

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ○○○ ○○○○	

Kodowanie informacji

Reprezentacja liczb

prof. dr hab. inż. Joanna Józefowska

Poznań, rok akademicki 2010/2011

prof. dr hab. inż. Joanna Józefowska

Kodowanie informacji

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	● ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ○○○ ○○○○	

Reprezentacja liczb

- Stałoprzecinkowa
 - bez znaku
 - ze znakiem
- Zmiennoprzecinkowa
 - pojedynczej precyzji
 - podwójnej precyzji
 - rozszerzonej precyzji

prof. dr hab. inż. Joanna Józefowska

Kodowanie informacji

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ○○○ ○○○○	

Plan wykładu

- 1 Plan wykładu
- 2 Liczby naturalne
 - Klasyfikacja
 - Kod BCD
 - Dodawanie w systemie BCD
- 3 Liczby całkowite
 - Kod ZM
 - Kod U1
 - Kod U2
- 4 Liczby rzeczywiste
 - Podejście stałoprzecinkowe
 - Podejście zmiennoprzecinkowe
 - Błędy zaokrągleń
- 5 Pytania

prof. dr hab. inż. Joanna Józefowska

Kodowanie informacji

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ○○○ ○○○○	

Reprezentacja binarna

- Liczby naturalne przechowywane są dokładnie tak, jak zapisalibyśmy je w systemie dwójkowym.
- Każdy bit stanowi pojedynczą cyfrę w tym systemie.
- Zapis binarny prosty pozwala za pomocą n cyfr zapisać liczby z zakresu: $0 \leq L_{10} \leq 2^n - 1$.
 - 1 bajt to 2^8 kombinacji, czyli zakres 0-255,
 - 2 bajty to 2^{16} kombinacji, czyli zakres 0-65536,
 - 4 bajty to ponad 4 miliardy kombinacji.

prof. dr hab. inż. Joanna Józefowska

Kodowanie informacji

Kod BCD

Systemy inne niż dziesiętny nie są wygodne dla człowieka.

- Ile wynosi $1/3$ z liczby 111100_2 ?
- Aby ułatwić konwersję między systemem dwójkowym i dziesiętnym wprowadzono system kodowania BCD (ang. Binary-Coded Decimal).
- W tym systemie liczby dziesiętne kodujemy za pomocą ciągu bitów przypisując każdej cyfrze dziesiętnej odpowiadający jej ciąg 4 bitów.
- Ponieważ mamy jedynie 10 możliwości więc wystarczy zapamiętać tylko tyle, aby móc sprawnie i szybko przechodzić z systemu dziesiętnego do BCD i odwrotnie.

Cyfry systemu dziesiętnego kodzie BCD

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

Znaki nadmiarowe!

Przykład kodowania BCD

$$120_{10} = 1111000_2$$

$$\begin{aligned} 120 &= 2 * 60 + 0 \\ 60 &= 2 * 30 + 0 \\ 30 &= 2 * 15 + 0 \\ 15 &= 2 * 7 + 1 \\ 7 &= 2 * 3 + 1 \\ 3 &= 2 * 1 + 1 \\ 1 &= 2 * 0 + 0 \end{aligned}$$

$$120_{10} = 0001\ 0010\ 0000 = 000100100000_{BCD}$$

Własności kodu BCD

- Pomimo, iż kod BCD do zapisu używa cyfr dwójkowych, to otrzymany ciąg różni się od odpowiadającej liczby w systemie dwójkowym.
- Konwersja $10 \leftarrow BCD$ lub $BCD \leftarrow 10$ odbywa się na podobnych zasadach jak konwersja $2 \leftarrow 16$ lub $16 \leftarrow 2$: grupujemy bity po 4.
- Kod BCD wymaga więcej pamięci niż tradycyjny zapis w systemie dwójkowym.
- Konwersja jest naturalna i odbywa się prawie natychmiast.
- Kod BCD znajduje zastosowanie w urządzeniach elektronicznych z wyświetlaczem cyfrowym (np. w kalkulatorach, miernikach cyfrowych), w zastosowaniach finansowych informatyki (ujednoznacznia zapis części ułamkowych kwot i ułatwia dziesiętne zaokrąglanie).

