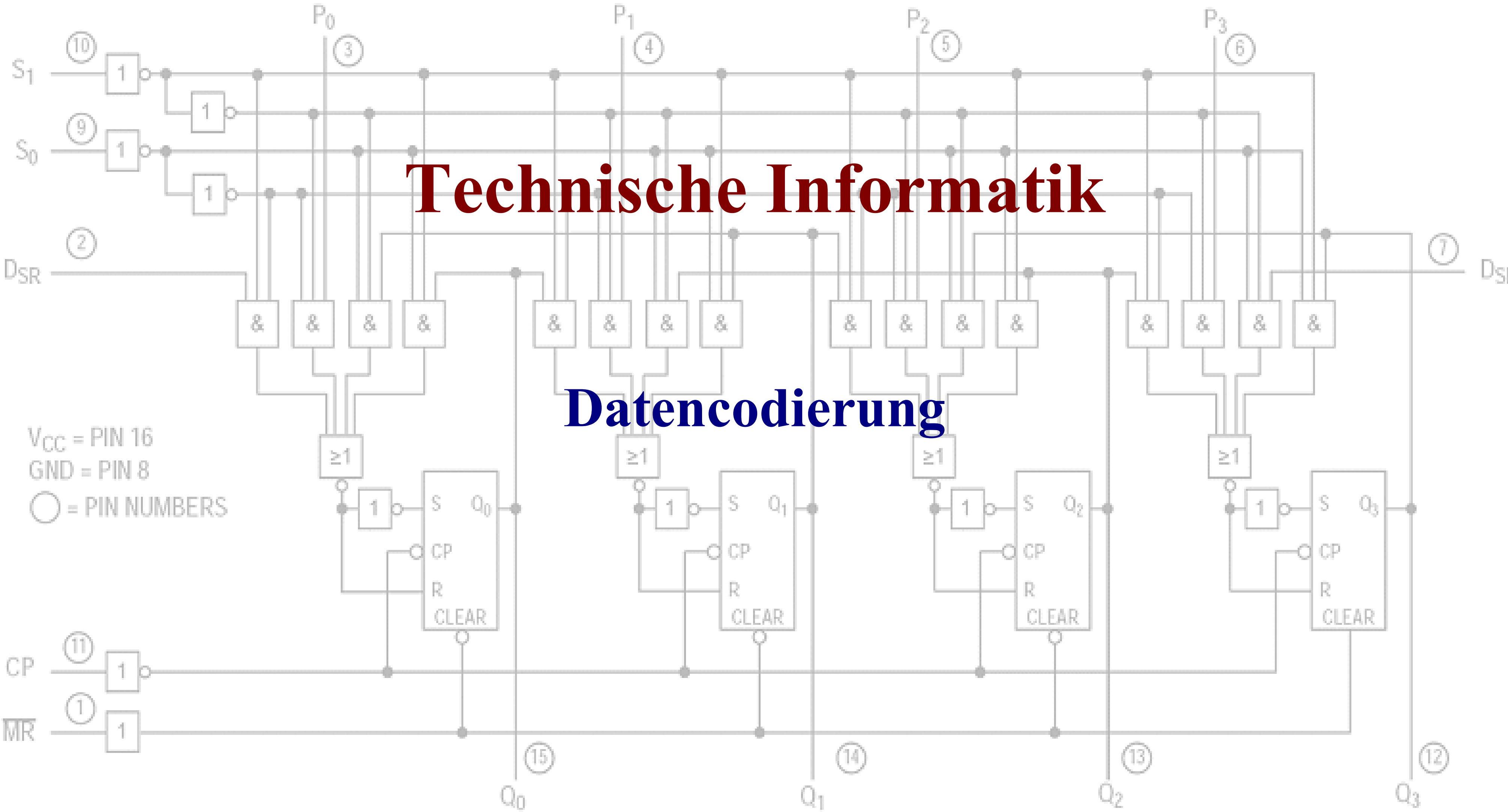
Technische Informatik

Datencodierung



Grundbegriffe

- Zeichenvorräte und Codierung sind prinzipiell willkürliche Vereinbarungen, die für die jeweilige Anwendung optimiert werden

