

Arvusüsteemid

⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	1	2	3	4	5	6	7	8	9	
○	9	2	3	8	4	6	9	7	9	
·	1	2	3	4	5	6	7	8	9	
○	一	二	三	四	五	六	七	八	九	
零	壹	貳	叁	肆	伍	陆	柒	捌	玖	
—	I	II	III	IV	V	VI	VII	VIII	IX	

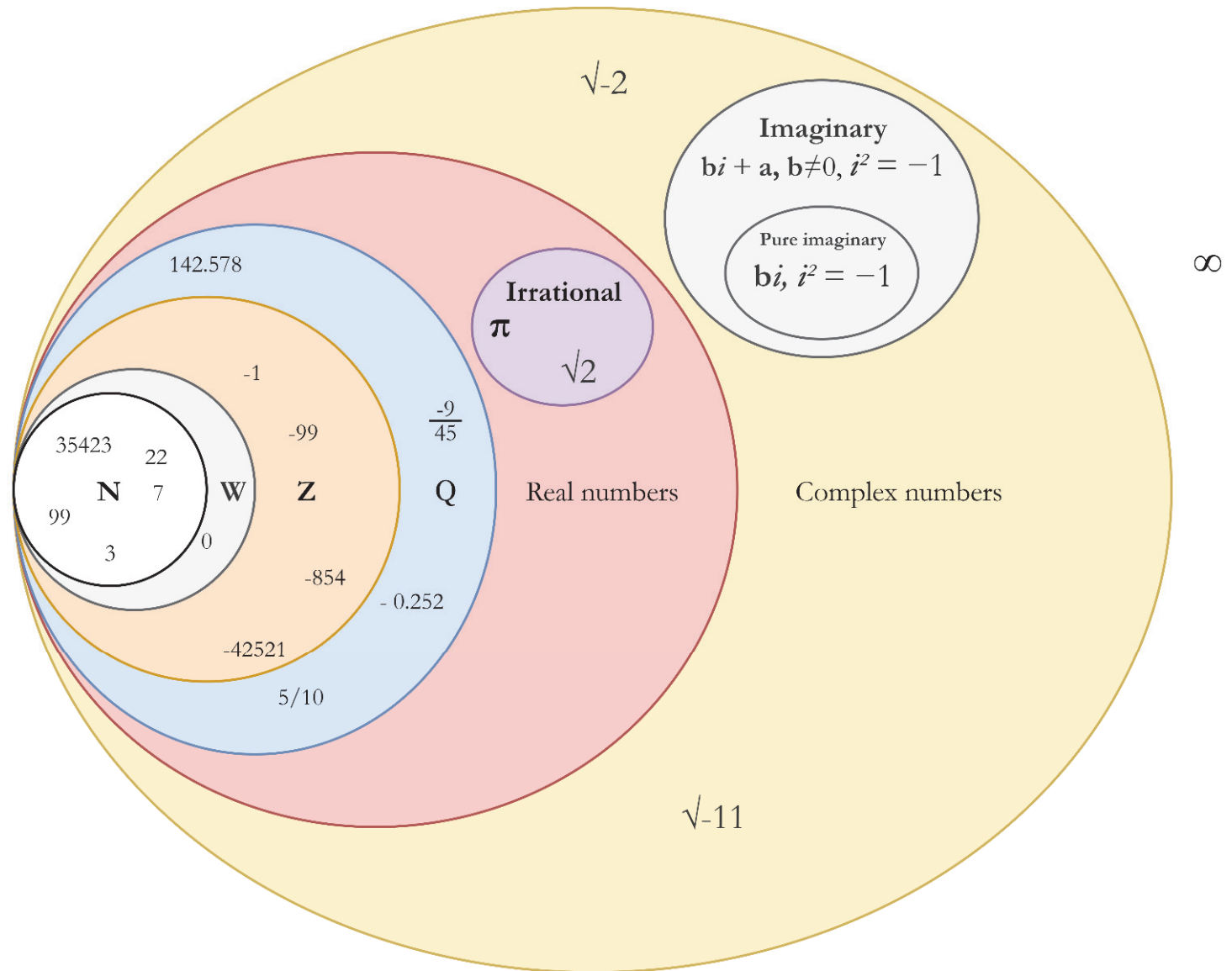
0	1	2	3	4
	•	••	•••	••••
5	6	7	8	9
—	•	••	•••	••••
10	11	12	13	14
—	•	••	•••	••••
15	16	17	18	19
—	•	••	•••	••••

