TECHNISCHE UNIVERSITÄT DRESDEN

VLSI-EDA

Fakultät Informatik, Institut für Technische Informatik, Professur VLSI-Entwurfssysteme, Diagnostik und Architektur

Rechnerstrukturen und -organisation

Informations-codierung

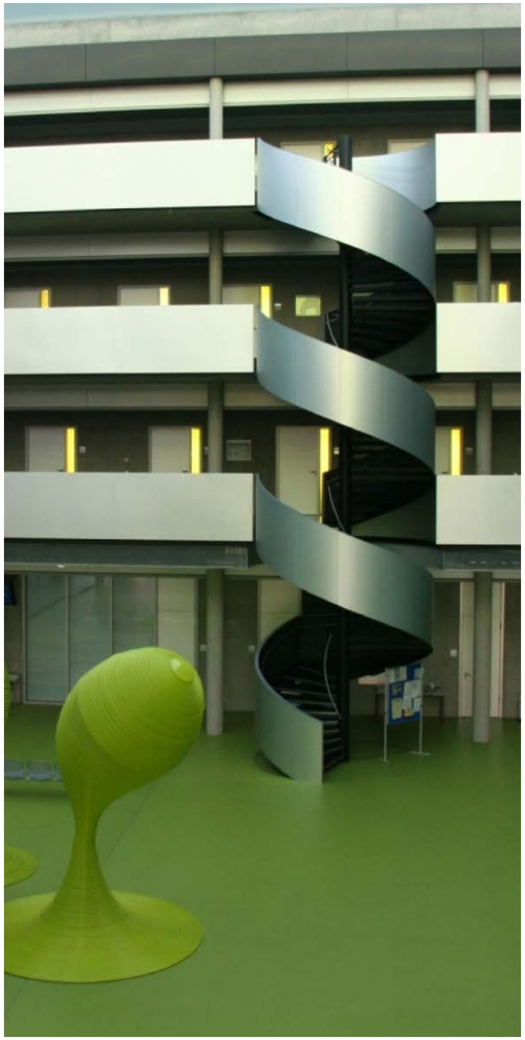
Rainer G. Spallek

TU Dresden, 29.10.2020

Rainer.Spallek@tu-dresden.de

http://vlsi-eda.inf.tu-dresden.de

Gliederung



1 Zielstellung

- Erlangung eines Grundverständnisses für die Codierungstheorie
- Auseinandersetzung mit den Begriffen Codierung, Codewort, Abbildung
- Kennenlernen des Zusammenhanges Information -- Signal
- Erwerb von Grundlagen über binäre Codierungen und binäre Blockcode
- Verständnis im Umgang mit üblichen binären Blockcode

2 Codierung

Informationen können in codierter Form als Zeichen bzw. Zeichenfolgen einer Zeichenmenge *Z* oder als Signale bzw. Signalfolgen einer Signalmenge *S* vorliegen.

Codierung ist eine injektive (linkseindeutige) Abbildung κ einer endlichen Menge von Zeichen Z_A eines Alphabetes *A* in eine geeignete Folge (*n*-faches kartesisches Produkt) der Signalmenge *S* ($S^n = S \times \cdots \times S$) bzw. der Zeichenmenge Z_B (${Z_B}^n = ZB \times \cdots \times ZB$) eines anderen Alphabetes *B.*

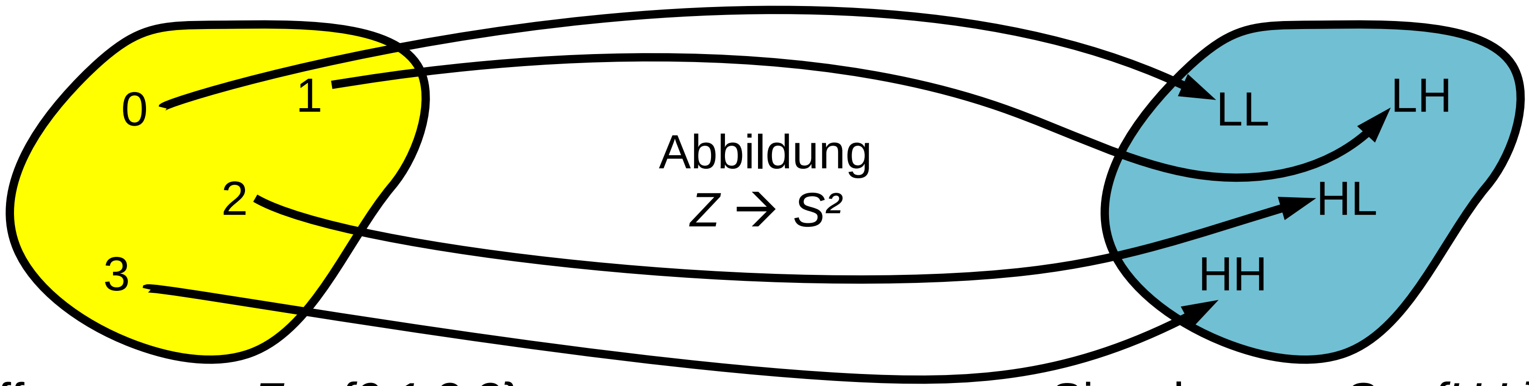
Codierung als Abbildung $\kappa : A \rightarrow S^n$ bzw. $\kappa : A \rightarrow {Z_B}^n$

