CENTER FOR ADVANCING ELECTRONICS DRESDEN

Rechner Architektur I (RAI)

Informationsverarbeitung

Prof. Dr. Akash Kumar

Chair for Processor Design

Inhalt

- Zahlensysteme
- Umrechnung zwischen Stellenwertsysteme
- Zahlendarstellung im Computer
- Vorzeichenlose natürliche Zahlen
- Vorzeichenbehaftete ganze Zahlen

Zahlensysteme (Stellenwertsysteme)

Polyadische Zahlensysteme (Positional Number System)

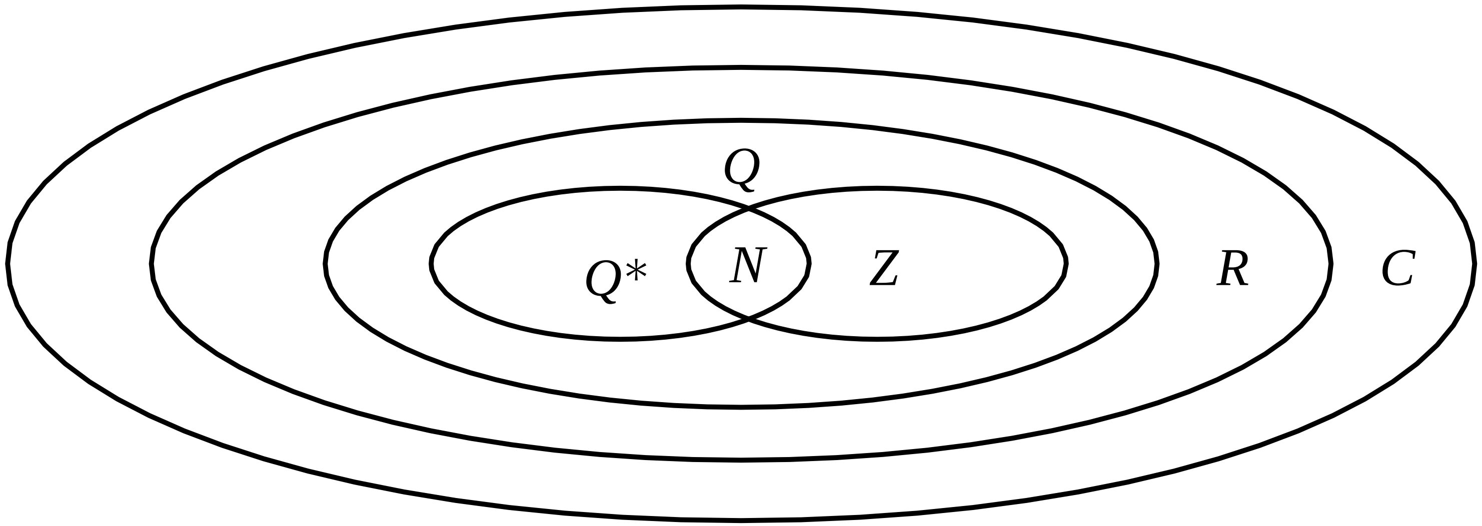
Ziffernfolge (Vor- und Nachkomma): $\sum_{k=-\infty}^{n} a_k B^k$ mit $B \geq 2,\ 0 \leq a_k < B,\ a_k \in N$

Stellenwertsysteme sind gekennzeichnet durch:

- Darstellung einer Zahl durch eine Ziffernfolge (Aneinanderreihung).
- Der Wert einer Ziffer hängt von der Stellung innerhalb der Ziffernfolge ab.
- Die Wertebildung erfolgt zu einer einheitlichen Basis B.
- Die Basis ist die kleinste, nicht mehr durch eine Ziffer darstellbare Zahl.
- Der Wert der Zahl ergibt sich durch Aufsummierung ihrer Ziffernwerte.
- Dezimalsystem ($B = 10$), Dualsystem ($B = 2$), Hexadezimalsystem ($B = 16$)

Zahlenbereiche

- Menge von Zahlen, in der eine Ordnung erklärt ist und gewisse mathematische Verknüpfungen, Operationen uneingeschränkt ausführbar sind (keine natürliche Ordnung für komplexe Zahlen).