Zeichenvorrat	Eine endliche Menge von Zeichen
Alphabet	Ein geordneter Zeichenvorrat
Wort	Eine endliche Folge von Zeichen eines Alphabets (Zeichenkette)
Codierung	Funktion (Abbildungsvorschrift), die die Elemente eines Zeichenvorrates ***A*** abbildet auf Wörter über einen anderen Zeichenvorrat ***B***. Die Codierung kann z.B. durch eine *Codetabelle* oder eine *Umwandlungsvorschrift* beschrieben werden.
Codewort	Element des Bildbereiches ***B*** der Codierung
Code	Menge aller Codewörter
Binärcode	Codierung mit dem Bildbereich ***B*** *={0,1}* (Der Zeichenvorrat des Bildbereiches umfasst genau zwei Zeichen). Wird eine feste Codewortlänge *n* verwendet, liegt eine *n-Bit-Codierung* vor

Grundbegriffe

- Optimalcodierung
  - Minimaler Zeichenaufwand
- Systematische Codes
  - Enthalten Redundanzen zur Fehlererkennung und -korrektur
  - Vergl. *Hamming-Distanz* (= Anzahl der Stellen, in denen sich zwei Codeworte unterscheiden)
- Zyklische und lineare Codes
  - Systematische Codes, bei denen die Verknüpfung zweier gültiger Codeworte wieder ein gültiges Codewort ergibt
- Übliche Wortbreiten binärer Codierung:
  - Byte 8 Bit
  - Word 16 Bit (high byte + low byte)
  - Doubleword 32 Bit (high word + low word)
  - Quadword 64 Bit (high doubleword + low doubleword)

Vorzeichenlose Ganzzahl: Dualsystem

- Dualsystem
  - Stellenwertsystem mit der Basis 2
  - Die i-te Ziffer von rechts (x_i) hat die Wertigkeit 2^i

Ziffer	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
Wertigkeit	2^7 = 128	2^6 = 64	2^5 = 32	2^4 = 16	2^3 = 8	2^2 = 4	2^1 = 2	2^0 = 1

Codetabelle:

Dezimal	***Dual***		
	x_2	x_1	x_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

- Codierung:

$$X = x_{N-1} \cdot 2^{N-1} + x_{N-2} \cdot 2^{N-2} + \ldots + x_1 \cdot 2^1 + x_0 \cdot 2^0$$

- Beispiel:

$$75_{dez} = 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 0100\,1011_{bin}$$

- Anmerkung:
  - Anwendbar für natürliche Zahlen (ohne Vorzeichen)
  - Verschiedene Wortbreite
  - Carry (Übertrag): $\texttt{000...0}_{\texttt{bin}} \leftrightarrow \texttt{111...1}_{\texttt{bin}}$

Vorzeichenlose Ganzzahl: Oktal- und Hexadezimalsystem

- Oktalsystem:
  - Stellenwertsystem mit der Basis 8
  - Zeichenvorrat = {0,1,2,3,4,5,6,7}

Ziffer	x_3	x_2	x_1	x_0
Wertigkeit	8^3 = 512	8^2 = 64	8^1 = 8	8^0 = 1

- Hexadezimalsystem:
  - Stellenwertsystem mit der Basis 16
  - Zeichenvorrat = {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Ziffer	x_3	x_2	x_1	x_0
Wertigkeit	16^3 = 4096	16^2 = 256	16^1 = 16	16^0 = 1

- Beispiel:

dezimal	156							
dual	1	0	0	1	1	1	0	0
hexadezimal	9				C			

156 = 1*128 + 1*16 + 1*8 + 1*4

156 = 9*16 + 12*1

- Anmerkung:
  - Eine Dualzahl kann in eine Hexadezimalzahl umgewandelt werden, indem jeweils 4 Bit zusammengefasst werden. Entsprechendes gilt für die Oktalzahl (3 Bit)

