Bits und Bytes

chp

FHNW, HS 2015

Mr

Bits und Bytes

- Analoge und digitale Signale
- Codierung
- Zahlensysteme
- Hexadezimalzahlen
- Ganzzahlen-Arithmetik
- Darstellung von Rechnerdaten

Analoge und digitale Signale

Analoge Signale:

Amplitudenwerte eines Signals können beliebige Werte annehmen.

Das Signal gibt zu jedem beliebigen Zeitpunkt den exakten Wert wieder.

In der analogen Welt existieren Werte- und Zeitkontinuum.

Analoge und digitale Signale

Digitale Signale:

Die Amplitudenwerte eines Signals sind nötigenfalls diskretisiert (digitalisiert).

Die Zeit ist quantisiert. Innerhalb eines Zeitquantes ist ein Signal konstant aufgefasst.

In der digitalen Welt existieren kein Werte- und kein Zeitkontinuum.

Analoge und digitale Signale

Vor- und Nachteile digitaler Systeme:

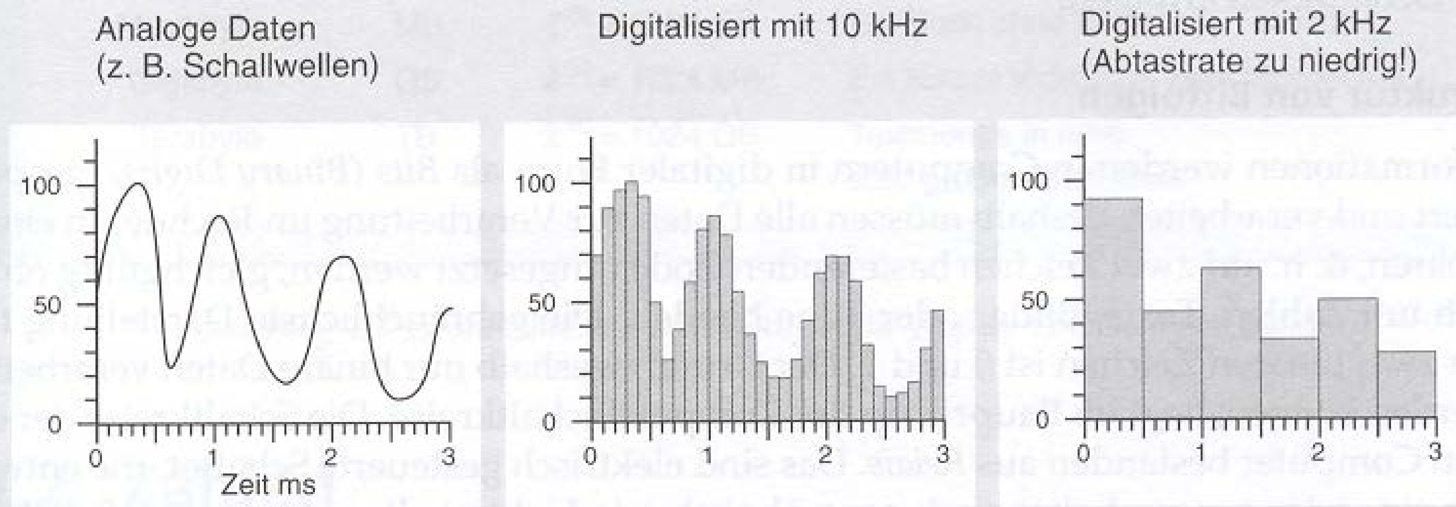
+ Werte sind endlich abzählbar.
+ Ereignisse über eine Dauer lassen sich mit endlich vielen Zeitintervallen beschreiben.

– Sind nicht absolut genau bezüglich Werten und Zeiten.

– Bedürfen mehr Zeit, da z.B. A/D- und D/A-Wandlung nötig ist.

Liste nicht abschliessend.

Digitalisierung



Bei einer Abtastrate von 10 kHz werden in jeder Sekunde 10 000 Meßwerte gewonnen, aus denen der Verlauf der Wellenform abgelesen werden kann.

Wertetabelle

Zeit	Pegel
0.1	69
0.2	90
0.3	95
0.4	99
0.5	93
0.6	50
...	...

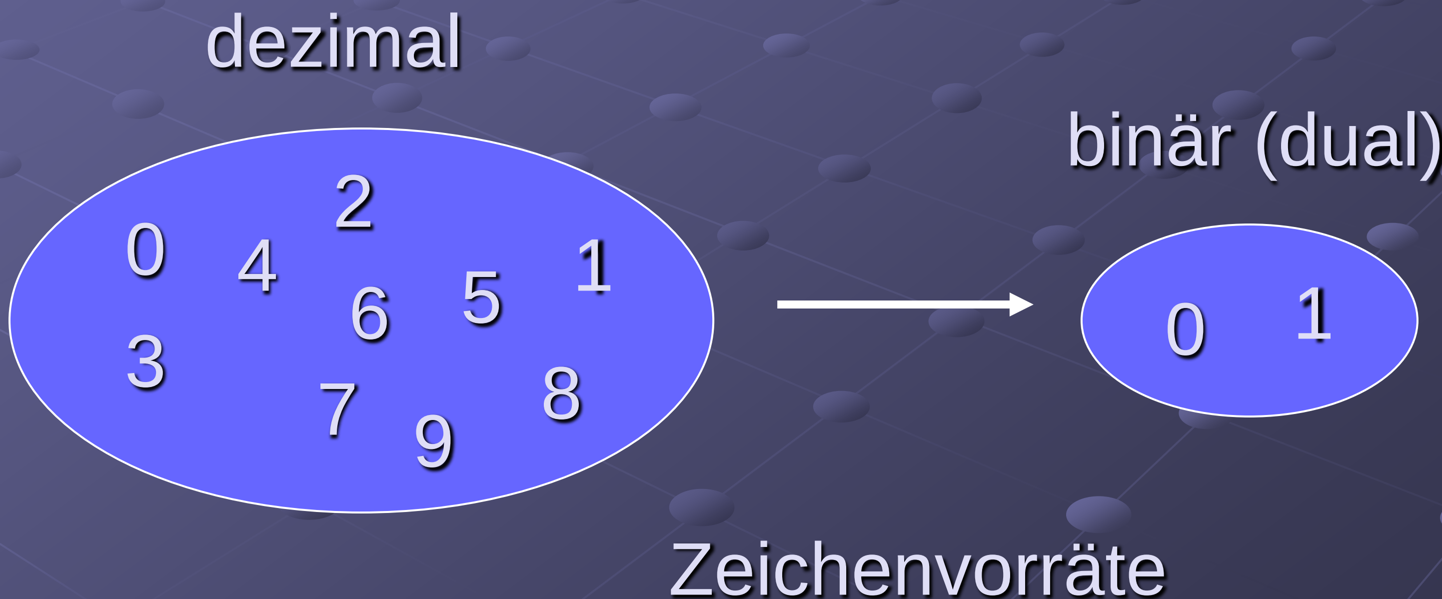
Wertetabelle

Zeit	Pegel
0.5	92
1.0	45
1.5	62
2.0	31
2.5	50
3.0	28
...	...