$N \subset Q^* \subset Q \subset R \subset C$ bzw. $N \subset Z \subset Q \subset R \subset C$

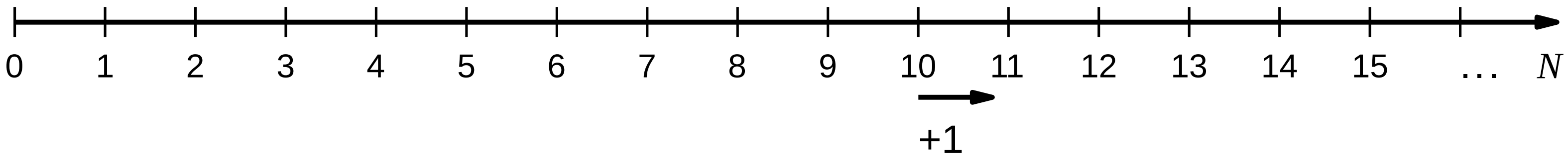
Zahlenbereiche

- Die Menge der natürlichen Zahlen N dient dem Abzählen.
- Die Menge der ganzen Zahlen Z entsteht aus N indem man negative Zahlen als Inverse bzgl. der Addition konstruiert.
- Die Menge der nichtnegativen Brüche Q^* entsteht aus N indem man Bruchzahlen als Inverse der Multiplikation konstruiert.
- Die Menge der Brüche oder rationalen Zahlen Q entsteht aus Q^* durch Hinzunahme der Inversen bzgl. der Addition oder aus Z durch Hinzunahme der Inversen bzgl. der Multiplikation.
- Die Menge der reellen Zahlen R entsteht aus Q durch topologische Vervollständigung (z.B. Werte für e, π).
- Die Menge der komplexen Zahlen C besteht aus Paaren reeller Zahlen (a, b), die in der Schreibweise $a + bi$ mit $i^2 = -1$ den üblichen Rechengesetzen genügen

Zahlenbereiche

	Zahlenbereich	Definition	uneing. Operation
N	natürliche Zahlen	$N = \{0, 1, 2, \ldots\}$	$+, \cdot, <$
Z	ganze Zahlen	$Z = \{m - n,\ \ m, n \in N\}$	$+, -, \cdot, <$
Q^*	gebrochene Zahlen	$Q^* = \{\frac{m}{n},\ \ m, n \in N,\ \ n \neq 0\}$	$+, \cdot, /, <$
Q	reelle Zahlen	$Q = \{\frac{a}{b},\ \ a, b \in Z,\ \ b \neq 0\}$	$+, -, \cdot, /, <$
R	rationale Zahlen	$a_0, a_1 a_2 \ldots \infty$ -Dezimalbruch	$+, -, \cdot, /, <, lim$
C	komplexe Zahlen	$C = [a; b],\ \ a, b \in P$ (Paare)	$+, -, \cdot, /, lim$

Zahlenstrahl natürliche Zahlen (Ordnung: Nachfolger der Zahl n ist $n+1$)



Useful link:
http://www.montereyinstitute.org/courses/DevelopmentalMath/COURSE_TEXT2_RESOURCE/U09_L1_T3_text_final.html

Vier wichtige Nummernsysteme
Four Important Number Systems

System	Warum? Why?	Anmerkungen Remarks
Dezimal Decimal	10 Finger fingers	
Binär Binary	AN/AUS Systeme ON/OFF systems	3 mal mehr Ziffern als dezimal 3 times more digits than decimal
Oktal Octal	Kurzschreibweise für die Arbeit mit Binärdateien Shorthand notation for working with binary	3 mal weniger Ziffern als dezimal 3 times less digits than binary
Hex	------- do ---------	4 mal weniger Ziffern als dezimal 4 times less digits than binary

