

Open CheatSheet: GdI-Edition

jm

29. Januar 2017

1 Tabellenwerk

Tabelle 1: Umrechnung zu dezimal bis b^{12}

	b^0	b^1	b^2	b^3	b^4	b^5	b^6	b^7	b^8	b^9	b^{10}	b^{11}	b^{12}
$b = 2$	1	2	4	8	16	32	64	128	256	512	1024	2048	4096
$b = 8$	1	8	64	512	4096	-	-	-	-	-	-	-	-
$b = 16$	1	16	256	4096	65536	-	-	-	-	-	-	-	-

Tabelle 2: Gängige Zahlensysteme: Darstellungen bis Wert 15

dez	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<i>bin</i>	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111
<i>oct</i>	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
<i>hex</i>	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Tabelle 3: vielfache von 10_{10} (Nützlich für die Berechnung von Dezimalzahlen im Quellsystem)

Basis	10	2	3	4	5	6	7	8	9
	10	1010	101	22	20	14	13	12	11
	20		202	110	40	32	26	24	22
	30			132	110	50	42	36	33
	40				130	104	55	50	44
	50					122	101	62	55
	60						114	74	66
	70							106	77
	80								88

2 Umrechnen

2.1 Im Dezimalsystem

2.1.1 $b \neq 10$ zu $b = 10$ (beliebige Basis ungleich 10 zu Basis 10)

Stellen addieren nach folgender Vorschrift:

$$\sum_{n=0}^{N-1} (a_n * b^n), a_n \in \{0, \dots, b-1\}, i \in \{0, \dots, N-1\}$$

wobei N : Anzahl der Stellen, n : Stelle der Zahl und a_n : Wert der Ziffer an Stelle n .

2.1.2 Ein Beispiel: $(124032)_5$

$$(124032)_5 = (1*5^5) + (2*5^4) + (4*5^3) + (0*5^2) + (3*5^1) + (2*5^0) = 3125 + 1250 + 500 + 0 + 15 + 2 = 4892$$

2.2 Umrechnen im Quellsystem: Von $b \neq 10$ zu $b = 10$

Rechnen im Quellsystem zur Umwandlung von Zahlen beliebiger b-adischer Systeme zum dezimal System

2.2.1 Der Algorithmus in Worten

1. Teile die umzurechnende Zahl durch die Repräsentation der $(10)_{10}$ im Quellsystem bis das Ergebnis der Subtraktionen (innerhalb des Divisionsalgorithmus) nicht mehr durch "10teilbar" sind. Das letzte Ergebnis der Subtraktion ist der Divisionsrest.
2. Wiederhole Schritt 1 für jedes Ergebnis (ohne Rest) bis das Resultat nicht mehr durch die Repräsentation von $(10)_{10}$ teilbar ist.
3. Die Reste der Divisionen sind die Quellsystemrepräsentanten der einzelnen Stellen der zu berechnenden Dezimalzahl. Die hochwertigste Stelle entspricht der zuletzt vorgenommenen Division.

2.2.2 Ein Beispiel: $(120102)_3$ zu $(416)_{10}$

Folgende Tabelle hilft bei der Berechnung der Divisionen. Man braucht nur vielfache bis $b - 1$ zum berechnen der Divisionen.

Tabelle 4: Vielfache von $(10)_{10}$ im 3-er System

Basis 10	basis 3
10	101
20	202

$$120102/101 = 1112 \text{ R } 20 \quad (1)$$

$$1112/101 = 11 \text{ R } 1 \quad (2)$$

$$11/101 = 0 \text{ R } 11 \quad (3)$$

$$\text{Aus (3)} \Rightarrow (11)_3 \hat{=} (4)_{10}$$

$$\text{Aus (2)} \Rightarrow (1)_3 \hat{=} (1)_{10}$$

$$\text{Aus (1)} \Rightarrow (20)_3 \hat{=} (6)_{10}$$

$$\Rightarrow (416)_{10}$$

3 Vorzeichenbehaftete Darstellungen

3.1 kleinste / größte darstellbare Zahl mit N Bits

Exzess-N kleinste: -N; Größte: N-1

b-1-Komplement $[-(2^{N-1} - 1) \dots + (2^{N-1} - 1)]$

doppelte null: 0000 und 1111 sind beides null

b-Komplement $[-2^{N-1} \dots + 2^{N-1} - 1]$

3.2 Rechnen mit dem einer Komplement

Den einer Rücklauf beachten (carry bit von höchster stelle wird (falls vorhanden) zum Ergebnis addiert (Ergebnis + 1)

3.3 Rechnen mit dem zweier Komplement

nach der Komplementbildung eins addieren.