bzw. Alphabetwandlung $\kappa : A \rightarrow B$ *(für n*=1*)*

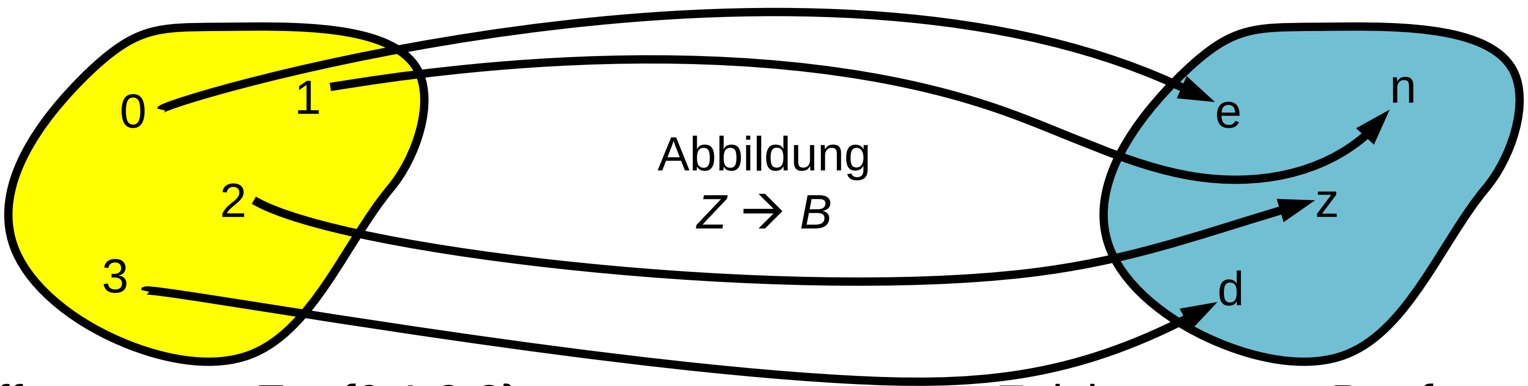
Die Elemente von S^n bzw Z^n werden als n-Tupel oder Codewörter bezeichnet:

$(s_1 s_2 \ldots s_\mu \ldots s_n) \in S^n$ mit $s_\mu \in S$ *bzw.* $(z_1 z_2 \ldots z_\mu \ldots z_n) \in Z^n$ mit $z_\mu \in Z$.

Beispiel: Codierung als Abbildung



Ziffernmenge $Z = \{0,1,2,3\}$ Signalmenge $S = \{H,L\}$



Ziffernmenge $Z = \{0,1,2,3\}$ Zeichenmenge $B = \{n,e,z,d\}$

Codetabelle

Codes können in Codetafeln, Codetabellen dargestellt werden.

Signalmenge: $S=\{L,H\}$
Ziffernmenge: $Z=\{0,1,2,3\}$
Abbildung: $Z \rightarrow S^2=\{LL,LH,HL,HH\}$

Z	S^2
0	LL
1	LH
2	HL
3	HH

L: low Pegel (niedriges Potential)
H: high Pegel (hohes Potential)

Ziffernmenge: $Z=\{0,1,2,3\}$
Zeichenmenge: $B=\{d,e,n,z\}$
Abbildung: $Z \rightarrow B$

Z	S^2
0	n
1	e
2	z
3	d

Beispiel: Morse-Alphabet 1

Signalmenge *M={●,–} = {kurzer Ton, langer Ton}*

Codierung: *μ: {A…Z, 0…9} → {●,–}*^{n} Pausezeichen ○ nicht mit codiert !

A	● –	J	● – – –	S	● ● ●	1	● – – – –
B	– ● ● ●	K	– ● –	T	–	2	● ● – – –
C	– ● – ●	L	● – ● ●	U	● ● –	3	● ● ● – –
D	– ● ●	M	– –	V	● ● ● –	4	● ● ● ● –
E	●	N	– ●	W	● – –	5	● ● ● ● ●
F	● ● – ●	O	– – –	X	– ● ● –	6	– ● ● ● ●
G	– – ●	P	● – – ●	Y	– ● – –	7	– – ● ● ●
H	● ● ● ●	Q	– – ● –	Z	– – ● ●	8	– – – ● ●
I	● ●	R	● – ●	0	– – – – –	9	– – – – ●

(z.B. SOS: ● ● ● ○ – – – ○ ● ● ● oder RGS: ● – ● ○ – – ● ○ ● ● ●)

Eigenschaften der Morse-Codierung

Eigenschaften:

- Die Codierung der alphanumerischen Zeichen erfolgt mit nur zwei Zeichen, Punkt und Strich.
- Es werden Codewörter (ungleichmäßig) unterschiedlicher Länge ($n = 1...5$) verwendet.
- Nicht alle möglichen Codewörter werden verwendet.
- Ziffern haben Codewörter fester Länge ($n = 5$) mit gewisser Systematik.
- Buchstaben haben Codewörter unterschiedlicher Länge ($n = 1…4$)
- die Pause (○) dient ausschließlich der Trennung der Codewörter und damit der Dekcodierbarkeit.