Vorzeichenbehaftete Ganzzahl: Offset-Dual-Darstellung

- Offset-Dual-Darstellung (binary offset, biased dual)
  - Abgeleitet vom Dualsystem
  - Subtraktion der größten Wertigkeit

Codetabelle:

Dezimal	*Offset-Dualzahl*		
	x_2	x_1	x_0
-4	0	0	0
-3	0	0	1
-2	0	1	0
-1	0	1	1
0	1	0	0
1	1	0	1
2	1	1	0
3	1	1	1

- Codierung:

$$X = x_{N-1} \cdot 2^{N-1} + \ldots + x_1 \cdot 2^1 + x_0 \cdot 2^0 \underline{- 2^{N-1}}$$

- Beispiel:

$$-53_{dez} = 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 - 2^7 = 01001011_{bin}$$

Vorzeichenbehaftete Ganzzahl: Betrag und Vorzeichen

- Darstellung mit Betrag und Vorzeichen
  - Abgeleitet vom Dualsystem
  - Höchstwertiges Bit (MSB) liefert das Vorzeichen (Vorzeichenbit: 0 = positiv, 1 = negativ)
  - Die Zahl 0 wird zweimal codiert („+0“ und „-0“)
  - Eindeutige Interpretation nur bei fester Wortbreite

Codetabelle:

Dezimal	Dualzahl mit Vorzeichenbit		
	x_2	x_1	x_0
+0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
-0	1	0	0
-1	1	0	1
-2	1	1	0
-3	1	1	1

- Codierung:

$$X = \underline{(-1)^{x_{N-1}}} \cdot (x_{N-2} \cdot 2^{N-2} + \ldots + x_1 \cdot 2^1 + x_0 \cdot 2^0)$$

- Beispiel:

$$-11_{dez} = (-1)^1 \cdot (0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0) = 1000\,1011_{bin}$$

Vorzeichenbehaftete Ganzzahl: Einerkomplement

- Einerkomplement
  - Abgeleitet vom Dualsystem
  - Eine negative Zahl wird durch das bitweise Komplement der entsprechenden positiven Zahl dargestellt
  - Vorzeichenwechsel:
    - Invertierung aller Bits

Keine praktische Anwendung!

Codetabelle:

Dezimal	*Einerkomplement*		
	x_2	x_1	x_0
+0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
-3	1	0	0
-2	1	0	1
-1	1	1	0
-0	1	1	1

- Codierung:

$$X = \underline{-}\, x_{N-1} \cdot \left(2^{N-1} - 1\right) + x_{N-2} \cdot 2^{N-2} + \ldots + x_0 \cdot 2^0$$

- Beispiel:

$$-7_{dez} = -1 \cdot (2^7 - 1) + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^2 + 0 \cdot 2^0 = 1111\,1000_{bin}$$

- Anmerkung:
  - Die Zahl 0 wird zweimal codiert („+0“ und „-0“)
  - Bei Addition/Subtraktion muss Vorzeichen als Sonderfall berücksichtigt werden

Vorzeichenbehaftete Ganzzahl: Zweierkomplement

- Zweierkomplement
  - Abgeleitet vom Dualsystem
  - Standarddarstellung für Rechnersysteme
  - Höchstwertiges Bit (MSB) hat *negative* Wertigkeit
  - Vorzeichenwechsel:
    - Invertierung aller Bits und Addition einer „1“

Codetabelle:

Dezimal	*Zweierkomplement*		
	x_2	x_1	x_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
-4	1	0	0
-3	1	0	1
-2	1	1	0
-1	1	1	1

- Codierung:

$$X = \underline{-}\, x_{N-1} \cdot 2^{N-1} + x_{N-2} \cdot 2^{N-2} + \ldots + x_0 \cdot 2^0$$

