Computer Systems in Engineering

Codes

Hendrik Weiss

Byron Worms

Raik Dankworth

Georg Jäger

Inhalt

Computer Systems in Engineering

- Information und Sprache
  - Definition Information
  - Sprache als Repräsentationsform von Information
- Zahlencodes
  - Umwandelung
  - BCD-Code
- Kodierung
  - Definitionen
  - Morse - , ASCII - Code usw.
- Gray-Code
  - Erläuterung
  - Umwandelung
- Fehlererkennung/Korrigieren
  - Drei Ansätze
  - Hamming

Information

Zitat: „Information ist der Unterschied, der einen Unterschied macht.“

(Math.: „Wissen2-Wissen1 = Information“)

Aufnehmen von Information bedeutet die Erweiterung des vorhandenen Wissens

Sprachdefinition

Computer Systems in Engineering

- Gegeben: Ein Alphabet A = {a1,a2,a3,...,an}
- A* bezeichnet die Wortmenge über dem Alphabet A

- Mithilfe der Wortmenge können nun Informationen in dieser Sprache dargestellt werden.

- Beispiel:
  - A = {0,1}

Zahlencodes

- Definiert durch eine Sprache X
  - |x| = Basis
- Wert hängt von der Stelle ab
  - $Faktor * Basis^{Position}$
- Aufbau für alle System gleich
  - Hexadezimal
  - Oktal
  - Binär
  - …

- X = {0…9} Dezimalzahlen
- $563 = 5*10^2 + 6*10^1 + 3*10^0$
- Hex
  - X = {0…F} mit Basis 16
- Oktal
  - X = {0…7} mit Basis 8
- Binär
  - X = {0, 1} mit Basis 2

BCD-Code

- Problematik: Dezimalzahlbasis 10 ist keine 2er Potenz
- Konsequenz: Die Codierung/Decodierung funktioniert nicht per einzelne Zahl

- Lösung:
  - Gleiches Vorgehen wie bei hexadezimale Zahlen
  - Jede Zahl hat 4-Bit in binärer Darstellung
    - 2361 = 0010.0011.0110.0001

- Vorteile:
  - Leicht und fehlerlos umwandelbar

- Nachteile
  - Weniger kompakt als die Binärdarstellung

Definition Kodierung

- Wörtliche Definition:
  - Die Darstellung einer Präsentation von einer Information
  - Für jeden Code speziell, so dass die Präsentation von einer Information in unterschiedlichen Codes anders aussehen können

- Mathematische Definition:
  - Die Abbildung von einer Wortmenge auf eine andere Wortmenge
  - c: $A^* \rightarrow B^*$, wobei jedem Wort aus A^* ein Wort aus B^* zugeordnet wird
  - Wenn c injektiv so ist die Kodierung entzifferbar
  - Die Umkehrfunktion ist dann der Dekodierer von dem Code

