INFORMATYKA

- Zajęcia organizacyjne
- Arytmetyka komputerowa

http://www.infoceram.agh.edu.pl
http://home.agh.edu.pl/~grzesik/

KONSULTACJE

Zbigniew Grzesik

środa, $9^{\underline{00}}$ – $10^{\underline{00}}$; A-3, p. 21

tel.: 617-2491

e-mail: grzesik@agh.edu.pl

OSOBY PROWADZĄCE ZAJĘCIA

Wykłady:

Prof. dr hab. inż. Zbigniew Grzesik

Ćwiczenia:

Dr inż. Grzegorz Smoła

Dr inż. Mirosław Stygar

Dr inż. Marek Zajusz

ORGANIZACJA ZAJĘĆ

Wykład:

15h, tj. 2h lekcyjne przez 0.5 semestru

Ćwiczenia:

- 1 nieobecność nieusprawiedliwiona
- kolokwia

Egzamin:

- egzamin pisemny + ewentualnie część ustna
- termin zerowy na prawach I terminu

TEMATYKA WYKŁADÓW

- Arytmetyka komputerowa
- Algorytmy
- Pseudo kod
- Schematy blokowe
- Struktury danych
- Języki programowania
- VBA

LITERATURA PODSTAWOWA I UZUPEŁNIAJĄCA

- http://www.infoceram.agh.edu.pl
- Wirth N., Algorytmy + struktury danych = programy Cormen T. Leiserson C. Rivest R., Wprowadzenie do algorytmów
- Knuth D., Sztuka Programowania, tom I, II, III
- Wróblewski P., Algorytmy, struktury danych i techniki programowania
- David Bourg, „Excel w nauce i technice. Receptury”, Helion 2006
- D. A. McQuarrie, „Matematyka dla Przyrodników i Inżynierów”, tomy 1-3, PWN Warszawa 2006.

PRZYKŁADOWE PROGRAMY DO PROJEKTOWANIA W INŻYNIERII MATERIAŁOWEJ I ENERGETYCE

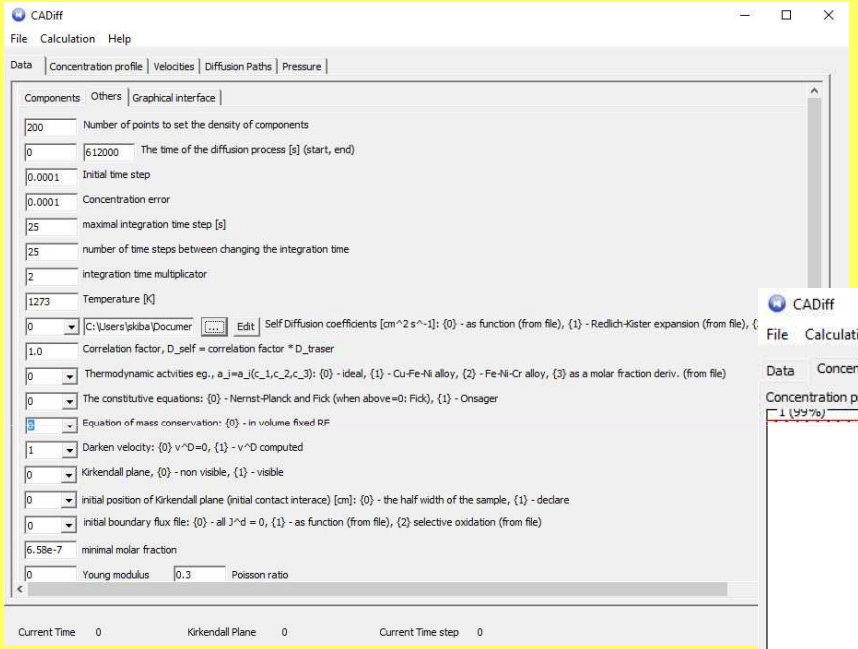
Modelowanie 1D i 2D:

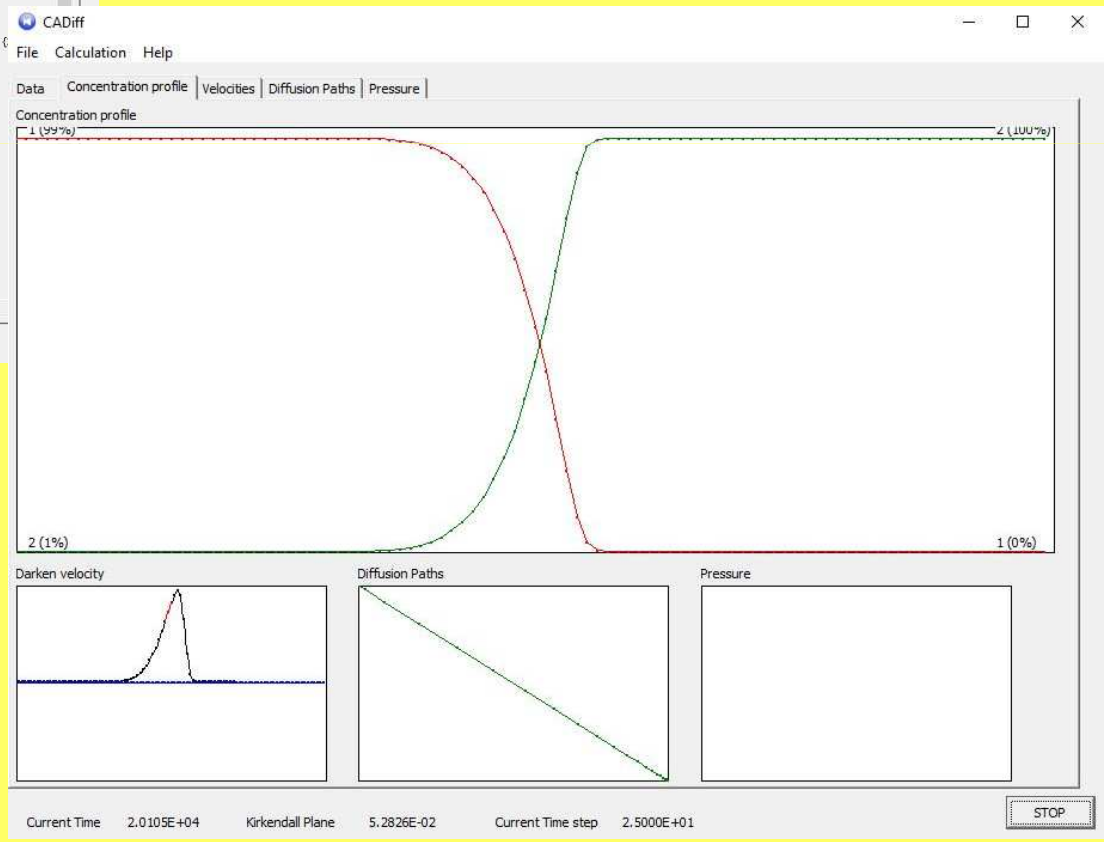
- CADiff (dyfuzja wzajemna)
- rCADiff (dyfuzja reakcyjna)
- MATLAB (zaawansowane środowisko matematyczne)
- MathCAD (zaawansowane środowisko matematyczne)

Modelowanie 1D, 2D i 3D:

- ANSYS Fluent (mechanika płynów, przepływ ciepła)
- Abaqus FEA (mechanika, przepływ ciepła)