Positionsnummernsysteme

Positional Number Systems

- Habe ein radix r (Base) mit Ihnen assoziiert
- Have a radix *r* (base) associated with them.
- Im Dezimalsystem ist r = 10 und es gibt 10 Symbole, 0 - 9
- In the decimal system, r = 10, and there are 10 symbols, 0 - 9.
- Was bedeutet 642.391_{10}??
- What does 642.391_{10} mean??

$$6 \times 10^2 + 4 \times 10^1 + 2 \times 10^0 \;.\; 3 \times 10^{-1} + 9 \times 10^{-2} + 1 \times 10^{-3}$$

Zunehmend +ve Krafte von radix

Increasingly +ve powers of radix

Radix Punkt

Radix point

Zunehmend –ve Krafte von radix

Increasingly -ve powers of radix

Positionsnummernsysteme

Positional Number Systems

- Was bedeutet 642.391_{10}??
- What does 642.391_{10} mean??

Radix Punkt point

100 (10^2)	10 (10^1)	1 (10^0)	0.1 (10^{-1})	0.01 (10^{-2})	0.001 (10^{-3})
6	4	2	3	9	1
600	40	2	0.3	0.09	0.001
= 600 + 40 + 2 + 0.3 + 0.09 + 0.001 = 642.391					

- Multiplizieren Sie jede Ziffer mit der entsprechenden Potenz der Basis -10 und fügen Sie sie zusammen
- Multiply each digit by appropriate power of base - 10 and add them together

Ganze Zahlen im Stellenwertsystem

Die Zifferndarstellung einer n-stelligen ganzen Zahlen Z_B lautet im Stellwertsystem der Basis B:

$$Z_B = \pm(z_{n-1} \ldots z_1 z_0)_B.$$

Der Wert von Z_B bestimmt sich durch :

$$Z_B = \pm \sum_{i=0}^{n-1} z_i \cdot B^i = \pm(z_{n-1} \cdot B^{n-1} + \cdots + z_1 \cdot B^1 + z_0 \cdot B^0)$$

Beispiele:

$$Z_2 = 1010011010_2 = \sum_{i=0}^{9} z_i \cdot 2^i = 1 \cdot 2^1 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^7 + 1 \cdot 2^9 = 666_{10}$$

$$Z_8 = -4321_8 = \sum_{i=0}^{3} z_i \cdot 8^i = -(1 \cdot 8^0 + 2 \cdot 8^1 + 3 \cdot 8^2 + 4 \cdot 8^3) = -2257_{10}$$

Echt gebrochene Zahlen im Stellenwertsystem

Die Zifferndarstellung einer gebrochenen Zahl R_B, die kleiner als 1 ist (Nachkommazahl), lautet im Stellenwertsystem der Basis B:

$$R_B = \pm(0,z_{-1}z_{-2}\ldots z_{-m})_B.$$

Der Wert von R_B bestimmt sich durch :

$$R_B = \pm\sum_{i=1}^{m} z_{-i} \cdot B^{-i} = \pm(z_{-1} \cdot B^{-1} + z_{-2} \cdot B^{-2} + \cdots + z_{-m} \cdot B^{-m})$$

Beispiele zu gebrochenen Zahlen:

$$R_{10} = -0,5362_{10} = -\sum_{i=1}^{4} z_{-i} \cdot 10^{-i} = -(5 \cdot 10^{-1} + 3 \cdot 10^{-2} + 6 \cdot 10^{-3} + 2 \cdot 10^{-4})$$

$$R_2 = 0,110_2 = \sum_{i=1}^{3} z_{-i} \cdot 2^{-i} = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 0,75_{10}$$

Positionsnummernsysteme

Positional Number Systems

Number system	Radix	Symbols
Binär Binary	2	{0,1}
Oktal Octal	8	{0,1,2,3,4,5,6,7}
Dezimal Decimal	10	{0,1,2,3,4,5,6,7,8,9}
Hexadecimal	16	{0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f}

There are 10 kinds of people in the world – those who understand binary and those who don‘t.