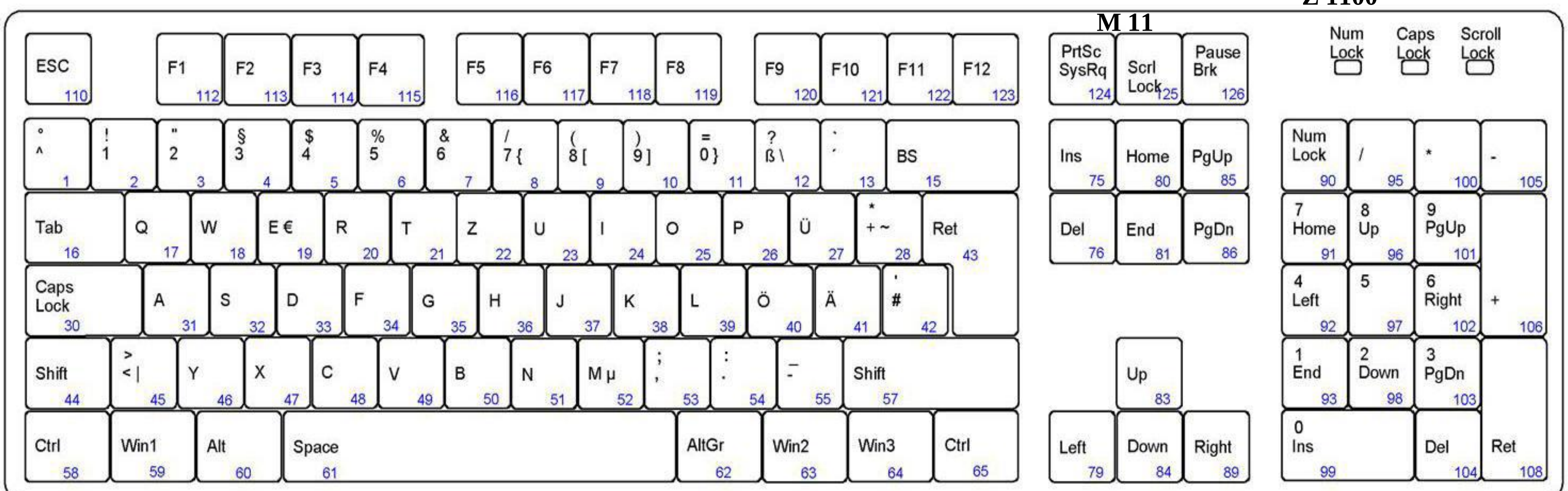
Morse-/ Tastaturcode

Computer Systems in Engineering

- Morse – Code:
  - A* = Lateinische Buchstaben
  - B* = {lang, kurz} (Pause) als Wortende
  - Code mit maximal 4 Bit – Darstellung
- Tastaturcode

A 01
B 1000
C 1010
D 100
E 0
F 0010
G 110
H 0000
I 00
J 0111
K 101
L 0100
M 11

N 10
O 111
P 0110
Q 1101
R 010
S 000
T 1
U 001
V 0001
W 011
X 1001
Y 1011
Z 1100



ASCII - /Unicode

- ASCII – Code:
  - Vorwiegend in der westlichen Welt benutzt
  - A* = lateinische Alphabet, arabische Ziffern und Sonderzeichen
  - B* = 8 Bit – Darstellung
  - |A*| = 100 und |B*| = 256
- Unicode:
  - Zur Darstellung von verschiedenen Alphabeten in der Welt
  - A* = verschiedene Sprachen z.B. hebräisch, kyrillisch
  - B* = 16 Bit – Darstellung
  - 16 Bit – Darstellung notwendig, da einige Sprachen mehrere tausend Wörter in ihrem Alphabet haben

Computer Systems in Engineering

ASCII Code

Scan-code	ASCII hex dez		Zeichen		Scan-code	ASCII hex dez		Zch.	Scan-code	ASCII hex dez		Zch.	Scan-code	ASCII hex dez		Zch.
	00	0	NUL			20	32	SP		40	64	@	0D	60	96	`
	01	1	SOH	^A	02	21	33	!	1E	41	65	A	1E	61	97	a
	02	2	STX	^B	03	22	34	"	30	42	66	B	30	62	98	b
	03	3	ETX	^C	29	23	35	#	2E	43	67	C	2E	63	99	c
	04	4	EOT	^D	05	24	36	$	20	44	68	D	20	64	100	d
	05	5	ENQ	^E	06	25	37	%	12	45	69	E	12	65	101	e
	06	6	ACK	^F	07	26	38	&	21	46	70	F	21	66	102	f
	07	7	BEL	^G	0D	27	39	'	22	47	71	G	22	67	103	g
0E	08	8	BS	^H	09	28	40	(	23	48	72	H	23	68	104	h
0F	09	9	TAB	^I	0A	29	41	)	17	49	73	I	17	69	105	i
	0A	10	LF	^J	1B	2A	42	*	24	4A	74	J	24	6A	106	j
	0B	11	VT	^K	1B	2B	43	+	25	4B	75	K	25	6B	107	k
	0C	12	FF	^L	33	2C	44	,	26	4C	76	L	26	6C	108	l
1C	0D	13	CR	^M	35	2D	45	-	32	4D	77	M	32	6D	109	m
	0E	14	SO	^N	34	2E	46	.	31	4E	78	N	31	6E	110	n
	0F	15	SI	^O	08	2F	47	/	18	4F	79	O	18	6F	111	o
	10	16	DLE	^P	0B	30	48	0	19	50	80	P	19	70	112	p
	11	17	DC1	^Q	02	31	49	1	10	51	81	Q	10	71	113	q
	12	18	DC2	^R	03	32	50	2	13	52	82	R	13	72	114	r
	13	19	DC3	^S	04	33	51	3	1F	53	83	S	1F	73	115	s
	14	20	DC4	^T	05	34	52	4	14	54	84	T	14	74	116	t
	15	21	NAK	^U	06	35	53	5	16	55	85	U	16	75	117	u
	16	22	SYN	^V	07	36	54	6	2F	56	86	V	2F	76	118	v
	17	23	ETB	^W	08	37	55	7	11	57	87	W	11	77	119	w
	18	24	CAN	^X	09	38	56	8	2D	58	88	X	2D	78	120	x
	19	25	EM	^Y	0A	39	57	9	2C	59	89	Y	2C	79	121	y
	1A	26	SUB	^Z	34	3A	58	:	15	5A	90	Z	15	7A	122	z
01	1B	27	Esc		33	3B	59	;		5B	91	[		7B	123	{
	1C	28	FS		2B	3C	60	<		5C	92	\		7C	124	|
	1D	29	GS		0B	3D	61	=		5D	93	]		7D	125	}
	1E	30	RS		2B	3E	62	>	29	5E	94	^		7E	126	~
	1F	31	US		0C	3F	63	?	35	5F	95	_	53	7F	127	DEL