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ●○○	○○○○○ ○○○○○ ○○○○○ ○○○○○	○ ○○○○ ○○○ ○○○	

Dodawanie bez przeniesienia

$$142_{10} = 000101000010_{BCD}$$

$$311_{10} = 001100010001_{BCD} +$$

$$453_{10} = 010001010011_{BCD}$$

$$\begin{array}{ccc} 0100 & 0101 & 0011 \\ 4 & 5 & 3 \end{array}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ●○○	○○○○○ ○○○○○ ○○○○○ ○○○○○	○ ○○○○ ○○○ ○○○	

Dodawanie z przeniesieniem

Jeżeli w wyniku dodawania otrzymamy czwórkę, która nie ma poprawnej interpretacji w kodzie BCD, to wykonujemy **korekcję**, która polega na dodaniu do tej czwórki bitów liczby $6_{10} = 0110_{BCD}$.

Na przykład:

$$9_{10} = 1001_{BCD}$$

$$6_{10} = 0110_{BCD} +$$

$$15_{10} = 1111_{BCD} ????$$

$$6_{10} = 0110_{BCD} +$$

$$15_{10} = 10101 = 0001\ 0101_{BCD}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○●	○○○○○ ○○○○○ ○○○○○ ○○○○○	○ ○○○○ ○○○ ○○○	

Dodawanie z przeniesieniem

$$60_{10} = 01100000_{BCD}$$

$$55_{10} = 01010101_{BCD} +$$

$$\begin{array}{ccc} & 10110101 & \\ 6_{10} & = 0110 & + \text{(korekcja)} \end{array}$$

$$115_{10} = 000100010101_{BCD}$$

$$\begin{array}{ccc} 0001 & 0001 & 0101 \\ 1 & 1 & 5 \end{array}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	●○○○○ ○○○○○ ○○○○○ ○○○○○	○ ○○○○ ○○○ ○○○	

Kodowanie liczb całkowitych

- Jak kodowany jest znak liczby?
- Istnieją trzy główne sposoby kodowania liczb ze znakiem:
 - kod ZNAK-MODUŁ,
 - kod U1,
 - kod U2.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	o o ooo	o●oooo oooooo ooooooo	o ooooo ooo oooo	

Kod Z-M

- Kod znak-moduł występuje też pod nazwami: ZM, Z-M, SM (Signed Magnitude), S+M.
- W kodzie ZM:
 - najstarszy bit liczby jest bitem znaku,
 - wszystkie bity liczby poza najstarszym mają takie samo znaczenie jak w kodzie binarnym.
- Jeśli przyjmimy do zapisu liczb format całkowity n-bitowy, to liczby w kodzie znak-moduł mają następującą postać:

$$\begin{array}{c}
 c_{n-1} \quad c_{n-2} c_{n-3} \dots c_2 c_1 c_0 \\
 \underbrace{\hspace{1cm}} \quad \underbrace{\hspace{1cm}} \\
 \text{znak} \quad \quad \text{moduł}
 \end{array}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	o o ooo	o●oooo oooooo ooooooo	o ooooo ooo oooo	

Kod Z-M

Wartość n-bitowej, całkowitej liczby w kodzie ZM obliczamy wg wzoru:

$$L_{ZM} = (-1)^{\text{bit znaku}} \times \text{moduł liczby}$$

Rozpisując bity otrzymujemy:

$$b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 = (-1)^{b_{n-1}} + b_{n-2}2^{n-2} + \dots + b_12^1 + b_02^0$$

gdzie:

gdzie b - bit, cyfra dwójkowa 0 lub 1,

n - liczba bitów w zapisie liczby,

bit znaku: 0 – liczba dodatnia, 1 – liczba ujemna.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	o o ooo	ooo●ooo oooooo ooooooo	o ooooo ooo oooo	

Zamiana liczb w systemie dziesiętnym na kod Z-M

- 1 Jeśli liczba jest dodatnia, to bit znaku ma wartość 0. W przeciwnym razie bit znaku ma wartość 1.
- 2 Obliczamy wartość bezwzględną liczby, czyli jej moduł.
- 3 Wyznaczamy bity modułu przeliczając liczbę dziesiętną na zapis dwójkowy.
- 4 Otrzymane bity modułu uzupełniamy w miarę potrzeby bitami o wartości 0, aby otrzymać ustaloną w formacie liczbę bitów dla modułu.
- 5 Do bitów modułu dodajemy bit znaku i otrzymujemy zapis Z-M.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	o o ooo	ooo●ooo oooooo ooooooo	o ooooo ooo oooo	

Przykłady

- 1 Zmieniamy liczbę dodatnią 93:
 - 1 Zmieniamy na liczbę binarną: $93_{10} = 1011101_2$
 - 2 Dodajemy bit znaku: $93_{10} = 01011101_{ZM}$
- 2 Zmieniamy liczbę ujemną -93:
 - 1 Zmieniamy moduł liczby na liczbę binarną: $|-93_{10}| = 93_{10} = 1011101_2$
 - 2 Dodajemy bit znaku: $93_{10} = 11011101_{ZM}$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○● ○○○○○ ○○○○○ ○○○○○	○ ○○○○ ○ ○○ ○○○	