Source: https://en.wikipedia.org/wiki/Numeral_system

- Peamiselt kümnendarvud & positsioonilised süsteemid
- John F. Wakerly, Digital design: principles and practices. Pearson/Prentice Hall.
- Vt ka Harri Lenseni materjale (Moodle's)

Arvusüsteemid

Number groups	
N	Natural numbers
W	Whole numbers
Z	Integers
Q	Rational numbers



Positsioonilised arvusüsteemid

- Väärtus = numbrite kaalutud summa
 - $173,42 = 1 \cdot 100 + 7 \cdot 10 + 3 \cdot 1 + 4 \cdot 0,1 + 2 \cdot 0,01$
 - $173,42 = 1 \cdot 10^2 + 7 \cdot 10^1 + 3 \cdot 10^0 + 4 \cdot 10^{-1} + 2 \cdot 10^{-2}$
 - Täisosa / murdosa
 - Eraldajaks koma / punkt
- Üldistatud esitusviis
 - $D = \sum d_i \cdot r^i, \quad i = -n, \dots, p-1$
 - $d_{p-1} d_{p-2} \dots d_2 d_1 d_0, d_{-1} d_{-2} \dots d_{-n}$
 - r - baas (radix)

Kahendarvud

- Binary numbers

- $D = \sum d_i \cdot 2^i, \quad i = -n, \dots, p-1$

- $101,001_2 =$

- $1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 + 0 \cdot 0,5 + 0 \cdot 0,25 + 1 \cdot 0,125 =$

- $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} =$

- $5,125_{10}$

- Vanim järk / MSB – most significant bit
 - Noorim järk / LSB – least significant bit

8- ja 16-arvud

- Octal numbers – $D = \sum d_i \cdot 8^i$
 - 0, 1, 2, 3, 4, 5, 6, 7
 - $100011001110_2 = 100\ 011\ 001\ 110_2 = 4316_8$
- Hexadecimal numbers – $D = \sum d_i \cdot 16^i$
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
 - $11101101110101001_2 =$
 $0001\ 1101\ 1011\ 1010\ 1001_2 = 1DBA9_{16}$
- Püsikomaarvud
 - $10,1011001011_2 = 010\ ,\ 101\ 100\ 101\ 100_2 = 2,5454_8$
 - $10,1011001011_2 = 0010\ ,\ 1011\ 0010\ 1100_2 = 2,B2C_{16}$
- Teisendused – 8-arv (16-arv) -> 2-arv
- 1 byte - 8 bits, 1 nibble - 4 bits

Teisendamine

- Teisendus tavaliselt üle 10-süsteemi
 - “loomulik arvusüsteem”
- Üldistatud teisendus
 - $d_{p-1} d_{p-2} \dots d_2 d_1 d_0$
 - $D = \sum d_i \cdot r^i \quad (i = 0, \dots, p-1) =$
 $d_{p-1} \cdot r^{p-1} + d_{p-2} \cdot r^{p-2} + \dots + d_2 \cdot r^2 + d_1 \cdot r^1 + d_0 \cdot r^0 =$
 $((((\dots ((d_{p-1}) \cdot r + d_{p-2}) \cdot r + \dots) \cdot r + d_2) \cdot r + d_1) \cdot r + d_0$
 - rekurssiivne jagamine
 - jääk annab koha väärtuse

