Akademia Górniczo-Hutnicza
Wydział Elektrotechniki, Automatyki,
Informatyki i Inżynierii Biomedycznej

Adrian Horzyk

WSTĘP DO INFORMATYKI

SYSTEMY KODOWANIA ORAZ REPREZENTACJA I ARYTMETYKA LICZB

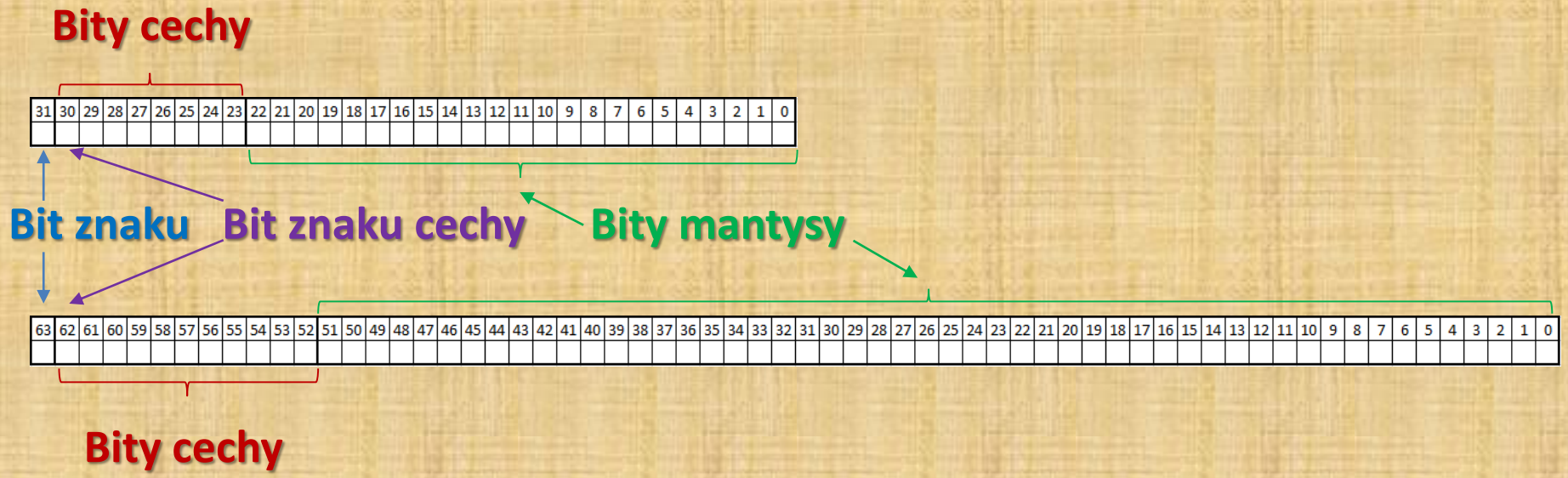
SYSTEMY KODOWANIA LICZB

W celu ujednolicenia obliczeń numerycznych na różnych platformach sprzętowych wprowadzono standard IEEE 754 dla zapisu liczb zmiennoprzecinkowych (o zapisie zmiennopozycyjnym).

Standard ten definiuje dwa rodzaje liczb zmiennoprzecinkowych:

- o pojedynczej precyzji (32-bitowe)
- o podwójnej precyzji (64-bitowe)

Kod binarny tych liczb podzielony jest na 3 części:



SYSTEMY KODOWANIA LICZB

Bit znaku *z* przyjmuje wartość:

0 – dla liczb dodatnich

1 – dla liczb ujemnych

Bit znaku cechy s przyjmuje wartość:

0 – dla liczb o wykładniku dodatnim (dużych liczb 2^c)

1 – dla liczb o wykładniku ujemnym (małych liczb 2^{-c})

Bity cechy określają:

wykładnik potęgowy dla podstawy dwójkowej

Bity mantysy:

Zawierają tylko bity ułamkowe (po przecinku dziesiętnym)

Liczba określona jest więc w następujący sposób: $l = (-1)^z \cdot m \cdot 2^{(-1)^S \cdot c}$

KODOWANIE STAŁO i ZMIENNOPOZYCYJNE

Kodowanie stałopozycyjne służy do zapisu liczb całkowitych l:

$$l = s\sum_{i=0}^{n} e_i 2^i, \quad (e_n \neq 0 \; dla \; l \neq 0)$$

Kodowanie zmiennopozycyjne służy do zapisu liczb wymiernych x: $rd(x) = s \cdot m_t \cdot 2^c$

$$m_t = \sum_{i=1}^{t} e_{-i} \cdot 2^{-i} + e_{-(t+1)} \cdot 2^{-t}$$

ograniczone do reprezentacji tylko t najistotniejszych bitów tych liczb w systemie binarnym. Powstaje więc pewien błąd reprezentacji $\tilde{\varepsilon} = rd(x) - x$
na skutek obcięcia najmniej znaczących bitów i zaokrąglenia mantysy do t bitów:

$$\left|\frac{rd(x)-x}{x}\right| \leq 2^{-t} \quad \Leftrightarrow \quad rd(x) = x \cdot (1+\varepsilon) \quad |\varepsilon| \leq 2^{-t}$$