[2]

Codierung

Eine Codevorschrift (Codierung) bildet die Zeichen mit einem ersten Zeichenvorrat auf Zeichen mit einem zweiten Zeichenvorrat ab.



Codierung

dez	dual	hex
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	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
16	10000	10

dez	dual	hex
17	10001	11
...	...	...
30	11110	1E
31	11111	1F
32	100000	20
...	...	...
100	1100100	64
200	11001000	C8
...	...	...
254	11111110	FE
255	11111111	FF
256	100000000	100
257	100000001	101
...	...	...
1000	1111101000	3E8
1023	1111111111	3FF
...	...	...

Codierung

Codierungsvorschriften können folgende Punkte beinhalten:

- Reihenfolge der Zeichen wichtig
- Basierend auf einem mathematischen Abbildungsgesetz
- Abbildungsliste
- etc.

Codierung

Die Codierung hat unterschiedliche Ziele:

- (Vereinfachtes) Festhalten einer Situation
- Geheimhaltung
- Komprimierung von Daten
- etc.

... und Kombinationen davon.

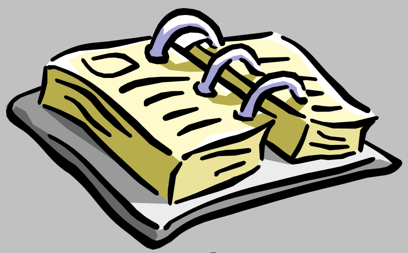
Codierung

Häufig soll auch eine Abbildung in beide Richtungen möglich sein.

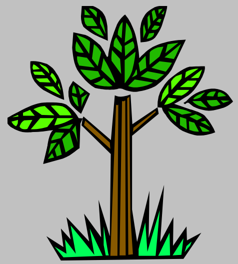
Typische Situation:

Beispiele

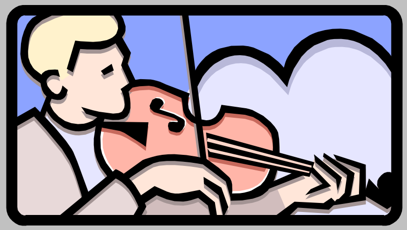
Dokumente



Bilder



Töne



Digitalisierung

Digitalisierung

010001110
101001010
100100101
001010011
101010110

010001110
101001010
100100101
001010011
101010110

010001110
101001010
100100101
001010011
101010110

Verarbeitung

Beispiele

Dez-Code	Name	Beschreibung	Dez-Code	Name	Beschreibung
0	NUL	Null-Zeichen	16	DEL	Löschen
1	SOH	Kopfzeilenanfang	17	DC1	Gerätesteuerzeichen 1 (Xon)
2	STX	Textanfang	18	DC2	Gerätesteuerzeichen 2
3	ETX	Textende	19	DC3	Gerätesteuerzeichen 3 (Xoff)
4	EOT	Übertragungsende	20	DC4	Gerätesteuerzeichen 3
5	ENQ	Aufford. Zur Datenübertrag.	21	NAK	Negative Rückmeldung
6	ACK	Positive Rückmeldung	22	SYN	Synchronisierung
7	BEL	Klingelzeichen	23	ETB	Datenübertragungsblock Ende
8	BS	Rückwärtsschritt	24	CAN	Ungültig-Kennung
9	HT	Horizontaler Tabulator	25	EM	Aufzeichnungsende
10	LF	Zeilenvorschub	26	SUB	Substitution
11	VT	Vertikaler Tabulator	27	ESC	Umschaltung
12	FF	Seitenvorschub	28	FS	Dateitrennzeichen
13	CR	Wagenrücklauf	29	GS	Gruppentrennzeichen
14	SO	Dauerumschaltung	30	RS	Satztrennzeichen
15	SI	Rückschaltung	31	US	Einheitentrennzeichen