Umrechnung zwischen Stellenwertsystemen
Conversion between Number Systems

Konvertierung von natürlichen Zahlen (1)

Beispiel: Konvertierung einer Dualzahl $X_2 = 1\ 0011\ 1010_2$ in die wertgleiche Dezimalzahl X_{10}

$$1\ 0011\ 1010_2 = \sum_{i=0}^{n-1} z_i \cdot 2^i$$

$$X_2 = 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0_2$$

$$X_{10} = 1 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

$$\underline{X_{10} = 314_{10}}$$

Konvertierung von natürlichen Zahlen (2)

Beispiel: Konvertierung einer Dezimalzahl $X_{10} = 314$ in die wertgleiche Dualzahl X_2

Sukzessive Division der Dezimalzahl durch 2 und notieren des Restes.
Mit dem Ergebnis der Division ist solange in gleicher Weise weiter zu verfahren, bis das Resultat 0 erreicht ist.
Der Rest, der nur die Werte 0 und 1 annehmen kann, bildet, mit dem letzten Rest der Division als MSB beginnend, die äquivalente Dualzahl.

$314 : 2$	=	157	Rest 0
$157 : 2$	=	78	Rest 1
$78 : 2$	=	39	Rest 0
$39 : 2$	=	19	Rest 1
$19 : 2$	=	9	Rest 1
$9 : 2$	=	4	Rest 1
$4 : 2$	=	2	Rest 0
$2 : 2$	=	1	Rest 0
$1 : 2$	=	0	Rest 1

Reste in umgekehrter Reihenfolge anordnen

$$314_{10} = 1\ 0011\ 1010_2$$

Konvertierung: Binary to Decimal

Binary $\longrightarrow$ Decimal

$1101.011_2 \longrightarrow (??)_{10}$

8 (2^3)	4 (2^2)	2 (2^1)	1 (2^0)	0.5 (2^{-1})	0.25 (2^{-2})	0.125 (2^{-3})
1	1	0	1	0	1	1
8	4	0	1	0	0.25	0.125
= 8 + 4 + 1 + 0.25 + 0.125 = 13.375						

$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \;.\; 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 13.375_{10}$$

Binary point

Konvertierung: Decimal to Binary

Dezimal Decimal ⟶ Binär Binary (Real)

$(27.375)_{10} \longrightarrow (??)_2$

Nummer number	/2	Reste Remainder
27	13	1
13	6	1
6	3	0
3	1	1
1	0	1

int part

$2 \times 0.375 = 0.75$ 0

$2 \times 0.75 = 1.5$ 1

$2 \times 0.5 = 1.0$ 1

Anordnen

Arrange

Reste in umgekehrter Reihenfolge anordnen

Arrange remainders in reverse order: 11011 $\Longrightarrow$ $27.375_{10} = 11011.011_2$

Konvertierung: Octal to Binary

Octal $\longrightarrow$ Binary

$345.5602_8 \longrightarrow (??)_2$

$$\underbrace{3}_{011}\ \underbrace{4}_{100}\ \underbrace{5}_{101}\ .\ \underbrace{5}_{101}\ \underbrace{6}_{110}\ \underbrace{0}_{000}\ \underbrace{2}_{010}$$

$345.5602_8 = 11100101.101110000010_2$

Konvertierung: Binary to Octal

Binary → Octal

$11001110.0101101_2 \longrightarrow (??)_8$

Schleppende Nullen beachten

Note trailing zeros

$\underbrace{11}\ \underbrace{001}\ \underbrace{110}\ .\ \underbrace{010}\ \underbrace{110}\ \underbrace{100}$

Gruppieren nach 3	Gruppieren nach 3
Group by 3's	Group by 3's
Fügen Sie ggf. Führende Nullen hinzu	Fügen Sie ggf. schleppende Nullen hinzu
Add leading zeros if necessary	Add trailing zeros if necessary

$$11001110.0101101_2 = 316.264_8$$

Konvertierung: Binary to Hex

Binary → Hex

$111001011101.1111010111_2 \longrightarrow (??)_{16}$

Schleppende Nullen beachten

Note trailing zeros

$$\underbrace{111}\,\underbrace{0010}\,\underbrace{1101}\;.\;\underbrace{1111}\,\underbrace{0101}\,\underbrace{1100}$$