Eigenschaften der Morse-Codierung

Schlussfolgerungen:

- Code mit Redundanz,
- Decodierbarkeit durch Pausenzeichen (Trennzeichen) gesichert,
- erleichterte Decodierbarkeit von Ziffern durch feste Codewortlänge und Systematik, → Ziffern im Blockcode codiert
- häufig vorkommende Buchstaben haben eine kurze Codewortlänge, selten vorkommende ein lange → Buchstaben in Entropiecodierung codiert
- Entropiecodierung z.B. Huffman-Code

Decodierbarkeit eines Codes

Eine Codierung ist decodierbar, wenn mindestens eine der folgenden Voraussetzungen gegeben ist:

- alle Codewörter sind gleich lang (Blockcode-Eigenschaft),
- Verwendung eines gesonderten Trennzeichens,
- kein (kurzes) Codewort ist Anfang bzw. Ende eines anderen (langen) Codewortes (Präfixeigenschaft).

Suffix-Code

Ungleichmäßiger Code, bei dem kein (kurzes) Codewort Ende (Suffix) eines anderen (langen) Codewortes darstellt.

Präfix-Code

Ungleichmäßiger Code, bei dem kein (kurzes) Codewort Anfang (Präfix) eines anderen (langen) Codewortes ist, (Präfixeigenschaft : z.B. Huffman-Code).

Zielstellung der Codierung

Beeinflussung der Informationsdarstellung durch gezielte Codierung:

- Lesbarkeit
- Verarbeitbarkeit
- Übertragbarkeit
- Fehlersicherheit
- Speicherbarkeit
- Vertraulichkeit

Anwendungsgebiete für die Codierung:

- Informationsdarstellung, -speicherung (Computertechnik)
- Informationsverschlüsselung (Kryptographie)
- Informationsübertragung (Kommunikationstechnik)
- Informationsverarbeitung (Computertechnik)

3 Binäre Codierung

Binär bedeutet zweiwertig, dual, bivalent. Codierungen für moderne elektronische Computer basieren praktisch ausschließlich auf der Menge $B = \{0,1\}$ der binären Zeichen 0 und 1. Die binären Zeichen $0,1 \in B$ werden physischen Signalen (zweiwertigen Zuständen) zugeordnet, z.B.:

Zeichen	0^+ (1^-)	1^+ (0^-)
Schalter	offen	geschlossen
Spannung	niedrig	hoch
Pegel	L (low)	H (high)
Kondensator	entladen	geladen
Magnetfeld	neg. Orientierung	pos. Orientierung

+ positive Logik, (- negative Logik)

➔ Im Weiteren soll nur noch positive Logik verwendet werden.

Binäre Codierung Darstellung

Mit $B = \{0,1\}$ können nur 2 Zeichen, *0* oder *1* dargestellt bzw. codiert werden.

➔ Übergang zu Zeichenfolgen, Codewörtern der binären Zeichen $0,1 \in B$.

Eine binäre Codierung ist eine injektive Abbildung einer endlichen Menge von Zeichen eines Alphabetes *A* in geeignete Folgen von nur zwei verschiedenen (binären) Zeichen der unterliegenden binären Zeichenmenge *B.*

$$\kappa : A \rightarrow B^n \;\; ; \;\; B^n = B \times \ldots \times B$$

B^n bezeichnet das n-fache kartesische Produkt über der Menge *B.*

Die Elemente von B^n sind n-Tupel, Zeichenfolgen mit:

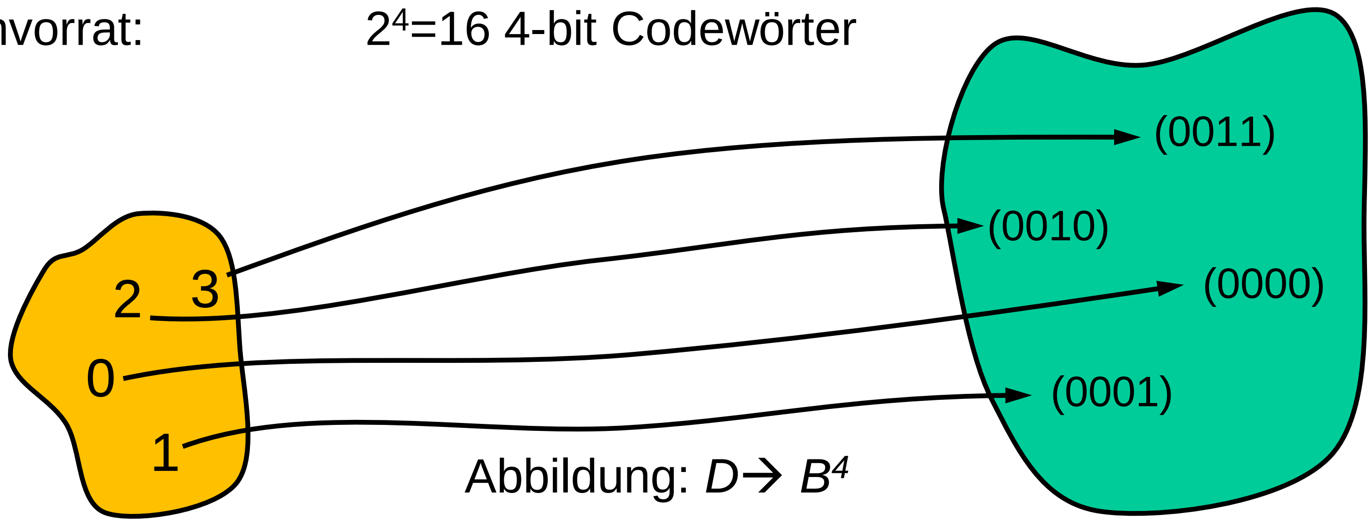
$$(b_1 b_2 \ldots b_\mu \ldots b_n) \in B^n \text{ und } b_\mu \in B$$