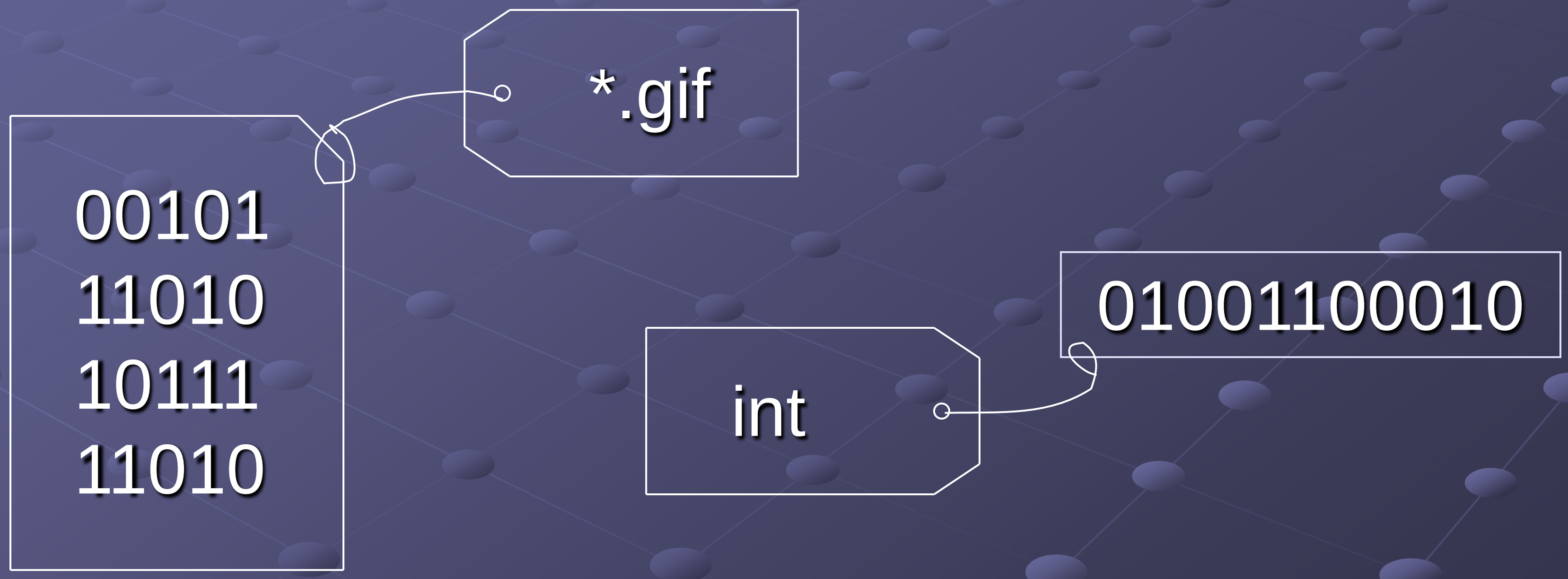
Typen

Zur Kennzeichnung von Bitfolgen wurden Typen eingeführt:

- Datentypen
- Erweiterungen bei Dateinamen (Extension)
- Protokolltypen
- etc.

Typen

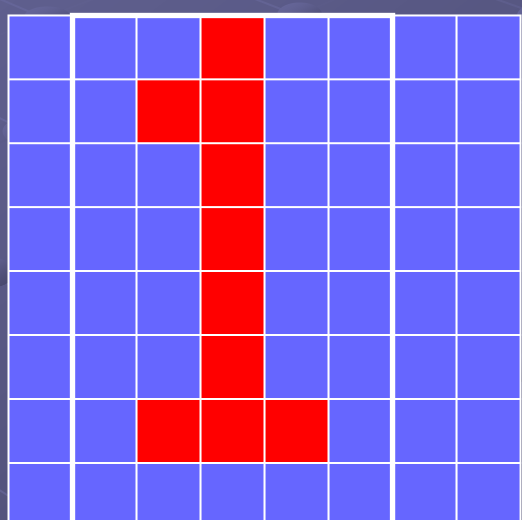
Typen lassen sich als 'Etikett' auffassen, die das 'Produkt' genauer charakterisieren.



Aufgabe

Die Darstellung einer Ziffer werde wie folgt codiert:

Zifferdarstellung



Code

11111111 11111111

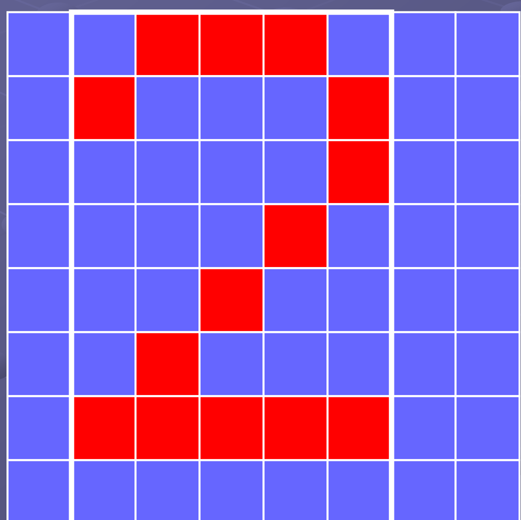
10111101 10000000

10111111 11111111

11111111 11111111

Aufgabe

Wie lautet der Code zur Darstellung der Zahl 2 ?

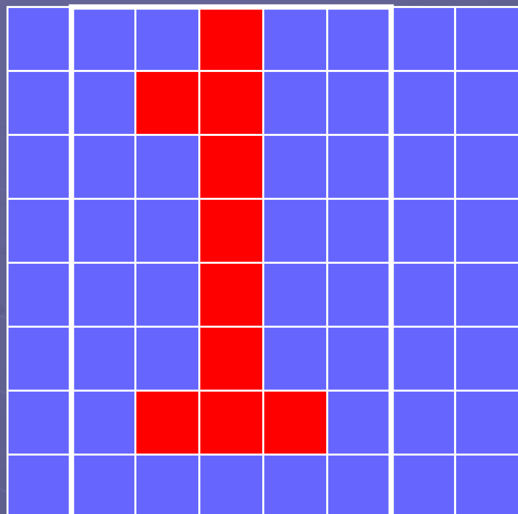


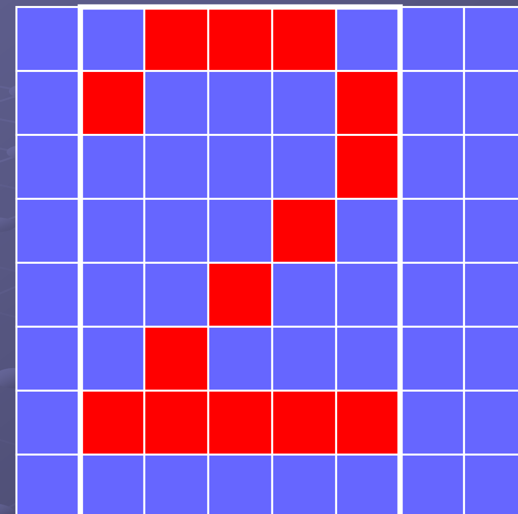
Ausgangslage?

Vereinbarungen?

Code?