Własności kodu Z-M

- 1 Zakres liczb reprezentowanych w n -bitowym kodzie Z-M:
 $-2^{n-1} + 1 \leq L_{ZM} \leq 2^{n-1} - 1$.
- 2 Kod Z-M jest koncepcyjnie prosty, lecz stwarza poważne problemy przy wykonywaniu operacji arytmetycznych.
- 3 Niejednoznaczność!!! - wartość 0 posiada dwa słowa kodowe: 0000 oraz 1000.
- 4 Bit znakowy posiada zupełnie inne znaczenie od pozostałych bitów i nie uczestniczy bezpośrednio w operacjach arytmetycznych.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ●○○○○ ○○○○○	○ ○○○○ ○ ○○ ○○○	

System uzupełnień do 1 - U1

- U1 (ang. 1C - One's Complement) - wszystkie bity zapisu liczby posiadają swoje wagi.
- Najstarszy bit jest bitem znaku i ma wagę równą $(-2^{n-1} + 1)$, gdzie n oznacza liczbę bitów w zapisie liczby.
- Pozostałe bity posiadają wagi takie jak w systemie binarnym.

Waga	$(-2^{n-1} + 1)$	(2^{n-2})	(2^{n-3})	...	2^2	2^1	2^0
Cyfra	b_{n-1}	b_{n-2}	b_{n-3}	...	b_2	b_1	b_0

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ●○○○○ ○○○○○	○ ○○○○ ○ ○○ ○○○	

Wartość liczby w systemie U1

Wartość liczby U1 obliczamy tak jak w systemie pozycyjnym - cyfry mnożymy przez wagi pozycji, na których się znajdują i dodajemy otrzymane iloczyny:

$$(b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0)_{U1} = b_{n-1}(-2^{n-1} + 1) + b_{n-2}2^{n-2} + b_{n-3}2^{n-3} + \dots + b_22^2 + b_12^1 + b_02^0$$

gdzie:

b - bit, cyfra dwójkowa 0 lub 1,

n - liczba bitów w zapisie liczby,

Najstarszy bit jest bitem znaku: 0 - liczba dodatnia, 1 - liczba ujemna.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ●○○○○ ○○○○○	○ ○○○○ ○ ○○ ○○○	

Własności systemu U1

- Wartość 0 posiada dwa słowa kodowe: 0000 oraz 1111.
- Liczba przeciwna zawsze powstaje w kodzie U1 przez negację wszystkich bitów. Zasada ta obowiązuje dla kodów U1 o dowolnej długości.
- W systemie U1 wykonywanie operacji arytmetycznych wciąż wymaga dodatkowych założeń.
- Przy dodawaniu do wyniku należy dodać przeniesienie poza bit znaku, aby otrzymać poprawny wynik.

$$\begin{aligned} 1_{10} &= 0001_{U1} & (-1)_{10} &= 1110_{U1} \\ 5_{10} &= 0101_{U1} & (-5)_{10} &= 1010_{U1} \\ 7_{10} &= 0111_{U1} & (-7)_{10} &= 1000_{U1} \end{aligned}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○	○○○○○ ○○○○○● ○○○○○○○	○ ○○○○ ○ ○○ ○○○	

Przeliczanie liczb dodatnich na zapis U1

- 1 Jeśli liczba jest dodatnia, znajdujemy jej reprezentację w naturalnym kodzie binarnym i uzupełniamy bitami o wartości 0 do uzyskania zadanej liczby bitów wymaganej przez przyjęty format zapisu U1.

Przykład: Wyznaczyć 8-mio bitowy zapis U1 liczby dziesiętnej 76.

$$76_{10} = 1001100_2$$

$$76_{10} = 01001100_{U1}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○	○○○○○ ○○○○○● ○○○○○○○	○ ○○○○ ○ ○○ ○○○	

Przeliczanie liczb ujemnych na zapis U1 - Wariant 1

- 1 Jeśli liczba jest ujemna, obliczamy jej moduł.
- 2 Moduł przedstawiamy w naturalnym systemie dwójkowym uzupełniając go bitami o wartości 0 do długości przyjętego formatu U1.
- 3 Następnie wszystkie bity zamieniamy na przeciwne i w wyniku otrzymujemy zapis U1.