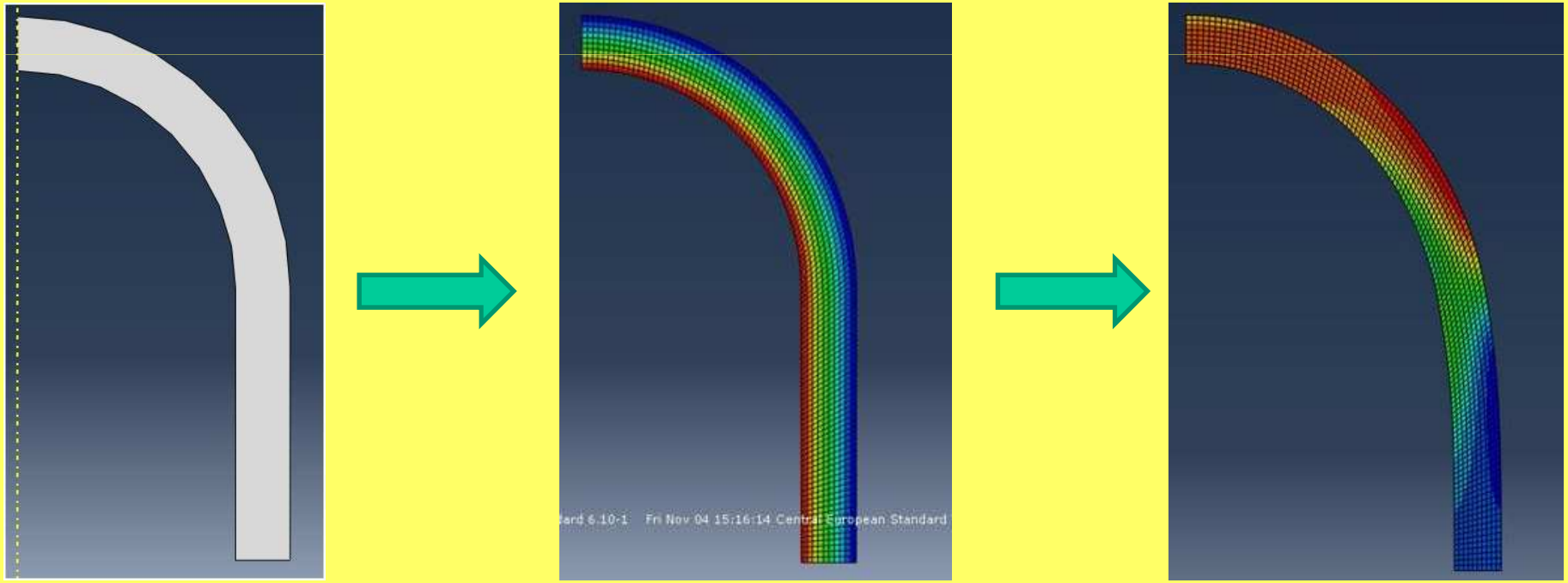
CADiff – dyfuzja wzajemna





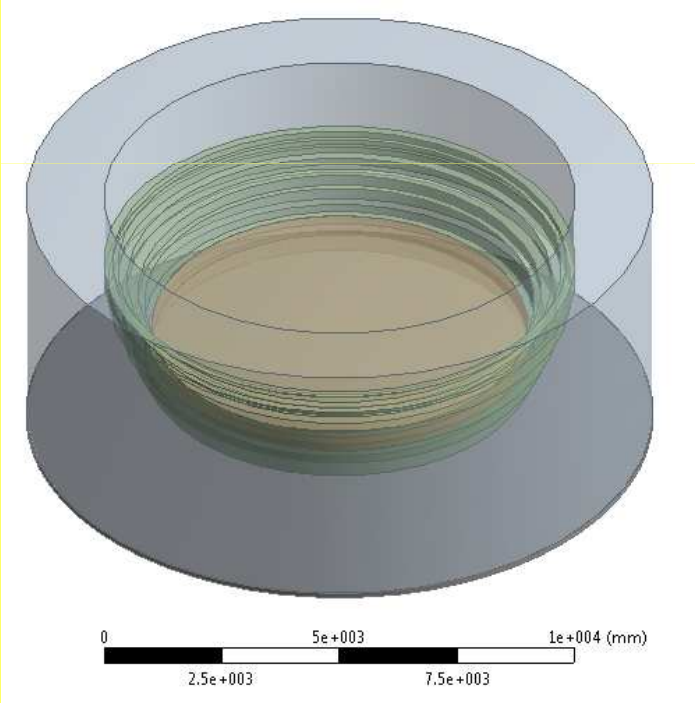
FEM – Abaqus FEA

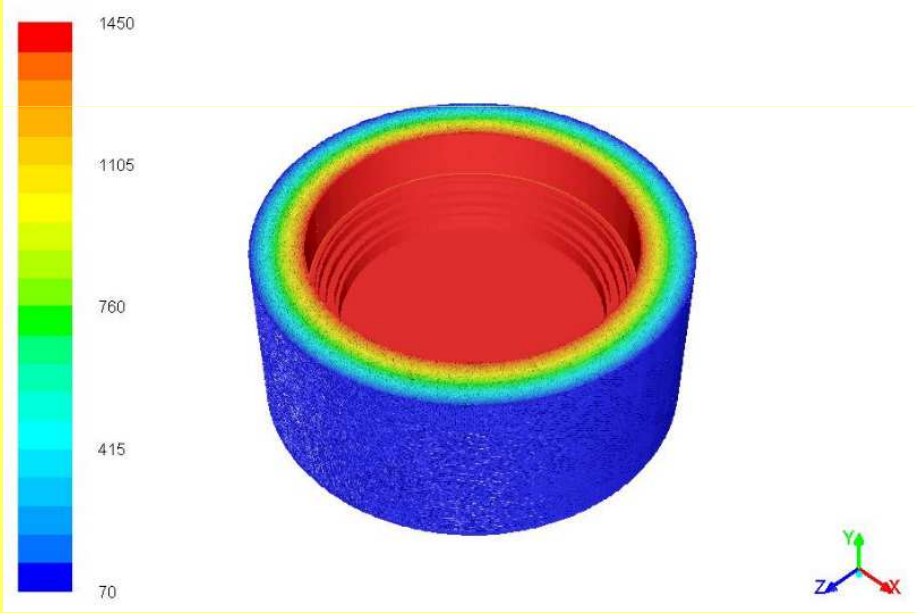
Obliczenie wzajemnej relacji między naprężeniami i dyfuzją wodoru z pojemnika – autor dr inż. Wojciech Skibiński



FVM – ANSYS Fluent

Obliczenie rozkładu temperatury dla wielkiego pieca (Australia, Port Kembla) – autor dr inż. Piotr Dziembaj





ARYTMETYKA KOMPUTEROWA
- kodowanie liczb w komputerze

DEFINICJA

Informatyka:

ogół dyscyplin naukowych i technicznych zajmujących się informacją, a w szczególności jej komputerowym przetwarzaniem.

Zakres:

- teorie informatyczne
- budowanie systemów informacyjnych, w tym programowanie
- budowanie i działanie sprzętu informatycznego
- zastosowanie metod informatycznych w różnych dziedzinach działalności ludzkiej

Podstawowe operacje maszyny cyfrowej

- Wszystkie informacje przetwarzane przez maszyny cyfrowe (komputery) muszą być odpowiednio zakodowane. Z powodów technicznych najwygodniej jest je przedstawiać w postaci ciągów podstawowych jednostek informacji (bitów), mogących przyjmować jeden z dwu stanów, zwyczajowo oznaczanych jako 0 i 1.
- **Bit** – najmniejsza jednostka informacji, może

  przyjmować jedną z dwóch wartości (znajdować się w jednym z dwóch stanów):

  0 lub **1**.
- **1 Bajt** = 8 bitów (oktet)
- 1 pojedyncze słowo = 16 bitów
- 1 Kb (kilobajt) = 1024 bajty
- 1 Mb (megabajt) = 1024 kilobajty