Computer Systems in Engineering

Gray-Code

- Für manche Anwendungen möchte man von einem Codewort zum nächsten eine minimale und konstante Anzahl an Änderungen:
- Beispiel: Modulo 8 Zähler:

Änderungen	Binär	Zahl	Gray-Code	Änderungen
1	000	0	000	1
2	001	1	001	1
1	010	2	011	1
3	011	3	010	1
1	100	4	110	1
2	101	5	111	1
1	110	6	101	1
3	111	7	100	1

Computer Systems in Engineering

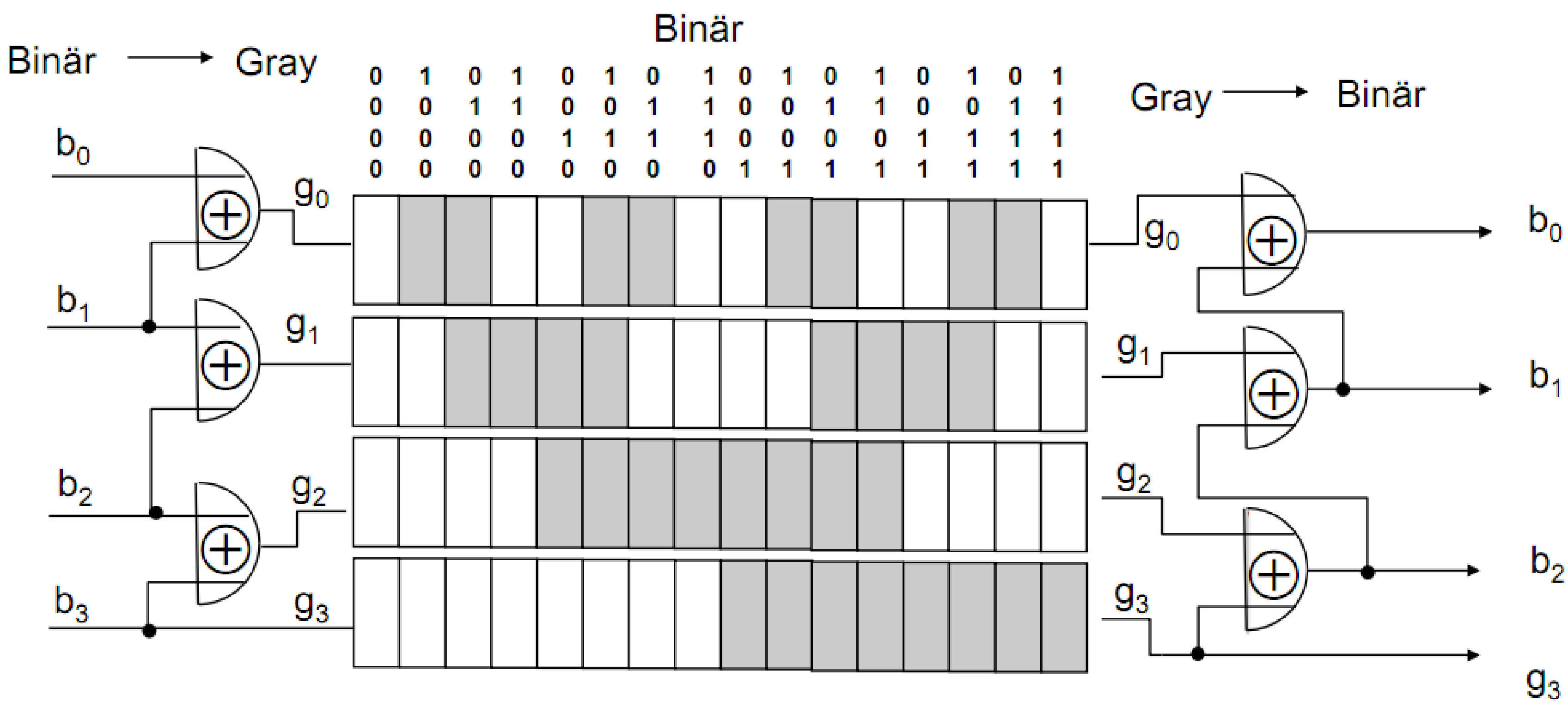
Gray-Code

- Codierung von Zahlen im Gray-Code (Beispiel: 5)