Przykład: Wyznaczyć 8-mio bitowy zapis U1 liczby dziesiętnej -113.

$$|-113| = 113$$

$$113_{10} = 01110001_2$$

$$-113_{10} = NOT 01110001 = 10001110_{U1}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○	○○○○○ ○○○○○● ○○○○○○○	○ ○○○○ ○ ○○ ○○○	

Przeliczanie liczb ujemnych na zapis U1 - Wariant 2

- 1 Wyznaczamy wartość $2^n - 1$ + liczba, gdzie n oznacza liczbę bitów w przyjętym formacie U1.
- 2 Wartość tę kodujemy w naturalnym systemie dwójkowym i otrzymujemy kod U1 liczby wyjściowej.

Przykład: Wyznaczyć 8-mio bitowy zapis U1 liczby dziesiętnej -113.

$$2^8 - 1 - 113 = 255 - 1 - 113 = 142$$

$$142_{10} = 10001110_2$$

$$-113_{10} = 10001110_{U1}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○	○○○○○ ○○○○○● ●○○○○○	○ ○○○○ ○ ○○ ○○○	

System uzupełnień do 2 - U2

- Zmieniamy wagę bitu znakowego z $-2^{n-1} + 1$ na -2^{n-1} , gdzie n oznacza liczbę bitów w formacie kodu.
- Wagi stają się teraz jednorodne - bit znakowy posiada wagę ujemną, lecz o wartości bezwzględnej takiej samej jak w systemie binarnym.
- Nowy kod nosi nazwę uzupełnień do podstawy 2 lub w skrócie U2 (ang. 2C - Two's Complement).

Waga	(-2^{n-1})	(2^{n-2})	(2^{n-3})	...	2^2	2^1	2^0
Cyfra	b_{n-1}	b_{n-2}	b_{n-3}	...	b_2	b_1	b_0

Wartość liczby w systemie U2

Wartość liczby U2 obliczamy tak jak w systemie pozycyjnym - cyfry mnożymy przez wagi pozycji, na których się znajdują i dodajemy otrzymane iloczyny:

$$(b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0)_{U2} = b_{n-1}(-2^{n-1}) + b_{n-2}2^{n-2} + b_{n-3}2^{n-3} + \dots + b_22^2 + b_12^1 + b_02^0$$

gdzie:

b - bit, cyfra dwójkowa 0 lub 1,

n - liczba bitów w zapisie liczby.

Liczba przeciwna w kodzie U2

- 1 Przejść do pierwszego od prawej strony bitu zapisu liczby.
- 2 Do wyniku przepisać kolejne bity 0, aż do napotkania bitu o wartości 1, który również przepisać.
- 3 Wszystkie pozostałe bity przepisać zmieniając ich wartość na przeciwną.

Przykład. Znaleźć liczbę przeciwną do liczby:

```
1100100010111010111010010100001000000U2
001101110100010100010110101111000000
```

Wartość liczby w systemie U2

Jeśli bit znaku jest równy 0, to liczba jest dodatnia i resztę zapisu możemy potraktować jak liczbę w systemie dwójkowym.

Przykład

$$01101011_{U2} = 64 + 32 + 8 + 2 + 1 = 107_{10}$$

Jeśli bit znaku ustawiony jest na 1, to liczba ma wartość ujemną. Bit znaku ma wagę (-2^{n-1}) , gdzie n oznacza liczbę bitów w wybranym formacie U2. Reszta bitów jest zwykłą liczbą w naturalnym kodzie dwójkowym. Wagę bitu znakowego i wartość pozostałych bitów sumujemy otrzymując wartość liczby U2.

Przykład

$$11101011_{U2} = (-2^7) + 64 + 32 + 8 + 2 + 1 = -128 + 107 = (-21)_{10}$$

Przeliczanie dodatnich liczb dziesiętnych na zapis U2

- 1 Znaleźć reprezentację dwójkową danej wartości liczbowej.
- 2 Uzupełnić ją bitami 0 do długości formatu kodu U2.

Przykład. Wyznaczyć 8-mio bitowy kod U2 dla liczby 27₁₀.

$$27_{10} = 16 + 8 + 2 + 1 = 11011_2 = 00011011_{U2}$$

Przeliczanie ujemnych liczb dziesiętnych na zapis U2

- Jeśli do liczby 2^n (n - liczba bitów w formacie U2) dodamy przetwarzaną liczbę dziesiętną, to w wyniku otrzymamy wartość kodu dwójkowego równoważnego bitowo (tzn. o takiej samej postaci) kodowi U2 przetwarzanej liczby.
- Wynik dodawania zapisujemy w postaci kodu binarnego.