Aufgabe





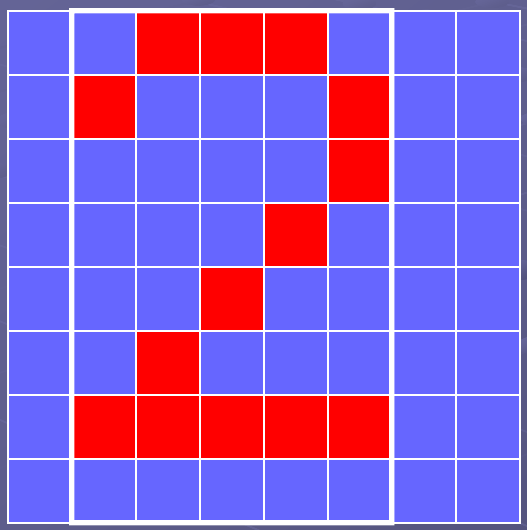
11111111 11111111

10111101 10000000

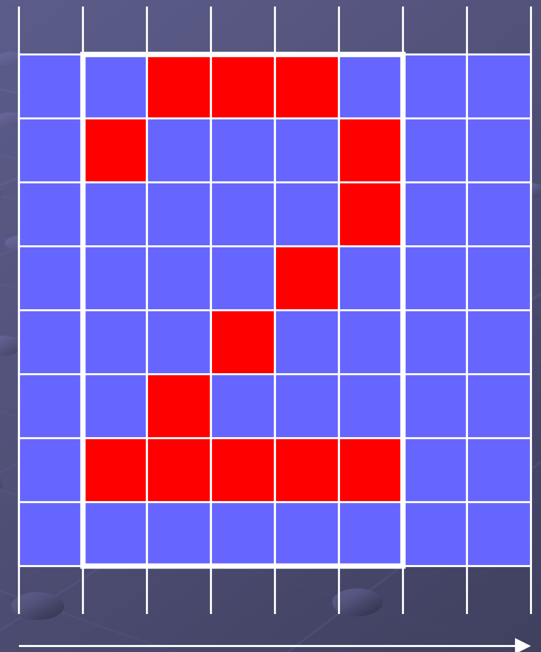
10111111 11111111

11111111 11111111

Lösung



Vereinbarungen:



Im 8 x 8 Feld entspricht jede Bitgruppe einer Spalte

11111111	10111101
10011110	10101110
10110110	10111001
11111111	11111111

codiert mit Bitwert 1

codiert mit Bitwert 0

Zahlensysteme

Eine Codierung, um die Menge abzählbarer 'Dinge' festzuhalten.

Wie Julius und Brutus notierten:

VII

XCV

MLIV

Die schwere Geburt der Null

Die Null wurde lange nach den andern Zahlen erfunden. Die Römer kannten sie überhaupt nicht, die Babylonier konnten nicht mit ihr umgehen, erst die Inder erkannten das Potential dieser bizarren Zahl, die alleine nichts ist, aber andern zur Grösse verhelfen kann.

...

Herbert Cerutti

[3]

Gewichteter Code

- Symbole werden aneinander gereiht.
- Jedes Symbol repräsentiert eine Stelle.
- Jede Stelle hat ein Gewicht.
- Die Stelle ganz rechts hat niedrigstes Gewicht, jene ganz links höchstes Gewicht.
- Eine Stelle ohne eigenen Beitrag wird mit Null (als Stellenhalter) dargestellt.
- Die Anzahl Stellen einer Zahl (Symbolfolge) wird als Wortlänge bezeichnet.

Gewichteter Code

Beispielhaft an einer Dezimalzahl:

Zahl (z):	1205
Zahlenbasis (b):	10
Ziffernvorrat (c_i):	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Wortlänge (k):	4

$$1205 = 1 * 10^3 + 2 * 10^2 + 0 * 10^1 + 5 * 10^0$$

$$z = c_{ik-1} * b^{k-1} + c_{ik-2} * b^{k-2} + c_{ik-3} * b^{k-3} + c_{ik-4} * b^{k-4}$$

Gewichteter Code

$1205 = 1 * 10^3 \quad + 2 * 10^2 \quad + 0 * 10^1 \quad + 5 * 10^0$

$z \quad = c_{ik-1} * b^{k-1} + c_{ik-2} * b^{k-2} + c_{ik-3} * b^{k-3} + c_{ik-4} * b^{k-4}$

Allgemein formuliert:

$$z = \sum_{j=0}^{j=k-1} c_{i_j} * b^j$$

Hexadezimalzahlen

Jedes gewichtete Zahlensystem hat so viele Ziffern nötig, wie die Basis gross ist.

Das Hexadezimale Zahlensystem (b = 16) benützt nebenstehende Ziffern:

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

Binär ↔ Hex - Wandlung

- Eine Binärzahl wird vorerst in Nibbles (Gruppen à 4 Bits) eingeteilt.
- Danach wird jedem Nibble die Ziffer mit dem entsprechenden Wert in Hex zugeordnet.

Beispiel:

$\underbrace{1001}_{9}'\underbrace{0011}_{3}'\underbrace{1101}_{D}'\underbrace{1111}_{F}$

Weitere Wandlungen

Wandlungen sind grundsätzlich besonders einfach, wenn zwischen Zahlensysteme mit Basen der Form 2^n zu wandeln ist:

Binär: $n = 1$

Oktal: $n = 3$

Hex: $n = 4$

Aufgabe

Wandeln Sie die Hex-Zahl 3B7 in eine Oktal-Zahl.

Hinweis:

Hex: 2^4

Oct: 2^3

Lösung

3 B 7	Hex
0011'1011'0111	Bin
001'110'110'111	Bin
1 6 6 7	Oktal

Binär → Dezimal - Wandlung

Jede Stelle wird mit ihrem Gewicht berück-sichtigt, wenn sie eine 1 aufweist.

Beispiel:

$1011'0111_2 =$

$2^7{}_{10} + 2^5{}_{10} + 2^4{}_{10} + 2^2{}_{10} + 2^1{}_{10} + 2^0{}_{10} =$

$128_{10} + 32_{10} + 16_{10} + 4_{10} + 2_{10} + 1_{10} =$

$\underline{\underline{183_{10}}}$

Dezimal $\rightarrow$ Binär - Wandlung

Variante 1:
Beim höchstem Gewicht beginnend werden die Beiträge beteiligter Gewichte ermittelt.

Beispiel:

$55_{10} = 32_{10} + (23_{10}) = 32_{10} + 16_{10} + (7_{10}) =$

$32_{10} + 16_{10} + 4_{10} + (3_{10}) =$

$32_{10} + 16_{10} + 4_{10} + 2_{10} + 1_{10} =$

$\underline{\underline{110111_2}}$

Dezimal → Binär - Wandlung

Variante 2:
Fortlaufende Ganzzahl-Division mit 2 und 'Sammeln' der Reste.

55 : 2 = 27	Rest: 1	LSB
27 : 2 = 13	Rest: 1	
13 : 2 = 6	Rest: 1	
6 : 2 = 3	Rest: 0	
3 : 2 = 1	Rest: 1	
1 : 2 = 0 (Abbruch)	Rest: 1	MSB

$$\underline{\underline{110111_2}}$$

Dezimal → Hex - Wandlung

Anwendung von Variante 2.