Gruppieren nach 4 Group by 4's Fügen Sie ggf. Führende Nullen hinzu Add leading zeros if necessary	Gruppieren nach 4 Group by 4's Fügen Sie ggf. schleppende Nullen hinzu Add trailing zeros if necessary

$= 72D.F5C_{16}$

Konvertierung: Hex to Binary

Hex $\longrightarrow$ Binary

$B9A4.E6C_{16} \longrightarrow (??)_2$

$$\underbrace{1011}_{B}\ \underbrace{1001}_{9}\ \underbrace{1010}_{A}\ \underbrace{0100}_{4}\ .\ \underbrace{1110}_{E}\ \underbrace{0110}_{6}\ \underbrace{1100}_{C}$$

$1011100110100100.111001101100_2$

Konvertierung: Hex to Decimal

Hex $\longrightarrow$ Decimal

$B63.4C_{16} \longrightarrow (??)_{10}$

16^2	16^1	16^0	16^{-1}	16^{-2}
B (=11)	6	3	4	C (=12)
= 2816 + 96 + 3 + 0.25 + 0.046875 = 2915.296875				

$$11 \times 16^2 + 6 \times 16^1 + 3 \times 16^0 \,.\, 4 \times 16^{-1} + 12 \times 16^{-2} = 2915.296875_{10}$$

Umrechnung ganzer Zahlen zwischen Stellenwertsystemen

$$
\begin{array}{rcrcl|l}
Z_D & = & Z_B & = & \pm(z_{n-1} \cdot B^{n-1} + z_{n-2} \cdot B^{n-2} \cdots + z_1 \cdot B^1 + z_0 \cdot B^0) & \text{Div.-Rest} \\
\hline
Q_0 & = & Z_D & = & \pm((((z_{n-1} \cdot B + z_{n-2}) \cdot B \ldots) \cdot B + z_1) \cdot B + z_0) & \\
Q_1 & = & \frac{Q_0 - z_0}{B} & = & \pm(((z_{n-1} \cdot B + z_{n-2}) \cdot B \ldots) \cdot B + z_1) & z_0 \\
Q_2 & = & \frac{Q_1 - z_1}{B} & = & \pm((z_{n-1} \cdot B + z_{n-2}) \cdot B) \ldots & z_1 \\
\ldots & & & & & \ldots \\
Q_{n-1} & = & \frac{Q_{n-2} - z_{n-2}}{B} & = & \pm\, z_{n-1} & z_{n-2} \\
Q_n & = & \frac{Q_{n-1} - z_{n-1}}{B} & = & \pm\, 0 & z_{n-1} \\
\hline
 & & Z_B & = & \pm \sum_{i=0}^{n-1} z_i \cdot B^i = \pm(z_{n-1} \ldots z_1 z_0)_B &
\end{array}
$$

Fortlaufende ganzzahlige Division durch B bis $Q_n = 0 \rightarrow$ Divisionsreste z_μ

