

114Spick_1

Codierung: Umwandlung von Daten/Information in ein anderes (oft ein allgemeineres) Medium, bzw. im gleichen Medium in eine andere Form ["Zeichen mit einem Symbol darstellen"]; **BSP:** Zeichensprache, Ampel, Morsecode; **Zweck:** Übertragung, Speicherung, Bearbeitung von Daten wird ermöglicht

Komprimierung: Komprimierung bezeichnet den Prozess der Reduzierung der Datenmenge, um Speicherplatz zu sparen oder die Übertragungseffizienz zu verbessern. Dabei werden redundante oder nicht wesentliche Informationen entfernt, um die ursprüngliche Datenmenge zu verkleinern. Es gibt aber Methoden, die Muster suchen und dann nur auf das erste Muster verweisen, oder häufig auftretender Information kürzere Codes zuweist (und umgekehrt); **BSP:** MP3, ZIP, Stenographie; **Zweck:** Spart Speicherplatz, verringert Übertragungszeiten

Verschlüsselung: Verschlüsselung ist ein Sicherheitsmechanismus, bei dem Informationen in eine unverständliche Form umgewandelt werden, um sie vor unbefugtem Zugriff zu schützen. Dieser Prozess erfolgt mit Hilfe von Algorithmen und Schlüsseln; **BSP:** Geheimsprache; **Zweck:** verhindert Zugriff auf Daten durch Unberechtigte

(Redundanz, Vernetzung)

Byte: Umfasst 8 Bit, also $2^8 = 256$ Zustände (= 2 Nibbles), mit 2 Hexadez. Zahlen darstellbar

Nibble: Ist eine Binäre Zahl mit 4 Stellen (im Hexadezimalsystem entspricht das einer Stelle)

Binär-, Oktal-, Hexadezimalsystem:

Binär	Oktal	Dez	Hexadez	Eigenes Tertiär bzw. ternär
0	0	0	0	N
1	1	1	1	a
10	2	2	2	b
11	3	3	3	aN
100	4	4	4	aa
101	5	5	5	ab
110	6	6	6	bN
111	7	7	7	ba
1000	10	8	8	bb
1001	11	9	9	aNN
1010	12	10	A	aNa
1011	13	11	B	aNb
1100	14	12	C	aaN
1101	15	13	D	aaa
1110	16	14	E	aab
1111	17	15	F	abN
1 0000	20	16	10	aba
1 0001	21	17	11	abb
1 0010	22	18	12	bNN
1 0011	23	19	13	bNa
1 0100	24	20	14	bNb

Bei der normalen Codierung von negativen Zahlen gilt: Das Komplement der positiven Zahl + 1 ergibt die negative Zahl

Exzess Darstellung von negativen Zahlen:

BSP: 8-Bit-Exzess-128-Codierung: 8 Bit heisst 256 Zahlen insgesamt, also-128-127 (wir zählen also für die Codierung einfach zu allen Zahlen 128 dazu, um alle Zahlen als ein Kontinuum darzustellen)

Binärzahlen addieren: Wie schriftlich Addieren, mit Übertrag und so.

Fliesskommazahlen: nach dem Schema der normalisierten Exponentialschreibweise (Wissenschaftl. Schreibweise):

$1.p * 2^e$: die ersten 8 Bits für den Exponenten (d. i. "p") die nächsten 23 Bits (und nicht 24, wegen dem negativen bzw. positiven Vorzeichen) von den anfänglichen 32 Bits für die Mantisse. Das erste Bit ist für das **Vorzeichen** reserviert. Der Exponent muss allerdings auch negativ sein können => deshalb wird das Bias/Exzess/Überschuss System verwendet (nach Standard immer plus 127 (bei 8Bit Kapazität) also sind Exp. -127 bis 128 möglich!); BSP:

5.75 \triangleq **0100 0000 1011 1000 0000 0000 0000 0000** (5.75 = 101.11, normalisiert: 1.0111, Exponent ist also 2, mit Ueberschuss also 129 = Binär 10000001)
 \triangleq **40B80000**

umgekehrt: **0xC1CC0000 = 1100 0001 1100 1100 0000 0000 0000 0000 = -1.10011*2⁽¹³¹⁻¹²⁷⁼⁴⁾ = -11001,1 = -25.5**

Umrechnungen:

Dividend	Quotient	Rest		Stelle	n für 2 ⁿ	Wert
1999	999	1		0	1	1
999	499	1		1	2	2
499	249	1		2	4	4
249	124	1		3	8	8
124	62	0		4	16	0
62	31	0		5	32	0
31	15	1		6	64	64
15	7	1		7	128	128
7	3	1		8	256	256
3	1	1		9	512	512
1	0	1		10	1024	1024
				Summe	1999	

Schreibweisen ohne tiefgestellte Ziffer: 0x7CF oder 7CFh (für Hexadezimalzahlen)

Hexadezimal <=> **Binär**: 4 Bit Gruppen bilden: 10(16) = 1 0000; 11000011(2) = 1100 (12) 11 (3) = C3

Oktal <=> **Binär**: 3er Gruppen bilden: 11(8) = 001 001 bzw. 1001; 110110001(2) = 661

Dezimalbrüche <=> **Binär**: Dezimalbruch mit 2 multiplizieren und ganze Zahl vor dem Komma notieren, mit den Nachkommastellen Verfahren wiederholen bis 1 resultiert:

0.75 x2 = **1**.5, 0.5x 2 = **1** => fertig, da keine Nachkommastellen [Reihenfolge hier übrignes nicht umgekehrt, die erste Ziffer betrifft die Zweitel, die zweite die Viertel etc.]

=> **11**(2)

Addition und Subtraktion vpn binär gespeicherten Zahlen: Beim Einerkomplement ist das Resultat um 1 zu klein, mit dem Zweierkomplement lässt sich aber „wie gewohnt“ (Subtraktion = wir addieren eine negative Zahl) rechnen.