Możemy więc reprezentować liczby z pewnego ograniczonego zakresu:

$$\frac{1}{2} \cdot 2^{c_{\min}} \leq |x| < 2^{c_{\max}} \qquad c_{\min} = -2^{d-t} + 1 \qquad c_{\max} = 2^{d-t} - 1$$

Mniejsze będą zastępowane zerem, co nazywamy błędem niedomiaru,
a większe będę powodować błąd nadmiaru, powodującym przerwanie obliczeń.

PRZYKŁAD

Obliczania modułu z liczby zespolonej: $z = a + bi \neq 0\,,\; i = \sqrt{-1}$

klasycznym algorytmem:

$$|z| = \sqrt{a^2 + b^2}$$

przekształconym algorytmem:

$$|z| = \begin{cases} a\sqrt{1 + \dfrac{b^2}{a^2}} & gdy\ a \geq b \\ b\sqrt{1 + \dfrac{a^2}{b^2}} & gdy\ a < b \end{cases}$$

W klasycznym algorytmie wystąpi nadmiar,

jeśli a lub b są co do modułu większe od $2^{c_{\max}/2}$

niedomiar, jeśli a lub b są co do modułu mniejsze od $\frac{1}{2} \cdot 2^{c_{\min}/2}$

zaś przekształcony algorytm pozwala na obliczenie modułu z liczby zespolonej z dla dowolnych a i b z zakresu reprezentowanych liczb.

ZAKRESY LICZB

Zakres liczb zmiennoprzecinkowych IEEE 754 w pojedynczej precyzji:

- $Z_{(IEEE\ 754)} = -3{,}4 \times 10^{38} \ldots 3{,}4 \times 10^{38}$
- Oferują precyzję 7 cyfr dziesiętnych

Zakres liczb zmiennoprzecinkowych IEEE 754 w podwójnej precyzji:

- $Z_{(IEEE\ 754)} = -1{,}8 \times 10^{308} \ldots 1{,}8 \times 10^{308}$
- Oferują precyzję 15-16 cyfr dziesiętnych

Interpretacja kodu IEEE 754			
Pole znaku	Pole cechy	Pole mantysy	Wartość
0	0...0	0..0	+0 (zero dodatnie)
0	0...0	0...1 : 1...1	Dodatnia liczba zdenormalizowana
0	0...1 : 1...0	0...0 : 1...1	Dodatnia liczba zmiennoprzecinkowa
0	1...1	0...0	+∞
0	1...1	0...01 : 01...1	SNaN (głośna nieliczba)
0	1...1	10...0 : 1...1	QNaN (cicha nieliczba)
1	0...0	0...0	-0 (zero ujemne)
1	0...0	0...1 : 1...1	Ujemna liczba zdenormalizowana
1	0...01 : 1...10	0...0 : 1...1	Ujemna liczba zmiennoprzecinkowa
1	1...1	0...0	-∞
1	1...1	0...01 : 01...1	SNaN (głośna nieliczba)
1	1...1	10...0 : 1...1	QNaN (cicha nieliczba)

Nie liczba (NaN):

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0/1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Nieskończoności:

- ✓ **Dodatnia:**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- ✓ **Ujemna:**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

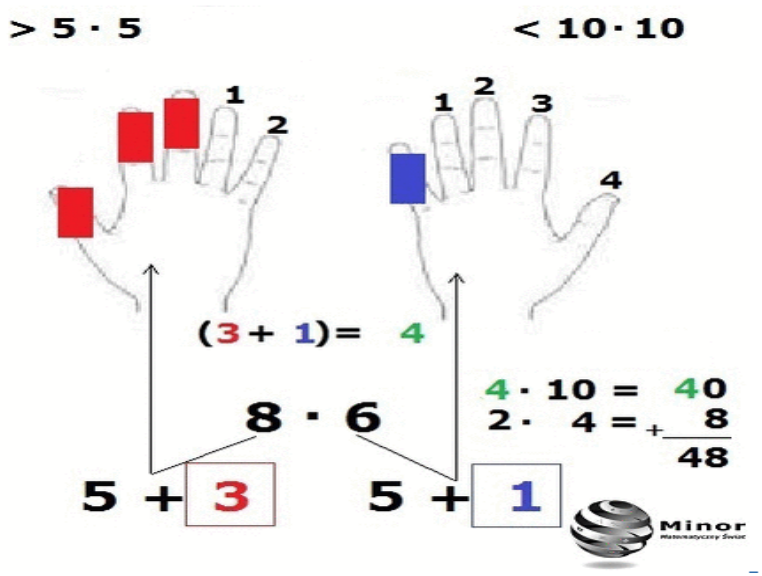
Liczenie na 10 palcach?