Teisendamine – näited

- $D = \sum d_i \cdot r^i \quad (i = 0, \dots, p-1) =$
 $((((\dots ((d_{p-1}) \cdot r + d_{p-2}) \cdot r + \dots) \cdot r + d_2) \cdot r + d_1) \cdot r + d_0$
 - $F1AC_{16} = (((15) \cdot 16 + 1) \cdot 16 + 10) \cdot 16 + 12 = 61868$
 - $61868 / 16 = 3866$ jääk 12
 - $3866 / 16 = 241$ jääk 10
 - $241 / 16 = 15$ jääk 1
- $54_{10} = ??_{13}$ (Kui palju on 6x9?)
- $54 / 13 = 4$ jääk 2
- $54_{10} = 42_{13}$

Teisendamine – näited

- $5,125_{10} \rightarrow 101,001_2$
 - Täisosa: $5 = ((1) \cdot 2 + 0) \cdot 2 + 1 = 101_2$
 - $5 / 2 = 2$ jääk 1
 - $2 / 2 = 1$ jääk 0
 - Murdosa: $0,125 = (((1)/2 + 0)/2 + 0)/2 = 0,001_2$
 - $0,125 \cdot 2 = 0,25$ – täisosa 0; murdosa 0,25
 - $0,25 \cdot 2 = 0,5$ – täisosa 0; murdosa 0,5
 - $0,5 \cdot 2 = 1$ – täisosa 1; murdosa 0
- $101,001_2 \rightarrow 5,125_{10}$
 - $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 5,125_{10}$
 - Täisosa: $101_2 / 1010_2 = 0_2$ jääk 101_2 😊

Teisendamine – näited

- $173,42_{10} \rightarrow AD,6B8..._{16}$
 - Täisosa: $173 = (10) \cdot 16 + 13 = AD_{16}$
 - $173 / 16 = 10$ jääk 13
 - Murdosa: $0,42 = (((...+8)/16+11)/16+6)/16 = 0,6B8...$
 - $0,42 \cdot 16 = 6,72$ – täisosa 6; murdosa 0,72
 - $0,72 \cdot 16 = 11,52$ – täisosa 11; murdosa 0,52
 - $0,52 \cdot 16 = 8,32$ – täisosa 8; murdosa 0,32
 - ...
- $AD,6B8_{16} \rightarrow 173,42_{10} ???$
 - $10 \cdot 16^1 + 13 \cdot 16^0 + 6 \cdot 16^{-1} + 11 \cdot 16^{-2} + 8 \cdot 16^{-3} = 173,419921875_{10} !!!$
- *NB! Täpsus teisendustel $\rightarrow$ ümardamine!*

Teisendamine – näited

- $8 \leftrightarrow 16$ – läbi 2-arvude
 - *NB! Alustada tuleb noorimast järgust!*
 - $4316_8 = 100\ 011\ 001\ 110_2 =$
 $1000\ 1100\ 1110_2 = 8CE_{16}$
 - $[4316_8 = 2254_{10} = 8CE_{16}]$
- $1DBA9_{16} = 0001\ 1101\ 1011\ 1010\ 1001_2 =$
 $011\ 101\ 101\ 110\ 101\ 001_2 =$
 355651_8
- $[1DBA9_{16} = 121769_{10} = 355651_8]$

Teisendamine – näited

- Murdosa ja ümardamine?
 - $2, B2C_{16} = 0010, 1011\ 0010\ 1100_2 = 2,6982421875_{10}$
- 10-arvud – $0,026 \rightarrow 0,03$ / $0,024 \rightarrow 0,02$
- 2-arvud – piiriks $0,0\dots5_{10}$ asemel $0,0\dots1_2$
- Mõjutab teisenduste täpsust!
 - $2,6_{10} = 10,10011001\textcolor{brown}{1}00110\dots_2$
 - $10,10011001_2 = 2,59765625_{10}$
 - $10,10011010_2 = 2,6015625_{10}$

Negatiivsed arvud?

