
4	Ą	Ě	Ĥ	Ĵ	ń	Ŕ	Ť	Ű
5	ą	ě	ĥ	ĵ	ņ	ŕ	ť	ű
6	Ć	Ę	Ħ	Ķ	ņ	Ŗ	Ŧ	Ŷ
7	ć	ę	ħ	ķ	ņ	ŗ	ŧ	ŷ

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 tshe
≡ 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
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	a	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

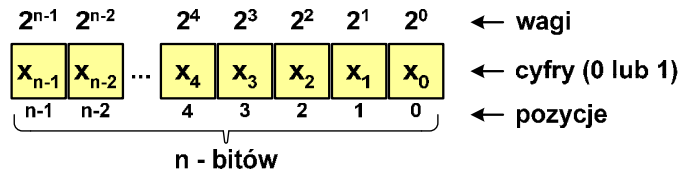
system szesnastkowy

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-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

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

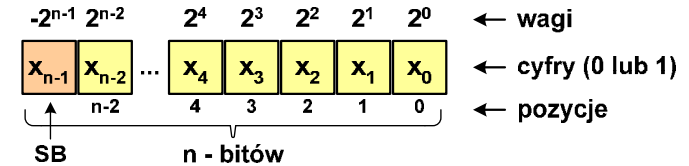
$$|-75_{(10)}| = 75_{(10)} = 01001011_{(U2)}$$

- negujemy wszystkie bity i dodajemy 1

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

U2 - liczby całkowite ze znakiem

- ZU2, uzupełnień do dwóch, two's complement**



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

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

BCD

- Binary-Coded Decimal** - 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

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

$$\overbrace{0001}^1 \overbrace{0110}^6 \overbrace{1000}^8$$

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

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

$$\overbrace{1001}^9 \overbrace{0101}^5 \overbrace{0011}^3$$

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

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

- Zapis liczby zmiennoprzecinkowej ma postać

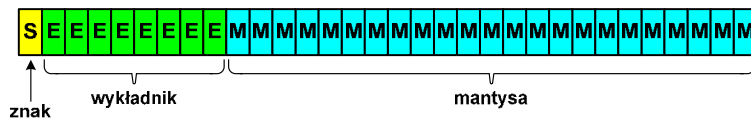
$$L = (-1)^S \cdot M \cdot B^E$$

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

Liczby zmiennoprzecinkowe w systemie binarnym

- Liczba bitów przeznaczonych na mantysę i wykładnik jest ograniczona



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

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

- Wykładnik jest zapisywany jako wartość przesunięta o pewną stałą (ang. **biased exponent**) - zapis z przesuniętym wykładnikiem
- Wartości przesunięcia: **127** (format 32-bit.), **1023** (format 64-bit.)

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

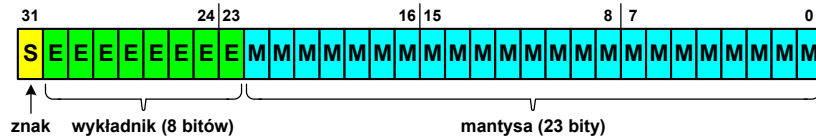
Standard IEEE 754

- 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)	≥ 43	1	≥ 11	$\geq 2^{\pm 1023} \approx 10^{\pm 308}$	≥ 31	≥ 10
Podwójna (Double Precision, binary64)	64	1	11	$2^{\pm 1023} \approx 10^{\pm 308}$	52	16
Podwójna rozszerzona (Double Extended)	≥ 79	1	≥ 15	$\geq 2^{\pm 16383} \approx 10^{\pm 4932}$	≥ 63	≥ 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

$$\begin{array}{c} 0 \quad 10000101 \quad 1001000000000000000000 \\ \text{S-bit znaku} \quad \text{E-wykładnik} \quad \text{M-mantysa (tylko część ułamkowa)} \end{array}$$

- określamy **znak liczby**

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

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

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

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:

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

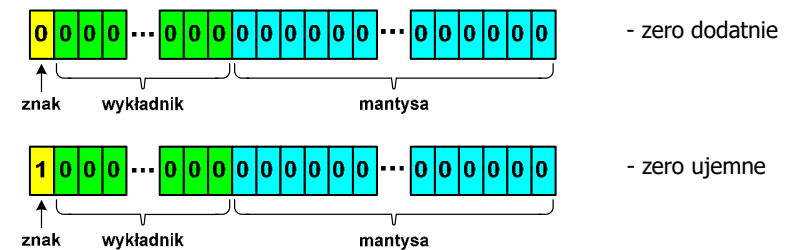
- podstawiając otrzymujemy:

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

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

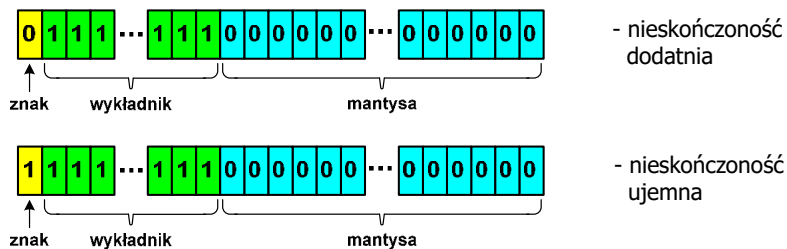
$$01000010110010000000000000000000_{(IEEE754)} = 100_{(10)}$$

Standard IEEE 754 - zero



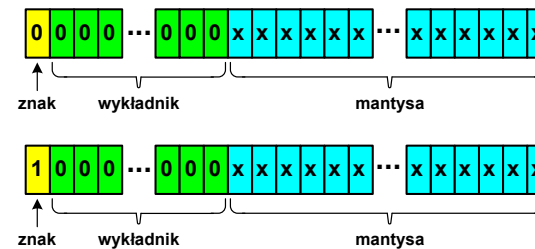
- bit znaku może przyjmować dowolną wartość
- przy porównaniach zero dodatnie i ujemne są traktowane jako równe sobie

Standard IEEE 754 - nieskończoność



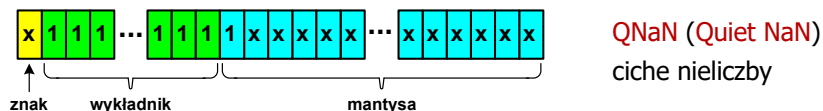
- bit znaku określa czy mamy nieskończoność dodatnią czy ujemną
- nieskończoność występuje w przypadku wystąpienia **nadmiaru** (przepełnienia) oraz przy dzieleniu przez zero

Standard IEEE 754 - liczba zdenormalizowana

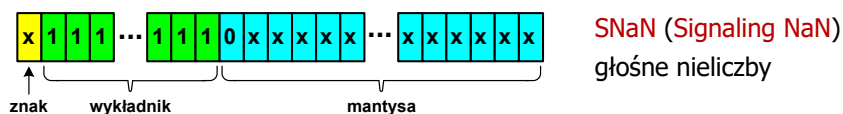


- pojawia się, gdy występuje **niedomiar** (ang. **underflow**), ale wynik operacji można jeszcze zapisać denormalizując mantysę
- wtedy mantysa nie posiada domyślnej części całkowitej równej 1, tzn. reprezentuje liczbę o postaci **0,xxx...xxx**, a nie **1,xxx...xxx**

Standard IEEE 754 - nieliczby



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



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