Z_D - Ist-System (dezimal), Z_B - Ziel-System , Q_ν - ganzzahlige Quotienten

Umrechnung echt gebrochener Zahlen zwischen Stellenwertsystemen

$$
\begin{array}{rcrcl|l}
R_D & = & R_B & = & \pm(z_{-1} \cdot B^{-1} + \cdots + z_{-m+1} \cdot B^{-m+1} + z_{-m} \cdot B^{-m}) & \text{ganzz.} \\
\hline
P_0 & = & R_D & = & \pm((((z_{-m} \cdot \frac{1}{B} + z_{-m+1}) \cdot \frac{1}{B} \ldots) \cdot \frac{1}{B} + z_{-1}) \cdot \frac{1}{B}) & \\
P_1 & = & P_0 \cdot B - z_1 & = & \pm(((z_{-m} \cdot \frac{1}{B} + z_{-m+1}) \cdot \frac{1}{B} \ldots) \cdot \frac{1}{B}) & z_{-1} \\
\ldots & & & & & \ldots \\
P_{m-1} & = & P_{m-2} \cdot B - z_{m-1} & = & \pm\, z_{-m} \cdot \frac{1}{B} & z_{-m+1} \\
P_m & = & P_{m-1} \cdot B - z_m & = & \pm\, 0 & z_{-m} \\
\hline
 & & R_B & = & \pm \sum_{i=1}^{m} z_{-i} \cdot B^{-i} = \pm(0{,}z_{-1}z_{-2}\ldots z_{-m})_B &
\end{array}
$$

Fortlaufende Multiplikation mit B bis $P_m = 0 \rightarrow$ ganzzahliger Anteil z_ν

R_D - Ist-System (dezimal), R_B - Ziel-System , P_ν - echt gebrochene Produkte

Quizzeit!

- Frage 1: Was ist (10111100.00001110)b in Octal form?
  A. (274.034)o
  B. (570.016)o
  C. (270.014)o
  D. (574.034)o

www.menti.com

Code: ...

Signierte Binärzahlen

Signed Binary Numbers

Negative Zahlen Darstellung

Negative numbers representation

- Drei Arten von Darstellungen sind üblich
- Three kinds of representations are common

1. Signierte Grösse Signed Magnitude (SM)
2. Eins Komplement One's Complement
3. Zwei Komplement Two's Complement

Signierte Magnituden-Darstellung (n bits)
Signed magnitude representation (n bits)

[0,1] {…………}

Sign bit (left most) — (n -1) magnitude bits

- 0 indicates +ve
- 1 indicates -ve

8 bit representation for +13 is 0 0001101

8 bit representation for -13 is 1 0001101

1 Komplement Notation (n bits)

1's Complement notation (n bits)

Sei N eine *n*-bit-Zahl und Ñ(1) sei das 1-C der Zahl. Dann,
Let N be an *n* bit number and Ñ(1) be the 1's C of the number. Then,

$$\tilde{N}(1) = 2^n - 1 - N$$

- Die Idee ist, positive Zahlen so zu belassen, wie sie sind, aber negative Zahlen durch die 1 von K ihrer Grösse darzustellen.
- The idea is to leave positive numbers as is, but to *represent negative numbers by the 1's C of their magnitude.*
- *Beispiel*: Sei $n = 4$. Wie Gross ist die 1-K-Darstellung für +6 und -6?
- *Example*: Let $n = 4$. What is the 1's C representation for +6 and -6?
  - +6 ist repräsentiert als is represented as 0110 (wie üblich in binär as usual in binary)
  - -6 wird durch das 1-Komplement seiner Grösse dargestellt is represented by 1's complement of its magnitude (6)

1 Komplement Notation (n bits)

1's Complement notation (n bits)

- Die C-Repräsentation von 1 kann auf zwei Arten berechnet werden:
- 1's C representation can be computed in 2 ways:
  - *Method 1*: 1's C representation of -6 is $2^4 - 1 - |N| = (16 - 1 - 6)_{10} = (9)_{10} = (1001)_2$
  - *Method 2*: For -6, the magnitude = 6 = $(0110)_2$
    - Die C-Darstellung von 1 wird durch Komplementieren der Bits der Grösse erhalten:
    - The 1's C representation is obtained by complementing the bits of the magnitude: $(1001)_2$
    - $2^4 - 1 - |N| = (16)_{10} - 1 - |N| = (15)_{10} - |N|$ $= (1111)_2 - |N|$

2 Komplement Notation (n bits)

2's Complement notation (n bits)

