



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



## **„Architektura komputerów” „Reprezentacja liczb. Kodowanie rozkazów.”**

Prezentacja jest współfinansowana przez  
Unię Europejską w ramach  
Europejskiego Funduszu Społecznego w projekcie pt.

*„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej -  
zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do  
zatrudniania osób niepełnosprawnych”*

Prezentacja dystrybuowana jest bezpłatnie





## Liczby zmiennoprzecinkowe

- Liczby zmiennoprzecinkowe są używane do reprezentacji liczb rzeczywistych
  - 1.23233, 0.0003002, 3323443898.3325358903
- Liczby zmiennoprzecinkowe to podzbiór zbioru liczb rzeczywistych
  - Zakres przechowywanych liczb jest ograniczony
    - Zależy od zastosowanej liczby bitów
  - Ograniczona precyzja
    - 12345678901234567890 --> 1234567890000000000000
    - Liczby zmiennoprzecinkowe to przybliżenie liczb rzeczywistych, podczas gdy liczby całkowite są reprezentowane dokładnie



## Notacja wykładnicza

$$+ 34.383 \times 10^2 = 3438.3$$

Znak      Mantysa      Wykładnik

$$+ 3.4383 \times 10^3 = 3438.3$$

Forma znormalizowana: jedna cyfra przed przecinkiem dziesiętnym

$$+3.4383000E+03 = 3438.3$$

Notacja zmiennoprzecinkowa

8-cyfrowa mantysa może reprezentować jedynie 8 cyfr znaczących



## Binarne liczby zmiennoprzecinkowe

 $+ 101.1101$ 

$$\begin{aligned} &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\ &= 4 + 0 + 1 + 1/2 + 1/4 + 0 + 1/16 \\ &= 5.8125 \end{aligned}$$

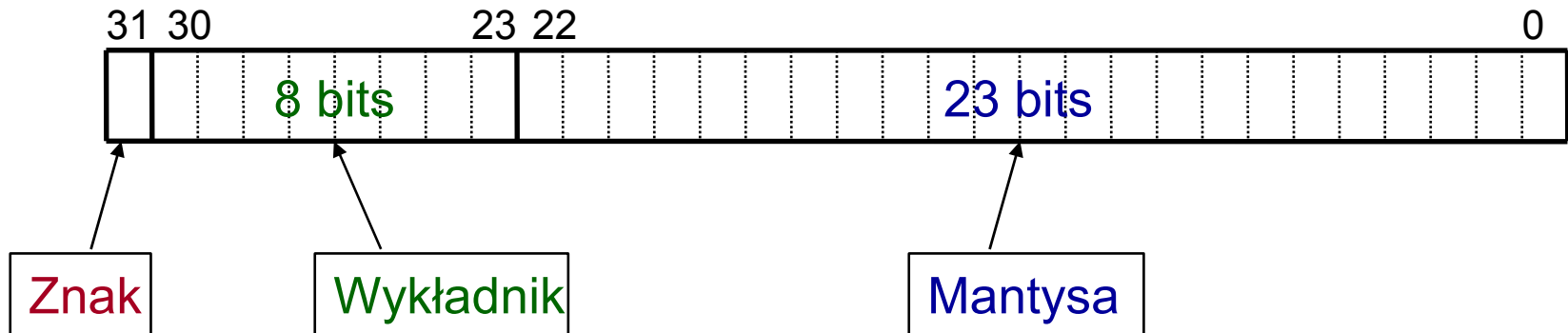
 $+1.011101 E+2$ 

Notacja znormalizowana, czyli taka, w której przecinek dwójkowy występuje zaraz za pierwszą cyfrą

Uwaga: pierwsza cyfra jest zawsze niezerowa  
--> pierwsza cyfra jest zawsze jedynką

## Format zmiennoprzecinkowy IEEE

The Institute of Electrical and Electronics Engineers



0: Dodatni  
1: Ujemny

Przesunięty o 127.