- Beispiel:

$$-7_{dez} = -1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^2 + 1 \cdot 2^0 = 1111\,1001_{bin}$$

- Anmerkung:
  - Keine getrennte Betrachtung des Vorzeichens bei Addition bzw. Subbtraktion
  - **overflow (Überlauf)**: $\texttt{011...1}_{\texttt{bin}} \leftrightarrow \texttt{100...0}_{\texttt{bin}}$

Binär Codierte Dezimalzahl (BCD)

- Binär Codierte Dezimalzahl:
  - Jede Ziffer einer Dezimalzahl wird auf eine 4-Bit Binärzahl abgebildet (8421-BCD).
  - Die „Pseudotetraden“ 1010_{bin} bis 1111_{bin} werden i.d.R. nicht verwendet
  - Auf beliebig viele Stellen erweiterbar
- Beispiel:

dezimal	3				7			
BCD	0	0	1	1	0	1	1	1

- Anwendung:
  - Finanzmathematik
  - Real-Time-Clock (Mikrocontroller)
  - 7-Segment-Anzeige

Codetabelle:

Dezimal	***BCD***			
	x_3	x_2	x_1	x_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Gray-Code

- Gray-Code:
  - Bei jedem Übergang von einer Zahl zur nächsten ändert sich genau ein Bit
  - Für arithmetische Operationen ungeeignet

- Anwendung:
  - Winkel- oder Längencodierung in der Messtechnik

Codetabelle:

Dezimal	***Gray-Code***		
	x_2	x_1	x_0
0	**0**	0	0
1	0	0	**1**
2	0	**1**	1
3	0	1	**0**
4	**1**	1	0
5	1	1	**1**
6	1	**0**	1
7	1	0	**0**

Festkommazahlen

- Festkommazahlen
  - Abgeleitet vom Dualsystem
  - Feste Anzahl M der Nachkommastellen
  - Die i-te Stelle hat die Wertigkeit 2^{i-M}

Ziffer	x_{N-1}	x_{N-2}	…	x_M	x_{M-1}	x_2	…	x_0
Wertigkeit	2^{N-M-1}	2^{N-M-2}	…	2^0	2^{-1} = 1/2	2^{-2} = 1/4	…	2^{-M}

  - Negative Zahlen mit Betrag und Vorzeichen oder als Zweierkomplement darstellbar

Codetabelle:

Dezimal	***Festkommazahl M = 2, ohne VZ***		
	x_2	x_1	x_0
0.00	0	0	0
0.25	0	0	1
0.50	0	1	0
0.75	0	1	1
1.00	1	0	0
1.25	1	0	1
1.50	1	1	0
1.75	1	1	1

- Codierung:

$$X = x_{N-1} \cdot 2^{N-1-M} + \ldots + x_M \cdot 2^0 + x_{M-1} \cdot 2^{-1} + \ldots + x_0 \cdot 2^{-M}$$

- Beispiel:

$$14/8 = 1.75_{dez} = 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} = 00001.110_{bin}$$

Gleitkommazahlen

- Gleitkommazahlen
  - Standard: IEEE 754
  - Eine Gleitkommazahl X besteht aus *Vorzeichen* S, *Mantisse* M und *Exponent* E

$$X = S \cdot M \cdot 2^E$$

- Normalisierte Darstellung
  - für die Mantisse M gilt: $1 \leq M < 2$
  - d.h.: M hat die Form $1.m_{k-1}m_{k-2}...m_0$
    - Folge: Die „1“ vor dem Komma muss nicht gespeichert werden (*hidden bit*)
- Denormalisierte Darstellung
  - Wird zur verbesserten Darstellung sehr kleiner Zahlen verwendet
  - M hat die Form $0.m_{k-1}m_{k-2}...m_0$
  - $E = 0$

