Repræsentation af tal

DM526

Rolf Fagerberg, 2009

Bitmønstre

$\boxed{01101011}0001100101011011\ldots$

Bitmønstre skal fortolkes for at have en betydning:

- Tal (heltal, kommatal)
- Bogstaver
- Computerinstruktion (program)
- Pixels (billedfil)
- Amplitude (lydfil)
- ⋮

Bitmønstre

$\boxed{01101011}00011001010110 11\ldots$

Bitmønstre skal fortolkes for at have en betydning:

- Tal (heltal, kommatal)
- Bogstaver
- Computerinstruktion (program)
- Pixels (billedfil)
- Amplitude (lydfil)
- ⋮

I dag: heltal og kommatal.

Talsystemer

Tital-systemet:

$$\begin{aligned} 4532 &= 4 \cdot 1000 &+& \quad 5 \cdot 100 &+& \quad 3 \cdot 10 &+& \quad 2 \cdot 1 \\ &= 4 \cdot 10^3 &+& \quad 5 \cdot 10^2 &+& \quad 3 \cdot 10^1 &+& \quad 2 \cdot 10^0 \end{aligned}$$

Talsystemer

Tital-systemet:

$$\begin{aligned} 4532 &= 4 \cdot 1000 + 5 \cdot 100 + 3 \cdot 10 + 2 \cdot 1 \\ &= 4 \cdot 10^3 + 5 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0 \end{aligned}$$

Grundtal: 10
Cifre: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (fordi $10 \cdot 10^i = 10^{i+1}$)

Talsystemer

Tital-systemet:

$$\begin{aligned} 4532 &= 4 \cdot 1000 + 5 \cdot 100 + 3 \cdot 10 + 2 \cdot 1 \\ &= 4 \cdot 10^3 + 5 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0 \end{aligned}$$

Grundtal: 10
Cifre: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (fordi $10 \cdot 10^i = 10^{i+1}$)

Syvtal-systemet:

$$\begin{aligned} 4532_7 &= 4 \cdot 7^3 + 5 \cdot 7^2 + 3 \cdot 7^1 + 2 \cdot 7^0 \\ &= 4 \cdot 343 + 5 \cdot 49 + 3 \cdot 7 + 2 \cdot 1 \\ &= 1640 \end{aligned}$$

Talsystemer

Tital-systemet:

$$\begin{aligned} 4532 &= 4 \cdot 1000 &&+ 5 \cdot 100 &&+ 3 \cdot 10 &&+ 2 \cdot 1 \\ &= 4 \cdot 10^3 &&+ 5 \cdot 10^2 &&+ 3 \cdot 10^1 &&+ 2 \cdot 10^0 \end{aligned}$$

Grundtal: 10
Cifre: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (fordi $10 \cdot 10^i = 10^{i+1}$)

Syvtal-systemet:

$$\begin{aligned} 4532_7 &= 4 \cdot 7^3 &&+ 5 \cdot 7^2 &&+ 3 \cdot 7^1 &&+ 2 \cdot 7^0 \\ &= 4 \cdot 343 &&+ 5 \cdot 49 &&+ 3 \cdot 7 &&+ 2 \cdot 1 \\ &= 1640 \end{aligned}$$

Grundtal: 7
Cifre: 0, 1, 2, 3, 4, 5, 6 (fordi $7 \cdot 7^i = 7^{i+1}$)

Total-systemet

$$\begin{aligned} 1011_2 &= 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 \\ &= 11 \end{aligned}$$

Grundtal: 2
Cifre: 0, 1 (fordi $2 \cdot 2^i = 2^{i+1}$)

Total-systemet

$$
\begin{aligned}
1011_2 &= 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\
&= 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 \\
&= 11
\end{aligned}
$$

Grundtal: 2
Cifre: 0, 1 (fordi $2 \cdot 2^i = 2^{i+1}$)

Relevante for computere fordi todelte valg er nemmest at repræsentere rent fysisk.

Total-systemet kaldes også det *binære talsystem*.

Addition

Addition fungerer ens i alle talsystemer, blot med grundtal udskiftet.

Tital-systemet:

$$
\begin{array}{r}
{\scriptstyle 1\,1\,1\,1} \\
5432 \\
+96781 \\
\hline
=102213
\end{array}
$$

Addition

Addition fungerer ens i alle talsystemer, blot med grundtal udskiftet.

Tital-systemet:

$$\begin{array}{r} {\scriptstyle 1\,1\,1\,1} \\ 5432 \\ +96781 \\ \hline =102213 \end{array}$$

Total-systemet:

$$\begin{array}{r} {\scriptstyle 1\,1\,1} \\ 1110 \\ +11100 \\ \hline =101010 \end{array}$$

Addition

Addition fungerer ens i alle talsystemer, blot med grundtal udskiftet.