A dlaczego właśnie na 10?

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

A może by tak na dwóch jak komputery?

Tabliczka mnożenie na palcach dla przykładów powyżej 5 razy 5



Każdy uczeń musi nauczyć się tabliczki mnożenia do 10 razy 10. Jak można usprawnić proces nauki tej tabliczki? Wystarczy znać tylko 25% pierwszych przykładów, żeby można było sprawnie liczyć do 100. Tym wielkim uproszczeniem w nauce jest tabliczka mnożenia na palcach. Z czasem, jak już wyćwiczy się wszystkie przykłady, to ta metoda nie będzie potrzebna. Jednak zawsze w razie potrzeby można sprawdzić dany przykład czy dobrze podaliśmy wynik.

http://www.matematyczny-swiat.pl/2013/10/tabliczka-mnozenia-na-palcach.html

Liczenie zależy od systemu liczbowego

W systemie 16-kowym tabliczka mnożenia wygląda inaczej:

1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Ile jest 8 * 9 w systemie 16-kowym? 48!

KONWERSJE POMIĘDZY SYSTEMAMI LICZBOWYMI

Tą samą liczbę możemy zapisać w różnych systemach liczbowych, np.:

$$11111001011[2] = 2201220[3] = 133023[4] = 30440[5] = 13123[6] =$$
$$= 5550[7] = 3713[8] = 2656[9] = 1995[10] = 1554[11] = 11A3[12] =$$
$$= BA6[13] = A27[14] = 8D0[15] = 7CB[16]$$

Konwersje pomiędzy systemami liczbowymi dokonujemy przede wszystkim w związku z możliwością wykonywania operacji matematycznych i porównań w określonym systemie liczbowym przez komputer w systemie 2-kowym, a przez człowieka w systemie 10-nym.

Przekształcenie liczby z systemu 10-nego na system 2-kowy poprzez dzielenie:

1995 : 2 = 997 r 1
997 : 2 = 498 r 1
498 : 2 = 249 r 0
249 : 2 = 124 r 1
124 : 2 = 62 r 0
62 : 2 = 31 r 0
31 : 2 = 15 r 1
15 : 2 = 7 r 1
7 : 2 = 3 r 1
3 : 2 = 1 r 1
1 : 2 = 0 r 1

(1995)[10] $= (11111001011)[2]$

W przekształceniu odwrotnym wykonujemy mnożenie przez podstawę systemu, z którego dokonujemy przekształcenia:

$(11111001011)[2]=$
$((((((((((1*2)+1)*2+1)*2+1)*2+1)*2+0)*2+0)*2+1)*2+0)*2+1)*2+1)*2$
$= (1995)[10]$

KONWERSJE LICZB WYMIERNYCH

Liczby wymierne w obcym systemie najpierw przekształcamy na iloraz liczb całkowitych, a następnie dokonujemy ich konwersji do znanego systemu, gdzie dokonujemy dzielenia:

$$(0,01101)[2] = \frac{(1101)[2]}{(100000)[2]} = \frac{(13)[10]}{(32)[10]} = (0,40625)[10]$$

W przypadku odwrotnego przekształcenia mnożymy przekształcaną liczbę przez podstawę systemu docelowego:

$$0,40625 * 2 = (0),81250$$
$$0,81250 * 2 = (1),62500$$
$$0,62500 * 2 = (1),25000$$
$$0,25000 * 2 = (0),50000$$
$$0,50000 * 2 = (1).00000$$

Takie liczby jak 0,45 czy 0,56 nie posiadają skończonego rozwinięcia dwójkowego, więc dalsze operacje wykonywana na takich liczbach mogą prowadzić do błędów numerycznych na skutek zaokrąglenia!

BIBLIOGRAFIA I LITERATURA UZUPEŁNIAJĄCA

1. L. Banachowski, K. Diks, W. Rytter: „Algorytmy i struktury danych”, WNT, Warszawa, 2001.
2. Z. Fortuna, B. Macukow, J. Wąsowski: „Metody numeryczne”, WNT, Warszawa, 1993.
3. J. i M. Jankowscy: „Przegląd metod i algorytmów numerycznych”, WNT, Warszawa, 1988.
4. A. Kiełbasiński, H. Schwetlick: „Numeryczna algebra liniowa”, WNT, Warszawa 1992.
5. M. Sysło: „Elementy Informatyki”.
6. A. Szepietowski: „Podstawy Informatyki”.
7. R. Tadeusiewicz, P. Moszner, A. Szydełko: „Teoretyczne podstawy informatyki”.
8. W. M. Turski: „Propedeutyka informatyki”.
9. N. Wirth: „Wstęp do programowania systematycznego”.
10. N. Wirth: „ALGORYTMY + STRUKTURY DANYCH = PROGRAMY”.