Przykład. Wyznaczyć 8-mio bitowy kod U2 dla liczby -45_{10} .

$$2^8 + (-45) = 256 - 45 = 211 = 11010011_2.$$

$$(-45)_{10} = 11010011_{U2}$$

Reprezentacja liczb ze znakiem – porównanie kodów

Kod	NKB	Z-M	U1	U2
0000	0	0	0	0
0001	1	1	1	1
0010	2	2	2	2
0011	3	3	3	3
0100	4	4	4	4
0101	5	5	5	5
0110	6	6	6	6
0111	7	7	7	7
1000	8	0	-7	-8
1001	9	-1	-6	-7
1010	10	-2	-5	-6
1011	11	-3	-4	-5
1100	12	-4	-3	-4
1101	13	-5	-2	-3
1110	14	-6	-1	-2
1111	15	-7	0	-1

Kodowanie stałoprzecinkowe

- Zapis pozycyjny można w prosty sposób rozszerzyć na liczby ułamkowe wprowadzając pozycje o wagach ułamkowych.
- Zakładamy, że przecinek jest za prawym skrajnym (najmłodszym bitem).

Wagi pozycji	10^3	10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Cyfry zapisu	3	5	7	9	,	8	2	9	1	4
Numery pozycji	3	2	1	0		-1	-2	-3	-4	-5
	Część całkowita					Część ułamkowa				

System pozycyjno-wagowy

- Liczbę stałoprzecinkową L w systemie pozycyjnym o podstawie (bazie) p zapisujemy w postaci: $(a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m})_p$.
- Unikalną reprezentacją liczby o wartości L jest ciąg cyfr $\{a_{n-1}, \dots, a_1, a_0, a_{-1}, \dots, a_{-m}\}$ takich, że: p - podstawa systemu, a - kolejne cyfry w liczbie, i - indeks kolejnych cyfr.

Przykład: Obliczyć wartość liczby stałoprzecinkowej $213,132_4$.

$$\begin{aligned}
 &213,132_4 \\
 &= 2 \times 4^2 + 1 \times 4^1 + 3 \times 4^0 + 1 \times 4^{-1} + 3 \times 4^{-2} + 2 \times 4^{-3} \\
 &= 2 \times 16 + 1 \times 4 + 3 \times 1 + 1 \times 1/4 + 3 \times 1/16 + 2 \times 1/64 \\
 &= 32 + 4 + 3 + 1/4 + 3/16 + 2/64 \\
 &= 39 + 16/64 + 12/64 + 2/64 \\
 &= 39 \frac{30}{64}
 \end{aligned}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○	○○○○○	○ ●○○○	
	○○○	○○○○○	○○○	
	○○○	○○○○○	○○○	

Uproszczenie

Część ułamkową potraktujemy jako liczbę całkowitą, obliczymy jej wartość i pomnożymy ją przez wagę ostatniej cyfry zapisu stałoprzecinkowego.

Przykład: Obliczyć wartość liczby stałoprzecinkowej $0,4231_5$.

$$4231_5 = 4 \times 5^3 + 2 \times 5^2 + 3 \times 5^1 + 1 \times 5^0 = 4 \times 125 + 2 \times 25 + 3 \times 5 + 1 \times 1 = 500 + 50 + 15 + 1 = 566$$

$$5^{-4} = \frac{1}{625}$$

$$0,4231_5 = 566 \times \frac{1}{625} = \frac{566}{625}$$

Sprawdzenie:

$$0,4231_5 = 0 \times 5^0 + 4 \times 5^{-1} + 2 \times 5^{-2} + 3 \times 5^{-3} + 1 \times 5^{-4}$$

$$= 0 \times 1 + 4 \times \frac{1}{5} + 2 \times \frac{1}{25} + 3 \times \frac{1}{125} + 1 \times \frac{1}{625}$$

$$= 0 + \frac{4}{5} + \frac{2}{25} + \frac{3}{125} + \frac{1}{625}$$

$$= \frac{500}{625} + \frac{50}{625} + \frac{15}{625} + \frac{1}{625}$$

$$= \frac{566}{625}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○	○○○○○	○ ●○○○	
	○○○	○○○○○	○○○	
	○○○	○○○○○	○○○	

Algorytm Hornera obliczania wartości liczby stałoprzecinkowej

Oznaczenia:

n - liczba cyfr w części całkowitej,

m - liczba cyfr w części ułamkowej,

p - podstawa systemu pozycyjnego, w którym jest zapisana liczba,

c_i - cyfra stojąca na i -tej pozycji; pozycja o numerze 0 jest pierwszą pozycją od strony prawej

w - obliczana wartość liczby.