Tital-systemet:

$$\begin{array}{r} {\scriptstyle 1\,1\,1\,1} \\ 5432 \\ +96781 \\ \hline =102213 \end{array}$$

Total-systemet:

$$\begin{array}{r} {\scriptstyle 1\,1\,1} \\ 1110 \\ +11100 \\ \hline =101010 \end{array}$$

Subtraktion, multiplikation, division fungerer også ens (ikke med i bog).

Konvertering til binært talsystem

Find cifrene fra højre til venstre i den binære representation af et positivt heltal N:

$X \leftarrow N$
Sålænge $X > 0$:
  (X, næste ciffer) $\leftarrow$ (kvotient, rest) ved heltalsdivision $X/2$

Konvertering til binært talsystem

Find cifrene fra højre til venstre i den binære representation af et positivt heltal N:

$X \leftarrow N$
Sålænge $X > 0$:
  (X, næste ciffer) $\leftarrow$ (kvotient, rest) ved heltalsdivision $X/2$

Eksempel: $N = 25$:

$$\begin{aligned} 25/2 &= (12,1) \\ 12/2 &= (6,0) \\ 6/2 &= (3,0) \\ 3/2 &= (1,1) \\ 1/2 &= (0,1) \end{aligned}$$

$$25 = 11001_2$$

Hvorfor virker det?

$$\begin{aligned} 25/2 &= (12,1) \\ 12/2 &= (6,0) \\ 6/2 &= (3,0) \\ 3/2 &= (1,1) \\ 1/2 &= (0,1) \end{aligned}$$

$$25 = 11001_2$$

Hvorfor virker det?

$$\begin{aligned} 25/2 &= (12,1) \\ 12/2 &= (6,0) \\ 6/2 &= (3,0) \\ 3/2 &= (1,1) \\ 1/2 &= (0,1) \end{aligned}$$

$$25 = 11001_2$$

$$25 = 2 \cdot 12 + 1$$

Hvorfor virker det?

$$\begin{aligned} 25/2 &= (12,1) \\ 12/2 &= (6,0) \\ 6/2 &= (3,0) \\ 3/2 &= (1,1) \\ 1/2 &= (0,1) \end{aligned}$$

$$25 = 11001_2$$

$$\begin{aligned} 25 &= 2 \cdot 12 + 1 \\ &= 2(2 \cdot 6 + 0) + 1 \end{aligned}$$

Hvorfor virker det?

$$\begin{aligned} 25/2 &= (12,1) \\ 12/2 &= (6,0) \\ 6/2 &= (3,0) \\ 3/2 &= (1,1) \\ 1/2 &= (0,1) \end{aligned}$$

$$25 = 11001_2$$

$$\begin{aligned} 25 &= 2 \cdot 12 + 1 \\ &= 2(2 \cdot 6 + 0) + 1 \\ &= 2(2(2 \cdot 3 + 0) + 0) + 1 \end{aligned}$$

Hvorfor virker det?

$$\begin{aligned}
25/2 &= (12,1)\\
12/2 &= (6,0)\\
6/2 &= (3,0)\\
3/2 &= (1,1)\\
1/2 &= (0,1)
\end{aligned}$$

$$25 = 11001_2$$

$$\begin{aligned}
25 &= 2 \cdot 12 + 1\\
&= 2(2 \cdot 6 + 0) + 1\\
&= 2(2(2 \cdot 3 + 0) + 0) + 1\\
&= 2(2(2(2 \cdot 1 + 1) + 0) + 0) + 1
\end{aligned}$$

Hvorfor virker det?

$$\begin{aligned}
25/2 &= (12,1)\\
12/2 &= (6,0)\\
6/2 &= (3,0)\\
3/2 &= (1,1)\\
1/2 &= (0,1)
\end{aligned}$$

$$25 = 11001_2$$

$$\begin{aligned}
25 &= 2\cdot 12 + 1\\
&= 2(2\cdot 6 + 0) + 1\\
&= 2(2(2\cdot 3 + 0) + 0) + 1\\
&= 2(2(2(2\cdot 1 + 1) + 0) + 0) + 1\\
&= 2(2(2(2(2\cdot 0 + 1) + 1) + 0) + 0) + 1
\end{aligned}$$

Hvorfor virker det?

$$\begin{aligned} 25/2 &= (12,1) \\ 12/2 &= (6,0) \\ 6/2 &= (3,0) \\ 3/2 &= (1,1) \\ 1/2 &= (0,1) \end{aligned}$$

$$25 = 11001_2$$

$$\begin{aligned} 25 &= 2 \cdot 12 + 1 \\ &= 2(2 \cdot 6 + 0) + 1 \\ &= 2(2(2 \cdot 3 + 0) + 0) + 1 \\ &= 2(2(2(2 \cdot 1 + 1) + 0) + 0) + 1 \\ &= 2(2(2(2(2 \cdot 0 + 1) + 1) + 0) + 0) + 1 \end{aligned}$$

Bemærk at sidste division altid er $1/2 = (0,1)$: Kvotienten bliver 1 på et tidspunkt, da man ikke ved en division kan komme fra heltal ≥ 2 til heltal ≤ 0.