1. Als Binärzahl umsetzen (Beispiel: 101)
2. Binärzahl um ein Bit nach rechts verschieben
   (Beispiel: 101 -> 10 -> 010)
3. Binärzahl und Verschiebung über XOR verknüpfen
   (Beispiel: 101 XOR 010 = 111)

- Wir haben den Gray-Code für 5: 111
- Diese Erzeugung ist einfach in Hardware umzusetzen

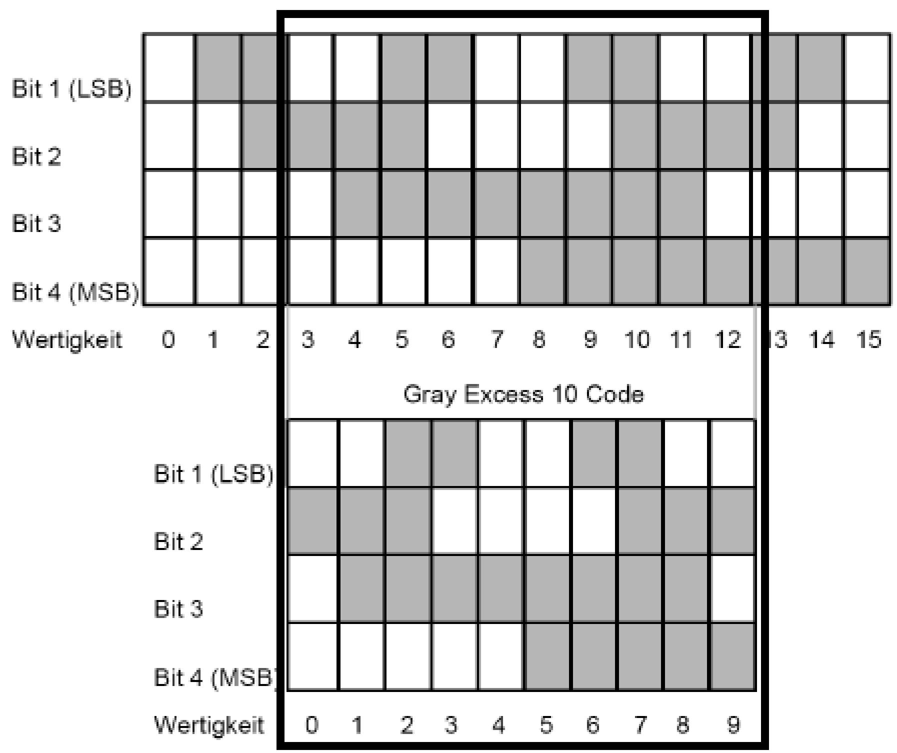
Gray-Code



Binär
Binär → Gray
Gray → Binär
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
b0
b1
b2
b3
g0
g1
g2
g3
b0
b1
b2
g3

Gray-Code

- Problem: Gray-Codes lassen sich nur für Zweierpotenzen erstellen.
- Lösung für gerade Zahlen: Symmetrisches Kappen



Computer Systems in Engineering

Fehlererkennung/Korrigieren

- Datenübertragung geht niemals Fehlerfrei
  - Streaming
  - Download aus dem Internet
  - CD/DVD/BluRay Player
- Ziel besteht in der Aufspürung und dem Korrigieren dieser Fehler

Drei Ansätze

- Jedes Bit zweimal
  - 1 → 11 und 0 → 00
  - Einzelne Fehler werden erkannt, aber nicht korrigierbar
    - 10: war es 00 oder 11?
- Jedes Bit dreimal
  - 1 → 111 und 0 → 000
  - Einzelne Fehler erkennbar und korrigierbar
  - Doppelfehle werden erkannt
- Jedes Bit viermal
  - 1 → 1100 und 0 → 0011
  - Einzelne Fehler erkennbar und korrigierbar
  - Doppelfehler nicht zwangsläufig erkennbar

Hamming

- Hamming-Gewicht
  - Die Anzahl der Stellen, die von „0“ unterschiedlich sind
    - W(110011) = 4
    - W(0001)   = 1
- Hamming-Distanz
  - Anzahl der Stellen, in denen sich zwei Wörter unterscheiden
    - D(1011, 0001) = w(1011) XOR w(0001) = w(1010) = 2