- Arvu summa tema pöördarvuga on 0
- Täiendkood (2's complement): $-A = (\sim A) + 1$
 - $-23_{10} = -10111_2 = (\sim 10111_2) + 1_2 = 11101001_2$
 - üks null: $00\dots 0$
 - $-13_{10} + 13_{10} = 10011_2 + 01101_2 = 00000_2$
- Pöördkood (1's complement): $-A = (\sim A)$
 - $-23_{10} = -10111_2 = 11101000_2$
 - kaks nulli: $00\dots 0$ ja $11\dots 1$
 - $-13_{10} + 13_{10} = 10010_2 + 01101_2 = 11111_2$
- Riistvara keerukus / võimsustarve edastamisel

Püsi- ja ujukomaarvud

- Täisarvud: $-2^n \dots 2^n - 1$ ($2^n - 2^0$)
- Püsikomaarvud: $-2^n \dots 2^n - 2^{-f}$
- Ujukomaarvud
 - eksponent: täisarv
 - mantiss: normaliseeritud püsikomaarv
- Põhitehted teostatavad täisarvude vahel
 - koma asukoha korrigeerimine:
 - püsikomaarvud – korrutamisel ja jagamisel
 - ujukomaarvud – normaliseerimine

Täis- / püsikoma- / ujukomaarv

- Täisarv
 - + lihtsad operatsioonid
 - puudub murdosa
- Püsikomaarv
 - + lihtsad liitmine ja lahutamine
 - normaliseerimine vajalik korrutamiseks ja jagamiseks
- Ujukomaarv
 - + paindlik vahemik
 - normaliseerimine vajalik

Täis- / püsikoma- / ujukomaarv

- Püsikomaarvud – operatsioonid
 - n bitti, m kohta pärast koma $(2^{n-m}-2^{-m}) \approx N/2^m$
 - $0,625 = 1/2 + 1/8 = 0000,10100000 = 160 / 256$
- Liitmine & lahutamine
 - $a+b = (A/2^m)+(B/2^m) = (A+B)/2^m$ – OK
- Korrutamine
 - $a*b = (A/2^m)*(B/2^m) = (A*B)/2^{2m}$ – “liiga väike”
 - n & n bitti $\rightarrow 2n$ bitti $\rightarrow m$ bitti sabast maha...
- Jagamine
 - $a/b = (A/2^m)/(B/2^m) = (A/B)/2^0$ – “liiga suur”

Täis- / püsikoma- / ujukomaarv

- Ujukomaarvud – operatsioonid
 - märk 1; eksponent k ja mantiss m bitti
 - $\pm(N/2^m)*2^K$; $1 > N \geq 0.5$ ja $2^{k-1} > K \geq -2^{k-1}$
 - $1,5 = 0,75*2^1 = 0|0001|11000000$
- Liitmine & lahutamine
 - $a+b = (A*2^{AE})+(B*2^{BE}) = (A*2^{AE})+(B*2^{AE-x}) = (A+B/2^x)*2^{AE}$
 - $AE > BE$ (e. $AE=BE+x$) ja $1 > \text{mantiss} > 0$
 - eel- ja järelkorrektsioonid võivad olla vajalikud!
- Korrutamine & jagamine
 - $a*b = (A*2^{AE})*(B*2^{BE}) = (A*B)*2^{AE+BE}$
- Normaliseerimine enne/pärast tehteid

Täis- / püsikoma- / ujukomaarv

- Tarkvara: 16 või 32-bitised täisarvud?
 - protsessori, siinide ja mälude arhitektuurid
 - kompilaatorid (optimeerimised)
- 16 bitti
 - + väiksem mäluvajadus
 - + kiirem edastus põhimälust vahemälusse
 - vajalik võib olla pakkimine
 - suhteline liiasus suvapöördustel
- 32 bitti
 - + andmed vastavad enamasti protsessori sõnale
 - infomahu raiskamine väikeste väärtuste korral
 - siini koormus!