Sei N eine *n*-bit-Zahl und Ñ(2) sei das 2-C der Zahl. Dann,

Let N be an *n* bit number and Ñ(2) be the 2's C of the number. Then,

$$\tilde{N}(2) = 2^n - N$$

- Wiederum ist die Idee, positive Zahlen so zu belassen, wie sie sind, aber negative Zahlen durch die 2 von C ihrer Grösse darzustellen.
- Again, the idea is to leave positive numbers as is, but to *represent negative numbers by the 2's C of their magnitude.*
- *Bespiel*: Sei $n = 5$. Wie Gross ist die 2-C-Darstellung für +11 und -13?
- *Example*: Let $n = 5$. What is the 2's C representation for +11 and -13?
  - +11 ist repräsentiert als is represented as 0110 (wie üblich in binär as usual in binary)
  - -13 wird durch das 2-Komplement seiner Grösse dargestellt is represented by 1's complement of its magnitude (13)

2 Komplement Notation (n bits)

2's Complement notation (n bits)

- Die C-Repräsentation von 1 kann auf zwei Arten berechnet werden:
- 2's C representation can be computed in 2 ways:
  - *Method 1*: 2's C representation of -13 is 2^5 - |N| = $(32 - 13)_{10} = (19)_{10} = (10011)_2$
  - *Method 2*: For -13, the magnitude = 13 = $(01101)_2$
    - Die C-Darstellung von 2 wird durch Komplementieren der Bits der Grösse erhalten:
    - The 2's C representation is obtained by adding 1 to the 1's C of the magnitude:
    - $2^5 - |N| = (2^5 - 1 - |N|) + 1 = 1\text{'s C} + 1$

$$01101 \xrightarrow{1's\,C} 10010 \xrightarrow{add\,1} 10011$$

Vergleichen aller signierten Notationen

Comparing all Signed Notations (4-bit)

4-bit No.	SM	1's C	2's C
0000	+0	+0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	-0	-7	-8
1001	-1	-6	-7
1010	-2	-5	-6
1011	-3	-4	-5
1100	-4	-3	-4
1101	-5	-2	-3
1110	-6	-1	-2
1111	-7	-0	-1

- In allen 2 Darstellungen hat eine –ve-Nummer eine 1 in MSB Position
- In all 3 representations, a –ve number has a 1 in MSB location
- Um –ve Zahlen mit n Bits zu behandeln,
- To handle –ve numbers using *n* bits,
  - $\cong 2^{n-1}$ symbols can be used for positive numbers
  - $\cong 2^{n-1}$ symbols can be used for negative umbers
- In der 2-C-Notation wird nur 1 Kombination für 0 verwendet
- In 2's C notation, only 1 combination used for 0

Vorzeichenlose natürliche Zahlen
unsigned binary numbers

Die Darstellung erfolgt als n-stellige Dualzahl.

Eine Darstellung von negativen Zahlen ist nicht direkt möglich.