Gleitkommazahlen

- Binärformate:
  - Vorzeichen (1 Bit): 0 = positiv, 1 = negativ
  - Mantisse (k Bit): M-1 als Festkommazahl mit k Nachkommastellen
  - Exponent: (j Bit): Offset-Dualzahl (Offset = 2^{j-1} -1)

- Binärdarstellung (hier: 32-Bit):

Bit	31							24	23							16	15							8	7							0
	s	e	e	e	e	e	e	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m

Exponent

Mantisse

Vorzeichenbit

- Beispiel:

Bit	31							24	23							16	15							8	7							0
	1	1	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$S = (-1)^{\mathbf{1}} = -1$$

$$E = \mathbf{1} \cdot 2^7 + \mathbf{1} \cdot 2^1 - (2^7 - 1) = 130 - 127 = 3$$

$$M = 1 + \mathbf{1} \cdot 2^{-1} + \mathbf{1} \cdot 2^{-3} + \mathbf{1} \cdot 2^{-4} = 1 + 0{,}6875$$

$$\Rightarrow X = -1 \cdot 1{,}6875 \cdot 2^3 = -13{,}5$$

Gleitkommazahlen

- Standardformate nach IEEE 754:

Format	*Genauigkeit*	*Exponent Größe*	*Exponent Bereich*	*Mantisse Größe*
single precision	32-bit	8 Bit	$10^{\pm 38}$	23+1 Bit
double precision	64-bit	11 Bit	$10^{\pm 308}$	52+1 Bit
intern	80-bit	16 Bit	$10^{\pm 4932}$	63+1 Bit

- Sonderfälle:

Typ	*Exponent*	*Mantisse*	*Wert*
infinity	$2^{j} - 1$ (alle Bits = 1)	$= 0$	unendlich
NAN (not a number)	$2^{j} - 1$ (alle Bits = 1)	$\neq 0$	nicht definiert
zero	0	$= 0$	0
denormalized	0	$\neq 0$	zwischen 0 und kleinster normalisierten Zahl

Gleitkommazahlen

- Interne Berechnungen:
  - Denormalisierte Darstellung mit höherer Genauigkeit
- Addition:
  - Angleichung des kleineren an den größeren Exponenten (erfordert Entnormalisierung)
  - Addition der Mantissen
  - Normalisierung des Ergebnisses
- Multiplikation:
  - Multiplikation der Vorzeichen
  - Multiplikation der Mantissen
  - Addition der Exponenten
  - Normalisierung des Ergebnisses

Alphanumerische Zeichen

- Alphanumerische Zeichen werden durch Code-Tabellen codiert
- ASCII (American Standard Code for Information Interchange)
  - Enthält 128 Zeichen (7 Bit)
  - Die ersten 32 Zeichen sind Steuerzeichen für Textausgabegeräte, z.B.:
    - 'HT' = Horizontal Tab
    - 'LF' = Line Feed
    - 'CR = Carrage Return,
  - Verschiedene Erweiterungen mit 8 Bit (Zeichen 128 bis 255)
    - Unterstützen verschiedene Zeichensätze und sprachspezifische Buchstaben
    - z.B. ISO 646, Codepage 437, ISO 8859,..

ersetzt durch

- Unicode
  - Einheitliche und umfassende Codierung für alle relevanten Schriftzeichen
  - 8-, 16- oder 32-Bit-Darstellung, organisiert in 17 Ebenen mit je 2^{16} Zeichen
  - UTF-8 (8-Bit Unicode Transformation Format) ist eine Teilmenge des Unicodes und abwärtskompatibel zu ASCII

Alphanumerische Zeichen

- ASCII-Tabelle:

Code	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1-	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2-	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5-	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7-	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

  - Die ersten 32 und das letzte Zeichen (Code 00H bis 1FH und 7FH) sind nicht darstellbare Steuerzeichen, SP (Code 20H) ist das Leerzeichen

- Beispiel:

Schriftzeichen	dezimal	hexadezimal	dual
A	65	41	0100 0001
{	123	7B	0111 1011