Vorlesung
Entwurf digitaler Schaltungen

Kapitel 2: Zahlensysteme und Kodes

Bernhard Rinner

30.07.2024 17:30

UNIVERSITÄT KLAGENFURT

Institut für Vernetzte und Eingebettete Systeme

Inhalt

1. Zahlendarstellung [DH: S 59-79]
2. Kodes und Kodierung [DH: S 80-85]
3. Binäre Arithmetik

UNIVERSITÄT KLAGENFURT

Zahlendarstellung

Notation

Typ	Schriftart	Beispiele
Variablen (Skalare)	kursiv	a, b, x, y
Funktionen	aufrecht	$f, g(x), \max(x)$
Vektoren	fett, Elemente zeilenweise	$\boldsymbol{a}, \boldsymbol{b} = \begin{pmatrix} x \\ y \end{pmatrix} = (x, y)^T$, $\mathbf{B}=(x, y, z)^T$
Matrizen	Schreibmaschine	$A, B = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$
Mengen	kalligrafisch	$\mathcal{A}, \mathcal{B} = \{a, b\}, b \in \mathcal{B}$
Zahlenbereiche, Koordinatenräume	doppelt gestrichen	$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{R}^2$

Kerben- und Strichsysteme

- Zahl x wird durch x-fache Wiederholung eines Zeichens dargestellt
- Strichsysteme sind ältestes bekannte Zahlensystem
- Praktisch verwendbar nur für kleine Zahlen
  (vgl. Kaffeeliste)
- Addition ergibt sich natürlich (reines Additionssystem)

Römische Zahlen

- Weiterentwicklung des Additionssystems
  - Berücksichtigung der Subtraktionsregel (relative Position der Symbole)

Zahlensymbol	Wertigkeit
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

IV = 5 − 1
V = 5
VI = 5 + 1

LXII = 50 + 10 + 1 + 1 = 62
XLII = 50 − 10 + 1 + 1 = 42

  - Große Zahlen lassen sich faktisch kaum darstellen

Dezimalsystem

- 10 Zahlensymbole aus der Menge $\{0,1,2,3,4,5,6,7,8,9\}$
- Absolute Position der Zeichen wichtig um Zahlenwert zu bestimmen (Stellenwertsystem)
  - Bsp. $123 \neq 321$
- Zahlenwert $w = z_{n-1} \cdot 10 \ldots 0 + z_2 \cdot 10^2 + z_1 \cdot 10^1 + z_0 \cdot 1$

$$= \sum_{i=0}^{n-1} z_i \cdot 10^i$$

- Stellenwertsystem zur Basis 10

b-adisches Stellenwertsystem

- Sei $b > 1$ eine natürliche Zahl, dann heißt die Menge $\{0,1,2,3,4, \dots, b-1\}$ das Alphabet des b-adischen Zahlensystems mit der Basis b

- Dezimalsystem

$$b = 10 \rightarrow \{0,1,2,3,4,5,6,7,8,9\}$$

- Dualsystem

$$b = 2 \rightarrow \{0,1\}$$

- Oktalsystem

$$b = 8 \rightarrow \{0,1,2,3,4,5,6,7\}$$

- Hexadezimalsystem

$$b = 16 \rightarrow \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$$

Die Buchstaben A bis F repräsentieren Ziffernwerte 10 bis 15

Dualsystem (Binärsystem)

- Binärsystem ($b = 2$) mit einem Alphabet von 2 Symbolen
  - Essentiell in der Informationstechnik
  - Symbole lassen sich leicht darstellen (Schalter geschlossen/geöffnet oder Strom fließt/fließt nicht)
  - Jede Stelle einer Binärzahl wird als Bit („binary digit“) bezeichnet

- 1 Bit ist die kleinste Informationsmenge, die gespeichert werden kann
  - 8 Bits werden als 1 Byte bezeichnet
- Die Bitfolge $(01001101)_2$ entspricht der Dezimalzahl $(77)_{10}$

$$(01001101)_2 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + \\ +0 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 + 0 \cdot 2^7 \\ = 1 + 4 + 8 + 64 = 77$$

Negative Zahlen

- Explizite Darstellung des Vorzeichens V (an höchstwertiger Stelle) und des Betrags B