Repræsentationer af heltal

Talrepræsentationer bruger (næsten altid) et fast antal bits (så operationer kan implementeres effektivt).

$$\boxed{k \text{ bits} = 2^k \text{ forskellige bitmønstre}}$$

Repræsentationer af heltal

Talrepræsentationer bruger (næsten altid) et fast antal bits (så operationer kan implementeres effektivt).

$$\boxed{k \text{ bits} = 2^k \text{ forskellige bitmønstre}}$$

Positive heltal: det binære talsystem giver en naturlig repræsentation.

$k = 4$:

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

Negative heltal

Forskellige forslag:

A Første bit $=$ fortegn, ellers binært talsystem

B Excess

C Two's complement

Negative heltal

Forskellige forslag:

- A Første bit = fortegn, ellers binært talsystem
- B Excess
- C Two's complement

	A	B	C
0111	-7	-1	7
0110	-6	-2	6
0101	-5	-3	5
0100	-4	-4	4
0011	-3	-5	3
0010	-2	-6	2
0001	-1	-7	1
0000	-0	-8	0

	A	B	C
1111	7	7	-1
1110	6	6	-2
1101	5	5	-3
1100	4	4	-4
1011	3	3	-5
1010	2	2	-6
1001	1	1	-7
1000	0	0	-8

Twos complement

Repræsentationen *two's complement* har mange gode egenskaber:

- Fortegn kan ses af første bit.
- Simpel metode til at skifte fortegn findes.
- Den almindelige metode til addition virker også for negative tal (evt. ekstra mente til sidst skal blot smides væk). Ingen ekstra logiske kredsløb for disse (sparer transistorer på CPU).
- Subtraktion kan laves ved at vende fortegn og addere. Ingen logiske kredsløb for subtraktion (sparer transistorer på CPU).

Uden bevis i bog. Prøv selv på eksempler.

Twos complement

Repræsentationen *two's complement* har mange gode egenskaber:

- Fortegn kan ses af første bit.
- Simpel metode til at skifte fortegn findes.
- Den almindelige metode til addition virker også for negative tal (evt. ekstra mente til sidst skal blot smides væk). Ingen ekstra logiske kredsløb for disse (sparer transistorer på CPU).
- Subtraktion kan laves ved at vende fortegn og addere. Ingen logiske kredsløb for subtraktion (sparer transistorer på CPU).

Uden bevis i bog. Prøv selv på eksempler.

Skifte fortegn:

Kopier bits fra højre til venstre til og med første 1-bit.
Resten af bits inverteres.

Eksempel: $6 = 0110 \rightarrow 1010 = -6$

Kommatal

Fast komma:

Tital-systemet:

$$
\begin{aligned}
45.32 &= 4 \cdot 10 + 5 \cdot 1 + 3 \cdot 1/10 + 2 \cdot 1/100 \\
&= 4 \cdot 10^{1} + 5 \cdot 10^{0} + 3 \cdot 10^{-1} + 2 \cdot 10^{-2}
\end{aligned}
$$

Kommatal

Fast komma:

Tital-systemet:

$$
\begin{aligned}
45.32 &= 4 \cdot 10 &+& \quad 5 \cdot 1 &+& \quad 3 \cdot 1/10 &+& \quad 2 \cdot 1/100 \\
&= 4 \cdot 10^1 &+& \quad 5 \cdot 10^0 &+& \quad 3 \cdot 10^{-1} &+& \quad 2 \cdot 10^{-2}
\end{aligned}
$$

Det binære talsystem:

$$
\begin{aligned}
10110.111 &= \quad 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 \\
&\quad + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} \\
&= \quad 1 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 \\
&\quad + 0 \cdot 1 + 1 \cdot 1/2 + 1 \cdot 1/4 + 1 \cdot 1/8 \\
&= \quad 22\tfrac{7}{8} \\
&= \quad 22.875
\end{aligned}
$$

Flydende komma

Flydende komma (alias videnskabelig notation):

Tital-systemet:

$$-0.00000456 = (-1) \cdot 4.56 \cdot 10^{-6}$$

Mantisse:	4.56
Eksponent:	-6

Flydende komma

Flydende komma (alias videnskabelig notation):

Tital-systemet:

$$-0.00000456 = (-1) \cdot 4.56 \cdot 10^{-6}$$

Mantisse:	4.56
Eksponent:	-6

Binært:

$$" - 0.01101 = (-1) \cdot 1.101 \cdot 2^{-2}"$$

Sign bit:	1	(sign bit 1 for negativt tal)
Mantisse bits:	(1)1010	(første bit underforstået)
Eksponent:	010	(-2 i excess notation (3 bits))

Der afsættes et fast antal bits til hver af de tre dele.