- ① $w \leftarrow 0$
- ② $i \leftarrow n - 1$
- ③ $w \leftarrow c_i + w \times p$
- ④ jeśli $i = -m$, to $w \leftarrow w \times p^{-m}$ i koniec.
- ⑤ $i \leftarrow i - 1$
- ⑥ wróć do punktu 3

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○	○○○○○	○ ○●○○	
	○○○	○○○○○	○○○	
	○○○	○○○○○	○○○	

Algorytm Hornera - przykład

- ① $w \leftarrow 0$
- ② $i \leftarrow n - 1$
- ③ $w \leftarrow c_i + w \times p$
- ④ jeśli $i = -m$, to $w \leftarrow w \times p^{-m}$ i koniec.
- ⑤ $i \leftarrow i - 1$
- ⑥ wróć do punktu 3

$$i \leftarrow 3 - 1 \quad i \leftarrow 2 - 1$$

$$i \leftarrow 1 - 1 \quad i \leftarrow 0 - 1$$

$$i \leftarrow -1 - 1 \quad i \leftarrow -2 - 1$$

Obliczyć za pomocą algorytmu Hornera wartość dziesiętną liczby $237,745$ ($p=8$).

$$w \leftarrow 0$$

$$w \leftarrow 2 + 0 \times 8 = 2$$

$$w \leftarrow 3 + 2 \times 8 = 19$$

$$w \leftarrow 7 + 19 \times 8 = 159$$

$$w \leftarrow 7 + 159 \times 8 = 1279 \text{ (od tego momentu występują cyfry części ułamkowej)}$$

$$w \leftarrow 4 + 1279 \times 8 = 10236$$

$$w \leftarrow 5 + 10236 \times 8 = 81893 \text{ (ostatnia cyfra, koniec)}$$

$$w \leftarrow 81893 \times 8^{-3} = 159 \frac{485}{512} = 159,947265625$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○	○○○○○	○ ○○○●	
	○○○	○○○○○	○○○	
	○○○	○○○○○	○○○	

Znajdowanie rozwinięcia liczby stałoprzecinkowej o podstawie p – przykład

Znaleźć rozwinięcie liczby $12 \frac{8}{9}$ na system trójkowy z dokładnością do dwóch miejsc po przecinku.

$$12 \frac{8}{9} \times 3^2 = 12 \frac{8}{9} \times 9 = 116 \text{ (mnożymy przez podstawę podniesioną do odpowiedniej potęgi)}$$

$$116/3 = 38 \text{ i reszta } 2$$

$$38/3 = 12 \text{ i reszta } 2$$

$$12/3 = 4 \text{ i reszta } 0$$

$$4/3 = 1 \text{ i reszta } 1$$

$$1/3 = 0 \text{ i reszta } 1 \text{ (koniec)}$$

Otrzymaliśmy kolejne cyfry 11022. Dwie ostatnie cyfry umieszczamy po przecinku.

$$\text{Ostatecznie: } 12 \frac{8}{9} = 110,22_3$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○● ○ ○○ ○○○	

Wady notacji stałoprzecinkowej

- Nie możemy prosto zapisywać na niewielkiej liczbie bajtów bardzo dużych i bardzo małych liczb.
- Jeśli chcemy rozszerzyć zakres i liczbę miejsc po przecinku musimy używać bardzo dużej liczby bajtów.
- Trudny do oszacowania błąd obliczeń.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ●○○○ ○○○	

Kodowanie zmiennoprzecinkowe

- Zapis bardzo dużych lub bardzo małych liczb w normalnej notacji pozycyjnej jest niewygodny ponieważ wymaga dużej liczby cyfr, np.
 - dwanaście bilionów: 12 000 000 000 000
 - trzydzieści trylionów: 30 000 000 000 000 000
 - jedna bilionowa: 0,000 000 000 001
- Znacznie prostsze jest przedstawienie powyższych liczb w postaci zmiennoprzecinkowej (ang. floating point numbers)
 - dwanaście bilionów: $12000000000000 = 1,2 \times 10^{13}$
 - trzydzieści trylionów: $30000000000000000 = 3,0 \times 10^{19}$
 - jedna bilionowa: $0,000000000001 = 1,0 \times 10^{-12}$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ○ ●○○ ○○○	

Zapis zmiennoprzecinkowy

Liczba zapisana w systemie zmiennoprzecinkowym składa się z dwóch części:

- liczby stałoprzecinkowej, której wartość bezwzględna jest mniejsza od wartości podstawy systemu pozycyjnego (dokładność),
- podstawy podniesionej do pewnej potęgi zwanej wykładnikiem lub cechą (zakres).

$$L = s \times m \times p^c$$

s – znak liczby (1 lub -1),
m – mantysa (liczba ułamkowa),
p – podstawa systemu,
c – wykładnik potęgowy, czyli cecha (liczba całkowita).