Sie werden als binäre Codewörter der Länge *n* bezeichnet (*n*-bit Codewort).

Binäre Codierung als Abbildung

Beispiel: Codierung von Dezimalziffern mit binären 4-bit Codewötern

Dezimale Ziffernmenge:	D={0,1,2,3}
Binäre Menge:	B={0,1}
Codierung:	$\kappa: D \rightarrow B^4$
Codewortlänge:	4
Zeichenvorrat:	2^4=16 4-bit Codewörter



Ziffernmenge D={0,1,2,3}

Teilmenge von B^4

Binäre Blockcodes

In der Computertechnik dominieren aufgrund der leichteren Decodierbarkeit binäre Codes fester Länge *n* (*n*-bit-Blockcode).

Codierung:	$\beta: A \rightarrow \{0,1\}^n$
Codewort:	$(b_1\, b_2 \ldots b_\nu \ldots b_n) \in B^n$ und $b_\nu \in B$, $B = \{0,1\}$
Codewortlänge:	n
Zeichenvorrat:	2^n verschieden *n*-bit Codewörter

Die Codewörter werden als Zeichenfolgen der Binärziffern *0* und*1* dargestellt. Eine Binärziffer wird bit (**bi**nary digi**t**) genannt.

Beispiel $n = 4$ → 16 verschiedene mögliche 4-bit Codewörter:

$$\{0,1\}^4 = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, \\ 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\}$$

Dichte / Volle Binäre Blockcodes

Ein binärer Blockcode mit n Stellen (n-bit Codierung)

mit $(b_1 b_2 \ldots b_\mu \ldots b_n) \in B^n$ *und* $b_\mu \in B$, $B = \{0,1\}$

realisiert maximal 2^n verschiedene n-bit Codewörter.

Dichter Code
Ein dichter Blockcode liegt vor, wenn q verschiedene Codewörter für die Abbildung benötigt werden und n die folgende Bedingung erfüllt:

$$2^{n-1} < q \leq 2^n.$$

Eine Reduktion von n ist hier nicht mehr möglich.

Voller Code
Ein Blockcode wird dann voll genannt, wenn q verschiedene Codewörter für die Abbildung benötigt werden und n genau die folgende Bedingung erfüllt:

$$q = 2^n.$$

Damit ist kein weiteres Codewort mehr darstellbar.

Übliche Formate binärer Blockcodes

Folgende Codewortlängen und Bezeichnungen sind für die binären Blockcode in der Computertechnik üblich:

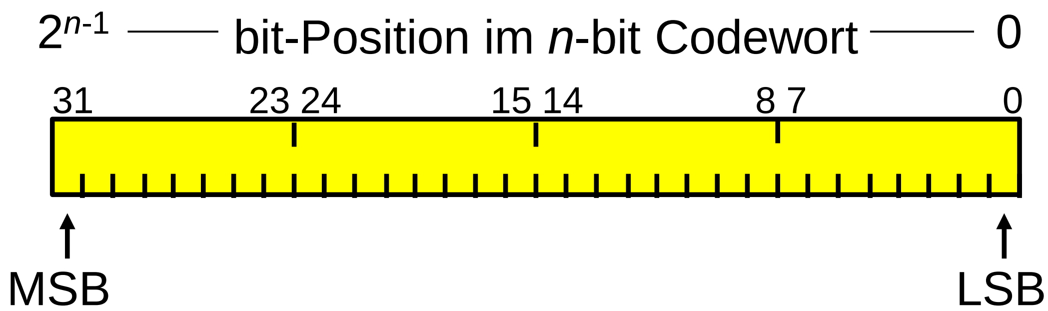
n= 1				x	bit
n= 4				xxxx	Halbbyte (Nibble)
n= 8			xxxx	xxxx	Byte
n= 16	xxxx	xxxx	xxxx	xxxx	16-bit Wort
n= 32					32-bit Wort
n= 64					64-bit Wort

Orientierung der Wortbreite an der Verarbeitungsbreite des Computers:

- 16-bit Mikroprozessoren → 16-bit Wort,
- 32-bit Mikroprozessoren → 32-bit Wort,
- 64-bit Mikroprozessoren → 64-bit Wort.