Begrænsninger

Heltal ($\mathbb{N}$, $\mathbb{Z}$) og reelle tal ($\mathbb{R}$) er uendelige talmængder.

Hvis der afsættes et fast antal (k) bits fås et endeligt antal (2^k) forskellige bitmønstre.

Begrænsninger

Heltal ($\mathbb{N}$, $\mathbb{Z}$) og reelle tal ($\mathbb{R}$) er uendelige talmængder.

Hvis der afsættes et fast antal (k) bits fås et endeligt antal (2^k) forskellige bitmønstre.

$\boxed{\text{Ikke alle tal kan repræsenteres!}}$

Begrænsninger

Heltal ($\mathbb{N}$, $\mathbb{Z}$) og reelle tal ($\mathbb{R}$) er uendelige talmængder.

Hvis der afsættes et fast antal (k) bits fås et endeligt antal (2^k) forskellige bitmønstre.

$\boxed{\text{Ikke alle tal kan repræsenteres!}}$

Viser sig f.eks. ved

- Overflow
  - maxInt + maxInt = ?
- Truncation errors
  - Stort tal + epsilon = stort tal.
  - $(a + b) + c \neq a + (b + c)$ hvis $a + b$ ikke kan repræsenteres exakt, mens $b + c$ godt kan.

Begrænsninger

Heltal ($\mathbb{N}$, $\mathbb{Z}$) og reelle tal ($\mathbb{R}$) er uendelige talmængder.

Hvis der afsættes et fast antal (k) bits fås et endeligt antal (2^k) forskellige bitmønstre.

$\boxed{\text{Ikke alle tal kan repræsenteres!}}$

Viser sig f.eks. ved

- Overflow
  - maxInt + maxInt = ?
- Truncation errors
  - Stort tal + epsilon = stort tal.
  - $(a + b) + c \neq a + (b + c)$ hvis $a + b$ ikke kan repræsenteres exakt, mens $b + c$ godt kan.

I praksis ses det sjældent pga. et stort antal bits i talrepræsentationerne.

Alternativt findes programmeringsbiblioteker der implementerer f.eks. vilkårligt store heltal (under brug af variabelt antal bits, samt tab af effektivitet).

Hexadecimal notation

Gruppér bits i grupper af 4 (dvs. 16 forskellige muligheder):

$\boxed{0110}\,\boxed{1010}\,\boxed{1110}\,01\ldots$

Hexadecimal notation

Gruppér bits i grupper af 4 (dvs. 16 forskellige muligheder):

$\boxed{0110}\boxed{1010}\boxed{1110}01\ldots$

Vælg 16 tegn:

0111	7	1111	F
0110	6	1110	E
0101	5	1101	D
0100	4	1100	C
0011	3	1011	B
0010	2	1010	A
0001	1	1001	9
0000	0	1000	8

Hexadecimal notation

Gruppér bits i grupper af 4 (dvs. 16 forskellige muligheder):

$$\boxed{0110}\boxed{1010}\boxed{1110}01\ldots$$

Vælg 16 tegn:

0111	7	1111	F
0110	6	1110	E
0101	5	1101	D
0100	4	1100	C
0011	3	1011	B
0010	2	1010	A
0001	1	1001	9
0000	0	1000	8

$$\boxed{0110}\boxed{1010}\boxed{1110}01\ldots = \texttt{6AE}\ldots$$

Hexadecimal notation

Gruppér bits i grupper af 4 (dvs. 16 forskellige muligheder):

$$\boxed{0110}\boxed{1010}\boxed{1110}01\ldots$$

Vælg 16 tegn:

0111	7	1111	F
0110	6	1110	E
0101	5	1101	D
0100	4	1100	C
0011	3	1011	B
0010	2	1010	A
0001	1	1001	9
0000	0	1000	8

$$\boxed{0110}\boxed{1010}\boxed{1110}01\ldots = \texttt{6AE}\ldots$$

NB: kan også bruges som cifre i et talsystem med grundtal 16.