  $w = (-1)^V \cdot B$

  - Beispiel: Binärsystem

    $\underbrace{1}_{V}\underbrace{1111011}_{B} = -(123)_{10}$ ($V = 1$ repräsentiert „−“)

    $\underbrace{0}_{V}\underbrace{1111011}_{B} = +(123)_{10}$ ($V = 0$ repräsentiert „+“)

  - Symmetrischer Zahlenbereich $[-z, \ldots, +z]$, jedoch nicht eindeutig: $+0 = 0000 \ldots 00, -0 = 1000 \ldots 00$

- Komplementdarstellung als Alternative
  - Einerkomplement: Invertieren jeder einzelnen Bits
  - Zweierkomplement: Addieren des Einerkomplements mit 1
- Offsetdarstellung (Charakteristik)
  - Zahlenbereich durch Addition verschoben (0 kleinste negative Zahl)

Komplemente und Charakteristik

- Einerkomplement entspricht Ergänzung auf $2^n - 1$
  - Differenz zu B an jeder Stelle
  - Beispiel: Dezimalzahl $(123)_{10}$

999	$10^3 - 1$
$\underline{-123}$	*Zahl*
876	*Einerkomplement*

Binärzahl $(01111011)_2$

11111111	$2^8 - 1$
$\underline{-01111011}$	*Zahl*
10000100	*Einerkomplement*

- Zweierkomplement entspricht Ergänzung auf 2^n und damit

$$Einerkomplement + 1$$

  - Eindeutiger, asymmetrischer Zahlenbereich: $[-B^{n-1}, \dots, +B^{n-1} - 1]$
- Charakteristik verschiebt den (ganzzahligen) Zahlenbereich durch Addition mit fixem Offset (meist ${}^{B^n}/_2$ oder ${}^{B^n}/_2 - 1$)
  - Beispiel: Dezimalzahl $(123)_{10}$

499	*Offset* ${}^{10^3}/_2 - 1$
$\underline{-123}$	*Zahl*
376	*Charakteristik (immer positiv)*

Binärzahl $(01111011)_2$

01111111	*Offset* $2^7 - 1$
$\underline{-01111011}$	*Zahl*
00000100	*Charakteristik*

Rationale Zahlen

- Zwei Darstellungen: Festkomma- und Gleitkommaformat
- Festkommaformat besteht aus Vorzeichen V und Mantisse M
  - Beispiel für Bitbreite von 16 Bit

  - Für Bitbreite n kann man die rationale Zahl wie folgt berechnen

$$w = (-1)^V \cdot 0, M = (-1)^V \cdot (\sum_{i=1}^{n-1} M_{n-1-i} \cdot \frac{1}{2^i})$$

und kann Zahlen aus dem offenen Intervall $]-1; +1[$ darstellen.

- Abstand zwischen allen benachbarten Zahlen beträgt $2^{-(n-1)}$ (äquidistantes Zahlenformat)

Gleitkommadarstellung

- Gleitkommadarstellung einer Zahl ($w \in \mathbb{Q}$) im b-adischen Zahlensystem aus Vorzeichen V, Mantisse M und Exponent E

  $w = (-1)^V \cdot 0,M \cdot b^E$

  $(-314{,}15)_{10} = (-1)^1 \cdot 0{,}31415 \cdot 10^3$

- Beispiel im Binärsystem mit Mantisse $M = (M_{m-1} M_{m-2} \dots M_0)$

  $$w = (-1)^V \cdot 0,M \cdot 2^E = (-1)^V \cdot \left(\sum_{i=1}^{m} M_{m-i} \cdot \frac{1}{2^i} \right) \cdot 2^E$$

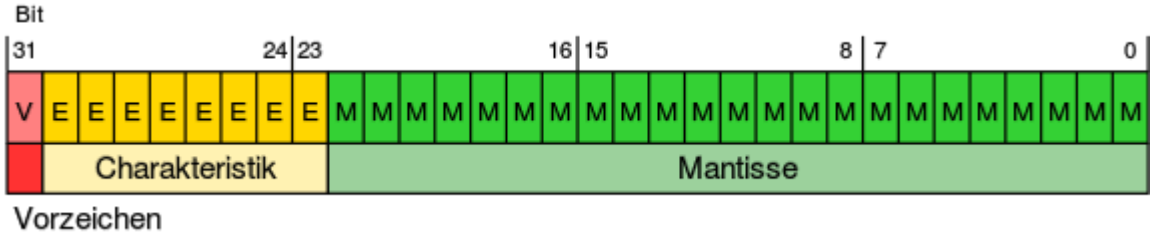
  $$= (-1)^V \cdot \sum_{i=1}^{m} M_{m-i} \cdot 2^{E-i}$$