Beispiel:

$181_{10} : 16_{10} = 11_{10}$ Rest: 5_{10} LSB

$11_{10} : 16_{10} = 0$ (Abbruch) Rest: $11_{10} = B_{16}$ MSB

$\underline{\underline{B5_{16}}}$

Ganzzahlen-Arithmetik

1. Summand:	0	1	0	1
2. Summand:	+ 0	+ 0	+ 1	+ 1
Summe:	= 0	= 1	= 1	= 1 0

	binär	dezimal
1. Summand:	1 0 0 1 1 1	3 9
2. Summand:	+ 1 1 1 0	+ 1 4
Übertrag:	*1 1 1*	*1*
Summe:	= 1 1 0 1 0 1	= 5 3

Das Einerkomplement

Das Einerkomplement (EK) einer Binärzahl wird gebildet, indem stellenweise eine 1 durch eine 0, bzw. eine 0 durch eine 1 ersetzt wird.

Beispiel:

EK(1001'1011) = 0110'0100

Das Zweierkomplement

Das Zweierkomplement (ZK) einer Binärzahl wird gebildet, indem vorerst das Einerkomplement gebildet und danach 1 addiert wird.

Beispiel:

ZK(1001'1011) = EK(1001'1011) + 1 =

0110'0100 + 1 = $\underline{\underline{\text{0110'0101}}}$

Zweierkomplement

- Das Zweierkomplement wird in Rechnern verwendet, um negative Ganzzahlen dar-zustellen.
- Bei der Komplement-Darstellung muss be-kannt sein, wie gross die Stellenzahl ist, sonst treten Fehlinterpretationen auf.

Beispiel

Stellen Sie die Dezimalzahl 49 als Binärzahl mit der Stellenzahl k = 8 dar.

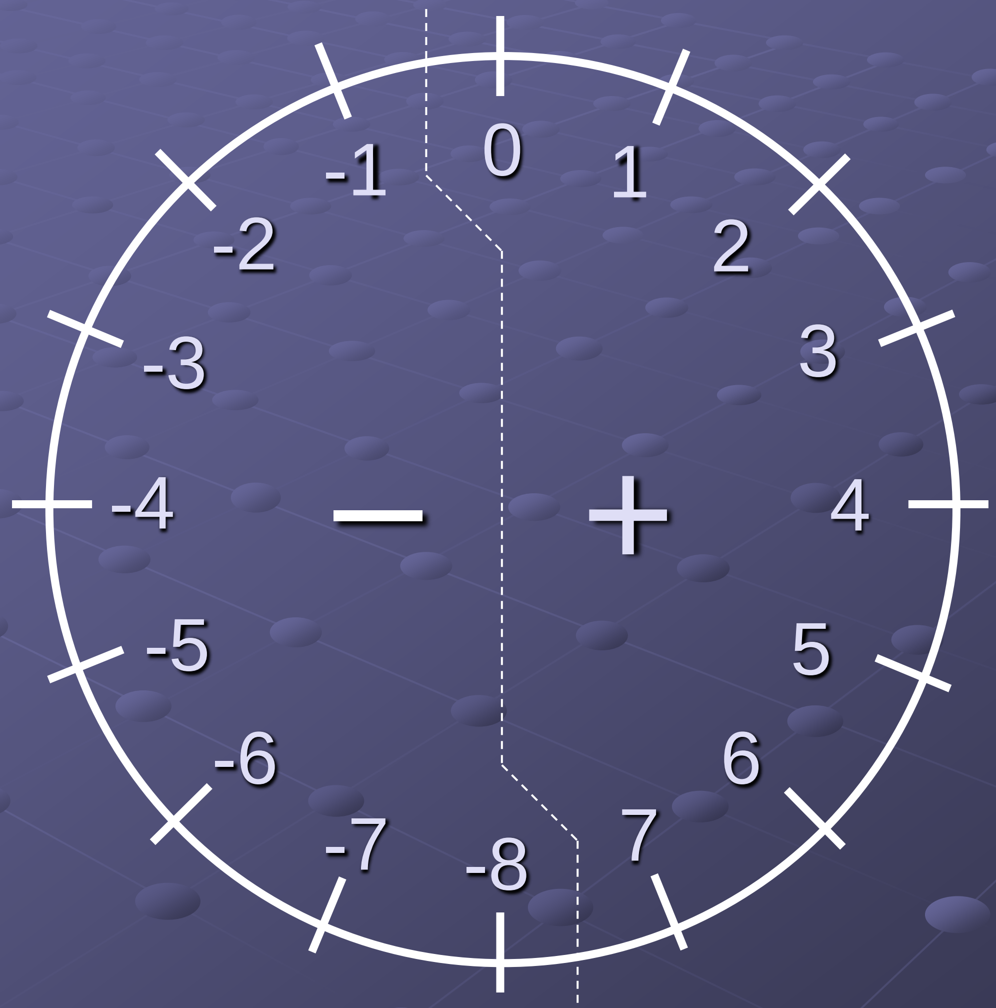
Dez: 49 = 32 + 16 + 1, also
Bin: 0011'0001

Ohne Angabe k = 8 ist die Anzahl führender 0 unklar!

Bilden Sie das ZK obiger Binärzahl.