Übliche Wortunterteilungen: Halbwort, Wort, Doppelwort, Quadwort.

Codewortdarstellung binärer Blockcode



MSB: most significant bit
LSB: least significant bit

Übliche Dimensionsangeben (1 Byte = 8 bit)

KiB	(Kibibyte)	:	2^{10} Byte			=	1024 Byte
MiB	(Mebibyte)	:	2^{20} Byte	=	1024 KiB	=	1 048 576 Byte
GiB	(Gibibyte)	:	2^{30} Byte	=	1024 MiB	=	1 073 741 824 Byte
TiB	(Tebibyte)	:	2^{40} Byte	=	1024 GiB	=	1 099 511 627 776 Byte
PiB	(Pebibyte)	:	2^{50} Byte	=	1024 TiB	=	1 125 899 906 842 624 Byte

ISO-Norm IEC 80000-13:2008 (bzw. DIN EN 80000-13:2009-01)

SI-Präfixe sind nur für SI-Einheiten standardisiert ➔ Binärpräfixe (IEC)

4 Wichtige binäre Codes

1. Binär-Code
2. BCD-Code
3. Hexadezimal-Code
4. Oktal-Code
5. M-aus-N-Code (1-aus-N-Code)
6. Gray-Code
7. ASCII-Code
8. Zahlendarstellung

4.1 Binär-Code (BIN)

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

Bildungsvorschrift:
Die Zuordnung erfolgt entsprechend dem Binäräquivalent (Dualzahl).

Anwendung:
Adressen in Computern,
Dualzahlendarstellung .

Binäre Zahlendarstellung als Dualzahl

Jeder Dezimalzahl wird die zu ihr wertmäßig äquivalente binär kodierte Dualzahl zugeordnet (→ Stellenwertsystem zur Basis 2).

Das dezimale Zahlensystem wird direkt auf das Dualzahlensystem abgebildet:

Dezimalziffern:	$z_i \in Z = \{0,1,2,3,4,5,6,7,8,9\}$
Dezimalzahlen:	$G \to Z^n, g \in G$
Dualziffern:	$b_j \in B = \{0,1\}$
Dualzahlen:	$D \to B^n, d \in D$
Kodierung:	$\delta : G \to D$

Wertgleiche Zuordnung:

$$g = \sum_{i=0}^{m-1} z_i \cdot 10^i = d = \sum_{j=0}^{n-1} b_j \cdot 2^j$$

Beispiel: $314_{10} = 0000\ 0001\ 0011\ 1010_2$

4.2 BCD-Code (Binary Coded Decimals)

DEZ	BCD	Aiken	3XS
0	0000	0000	0011
1	0001	0001	0100
2	0010	0010	0101
3	0011	0011	0110
4	0100	0100	0111
5	0101	1011	1000
6	0110	1100	1001
7	0111	1101	1010
8	1000	1110	1011
9	1001	1111	1100

Bildungsvorschrift:
Die Zuordnung erfolgt entsprechend dem Binäräquivalent 0-9.

Anwendung:
Darstellung von Dezimalziffern.

Die nicht in der Abbildung berücksichtigten Tetraden (10-15) werden als Pseudotetraden bezeichnet (6 Pseudotetraden).

Binäre Zahlendarstellung im BCD-Code

Jeder Dezimalziffer wird genau ein 4-bit langes binäres Codewort (Tetrade) entsprechend ihrem Binäräquivalent zugeordnet (auch 8241-Code).

Die Codierung einer n-stelligen Dezimalzahl erfolgt ziffernweise durch Aneinanderreihung der BCD-codierten Dezimalziffern.

Dezimalziffern: $z_i \in Z = \{0,1,2,3,4,5,6,7,8,9\}$
Codierung: $\delta : Z \rightarrow \{0,1\}^4$

Dem BCD-Code ähnliche Codes sind der Aiken-Code und der 3XS-Code.

Bei der Rechnung mit BCD-Zahlen sind die Pseudotetraden unbedingt zu beachten.

Beispiel: $314_{10} = 0011\ 0001\ 0100_{BCD}$

4.3 Hexadezimale-Code

DEZ	BIN	HEX
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Bildungsvorschrift:
Die Zuordnung erfolgt zu je 4 bit entsprechend dem Binär-äquivalent (Hexadezimalzeichen 0…9, A…F). Für 10 - 15 wird A - F codiert .

Anwendung:
Adressen in Computern,
Hexadezimalzahlendarstellung.

Hexadezimale Darstellung

Für die einfachere Darstellung binärer Codierungen mit großer Binärstellenzahl werden oft hexadezimale Codierungen verwendet.

Zahlendarstellung als Hexadezimalzahl: Stellenwertsystem zur Basis 16.

$$g = \sum_{i=0}^{l-1} h_i \cdot 16^i \quad \text{mit} \quad h_i \in H = \{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$$

Konvertierung binär → hexadezimal

Unterteilung der binären Codierung mit dem LSB beginnend in Vierergruppen. Ersetzen der Vierergruppen durch die entsprechenden Hexadezimalzeichen.