  $(-0{,}001101)_2 = (-1)^1 \cdot (0{,}1101) \cdot 2^{-2}$

- Gleitkommaformate decken größeren Wertebereich ab, haben jedoch einen ungleichen Abstand zwischen benachbarten Zahlen

Gleitkommaformat nach IEEE 754

- Normierte Darstellung (am Beispiel einfache Genauigkeit) mit insgesamt 32 Bit



  - Vorzeichen: $(-1)^V$
  - Charakteristik: Wert des Exponenten mit Konstante addiert (hier 127)
  - Mantisse: Normalisierung der Zahl in den Bereich $1 \leq m < 2$
    Daher ist immer genau eine 1 vor dem Komma, es werden nur die Nachkommabits als Mantisse gespeichert und spart damit 1 bit

- Beispiel

  $(-0{,}203125)_{10} = (-0{,}001101)_2 = (-1)^1 \cdot (1{,}101) \cdot 2^{-3}$

  V: 1; E:$(-3 + 127)_{10} = (01111100)_2$; M:10100000000000000000000

Darstellbare Zahlenbereich

- Wertebereich der Zahlenformate mit Bitbreite n

Format	Kleinste Zahl	Größte Zahl	Abstand
Natürliche Zahlen (unsigned)	0	$+2^n - 1$	1
Ganze Zahlen (Zweierkompl.)	-2^{n-1}	$+2^{n-1} - 1$	1
Ganze Zahlen (Vorzeichen & Betrag)	$-2^{n-1} + 1$	$+2^{n-1} - 1$	1
Festkomma	$-1 + 2^{-(n-1)}$	$+1 - 2^{-(n-1)}$	$2^{-(n-1)}$
Gleitkomma IEEE 754 (single precision 32 Bit)	ca. $-2 \cdot 10^{38}$	ca. $+2 \cdot 10^{38}$	$> 2 \cdot 10^{-38}$

Binär Präfixe

- Unterschiedliche Interpretation der Präfixe im dezimalen und binären Zahlensystem (vgl. 1 Kilobyte ≠ 1000 Bytes)

Name	Präfix	Wert
kibi	Ki	$2^{10} = 1024^1 = 1.024$
mebi	Mi	$2^{20} = 1024^2 = 1.048.576$
gibi	Gi	$2^{30} = 1024^3 = 1.073.741.824$
tebi	Ti	$2^{40} = 1024^4 = 1.099.511.627.776$
pebi	Pi	$2^{50} = 1024^5 = 1.125.899.906.842.624$
exbi	Ei	$2^{60} = 1024^6$ $= 1.152.921.504.606.846.976$
zebi	Zi	$2^{70} = 1024^7$ $= 1.180.591.620.717.411.303.424$

Umwandlung zw. Zahlensystemen

Entwicklung eines Algorithmus zum Umrechnen

1. Division durch Basis b

$$w = \sum_{i=0}^{n-1} z_i \cdot b^i \quad \Leftrightarrow \frac{w}{b} = \frac{1}{b}\sum_{i=0}^{n-1} z_i \cdot b^i = \sum_{i=0}^{n-2} z_{i+1} \cdot b^i + \frac{z_0}{b}$$

   - Der erste Summand ist das ganzzahlige Ergebnis der Division durch b
   - z_0 der Rest der Division (letzte Ziffer der Zahl w)

2. Berechnung aller Ziffern mittels Rekursion

Umwandlung zw. Zahlensystemen

Beispiel: Umrechnung von $(6485)_{10}$ ins Binärsystem

6485	/ 2 = 3242	Rest 1
3242	/ 2 = 1621	Rest 0
1621	/ 2 = 810	Rest 1
810	/ 2 = 405	Rest 0
405	/ 2 = 202	Rest 1
202	/ 2 = 101	Rest 0
101	/ 2 = 50	Rest 1
50	/ 2 = 25	Rest 0
25	/ 2 = 12	Rest 1
12	/ 2 = 6	Rest 0
6	/ 2 = 3	Rest 0
3	/ 2 = 1	Rest 1
1	/ 2 = 0	Rest 1