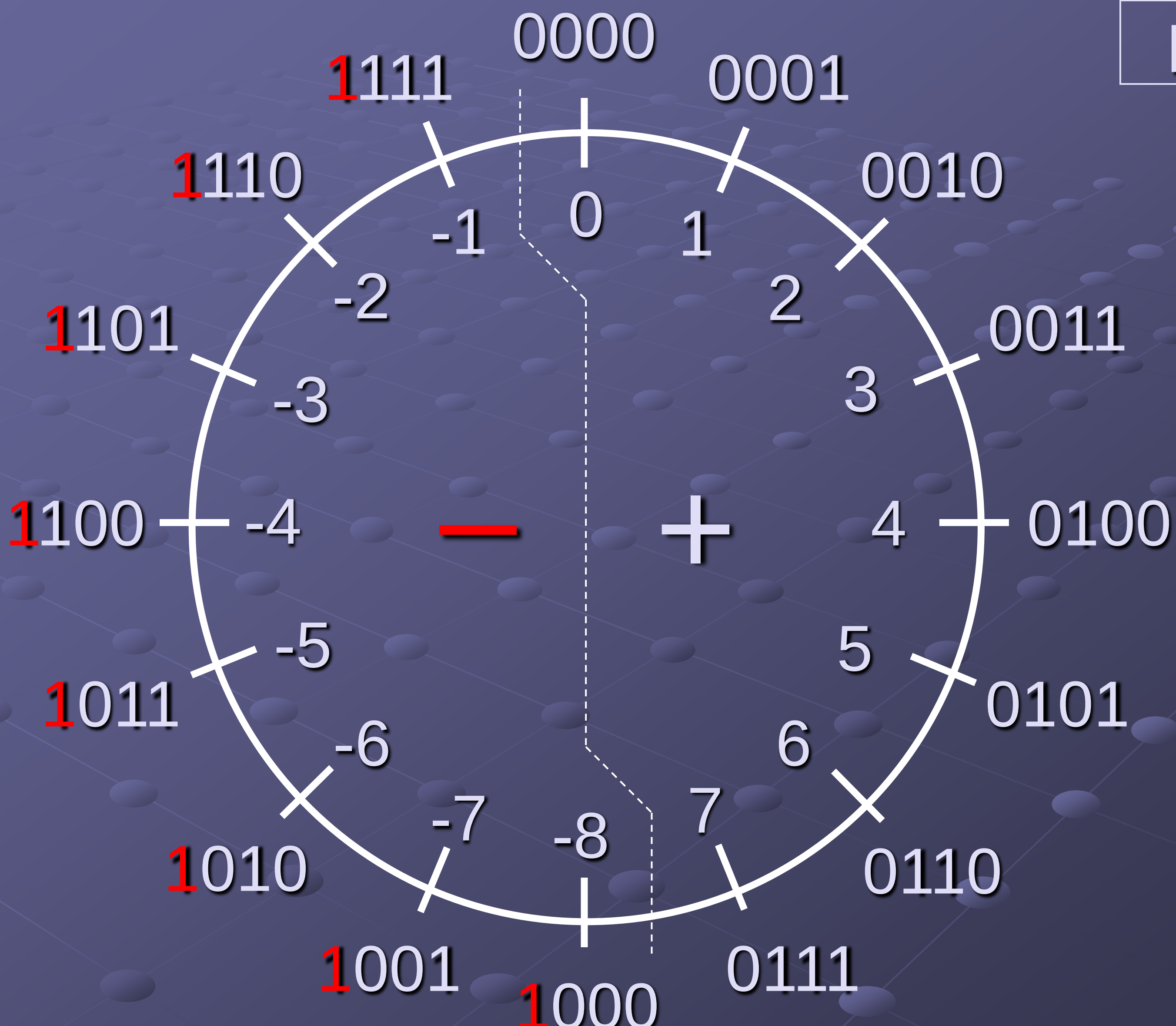
Bin: 1100'1111

Der Zahlenkreis



Der Zahlenkreis

k = 4



Addition/Subtraktion

Fallbetrachtungen für k = 4:

a) $\underline{0}100 + \underline{0}011 = \underline{0}111$ pos + pos = pos √

$4 + 3 = 7$

b) $\underline{0}100 + \underline{0}101 = \underline{1}001$ pos + pos = neg f

$4 + 5 \neq -7$

c) $\underline{0}100 + \underline{1}101 = \cancel{1}\underline{0}001$ pos + neg = pos √

$4 + -3 = 1$

d) $\underline{0}100 + \underline{1}010 = \underline{1}110$ pos + neg = neg √

$4 + -6 = -2$

Addition/Subtraktion

Fortsetzung Fallbetrachtungen für k = 4:

e) $\underline{1}100 + \underline{1}101 = \cancel{1}\underline{1}001$ neg + neg = neg √

$-4 + -3 = -7$

f) $\underline{1}100 + \underline{1}011 = \cancel{1}\underline{0}111$ neg + neg = pos f

$-4 + -5 \neq 7$

g) $\underline{1}100 + \underline{0}011 = \underline{1}111$ neg + pos = neg √

$-4 + 3 = -1$

h) $\underline{1}100 + \underline{0}110 = \cancel{1}\underline{0}010$ neg + pos = pos √