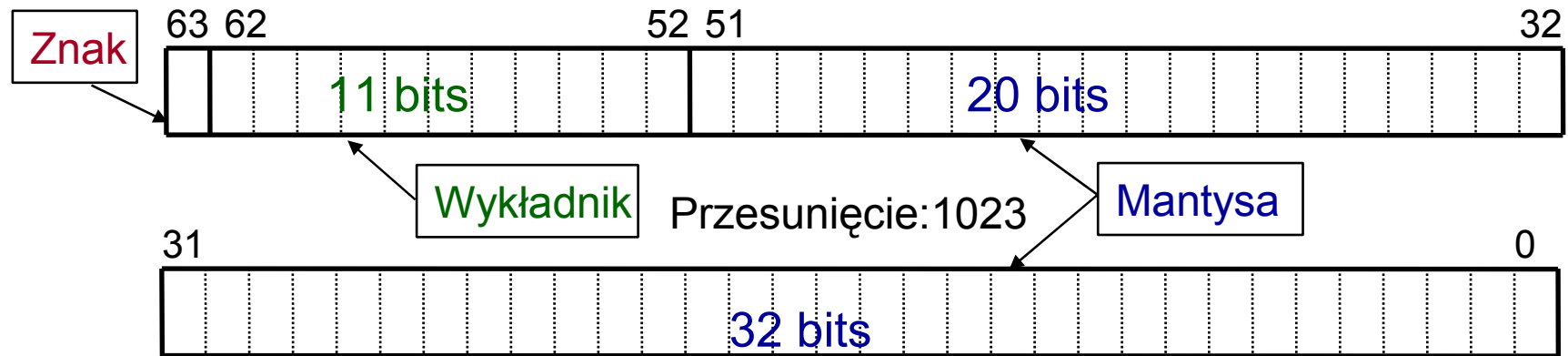
Wiodące '1' występuje niejawnie,  
nie jest reprezentowane w pamięci

$$\text{Liczba} = -1^{\text{S}} * (1 + \text{M}) \times 2^{\text{E}-127}$$

- Pozwala na reprezentację liczb w zakresie  $2^{-127}$  do  $2^{+128}$  ( $10^{\pm 38}$ )
- Ponieważ **mantysa** zawsze zaczyna się od '1', nie trzeba tego jawnie zapisywać
  - **Mantysa** ma efektywnie 24 bity



## Format IEEE o podwójnej precyzji



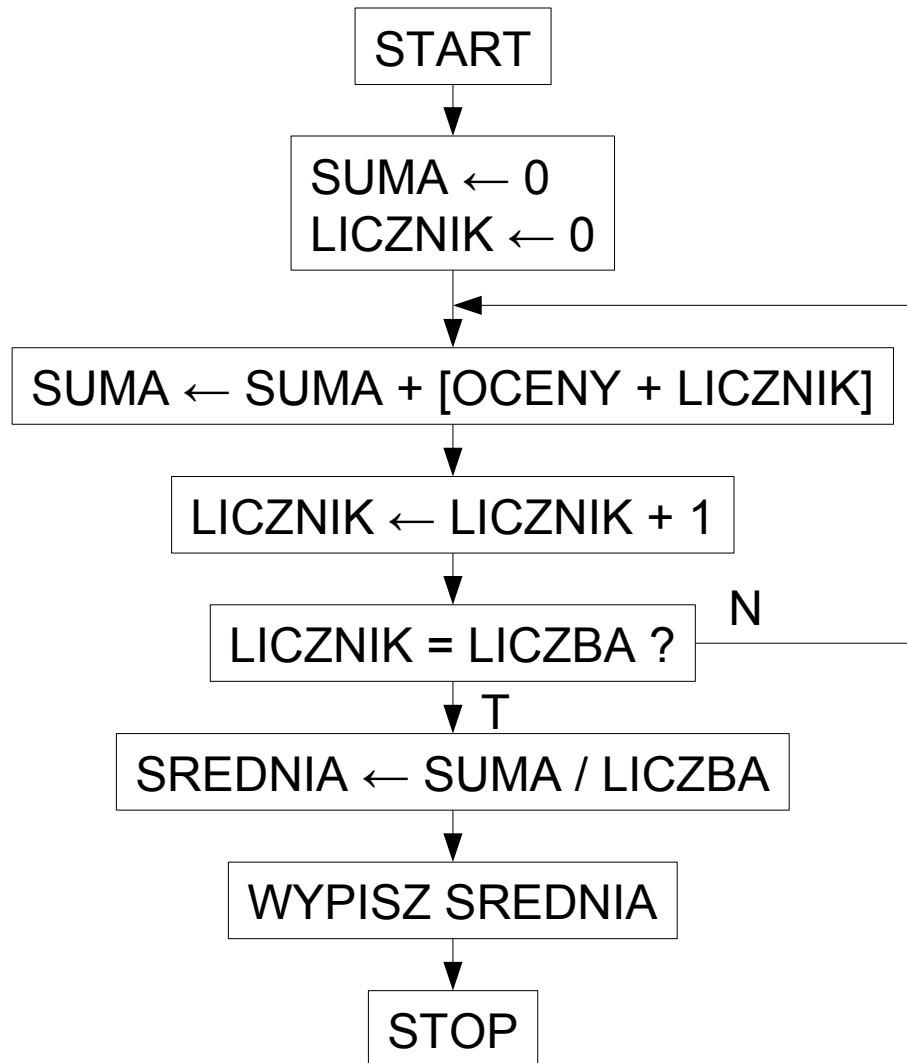
$$\text{Liczba} = -1^{\text{S}} * (1 + \text{M}) \times 2^{\text{E}-1023}$$