3.4 Rechnen im Exzesscode

3.5 Welche Zahlen sind in einem b-adischen System negativ

bei geraden basen ist die erste Ziffer größergleich $b/2$ gilt die Zahl als negativ ansonsten ist sie positiv

So sind die vorzeichenlosen Zahlen $1 \dots (b^N/2) - 1$ im b-Komplement positiv, $(b^N/2) \dots b^N - 1$ sind negativ, 0 ist null.

Im (b-1)-Komplement gilt, dass die Zahlen $(b^N/2) \dots b^N - 2$ negativ sind, $b^N - 1$ ist minus null.

bei ungeraden basen Hier wird die Trennlinie bei der Zahl $(aaa\dots a)_b$ gezogen, wobei $a = (b-1)/2$ ist. Bei $b = 3$ ist es die Zahl $(111\dots 1)_3$.

4 Reelle Zahlen

4.1 2-adische Entwicklung (Dezimal zu Binär wandeln)

1. Teile durch 2
2. Erstes Ergebnis: Erste Stelle nach dem Komma
3. Wenn Ergebnis < 0 bedeutet dies eine 0 für diese Nachkommastelle

Trick: Wenn die Zahl welche binär dargestellt werden soll eine Zweierpotenz im Nenner hat, dann stellt der Wert des Exponenten die Anzahl der Nachkommastellen dar. Also bspw. $1/8 = 1/2^3$ dar. Daraus folgt 3 Nachkommastellen (weil der Exponent die drei ist). Die Zahl auf dem Bruch wird dann von rechts eingeschoben also bspw. $2/8 = 2/2^3 \Rightarrow$ drei Nachkommastellen und $2 = 01$ daraus folgt das Ergebnis: $0.010 = 1/4$

4.2 Konvertieren zwischen Binär/Hexadezimal/Dezimal

Folgende Beispiele beschreiben das Schema:

Vorkomma Anteil (siehe Kapitel 2 und 3 - Umrechnen ganzer Zahlen)

$$10110.110010 \quad (4)$$

$$0001 = (1)_{16} \quad (5)$$

$$0110 = (6)_{16} \quad (6)$$

$$\Rightarrow (16)_{16} \quad (7)$$

$$(16)_{16} = (22)_{10} \quad (8)$$

Nachkomma Anteil zu dez Aufsummieren: $N * b^{-1} + N * b^{-2} + \dots + N * b^{-i}$ wobei i die letzte Nachkommastelle ist und N der Wert der Stelle

Shortcut bei Zahlen mit nur Einsen nach dem Komma Wert von 0.1111 (also vier Stellen nach dem Komma): $(1 - 2^{-4}) = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} = \frac{15}{16}$

Nachkommaanteil von bin zu hex Vierergruppchen bilden: $\Rightarrow (10110.110010)_2 = (0001)(0110).(1100)(1000) = (16.C8)_{16}$

4.3 Gleitkommazahlen

normalisiert ist eine Gleitkommazahl wenn die Ziffern nach dem Komma sich nicht mehr weiter nach links verschieben lassen ohne das Werte über das Komma "hinausrutschen". Die folgenden Werte sind bereits normalisiert:

$$0.11001 * 2^{-39} \quad (9)$$

$$0.001011 * 8^4 \quad (10)$$

Unterlauf- und Überlaufbereich Unterlauf: Kleinste zulässige(!) Darstellung wählen und Dezimalwert berechnen. Dabei darauf achten: wenn Normalisierung gefordert wird, ist die kleinste zulässige Mantisse nicht 0.000001 sondern 0.1. In IEEE ist bspw. keine Normalisierung gefordert, daher kann man hier wirklich den kleinst möglichen Mantissenwert annehmen.

4.3.1 IEEE-Single-Precision Floating Point

Umrechnung IEEE zu Dez von links: Bit 1: Vorzeichen, Bits 2-9 (8bit): Exponent, Bits 10-32 (23bit): Mantisse.

$$z = \pm 1.m_1 \dots m_{23} * 2^{e_1 \dots e_8 - 127} \text{ mit VZ } 1 \stackrel{\wedge}{=} - \text{ und } 0 \stackrel{\wedge}{=} +$$