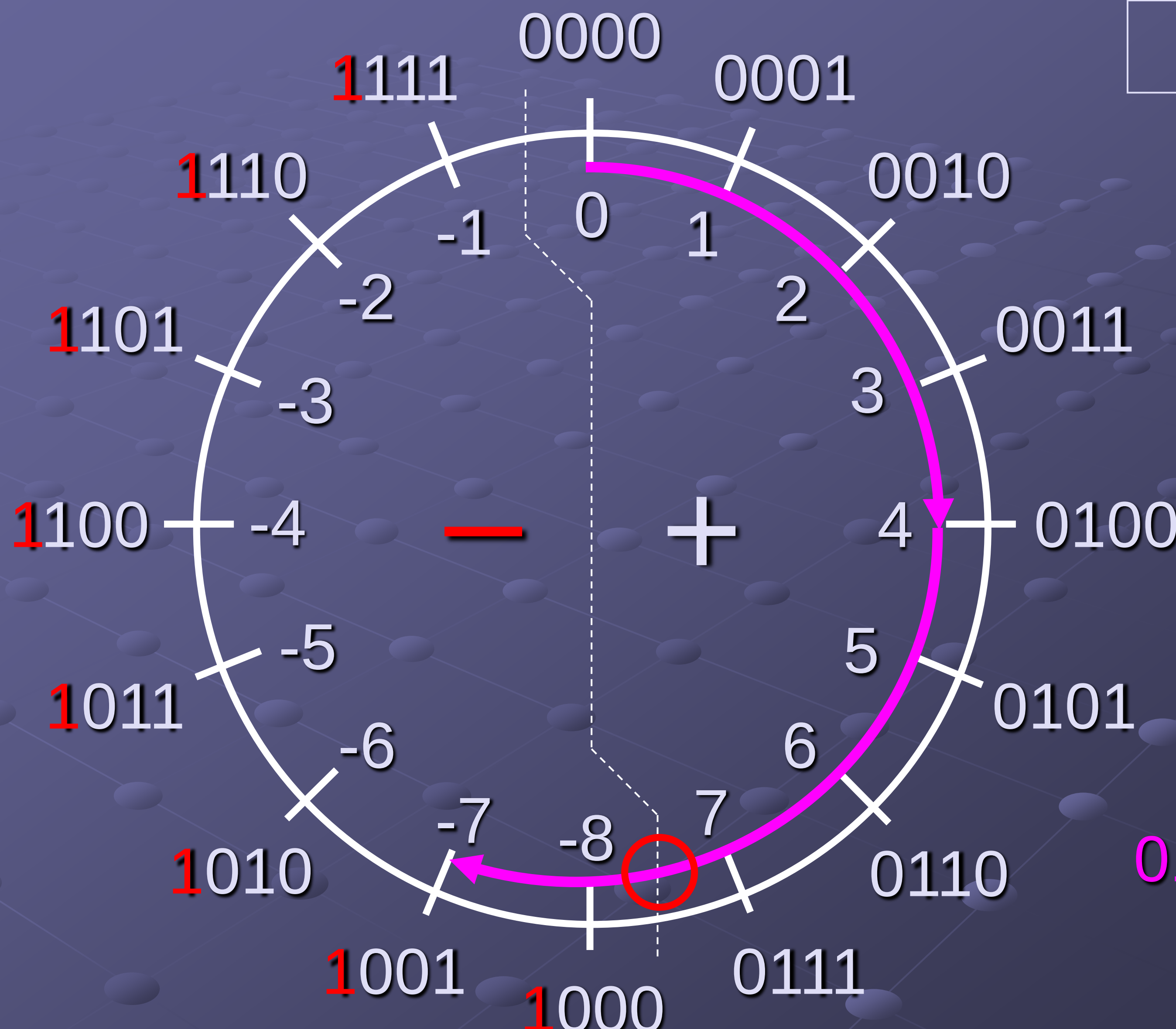
$-4 + 6 = 2$

Addition/Subtraktion

- Der Zahlenkreis eignet sich gut, um die Fallunterscheidungen zu illustrieren.
- Die Resultate bei der Addition/Subtraktion sind falsch, wenn es zu einer Bereichsüber-schreitung kommt.
- Eine Bereichsüberschreitung findet statt, wenn vom positiven in den negativen Halb-kreis via 7/-8 gewechselt wird - oder um-gekehrt (für k = 4).

Bereichsüberschreitung bei der Addition

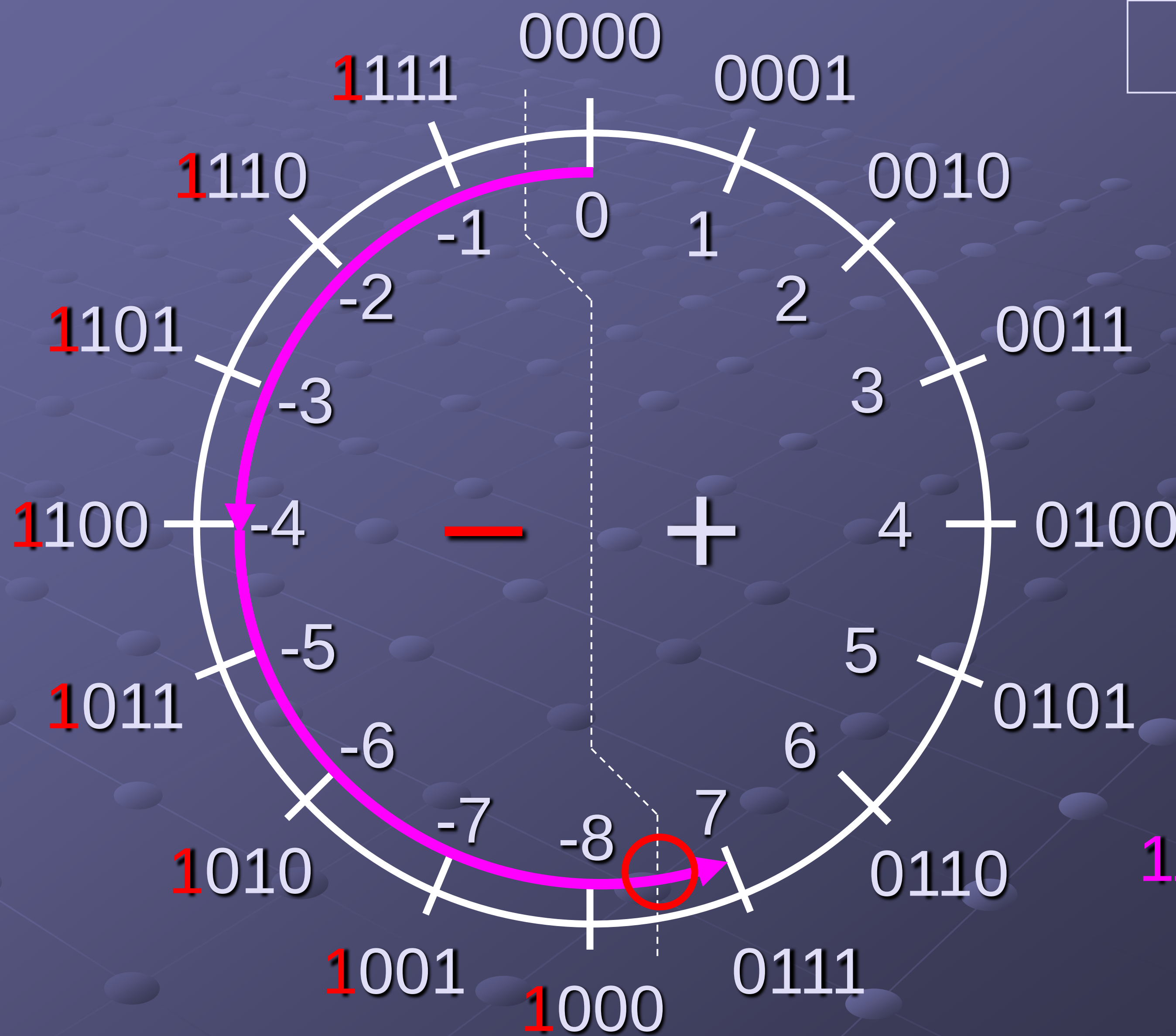
k = 4



0100 + 0101

Bereichsüberschreitung bei der 'Subtraktion'