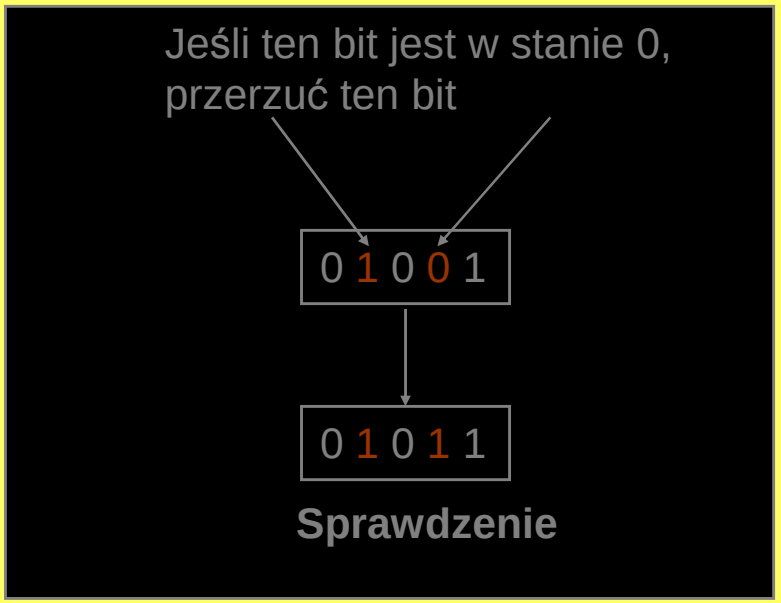
Architektura systemu komputerowego

Pamięć komputera złożona jest z szeregu pogrupowanych w porcje bitów (np. w przypadku komputera z ośmiobitowym procesorem zgrupowanych jest po osiem bitów). Jedne z najwcześniejszych procesorów firmy Intel z 1974 r. (8008 i 8080) były ośmiobitowe. Fragment ich pamięci można zilustrować w następujący sposób:

1	0	0	0	1	1	0	0
1	0	1	1	0	1	0	0
1	1	0	0	1	0	1	0
0	1	0	1	0	0	1	1

Podstawowe operacje maszyny cyfrowej

- Każda maszyna cyfrowa może wykonywać bezpośrednio tylko niewielką liczbę **operacji**:
  - przerzucenie bitu – zmienia wartość bitu
  - wyzerowanie bitu – przypisanie wartości 0
  - sprawdzenie bitu – wykonuje pewną czynność jeśli bit jest w stanie 0, a inną jeśli w stanie 1.



Kodowanie podstawowych pojęć w komputerze

Efektywne przetwarzanie danych, np. przy użyciu tzw. języków programowania stosowanych do pisania odpowiednich programów, wymaga możliwości używania szeregu abstrakcyjnych pojęć, takich jak liczby całkowite, liczby wymierne (z „częścią po przecinku dziesiętnym”), znaki tekstowe. Wszystkie te obiekty muszą być jednak zapamiętane w postaci ciągów zer i jedynek.

Kodowanie liczb w komputerze

Pomimo, iż z czysto matematycznego punktu widzenia zbiór liczb wymiernych stanowi rozszerzenie zbioru liczb całkowitych, to w informatyce stosowane są zupełnie inne sposoby kodowania dla liczb całkowitych i liczb wymiernych. Z powodów technicznych używa się podziału na dwa typy liczb: liczby *stałoprzecinkowe* i liczby *zmiennoprzecinkowe*.

Kodowanie liczb całkowitych

Liczby są kodowane w komputerach przy użyciu tzw. naturalnego kodu binarnego (NKB). Jest to kod o strukturze analogicznej do tej, stosowanej w życiu codziennym do zapisu liczb w tzw. systemie dziesiętnym, tj. systemie pozycyjnym o podstawie 10.

Kodowanie liczb

System pozycyjny:

metoda zapisywania liczb w taki sposób,
że w zależności od pozycji danej cyfry w ciągu,
oznacza ona wielokrotność potęgi pewnej liczby
uznawanej za bazę danego systemu.

Powszechnie używa się systemu dziesiętnego,
w którym za bazę przyjmuje się liczbę dziesięć.
Tym samym zapis 1946532 oznacza:

$$1 \times 10^6 + 9 \times 10^5 + 4 \times 10^4 + 6 \times 10^3 + 5 \times 10^2 + 3 \times 10^1 + 2 \times 10^0 = 1946532$$

Przykład liczby w systemie dwójkowym:

$$(01100110)_2 = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^6 + 0 \cdot 2^7 = 102.$$

Przeliczanie liczb z systemu dziesiętnego na dwójkowy

$524 \quad \rightarrow (\qquad)_2$

524	**262**	**0**
262	**131**	**0**
131	**65**	**1**
65	**32**	**1**
32	**16**	**0**
16	**8**	**0**
8	**4**	**0**
4	**2**	**0**
2	**1**	**0**
1	**0**	**1**