Konvertierung hexadezimal → binär

Ersetzen der Hexadezimalzeichen durch ihre Binärcodierung (Bitstellen).

Beispiel: $314_{10} = 013A_{16}$

4.4 Oktal-Code

DEZ	BIN	OKT
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Bildungsvorschrift

Die Zuordnung erfolgt zu je 3 bit entsprechend dem Binäräquivalent (Oktalzeichen 0 – 7).

Anwendung

Adressen in Computern,
Oktalzahlendarstellung.

Oktale Darstellung

Für dein einfachere Darstellung binärer Codierungen mit großer Binärstellenzahl wurden teilweise auch die oktale Codierungen verwendet.

Zahlendarstellung als Oktalzahl: Stellenwertsystem zur Basis 8.

$$g = \sum_{i=0}^{k-1} o_i \cdot 8^i \quad \text{mit} \quad o_i \in O = \{0,1,2,3,4,5,6,7,8\}$$

Konvertierung binär → oktal

Unterteilung der binären Codierung mit dem LSB beginnend in Dreiergruppen. Ersetzen der Dreiergruppen durch das entsprechende Oktalzeichen.

Konvertierung oktal → binär

Ersetzen der Oktalzeichen durch ihre Binärcodierung (Bitstellen).

Beispiel: $314_{10} = 00472_8$

4.5 M-aus-N-Code

Der M-aus-N-Code hat genau $\binom{N}{M}$ Codewörter der Länge *N*, die jeweils genau *M* 1-bit enthalten, sonst alles 0-bits

Codewortlänge: *N*; Anzahl der 1-bit-Stellen: M

Anzahl der Codewörter: $\binom{N}{M} = \frac{N!}{M!(N-M)!}$

Beispiel: 2 aus 4 Code

$\binom{4}{2} = \frac{4!}{2!(2-4)!} = 2 \cdot 3 = 6$ Codewörter: 1100, 1010, 1001, 0101, 0011, 0110.

Sonderfall: 1-aus-N-Code (one hot code)

Anwendung: Spezielle Codierungen, Adressdecodierung, Zahlendarstellung.

Beispiel: 314_{10} = $0000001000\ 0000000010\ 0000010000_{\text{1-aus-10}}$

1-aus-N-Code (one-hot-code)

DEZ	1-aus-10
0	00000 00001
1	00000 00010
2	00000 00100
3	00000 01000
4	00000 10000
5	00001 00000
6	00010 00000
7	00100 00000
8	01000 00000
9	10000 00000

Bildungsvorschrift 1-aus-10-Code:

Die Dezimalzahl von rechts (LSB) entsprechende Bitstelle ist 1 alle anderen sind 0.

Anwendung:

Adressdecoder, einfache Zahlendarstellung

Der 1-aus-N-Code (**one-hot-code**) hat genau *N* Codewörter der Länge *N*, die jeweils nur ein 1-bit enthalten, sonst alles 0-bit. Die einzelnen Codewörter unterscheiden sich untereinander immer nur in jeweils 2 Bitstellen.

