



Universiteit Antwerpen
| Faculteit Wetenschappen

Computersystemen en -architectuur

Datarepresentatie

Kasper Engelen

kasper.engelen@uantwerpen.be

Academiejaar 2022 – 2023

Opdracht

- PDF met instructies (te vinden op de MSDL-website, onder “Data Representation”)
- <http://msdl.uantwerpen.be/people/hv/teaching/ComputerSystemsArchitecture>
- Individueel oplossen
- Oplossingen duidelijk opschrijven
- **Deadline:** donderdag 17 november 2022 22u00
- Indienen via **Blackboard**
- Telt mee voor eindtotaal
- Belangrijke voorbereiding voor het examen!

Naar base 10

Polynomial method

$$1100.01_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} = 12.25_{10}.$$

$$8325.71_9 = 8 \cdot 9^3 + 3 \cdot 9^2 + 2 \cdot 9^1 + 5 \cdot 9^0 + 7 \cdot 9^{-1} + 1 \cdot 9^{-2} = 6098.79_{10}$$

$$\dots b_3 b_2 b_1 b_0 . b_{-1} b_{-2} b_{-3} \dots$$

$$\dots + b_3 \cdot k^3 + b_2 \cdot k^2 + b_1 \cdot k^1 + b_0 \cdot k^0 + b_{-1} \cdot k^{-1} + b_{-2} \cdot k^{-2} + b_{-3} \cdot k^{-3} + \dots$$

Base 10 naar base 8

359.732₁₀

Remainder method

Quotiënt	Rest	Positie
359/8 = 44	7	LSB
44/8 = 5	4	
5/8 = 0	<u>5</u>	MSB

Multiplication method

$$0.732 \cdot 8 = \underline{5}.856$$

$$0.856 \cdot 8 = \mathbf{6}.848$$

$$0.848 \cdot 8 = \mathbf{6}.784$$

$$0.784 \cdot 8 = \mathbf{6}.272$$

$$0.272 \cdot 8 = \mathbf{2}.176$$

...

547.56662₈

Base 3 naar base 7

1. Zet om naar base 10 (polynomial method)

$$2110.12_3 = 2 \cdot 3^3 + 1 \cdot 3^2 + 1 \cdot 3^1 + 0 \cdot 3^0 + 1 \cdot 3^{-1} + 2 \cdot 3^{-2} = 66.55 \dots_{10}$$

2. Zet daarna om naar base 7 (remainder en multiplication method)

$$66.55 \dots_{10}$$

Quotiënt	Rest	Positie
$66/7 = 9$	3	LSB
$9/7 = 1$	2	
$1/7 = 0$	1	MSB

$$0.5555 \cdot 7 = \underline{3}.8885$$

$$0.8885 \cdot 7 = \underline{6}.2195$$

$$0.2195 \cdot 7 = \underline{1}.5365$$

$$0.5365 \cdot 7 = \underline{3}.7555$$

...

$$\underline{1}23.\underline{3}613_7$$

Negatieve getallen in binair

- **Signed magnitude:** eerste bit is sign bit (0 = positief, 1 = negatief)

$$25 = 0001\ 1001 \implies -25 = 1001\ 1001_2$$

- **Ones' complement:** invert alle bits

$$25 = 0001\ 1001 \implies -25 = 1110\ 0110_2$$

- **Two's complement:** invert alle bits, tel één erbij op

$$25 = 0001\ 1001 \implies -25 = 1110\ 0111_2$$

- **Excess-128:** tel 128 erbij op, zet om naar binair

- $12 \implies 12 + 128 = 140 = 10001100_2$

- $-12 \implies -12 + 128 = 116 = 01110100_2$

- **Excess- K**

Floating point

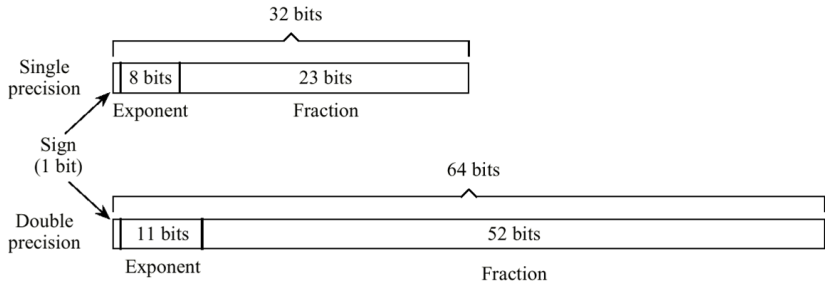
1. **Voorbeeld:** 26-bit normalised floating point formaat in base 5
2. **Conversie naar correcte base:** $-1878.48_{10} = -30003.22_5$
3. **Normalisatie:**
 - één cijfer voor de komma: $-3.000322 \cdot 5^4$
 - of een nul voor de komma: $-0.3000322 \cdot 5^5$
4. **Binaire voorstelling:**
 - Sign bit (negatief = 1)
 - 4-bit excess-7 exponent ($5+7=12$)
 - 7 significante 3-bit cijfers in base-5

$$\begin{array}{ccccccccccc} - & 12 & (0.) & 3 & 0 & 0 & 0 & 3 & 2 & 2 \\ \hline 1 & 1100 & & 011 & 000 & 000 & 000 & 011 & 010 & 010 \end{array}$$

IEEE-754

Overzicht

- Floating point in base 2
- Exponent: excess-127 (32-bit) of excess-1023 (64-bit)
- Mantisse met “verborgen” bit gelijk aan 1



IEEE-754 (single precision)

Conversie

1. **Normalised base 2 floating point:** $-1.100101_2 \cdot 2^4$
2. **Bereken sign bit:** negatief $\Rightarrow 1$
3. **Bereken exponent in excess-127:** $4 \Rightarrow 4 + 127 = 131_{10} = 10000011_2$
4. **Bereken mantisse:**
 - Verwijder eerste bit: $1.100101 \Rightarrow 100101$
 - Zet om naar 23-bits: $100101 \Rightarrow 10010100000000000000000$
5. **Voeg alles samen:** $1\ 10000011\ 10010100000000000000000$

IEEE-754

Denormalised

- Getallen kleiner dan 2^{-126} voorstellen in IEEE-754
- Exponent op 0000 0000 zetten (exponent is dan -126 en niet -127)
- Gevolg: hidden bit is 0
- Voorbeeld:

$$\begin{aligned} -1.01 \cdot 2^{-133} &= -1.01 \cdot 2^{-7} \cdot 2^{-126} \\ &= -0.\underline{000000101} \cdot 2^{-126} \\ &= 1\ 0000\ 0000\ \underline{000\ 0001\ 01}00\ 0000\ 0000\ 0000 \end{aligned}$$

IEEE-754

Speciale waarden

$-0 = 1\ 0000\ 0000\ 000\ 0000\ 0000\ 0000\ 0000\ 0000$
 $+NaN = 0\ 1111\ 1111\ 001\ 0010\ 0000\ 0100\ 1000\ 0000$
 $-\infty = 1\ 1111\ 1111\ 000\ 0000\ 0000\ 0000\ 0000\ 0000$