- Algorithmus endet, wenn Division 0 ergibt
- Ergebnis von unten nach oben (1100101010101)

Umwandlung von rationalen Zahlen

Algorithmus zum Umrechnen des Nachkommaanteiles $\widetilde{w}$

1. Multiplikation mit Basis b

$$\widetilde{w} = \sum_{i=-m}^{-1} z_i \cdot b^i \quad \Leftrightarrow \widetilde{w}.b = b \cdot \sum_{i=-m}^{-1} z_i \cdot b^i$$

$$= \sum_{i=-m}^{-2} z_i \cdot b^{i+1} + z_{-1}$$

   - Der Summand z_{-1}kann ≥ 1 sein. Damit ist der Vorkommaanteil von $\widetilde{w} \cdot b = z_{-1}$
   - Nachkommaanteil von $\widetilde{w} \cdot b$ stellt Wert der verbleibende Ziffern dar

2. Berechnung aller Ziffern mittels Rekursion

Umwandlung von rationalen Zahlen

Beispiel: Umrechnung von $(0{,}6875)_{10}$ ins Binärsystem

0,6875 * 2 = 1,375
0,375 * 2 = 0,75
0,75 * 2 = 1,5
0,5 * 2 = 1

- Algorithmus endet, wenn Nachkommaanteil 0 ergibt
- Ergebnis von oben nach unten $(0{,}1011)_2$

Umwandlung von rationalen Zahlen

Beispiel: Umrechnung von $(0{,}1)_{10}$ ins Binärsystem

0,1	* 2 = 0,2
0,2	* 2 = 0,4
0,4	* 2 = 0,8
0,8	* 2 = 1,6
0,6	* 2 = 1,2
0,2	* 2 = 0,4
0,4	* 2 = …

- Algorithmus endet nie
- Nachkommaanteil ist im Binärsystem periodisch, obwohl er es im Dezimalsystem nicht ist
- Beim Runden (Begrenzung der Stellen) kommt es zu Rundungsfehler
  - Bsp: Bei 4 Nachkommastellen $(0{,}0001)_2 = (0{,}0625)_{10}$
  - Fehler: $0{,}1 - 0{,}0625 = 0{,}0375$

Umwandlung zw. binären, oktalen und hexadezimalen Zahlen

- Einzelnen Ziffern im oktalen oder hexadezimalen Zahlensystem können direkt als Binärzahl dargestellt werden

$$(\underbrace{3}_{0011}\ \underbrace{A}_{1010}\ \underbrace{7}_{0111}\ \underbrace{E}_{1110})_{16} = (0011\ 1010\ 0111\ 1110)_2$$

$$(\underbrace{3}_{011}\ \underbrace{5}_{101}\ \underbrace{7}_{111})_8 = (011\ 101\ 111)_2$$

- Gruppenbildung von der Kommastelle und fehlende Stellen mit 0 auffüllen

$$(10\ 1110\ 1111{,}101)_2 = (\underbrace{2}_{0010}\ \underbrace{E}_{1110}\ \underbrace{F}_{1111}\ ,\ \underbrace{A}_{1010})_{16}$$

- Keine Umwandlung mit rekursiver Division erforderlich

Kodes und Kodierung

Kodierung

- Kodierung beschreibt eine Umwandlungsvorschrift, die jedem Zeichen eines Zeichenvorrats $\mathcal{A}$ eindeutig ein Zeichen aus einem anderen Zeichenvorrat $\mathcal{B}$ zuordnet
  $c: \mathcal{A} \rightarrow \mathcal{B}$
- Binäre Kodes: Zeichen werden als Bitmuster dargestellt
- Zeichenvorrat für Zahlen
  - Darstellung von Zahlen für bestimmten Bereich
  - Enge Beziehung mit binärer Darstellung
  - Unterstützung der arithmetischen Verknüpfungen
- Nicht-numerischer Zeichenvorrat
  - Buchstaben, Texte, Farben, ...
  - Große Flexibilität, da arithmetische Verknüpfungen nicht unterstützt werden müssen