4.6 Gray-Code

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

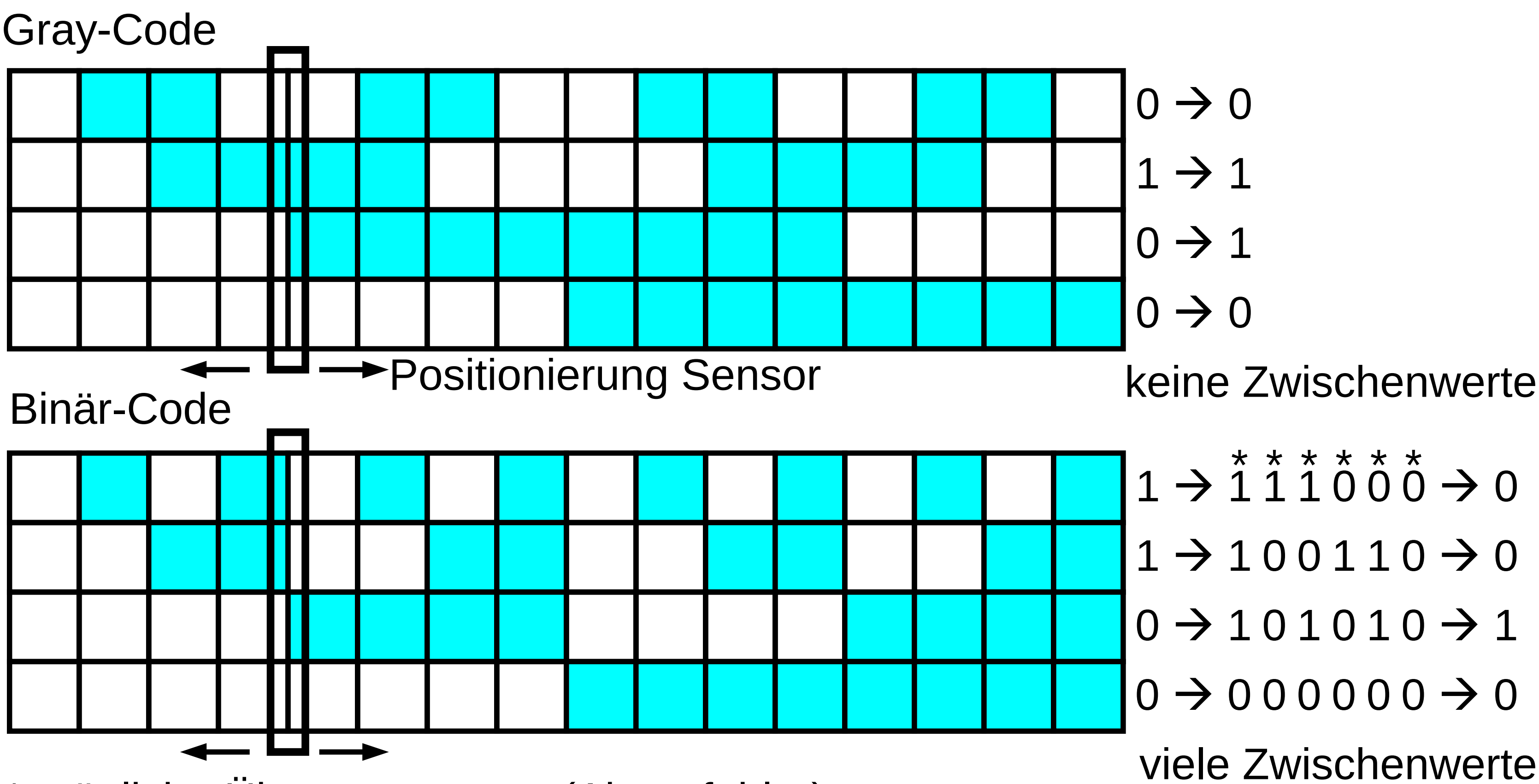
In einer Folge von Codewörtern unterscheiden sich benachbarte Codewörter jeweils nur in einer einzigen Bitstelle.

Bildungsvorschrift:
Generierung aus Binärcode
$X_{Gray} := (X_{BIN}\ \text{XOR}\ (1/2 * X_{BIN}))$.

1/2 → 1bit rechts schieben

Anwendung:
Messtechnik, Zähler,
- Erhöhung der Störsicherheit,
- Reduzierung der Verlustleistung.

Positionssensor zur absoluten Lageposition



* mögliche Übergangswerte (Abtastfehler)

4.7 ASCII-Code

American Standard Code for Information Interchange (ASCII, auch ANSI X3.4-1986) ist ein 7-bit binärer Blockcode.

Die Zeichencodierung des ASCII-Code definiert 128 Zeichen, 33 nicht-druckbare und 95 druckbare.

ASCII-Zeichen: $a \in A$

ASCII-Codierung: $\alpha : A \rightarrow \{0,1\}^7$

Werden ASCII-Zeichen mit 7-bit codiert und im Byte-Format dargestellt, wird die Bitposition des MSB durchgängig mit 0 oder einem Paritätsbit aufgefüllt.

ASCII-Code ist als gemeinsamer Subcode in fast allen 8-bit Zeichencodierungen und auch im Unicode enthalten (Ausnahme **EBCDIC-Code**, Extended Binary Coded Decimal Interchange Code, IBM).

UTF-8 ist eine 8-bit Codierung von Unicode, die zum ASCII-Code abwärtskompatibel ist. Ein Zeichen kann dabei 1 bis 4 8-bit-Wörter einnehmen. → Codierung variabler Länge, kein reiner Blockcode.

Umschaltzeichen

Mit *n*-bit Codewortlänge können maximal 2^n Zeichen binär codiert werden.

Erweiterungen bzw. Erhöhung der Anzahl der codierbaren Zeichensätze durch:

- Erhöhung der Bitstellenanzahl *n* (Codewortlänge),
- Einführung von Umschaltzeichen (Zeichensatzanzahl).

Mit *u* Umschaltzeichen können *u* verschiedene Zeichensätze mit jeweils $2^n - u$ Zeichen codiert werden.

Zeichenvorrat:	$u(2^n - u)$
Maximaler Zeichenvorrat:	2^{2n-2} bei $u = 2^{n-1}$

Umschaltzeichen werden kaum noch verwendet.

