1 Zahlensystem

1.1 Zahlendarstellung

Zahlendarstellung abhängig durch verwendete Basis. Dabei lässt sich jede positive Zahl als Produkt einer jeden Basis durch diese Form darstellen:

$$\left\{ \sum_{i=0}^{\infty} z \cdot b^i \middle| z \in Z_{\mathrm{b}}, b \in N \setminus \{1\} \right\}$$

Hexadezimal ($b = 16$) eignet sich besonders gut, da 4 Bit folgen gut mit dieser Basis repräsentiert werden können. Somit können 2^n Bitfolgen effektiv realisiert werden

Die Darstellung negativer Zahlen kann durch 2 Möglichkeiten erfolgen:

1.
Nutzung des führenden, höchsten Bits als Vozeichen. Dabei steht deine führende 0 für eine positive Zahl und eine führende 1 für eine negative Zahl

2.
Nutzung des Zweierkomplements. Dabei das negative Gegenstück einer Binärzahl geschaffen indem man das vorhandene, positive Äquivalent verwendet, die 1er und 0er invertiert und man auf die resultiernede Zahl +1 addiert:

$1 \equiv$ 0000 0001
$-1 \equiv inv(0000\ 0001) + 0000\ 0001 = 1111\ 1110 + 0000\ 0001 = 1111\ 1111$

Das resultierende Problem ist nun dass nurnoch dieser Zahlenraum darstellbar ist:

$$\left\{ \left[-2^{\text{n-1}},\ 2^{\text{n-1}} - 1 \right]\ n \equiv Anzahl\ der\ Bits \right\}$$

Die Darstellung reeller Zahlen kann durch 2 Möglichkeiten erfolgen:

1.
Die Fest-Komma Darstellung ist ein Tupel bestehend aus zwei Zahlen. Dabei stellt die erste Zahl die Zahl vor dem Komma und die zweite Zahl die Zahl nach dem Komma da.

Nachteile: $-$ *Begrenzte Reichweite*
$-$ *Bestimmte, festzulegende Genauigkeit (Kommasetzung)*

2.
Die Fließ-Komma Darstellung verfolgt das Ziel einer höheren Flexibilität. Die meisten Systeme halten sich an die Norm von IEEE-754

Single-Point-Precision: 32Bit - Float
Double-Point-Precision: 64Bit - Double

Nachteile: $-$ *Ungleichmäßige Verteilung*
$-$ *Ungenauigkeit bei arithmetischen Operationen*

$$\left\{M \cdot b^{e} \middle| b = Basis, e = Exponent\right\}$$

1.2 Arithmetik

1.
Die Addition erfolgt trivial

2.
Die Subtraktion ist trivial solange man mit unsigned Zahlen arbeitet. Die Subtraktion mit Zahlen des Zweierkomplements erfolgt durch die Addition der invertierten Zahl +1:

01001-00111=01001+inv(00111)+1=01001+11000+1=00010

Formalitäten:

	Bits	Unsigned	Signed	Typname in C
Bit	1	$2^1 - 1$	nicht möglich	-
Nibble	4	$2^4 - 1$	-2^{4-1}, $2^{4-1}-1$	-
Byte	8	$2^8 - 1$	-2^{8-1}, $2^{8-1}-1$	char
Word	16	$2^{16} - 1$	-2^{16-1}, $2^{16-1}-1$	short
Double Word	32	$2^{32} - 1$	-2^{32-1}, $2^{32-1}-1$	int
Quad Word	64	$2^{64} - 1$	-2^{64-1}, $2^{64-1}-1$	long long