k = 4



1100 + 1011

Resultatauswertung

$a + b = r \qquad k > 1$

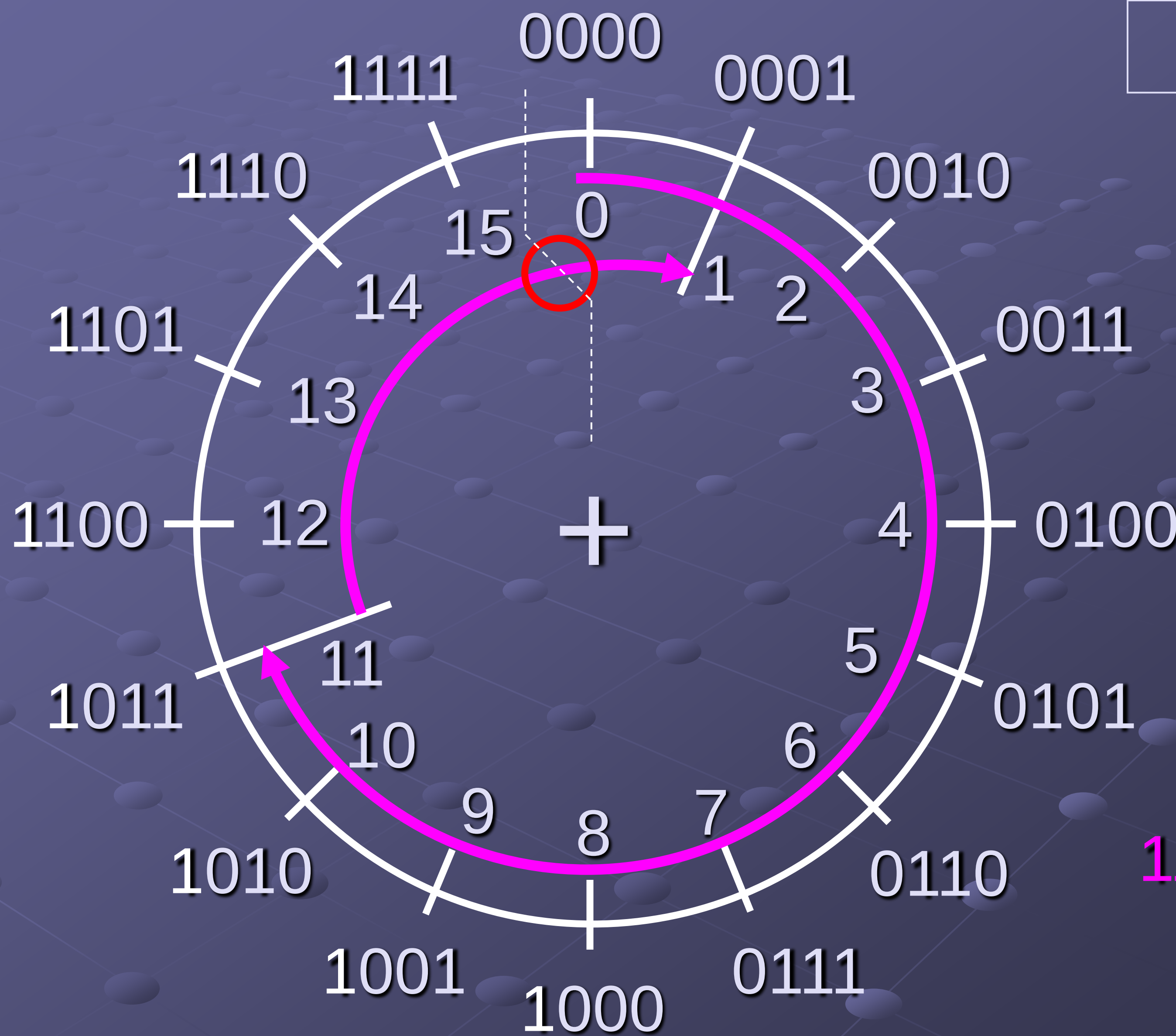
Fall	MSB(a)	MSB(b)	MSB(r)	n/ok
a	0	0	0	√
b	0	0	1	f
c	0	1	0	√
d	0	1	1	√
e	1	1	1	√
f	1	1	0	f
g	1	0	1	√
h	1	0	0	√

Aufgabe

- Zeichnen Sie einen Zahlenkreis für k = 4 für nur positiv zu interpretierende Zahlen (unsigned int).
- Zeigen Sie im Kreis, wo es zu Bereichs-überschreitungen kommen kann.
- Begründen Sie, dass es sich um Bereichs-überschreitungen handelt.

Teillösung

k = 4



1101 + 0110

Darstellung Rechnerdaten

- Einfachere Rechner arbeiten ausschliesslich mit Ganzzahlen-Arithmetik: Mit *unsigned* und *signed int*, allenfalls mit Binary Coded Decimal (BCD).
- Leistungsfähigere Rechner benutzen Numerische Koprozessoren (Number Cruncher) oder haben diese gar integriert (Pentium).

Coprozessor Intel 8087

Folgende Datentypen sind definiert:

Coprozessor Intel 8087

Bereiche

Typ	Genauigkeit	Bereich (grob)
Word Integer	16 Bit	$\pm 10^{4}$
Short Integer	32 Bit	$\pm 10^{9}$
Long Integer	64 Bit	$\pm 10^{18}$
Short Real	24 Bit	$\pm 10^{\pm 38}$
Long Real	53 Bit	$\pm 10^{\pm 308}$
Temporary Real	64 Bit	$\pm 10^{\pm 4932}$
Packed BCD	18 Digit	$\pm 10^{18}$

Literatur

[1] Mikroelektronik, Rolf Enderlein, Spektrum, Akademischer Verlag GmbH, Heidelberg, 1993
ISBN 3-86025-037-X

[2] EDV-Grundwissen, M. Precht, N. Meier, J. Kleinlein, Addison-Wesley-Longman, 1998
ISBN 3-8273-1422-2

[3] NZZ-Folio, Neue Zürcher Zeitung, Zürich, Februar 2002
ISSN 1420-5262

Literatur

[4] Digitaltechnik, Klaus Fricke, Verlag vieweg, Braunschweig/Wiesbaden, 1999
ISBN 3-528-03861-6

[5] Mikroprozessortechnik, Müller/Walz, Vogel Verlag, Würzburg, 1992
ISBN 3-8023-1453-0

[6] Rechnerarchitektur, Helmut Malz, vieweg Verlag, Braunschweig/Wiesbaden, 2001
ISBN 3-528-03379-7