Täis- / püsikoma- / ujukomaarv

- Riistvara: täis-, püsikoma- või ujukomaarvud?
 - täisarv – 1+15 bitti $\approx$
-32000 kuni +32000, täpsus 1
 - püsikomaarv – 1+5+10 bitti $\approx$
-32 kuni +32, täpsus ~ 0.001 ($\sim 0.03\%$)
 - normaliseerimine == nihutamine
 - ujukomaarv – 1+5+10 bitti $\approx$
-64000 kuni +64000, täpsus $\sim 0.1\%$
 - normaliseerimine == analüüs + nihutamine

Liitmine ja lahutamine

- Liitmine

	0011000.	ülekanne
13	00001101	
+ 24	+ 00011000	
37	00100101	tulemus

- Lahutamine

	1110000.	Laenamine		1
13	00001101		00001101	00001101
- 24	- 00011000		11101000	11100111
-11	11110101	tulemus	11110101	11110101
	[A - B]		[A + -B]	[A+~B+1]

Korrutamine

- 2-nd arvude korrutamine
 - analoogne 10-nd arvude korrutamisele

13 * 24

52

26

312

00001101 * 00011000

00001101...

00001101

000100111000

- Liitmine ja nihutamine
- Kõik arhitektuurid põhinevad sellel algoritmil

Korrutamine

- Ühe biti kaupa

```
      00011000 * 00001101
      -----
1.           00000000
2.           00000000.
3.           00000000..
4.           00001101...
5.           00001101....
6.           00000000.....
7.           00000000.....
8.           00000000.....
      -----
           000000100111000
```

- summaator
- nihutaja
 - register
 - tulemuse nihutamine
 - operandi nihutamine
- kontrollor

Jagamine

- 2-nd arvude jagamine
 - analoogne 10-nd arvude jagamisele

$$312 / 13 = 24$$

26

52

52

0

===

$$0100111000 / 01101 = 011000$$

- 01101

01101

- 01101

0000000000 [jääk]

=====

- Lahutamine, kontroll ja nihutamine
- Kõik arhitektuurid põhinevad sellel algoritmil

Muud arvusüsteemid

- Gray kood
 - Frank Gray
 - 1 järk muutub korraga
- 0000 -> 0001 -> 0011 -> 0010 -> 0110 -> 0111 -> 0101 -> ...

- “ribakood”



0000	0001	0011	0010
0100	0101	0111	0110
1100	1101	1111	1110
1000	1001	1011	1010

- BCD
 - *Binary-Coded Decimal*
 - *kahend-kümnendkood*
- Pakitud BCD
 - kümnendjärk -> 4 bitti
- Pakkimata BCD
 - kümnendjärk -> 8 bitti

https://en.wikipedia.org/wiki/Binary-coded_decimal

Muud arvusüsteemid

- Mitmevalentne loogika

- *ternary*

- Mitte-positsioonilised arvusüsteemid

- *residue*
 - *algarvud*
- *set-interval*
 - *vahemikud & hulgad*

- Ujukoma täiendused

- *unum, posit, ...*

- Sümmeetriline 3nd-loogika

6		+ - 0		$1 \cdot 3^2$	$-1 \cdot 3^1$	$+0 \cdot 3^0$
5		+ - -		$1 \cdot 3^2$	$-1 \cdot 3^1$	$-1 \cdot 3^0$
4		++			$1 \cdot 3^1$	$+1 \cdot 3^0$
3		+ 0			$1 \cdot 3^1$	$+0 \cdot 3^0$
2		+ -			$1 \cdot 3^1$	$-1 \cdot 3^0$
1		+				$1 \cdot 3^0$
0		0				$0 \cdot 3^0$
-1		-				$-1 \cdot 3^0$
-2		- +			$-1 \cdot 3^1$	$+1 \cdot 3^0$
-3		- 0			$-1 \cdot 3^1$	$+0 \cdot 3^0$
-4		--			$-1 \cdot 3^1$	$-1 \cdot 3^0$
-5		- ++		$-1 \cdot 3^2$	$+1 \cdot 3^1$	$+1 \cdot 3^0$
-6		- + 0		$-1 \cdot 3^2$	$+1 \cdot 3^1$	$+0 \cdot 3^0$

+3		0 + 0		+5		+ - -
-5		- ++		-4		0 - -
= 0 - + = -2				= 0 0 + = 1		