ASCII-Code (ISO-7-Bit-Code)

HEX	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	
0	NUL	DLE	SP	0	@/§	P	`	p	0000
1	SOH	DC1	!	1	A	Q	a	q	0001
2	STX	XON	"	2	B	R	b	r	0010
3	ETX	DC3	#	3	C	S	c	s	0011
4	EOT	XOF	$	4	D	T	d	t	0100
5	ENQ	NAK	%	5	E	U	e	u	0101
6	ACK	SYN	&	6	F	V	f	v	0110
7	BEL	ETB	'	7	G	W	g	w	0111
8	BS	CAN	(	8	H	X	h	x	1000
9	HT	EM	)	9	I	Y	i	y	1001
A	LF	SUB	*	:	J	Z	j	z	1010
B	VT	ESC	+	;	K	[/Ä	k	{/ä	1011
C	FF	FS	,	<	L	\/Ö	l	|/ö	1100
D	CR	GS	-	=	M	]/Ü	m	}/ü	1101
E	SO	RS	.	>	N	^	n	~/ß	1110
F	SI	US	/	?	O	_	o	DEL	1111
	000	001	010	011	100	101	110	111	BIN

(ASCII – American Standard Code for Information Interchange) (ISO 7-Bit-Code 646 oder DIN-Norm 66003)

ASCII-Control-Code

Dec	Code	HEX	Name
0	NUL	00	Null
1	SOH	01	Start of heading
2	STX	02	Start of text
3	ETX	03	End of text
4	EOT	04	End of transmission
5	ENQ	05	Enquiry
6	ACK	06	Acknowledge
7	BEL	07	Bell
8	BS	08	Back space
9	HT	09	Horizontal tab
10	LF	0A	Linefeed
11	VT	0B	Vertical tab
12	FF	0C	Formfeed
13	CR	0D	Carriage return
14	SO	0E	Shift out
15	SI	0F	Shift in
16	DLE	10	Data link escape

Dec	Code	Hex	Name
17	DC1	11	Device control 1 (XON)
18	DC2	12	Device control 2
19	DC3	13	Device control 3 (XOFF)
20	DC4	14	Device control 4
21	NAK	15	Negative acknowledge
22	SYN	16	Synchronous idle
23	ETB	17	End transmission block
24	CAN	18	Cancel
25	EM	19	End of medium
26	SUB	1A	Substitute (EOF)
27	ESC	1B	Escape
28	FS	1C	File separator
29	GS	1D	Group separator
30	RS	1E	Record separator
31	US	1F	Unit separator
...			
127	DEL	7F	Delete

ANSI (8-Bit) Code Table for Windows

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	NUL	DEL	SP	0	@	P	`	p				°	À	Ð	à	ð	**0000**
1	SOH	DC1	!	1	A	Q	a	q		`	¡	±	Á	Ñ	á	ñ	**0001**
2	STX	DC2	"	2	B	R	b	r	,	´	¢	^{2}	Â	Ò	â	ò	**0010**
3	ETX	DC3	#	3	C	S	c	s	f	“	£	^{3}	Ã	Ó	ã	ó	**0011**
4	EOT	DC4	$	4	D	T	d	t	“	“	¤	´	Ä	Ô	ä	ô	**0100**
5	ENQ	NAK	%	5	E	U	e	u	...	*	¥	µ	Å	Õ	å	õ	**0101**
6	ACK	SYN	&	6	F	V	f	v	?	-	¦	¶	Æ	Ö	æ	ö	**0110**
7	BEL	ETB	'	7	G	W	g	w	?	_	§	·	Ç	×	ç	÷	**0111**
8	BS	CAN	(	8	H	X	h	x		~	¨	¸	È	Ø	è	ø	**1000**
9	HT	EM	)	9	I	Y	i	y	?	[tm]	©	^{1}	É	Ù	é	ù	**1001**
A	LF	SUB	*	:	J	Z	j	z	S	s	ª	º	Ê	Ú	ê	ú	**1010**
B	VT	ESC	+	;	K	[	k	{	<	>	«	»	Ë	Û	ë	û	**1011**
C	FF	FS	,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü	**1100**
D	CR	GS	-	=	M	]	m	}				½	Í	Ý	í	ý	**1101**
E	SO	RS	.	>	N	^	n	~			®	¾	Î	Þ	î	þ	**1110**
F	SI	US	/	?	O	_	o	DEL		Y	¯	¿	Ï	ß	ï	ÿ	**1111**
	0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**	**Bin**

4.8 Codierung zur Zahlendarstellung

OCT	3	0	5	4	1	7	0	2	6	3	1
BIN	11	000	101	100	001	111	000	010	110	011	001

BIN	1100	0101	1000	0111	1000	0101	1001	1001
HEX	C	5	8	7	8	5	9	9

$1100\ 0101\ 1000\ 0111\ 1000\ 0101\ 1001\ 1001_2$ Dualzahl

$= C5\ 87\ 85\ 99_{16}$ Hexadezimalzahl

$= 305\ 4170\ 2631_8$ Oktalzahl

$= 3\ 313\ 993\ 113_{10}$ Dezimalzahl

DEC	3	3	1	3	9	9	3	1	1	3
BCD	0011	0011	0001	0011	1001	1001	0011	0001	0001	0011

8 Zusammenfassung

- Codierung ist zwingend notwendig, um Information durch Signale darzustellen, abzubilden.
- Die Codierung bestimmt die Darstellungsvorschrift von Informationseinheiten mit Hilfe der Zeichen eines Alphabetes.
- Binäre Blockcode werden in der Computertechnik bevorzugt.
- Eine Codetabelle stellt als tabellarische Zuordnung die Abbildungsvorschrift dar.
- Es gibt verschiedene Standardcodes für Zahlendarstellungen, Textdarstellungen, Adressen, Auswahlentscheidungen usw.