Binary Coded Decimals (BCD)

- Im BCD Kode wird jede dezimale Ziffer durch 4 Bit dargestellt

Dezimalzahl	BCD-8421-Zahl	Hexadezimalzahl	Bezeichnung
0	0000	0	
1	0001	1	
2	0010	2	
3	0011	3	
4	0100	4	Tetraden
5	0101	5	
6	0110	6	
7	0111	7	
8	1000	8	
9	1001	9	
Folgende Kombinationen werden nicht verwendet			
10	1010	A	
11	1011	B	
12	1100	C	Pseudotetraden
13	1101	D	
14	1110	E	
15	1111	F	

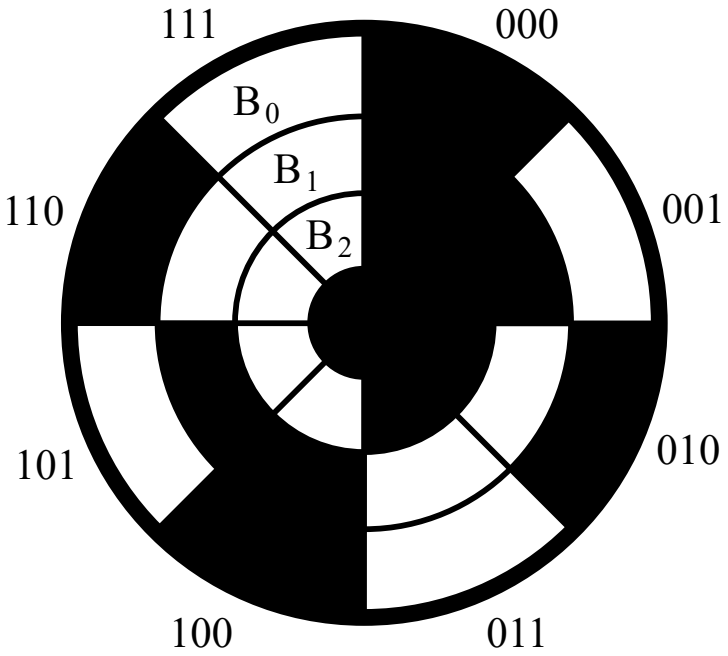
Gray Kode

- Der Gray Kode ändert sich nur um 1 Bit zwischen benachbarten Zahlen

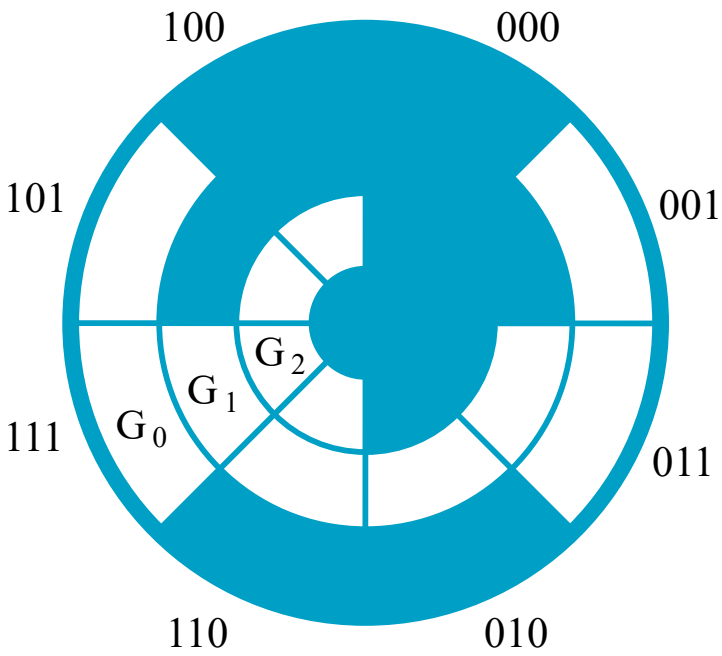
Dezimalzahl	BCD-8421-Zahl	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101