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ○ ○●○ ○○○	

Zapis zmiennoprzecinkowy

- Położenie przecinka w mantysie nie jest ustalone i może się dowolnie zmieniać.
- Następujące zapisy oznaczają tę samą liczbę:
 $325 \times 10^{20} = 32,5 \times 10^{21} = 3,25 \times 10^{22} = 0,325 \times 10^{24}$, itd.
- Zmiana położenia przecinka w mantysie wpływa na wartość cechy liczby.
 - Przesunięcie przecinka o 1 pozycję w lewo wymaga zwiększenia cechy o 1.
 - Przesunięcie przecinka o 1 pozycję w prawo wymaga zmniejszenia cechy o 1.

Przykład dla systemu dziesiętnego:

$$32.246 = 0.32246 \times 10^2$$

$$0.000183 = 0.183 \times 10^{-3}$$

$$-84239253.5 = -0.842392535 \times 10^8$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○	○○○○○ ○○○○○ ○○○○○	○ ○○○○ ○○○ ○○○	

Znormalizowana liczba zmiennoprzecinkowa

Ponieważ liczbę zmiennoprzecinkową można zapisywać na różne sposoby, przyjęto tzw. postać znormalizowaną.

Definicja

Znormalizowana liczba zmiennoprzecinkowa to taka, w której mantysa spełnia nierówność:

$$p > |m| \geq 1$$

Według tej definicji postacią znormalizowaną dla zapisów:

$$325 \times 10^{20} = 32,5 \times 10^{21} = 3,25 \times 10^{22} = 0,0325 \times 10^{24}$$

jest jedynie zapis $3,25 \times 10^{22}$.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○	○○○○○ ○○○○○ ○○○○○	○ ○○○○ ○○○ ○○○	

Zapis zmiennoprzecinkowy

- Zapisujemy mantysę przy wykładniku równym zero.
- Normalizujemy mantysę modyfikując wykładnik liczby.
- Dodatkowo może nastąpić obcięcie albo zaokrąglenie mantysy do zadanej liczby cyfr.

Przykład

$$(1523,169)_{10}$$

$$(1523,169 \times 10^0)_{10}$$

$$(1,523169 \times 10^3)_{10}$$

$$\text{Zaokrąglenie: } (1,5232 \times 10^3)_{10}$$

$$\text{Obcięcie: } (1,5231 \times 10^3)_{10}$$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○	○○○○○ ○○○○○ ○○○○○	○ ○○○○ ○○○ ○○○	

Zmiennoprzecinkowe liczby binarne

Liczby są kodowane w jednym słowie maszyny podzielonym na trzy części:

- znak liczby (zwykle 1 bit),
- jedna część (zwykle znacznie dłuższa) zawiera kod mantysy liczby i jej znak
- druga część (kilkubitowa) - kod cechy liczby i jej znak.

$$L = (-1)^s \times m \times 2^c$$

s – bit znaku (0 oznacza znak +, 1 znak -)

m – znormalizowana liczba zapisana jako binarna liczba naturalna

c – wykładnik potęgi, czyli informacja o ile miejsc przesuwamy przecinek; może być dodatnia lub ujemna; jest zapisana w kodzie uzupełnień do 2.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○	○○○○○ ○○○○○ ○○○○○	○ ○○○○ ○○○ ○○○	

Zmiennoprzecinkowe liczby binarne

- W praktycznych realizacjach zapisu zmiennoprzecinkowych przyjmuje się ograniczony zakres na mantysę i cechę,
- Reprezentacja jest znormalizowana, tzn. bity mantysy są przesunięte tak, aby pierwsze miejsce po przecinku zajmował pierwszy znak 1 ciągu mantysy. $(0.5)_{10} = |mantysa| < 1$
- Z powyższego powodu liczba w zapisie zmiennoprzecinkowym jest określona z pewną dokładnością i może występować tylko w określonym zakresie.

Błąd zaokrąglenia

Błąd zaokrąglenia

Wystąpi wtedy, gdy część wartości liczby zostaje utracona w wyniku braku wystarczającej długości mantysy.