$524 \quad \rightarrow (1000001100)_2$

Sprawdzenie:

$$(1000001100)_2 = 0\cdot 2^0 + 0\cdot 2^1 + 1\cdot 2^2 + 1\cdot 2^3 + 0\cdot 2^4 + 0\cdot 2^5 + 0\cdot 2^6 + 0\cdot 2^7 + 0\cdot 2^8 + 1\cdot 2^9 = 4 + 8 + 512 = 524$$

Kodowanie ujemnych liczb całkowitych kod U1

$$-52 \rightarrow (\qquad)_2$$

$$52 \rightarrow (110100)_2$$

52	26	0
26	13	0
13	6	1
6	3	0
3	1	1
1	0	1

Sprawdzenie:

$$(110100)_2 = 0\cdot 2^0 + 0\cdot 2^1 + 1\cdot 2^2 + 0\cdot 2^3 + 1\cdot 2^4 + 1\cdot 2^5 = 4 + 16 + 32 = 52$$

Kod U1

$-52 \rightarrow (\quad\quad)_2$ $\qquad\qquad$ $52 \rightarrow (110100)_2$

Jednym ze stosowanych rozwiązań jest wprowadzenie tzw. bitu znaku: jest to pierwszy bit (licząc od lewej strony) w danym ciągu bitów (np. w ciągu ośmiu bitów dla procesorów ośmiobitowych), który nie informuje o wartości, lecz o znaku:

1 oznacza – (minus)

0 oznacza + (plus).

Taki sposób kodowania liczb ujemnych nazywany jest kodem uzupełnień do jeden (U1). Tak więc:

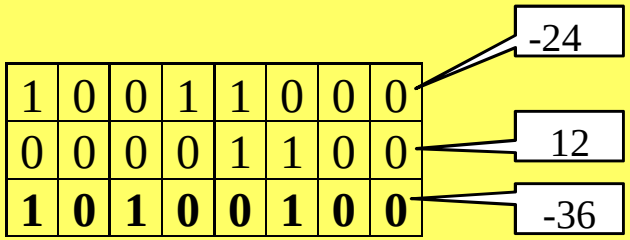
$-52 \rightarrow (10110100)_{U1}$

Kod U1

Należy podkreślić, iż ciąg znaków 10110100 jedynie w kodzie U1 oznacza liczbę -52, gdyż w kodzie NKB oznacza on liczbę 180.
Warto też zwrócić uwagę, iż liczbę 0 (zero) można przedstawić w kodzie U1 na dwa różne sposoby:
$(10000000)_{U1} = -0 = 0$
$(00000000)_{U1} = +0 = 0$

Kod U1

Kod U1 pomimo swej prostoty nie jest używany w praktyce, z powodu trudności z wykonywaniem operacji dodawania, czy mnożenia na liczbach zapisanych tym kodem. W celu dodania dwu liczb (np. -24 + 12) nie wystarczy zwykła suma bitowa z przeniesieniem:



Uzyskany wynik dodawania bitowego z przeniesieniem jest błędny, co oznacza że w przypadku liczb ujemnych kodowanych w systemie U1 należałoby wykorzystać bardziej skomplikowane algorytmy (zrealizowane sprzętowo lub programowo).

Kodowanie ujemnych liczb całkowitych – kod U2

Ciąg zer i jedynek o długości n zakodowanych w kodzie U2 oznacza liczbę, której wartość jest następująca:

$$(x_{n-1}x_{n-2}...x_1x_0)_{U2} = -x_{n-1}\cdot 2^{n-1} + x_{n-2}\cdot 2^{n-2} + ... + x_1\cdot 2^1 + x_0\cdot 2^0.$$

a zatem:

$$(x_{n-1}x_{n-2}...x_1x_0)_{U2} = \begin{cases} (x_{n-2}\ldots x_1x_0)_2 & \text{gdy } x_{n-1}=0, \\ -2^{n-1} + (x_{n-2}\ldots x_1x_0)_2 & \text{gdy } x_{n-1}=1. \end{cases}$$