Anwendungsbeispiel: Gray Kode

- Optischer Winkelgeber



(a) Binärer Kode für Positionen 0 bis 7



(b) Gray Kode für Positionen 0 bis 7

„Falsche“ Kodewörter bei ungenauer Lage der optischen Sensoren möglich

Zeichenkodierung

- Um einen Text darzustellen, muss jeder Buchstabe binär kodiert werden
- Je nachdem wie viele Bits pro Zeichen verwendet werden, können unterschiedlich viele verschiedene Zeichen dargestellt werden
  - 7 Bits: $2^7 = 128$ Zeichen
  - 8 Bits: $2^8 = 256$ Zeichen
  - 16 Bits: $2^{16} = 65536$ Zeichen

ASCII Kode

- Der ASCII-Kode (American Standard Code for Information Interchange) ist eine 7-Bit-Zeichenkodierung, die 1963 von der American Standards Association (ASA) beschlossen wurde
- Ein Zeichen wird jedoch immer als 1 Byte (=8 Bits) abgelegt, d.h. das höchstwertige (8.) Bit ist immer Null
- Insgesamt gibt es 128 Zeichen, davon 95 druckbare und 33 Steuerzeichen

ASCII Kodetabelle

- Die Tabelle zeigt alle 128 ASCII-Zeichen
  - Mit hexadezimaler Spalten- und Zeilenbeschriftung

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

- Beispiel: Das Zeichen „A“ hat den Hexadezimalwert $(41)_{16}$

ISO 8859

- In der ASCII-Codierung werden nur 7 der 8 Bits eines Bytes genutzt
- Restliche Zahlenbereich (128 bis 255) kann für weitere Zeichen verwendet werden
- Die International Organization for Standardization definiert in ISO 8859 insgesamt 15 ASCII-Erweiterungen
- ISO 8859-1 enthält z.B. die für uns in Deutschland wichtigen Buchstaben: "ä", "ö", "ü", "ß"

ISO 8859-1	Westeuropäisch (Latin-1)
ISO 8859-2	Mitteleuropäisch (Latin-2)
ISO 8859-3	Südeuropäisch (Latin-3)
ISO 8859-4	Nordeuropäisch (Latin-4)
ISO 8859-5	Kyrillisch
ISO 8859-6	Arabisch
ISO 8859-7	Griechisch
ISO 8859-8	Hebräisch
ISO 8859-9	Türkisch (Latin-5)
ISO 8859-10	Nordisch (Latin-6)
ISO 8859-11	Thai
ISO 8859-12	verworfen
ISO 8859-13	Baltisch (Latin-7)
ISO 8859-14	Keltisch (Latin-8)
ISO 8859-15	Westeuropäisch (Latin-9)
ISO 8859-16	Südosteuropäisch (Latin-10)

Paritätsbit

- Bei Übertragung bzw. Speicherung von Zeichen kann es zu Fehlern f kommen, beispielsweise:
  $(1000\ 0001)_2 \rightsquigarrow (100\ \underbrace{1}_{f}\ 0001)_2$

  Bei ASCII-Kodierung wird dann aus Zeichen „A“ das Zeichen „Q“
- Paritätsbit P: zusätzliches Bit damit Anzahl der „1“ immer
  - gerade (even) oder
  - ungerade (odd) bleibt
- Beispiel: nun mit Paritätsbit P am Ende (gerader Parität):
  $(1000\ 0001\ \underbrace{0}_{P})_2 \rightsquigarrow (100\ \underbrace{1}_{f}\ 0001\ \underbrace{0}_{P})_2$
  - Aus Kodewort mit gerader Parität (zwei „1“) wird ein (ungültiges) Kodewort mit ungerader Parität (drei „1“)
  - Dadurch kann Fehler erkannt werden

UNIVERSITÄT KLAGENFURT

Binäre Arithmetik

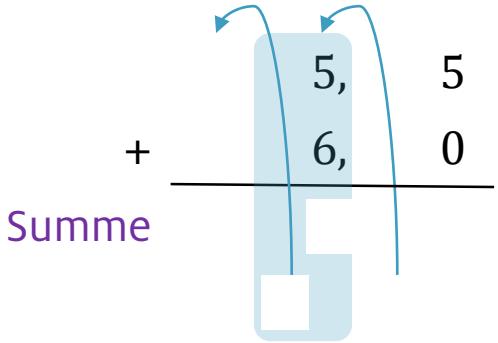
Binäre Arithmetik

- Sinngemäß wie im Dezimalsystem mit Regeln
  - für stellenweise Operation
  - für Übergang von einer zur benachbarten Stelle
- Regeln abhängig von Zahlendarstellung
  - binäre Zahlendarstellung
  - Komplementdarstellung
  - Fixkomma- und Gleitkommadarstellung
- Beispiel: dezimale Addition mit Übertrag