- Pozwala na reprezentację liczb w zakresie od  $2^{-1023}$  do  $2^{+1024}$  ( $10^{\pm 308}$ )
- Większa **mantysa** pozwala na większą precyzję





## Algorytm liczenia średniej



LICZBA: 3  
OCENY: 3.5  
4  
4.5



## Program po przetłumaczeniu na język procesora

1. WSTAW WARTOŚĆ 0
2. PRZENIEŚ DO LICZNIK
3. WSTAW WARTOŚĆ 0
4. PRZENIEŚ DO SUMA
5. L1: WSTAW WARTOŚĆ OCENY
6. DODAJ LICZNIK
7. UMIEŚĆ\*
8. DODAJ SUMA
9. PRZENIEŚ DO SUMA
10. WSTAW WARTOŚĆ 1
11. DODAJ LICZNIK
12. PRZENIEŚ DO LICZNIK
13. JEŚLI = LICZBA SKOCZ DO L2
14. SKOCZ DO L1
15. L2: UMIEŚĆ SUMA
16. PODZIEL PRZEZ LICZNIK
17. PRZENIEŚ DO ŚREDNIA
18. WYPISZ
19. STOP
20. ŚREDNIA:
21. SUMA:
22. LICZNIK:
23. LICZBA: 3
24. OCENY: 3.5
25. 4
26. 4.5







## Kodowanie rozkazów

- Aby rozkazy były zrozumiałe dla procesora, trzeba je zakodować w postaci liczb
- Zakładamy, że komórki pamięci mają po 8 bitów (1 bajt)
- Rozkazy bez parametrów będą zajmować jeden bajt, rozkazy zawierające daną lub adres dwa bajty, rozkazy zawierające dwa adresy trzy bajty
- Kody rozkazów i adresy zwykle zapisywane są szesnastkowo
- Postać taką nazywa się kodem maszynowym

Rozkaz	Kod
NIC NIE RÓB	00
WSTAW WARTOŚĆ	01 WARTOŚĆ
PRZENIEŚ DO	02 ADRES
UMIEŚĆ	03 ADRES
UMIEŚĆ*	04
DODAJ	05 ADRES
ODEJMIJ	06 ADRES
POMNÓŻ PRZEZ	07 ADRES
PODZIEL PRZEZ	08 ADRES
SKOCZ DO	09 ADRES
WYPISZ	0A
JEŚLI = SKOCZ DO	0B ADRES ADRES
JEŚLI ≠ SKOCZ DO	0C ADRES ADRES
JEŚLI > SKOCZ DO	0D ADRES ADRES
JEŚLI < SKOCZ DO	0E ADRES ADRES
JEŚLI ≥ SKOCZ DO	0F ADRES ADRES
JEŚLI ≤ SKOCZ DO	10 ADRES ADRES
STOP	11



- |                                |                           |
|--------------------------------|---------------------------|
| 1. WSTAW WARTOŚĆ 0             | 14. SKOCZ DO L1           |
| 2. PRZENIEŚ DO LICZNIK         | 15. L2: UMIEŚĆ SUMA       |
| 3. WSTAW WARTOŚĆ 0             | 16. PODZIEL PRZEZ LICZNIK |
| 4. PRZENIEŚ DO SUMA            | 17. PRZENIEŚ DO ŚREDNIA   |
| 5. L1: WSTAW WARTOŚĆ OCENY     | 18. WYPISZ                |
| 6. DODAJ LICZNIK               | 19. STOP                  |
| 7. UMIEŚĆ*                     | 20. ŚREDNIA:              |
| 8. DODAJ SUMA                  | 21. SUMA:                 |
| 9. PRZENIEŚ DO SUMA            | 22. LICZNIK:              |
| 10. WSTAW WARTOŚĆ 1            | 23. LICZBA: 3             |
| 11. DODAJ LICZNIK              | 24. OCENY: 3.5            |
| 12. PRZENIEŚ DO LICZNIK        | 25. 4                     |
| 13. JEŚLI = LICZBA SKOCZ DO L2 | 26. 4.5                   |

Rozkaz	Kod
NIC NIE RÓB	00
WSTAW WARTOŚĆ	01 WARTOŚĆ
PRZENIEŚ DO	02 ADRES
UMIEŚĆ	03 ADRES
UMIEŚĆ*	04
DODAJ	05 ADRES
ODEJMIJ	06 ADRES
POMNÓŻ PRZEZ	07 ADRES
PODZIEL PRZEZ	08 ADRES
SKOCZ DO	09 ADRES
WYPISZ	0A
JEŚLI = SKOCZ DO	0B ADRES ADRES
JEŚLI ≠ SKOCZ DO	0C ADRES ADRES
JEŚLI > SKOCZ DO	0D ADRES ADRES
JEŚLI < SKOCZ DO	0E ADRES ADRES
JEŚLI ≥ SKOCZ DO	0F ADRES ADRES
JEŚLI ≤ SKOCZ DO	10 ADRES ADRES
STOP	11