Format IEEE 754

- Aby ujednolicić wyniki obliczeń numerycznych wykonywanych na różnych platformach sprzętowych, wprowadzono ściśle określony standard zapisu zmiennoprzecinkowego IEEE 754.
- Pełna nazwa standardu to: IEEE Standard for Binary Floating-Point Arithmetic (ANSI/IEE Std 754-1985).
- Obecnie praktycznie wszystkie implementacje sprzętowe liczb zmiennoprzecinkowych oparte są na tym standardzie.
- Standard IEEE 754 definiuje dwie podstawowe klasy liczb zmiennoprzecinkowych:
 - 32 bitowe - pojedynczej precyzji (ang. single-precision),
 - 64 bitowe - podwójnej precyzji (ang. double-precision).

Format	Bit znaku	Bity cechy	Bity mantysy
32 bity - pojedyncza precyzja	1 bit	8 bitów	23 bity
64 bity - podwójna precyzja	1 bit	11 bitów	52 bity

Format IEEE 754

- W standardzie IEEE 754 zdefiniowane zostały także inne klasy liczb zmiennoprzecinkowych:
 - pojedynczej rozszerzonej precyzji (ang. single-extended precision) – liczby ≥ 43 -bitowe, nie są powszechnie stosowane,
 - podwójnej rozszerzonej precyzji (ang. double-extended precision) – liczby ≥ 79 -bitowe, zazwyczaj implementowane jako 80-bitowe.
- Standard IEEE 754 definiuje także trzy formaty dwójkowe stałoprzecinkowe i jeden stałoprzecinkowy format dziesiętny BCD:
 - 16-bitowy format całkowity (ang. short integer),
 - 32-bitowy format całkowity (ang. integer),
 - 64-bitowy format całkowity (ang. extended integer),
 - 80-bitowy format dziesiętny BCD (kodowanie 18-cyfrowej liczby całkowitej dziesiętnej oraz znaku na najbardziej znaczącej pozycji),

Format IEEE 754

Standard IEEE 754 definiuje nie tylko sposób reprezentacji liczb, ale także:

- sposób reprezentacji specjalnych wartości, np. nieskończoności, zera,
- sposób wykonywania działań na liczbach zmiennoprzecinkowych,
- sposób zaokrąglania liczb,
- Norma IEEE 754 jest standardem dwójkowym, którym wymaga się, aby bazą wszystkich reprezentacji było 2.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ○○○ ○○○○	

Format IEEE 854

Istnieje także norma IEEE 854 będąca standardem uniwersalnym, niezależnym od bazy:

- obejmuje arytmetykę dwójkową i dziesiętną,
- sposób wykonywania działań na liczbach zmiennoprzecinkowych,
- nie precyzuje dokładnie przyporządkowania poszczególnych bitów ani sposobu kodowania liczb zmiennoprzecinkowych.

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ○○○ ○○○○	

Przykład

Zapisać liczbę 1984,0415 w kodzie zmiennoprzecinkowym pojedynczej precyzji

Przeliczamy liczbę 1984,0415:

$1984 = 11111000000$

$0,0415 = 0.00001010101$

$1984.0415 = 11111000000.00001010101$

Normalizacja 1.111100000000001010101

Uzupełniamy 11111000000 do 23 bitów: 11110000000000000000000

Cecha: przecinek przesunęliśmy o 10 miejsc w lewo, więc jako cechę trzeba będzie zapisać liczbę 10 w kodzie U2: $c = (01010)_{U2}$

Ostatecznie:

$1984,0415 = 0\ 01010\ 1111000000$

Plan wykładu	Liczby naturalne	Liczby całkowite	Liczby rzeczywiste	Pytania
	○ ○ ○○○	○○○○○ ○○○○○ ○○○○○○○	○ ○○○○ ○○○ ○○○○	

Pytania

- Zapisać liczbę w kodzie BCD.
- Wykonać dodawanie na liczbach w kodzie BCD.
- Zapisać liczbę dziesiętną w (ośmiobitowym) kodzie Z-M.
- Odczytać wartość dziesiętną liczby zapisanej w kodzie Z-M
- Wyznaczyć 16 bitowy zapis U1 liczby dziesiętnej -4521.
- Wyznaczyć liczbę przeciwną w kodzie U2 do liczby 01101110_{U2} . (Aby się przekonać czy wyliczona została poprawna wartość należy wyliczyć wartości dziesiętne obu liczb.)
- Obliczyć wartość liczby stałoprzecinkowej $537,462_8$.
- Znaleźć rozwinięcie liczby $1/10$ na system dwójkowy z dokładnością do 8 miejsc po przecinku.
- Obliczyć wartość zmiennoprzecinkowej liczby trójkowej $2,21 \times 10^{21}$.
- Zapisać liczbę 984,415 w kodzie zmiennoprzecinkowym pojedynczej precyzji