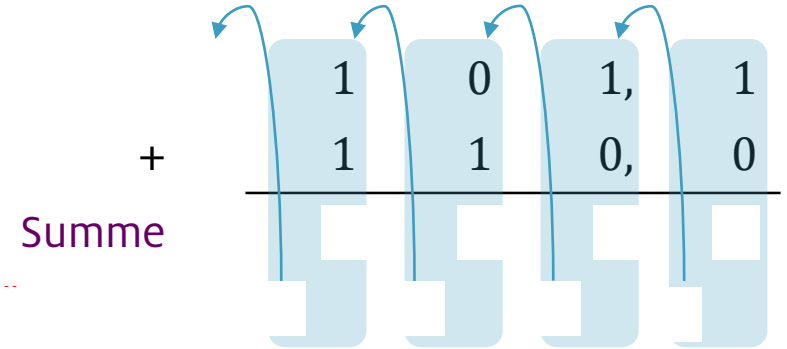
1	1	0	Übertrag
1	4	3	1. Summand
3	6	9	2. Summand
5	1	2	Summe

Rechnen mit Binärzahlen

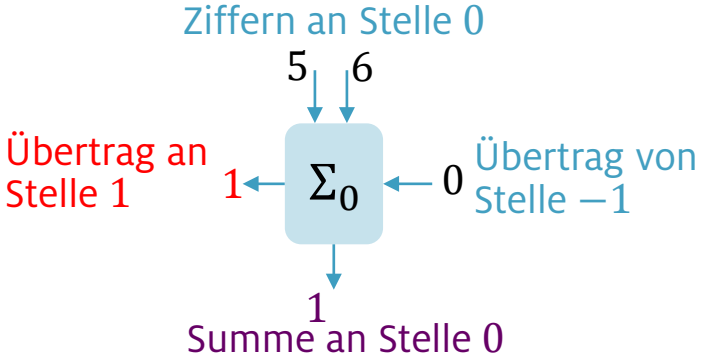
- Addieren von Dezimalzahlen



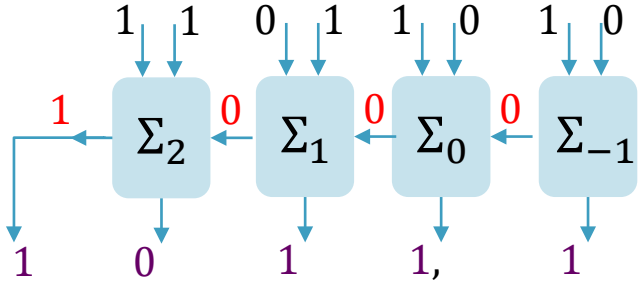
Addieren von Binärzahlen



- Addition an Stelle 0



Addition an Stellen $2, \ldots, -1$



Binäre Addition

- Addition zweier 1 Bit Zahlen X und Y (mit Übertrag Z) Ergebnis S und Übertrag C
  - Übertrag (carry in) $Z = 0$

Z	0	0	0	0
X	0	0	1	1
$+ Y$	$+ 0$	$+ 1$	$+ 0$	$+ 1$
$C S$	0 0	0 1	0 1	1 0

  - Übertrag (carry in) $Z = 1$

Z	1	1	1	1
X	0	0	1	1
$+ Y$	$+ 0$	$+ 1$	$+ 0$	$+ 1$
$C S$	0 1	1 0	1 0	1 1

Mehrstufige binäre Addition

- Addition zweier 1 Bit Zahlen unter Berücksichtigung des Übertrags der vorherigen Stelle
- Übertrag der aktuellen Stelle wird weitergereicht
- Beispiele

01100	10110
10001	10111

Bernhard Rinner
E: bernhard.rinner@aau.at
W: http://nes.aau.at

UNIVERSITÄT KLAGENFURT
Institut für Vernetzte und Eingebettete Systeme