## Program po przetłumaczeniu na kod maszynowy (faza 1)

	01 00		02 ŚREDNIA
	02 LICZNIK		0A
	01 00		11
	02 SUMA	ŚREDNIA:	00
L1 :	01 OCENY	SUMA:	00
	05 LICZNIK	LICZNIK:	00
	04	LICZBA:	03
	05 SUMA	OCENY:	07
	02 SUMA		08
	01 01		09
	05 LICZNIK		
	02 LICZNIK		
	0B LICZBA L2		
	09 L1		
L2 :	03 SUMA		
	08 LICZNIK		





## Program po przetłumaczeniu na kod maszynowy (faza 2)

00	01 00	20	02 ŚREDNIA
02	02 LICZNIK	22	0A
04	01 00	23	11
06	02 SUMA	24 ŚREDNIA:	00
08 L1:	01 OCENY	25 SUMA:	00
0A	05 LICZNIK	26 LICZNIK:	00
0C	04	27 LICZBA:	03
0D	05 SUMA	28 OCENY:	07
0F	02 SUMA		08
11	01 01		09
13	05 LICZNIK		
15	02 LICZNIK		
17	0B LICZBA L2		
1A	09 L1		
1C L2:	03 SUMA		
1E	08 LICZNIK		





## Program po przetłumaczeniu na kod maszynowy (faza 3)

00	01 00	20	02 23
02	02 26	22	0A
04	01 00	23	11
06	02 25	24 ŚREDNIA:	00
08 L1:	01 28	25 SUMA:	00
0A	05 26	26 LICZNIK:	00
0C	04	27 LICZBA:	03
0D	05 25	28 OCENY:	07
0F	02 25		08
11	01 01		09
13	05 26		
15	02 26		
17	0B 27 1C		
1A	09 08		
1C L2:	03 25		
1E	08 26		





## Program po przetłumaczeniu na kod maszynowy (faza 4)

00 : 01 00 02 26 01 00 02 25  
08 : 01 28 05 26 04 05 25 02  
10 : 25 01 01 05 26 02 26 0B  
18 : 27 1C 09 08 03 25 08 26  
20 : 02 23 0A 11 00 00 00 03  
28 : 07 08 09



## Co robi poniższy program?

```
00: 01 00 02 24 01 01 02 21
08: 02 23 01 04 02 22 03 21
10: 05 21 02 21 03 22 06 23
18: 02 22 0c 24 0E 03 21 0a
20: 11 03 02 11 0d
```

Rozkaz	Kod
NIC NIE RÓB	00
WSTAW WARTOŚĆ	01 WARTOŚĆ
PRZENIEŚ DO	02 ADRES
UMIEŚĆ	03 ADRES
UMIEŚĆ*	04
DODAJ	05 ADRES
ODEJMIJ	06 ADRES
POMNÓŻ PRZEZ	07 ADRES
PODZIEL PRZEZ	08 ADRES
SKOCZ DO	09 ADRES
WYPISZ	0A
JEŚLI = SKOCZ DO	0B ADRES ADRES
JEŚLI ≠ SKOCZ DO	0C ADRES ADRES
JEŚLI > SKOCZ DO	0D ADRES ADRES
JEŚLI < SKOCZ DO	0E ADRES ADRES
JEŚLI ≥ SKOCZ DO	0F ADRES ADRES
JEŚLI ≤ SKOCZ DO	10 ADRES ADRES
STOP	11



## Model programowy procesora Z80

## Alternate Registers

Accumulator A	Flags F
B	C
D	E
H	L
Index Register IX	
Index Register IY	
Stack Pointer SP	
Program Counter PC	
Interrupt Vector I	Memory Refresh R

Accumulator A'	Flags F'
B'	C'
D'	E'
H'	L'

D7	D6	D5	D4	D3	D2	D1	D0
S	Z		H		P/V	N	C

S = Sign; Z = Zero; H = Half Carry

P/V = Parity/Overflow; N = Add/Subtract





**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



## **„Architektura komputerów” „Reprezentacja liczb. Kodowanie rozkazów.”**

Prezentacja jest współfinansowana przez  
Unię Europejską w ramach  
Europejskiego Funduszu Społecznego w projekcie pt.

*„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej -  
zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do  
zatrudniania osób niepełnosprawnych”*

Prezentacja dystrybuowana jest bezpłatnie