Zalety kodu U2:

- jednoznaczna reprezentacja liczby 0
- łatwa realizacja dodawania i odejmowania

Przykłady liczb zakodowanych kodem U2

1	1	0	1	1	0	0	1
0	0	0	0	1	1	1	0
0	1	1	0	0	0	1	0
1	0	1	1	1	0	0	1
1	1	0	1	1	0	1	0

Wartości tych liczb:

$$(11011001)_{U2} = -2^7 + 1\cdot 2^6 + 0\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 =$$
$$= -128 + 64 + 16 + 8 + 1 = -39.$$

$$(00001110)_{U2} = -0\cdot 2^7 + 0\cdot 2^6 + 0\cdot 2^5 + 0\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 0\cdot 2^0 =$$
$$= 8 + 4 + 2 = 14.$$

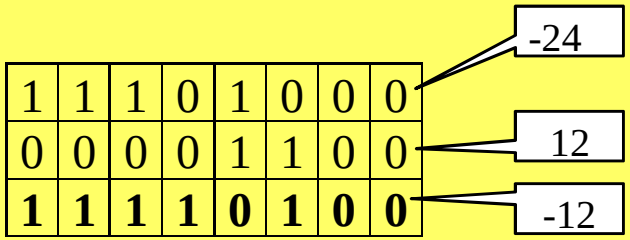
Liczby ujemne w kodzie U2 mają 1 na pierwszej (od lewej) pozycji.
Liczby dodatnie w kodzie U2 mają 0 na pierwszej (od lewej) pozycji.

Podstawowe własności kodu uzupełnień U2

- Najstarszy (pierwszy od lewej) bit informuje o znaku liczby. Bit 0 mają liczby nieujemne, bit 1 mają liczby ujemne. Np. $(1011)_{U2} = -2^3 + 2^1 + 2^0 = -5.$
- Zmiana znaku liczby zakodowanej w U2 dokonuje się przez negację poszczególnych bitów kodu i dodanie bitu 1 do najmłodszej (pierwszej od prawej) pozycji. Np. $(1011)_{U2} = -5.$ Zatem $\overline{(1011)}_{U2} + 1 = (0100)_{U2} + 1 = (0101)_{U2} = 2^2 + 2^0 = 5.$
- Powielanie najstarszego (pierwszego od lewej) bitu nie zmienia wartości zakodowanej liczby.
- Zakres liczb w kodzie U2 o długości n wynosi $[-2^{n-1}, 2^{n-1} - 1]$. Np. dla $n = 8$ daje to zakres $[-128, 127]$, a dla $n = 16$ zakres $[-32768, 32767]$.

Dodawanie dwóch liczb zapisanych w kodzie U2

Dodawanie dwóch liczb zapisanych w kodzie U2 (np. -24 + 12) można wykonać wykonując zwykłe dodawanie binarne z przeniesieniem.



Kodowanie liczb rzeczywistych

$$x = S \cdot M \cdot B^{C}$$

S – znak
M – mantysa
B – podstawa systemu
C – cecha, wykładnik

Kodowanie liczb rzeczywistych w układzie dziesiętnym
Liczby zmiennoprzecinkowe

$$x = S \cdot M \cdot 10^{C}$$

$$0{,}1 \leq M < 1$$

Przykład:

Przyjmijmy, że *B* = 10, liczba cyfr dziesiętnych przeznaczonych na mantysę wynosi 4, natomiast na cechę (wykładnik) 2.
Chcemy zapisać wartość **54,125367**.

Pierwszy etap to normalizacja mantysy, sytuacja przedstawia się następująco: ***M=54,125367; C=0***.
Mantysa *M* nie należy do zadanego przedziału ***[0,1;1)***, zatem należy przesuwać przecinek w lewo do chwili, aż wartość ***M*** będzie doń należała.
Przesuwanie przecinka w lewo wiąże się ze **zwiększaniem *C***.
Po normalizacji (przesunięciu przecinka o 2 pozycje w lewo) otrzymujemy:
M=0,54125367, C=2
Ostatnim krokiem jest odpowiednie **obcięcie** (ang. *truncate*), albo zaokrąglenie (ang. *round*) mantysy do zadanej ilości cyfr.

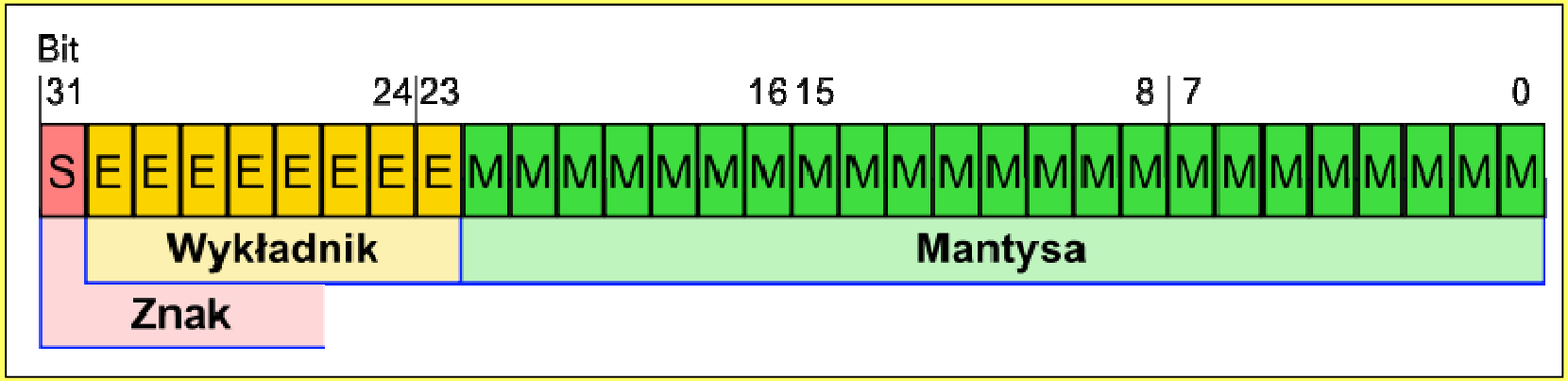
Kodowanie liczb rzeczywistych w układzie dwójkowym
Liczby zmiennoprzecinkowe

$$x = S \cdot M \cdot 2^{C}$$

$$2^{-1} \leq M < 1$$

Kodowanie liczb w komputerze

Liczby zmiennoprzecinkowe



Wartość liczby zmiennoprzecinkowej

$$x = S \cdot M \cdot B^{C}$$

S – znak
M – mantysa
B – podstawa systemu
C – cecha, wykładnik

Kodowanie znaków w komputerze

Zestaw znaków to zestawienie znaków pisma z odpowiadającymi im kodami liczbowymi. Zestawienie takie można następnie wykorzystać do przekształcenia tekstu na postać cyfrową, w szczególności w komputerze. Historycznie, istniało wiele różnych zestawów znaków.

Przykłady:

ASCII
Base32
Base64
EBCDIC
ISO 8859-2
JIS
KOI8-R
Szesnastkowy system liczbowy (Base16)
Unicode

Obecnie powszechny we wszystkich nowoczesnych aplikacjach i systemach operacyjnych jest międzynarodowy standard Unicode (najczęściej w połączeniu z kodowaniem UTF-8), zdolny przedstawić wszystkie pisma świata.

Kodowanie znaków w komputerze

ASCII (ang. *American Standard Code for Information Interchange*) – 7-bitowy kod przyporządkowujący liczby z zakresu 0-127: literom (alfabetu angielskiego), cyfrom, znakom przestankowym i innym symbolom oraz poleceniom sterującym. Na przykład litera "a" jest kodowana liczbą 97, a znak spacji jest kodowany liczbą 32.

Litery, cyfry oraz inne znaki drukowane tworzą zbiór znaków ASCII. Jest to 95 znaków o kodach 32-126. Pozostałe 33 kody (0-31 i 127) to tzw. kody sterujące służące do sterowania urządzeniem odbierającym komunikaty, np. drukarką czy terminalem.

KONIEC