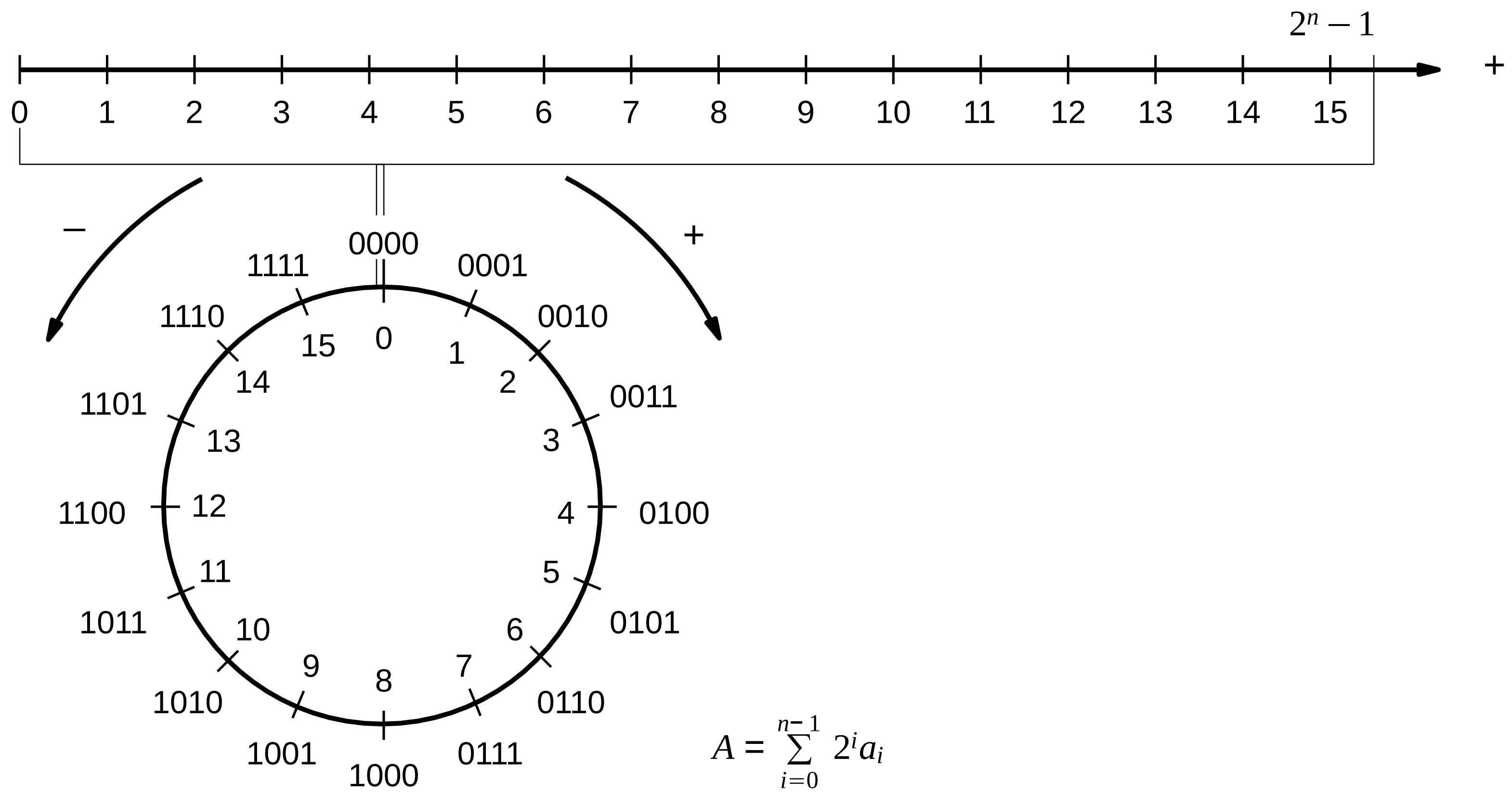
Wertebereich: $0 \leq x \leq 2^n - 1$

Beispiel für Dualzahlen-Darstellung $(n = 8)$**:**

dezimal		dual	dezimal		dual
$+0$	$-$	$0000\ 0000$	128	$-$	$1000\ 0000$
$+3$	$-$	$0000\ 0011$	131	$-$	$1000\ 0011$
$+127$	$-$	$0111\ 1111$	255	$-$	$1111\ 1111$

Anwendung: Speicheradresse

Zahlenkreis für 4-Bit-Zahlen als vorzeichenlose Dualzahl



$$A = \sum_{i=0}^{n-1} 2^i a_i$$

Vorzeichenbehaftere ganze Zahlen
signed binary numbers

Die Darstellung erfolgt als n-stellige Dualzahl, wobei das Vorzeichen durch verschiedene Verfahren realisiert wird:

- Vorzeichen-Wert-Darstellung
- Basiswert-Darstellung
- 1-Komplement-Darstellung
- 2-Komplement-Darstellung

Die Wertebereiche der Darstellungen unterscheiden sich je nach der Realisierung des Vorzeichens.

Anwendung: Integer-Arithmetik

Vorzeichen-Wert-Darstellung

Das höchstwertigste Bit (MSB) der Dualzahl wird für die Darstellung des Vorzeichens genutzt: (0-positiv, 1-negativ), die restlichen Stellen als $n-1$ Bit lange vorzeichenlose ganze Zahl.

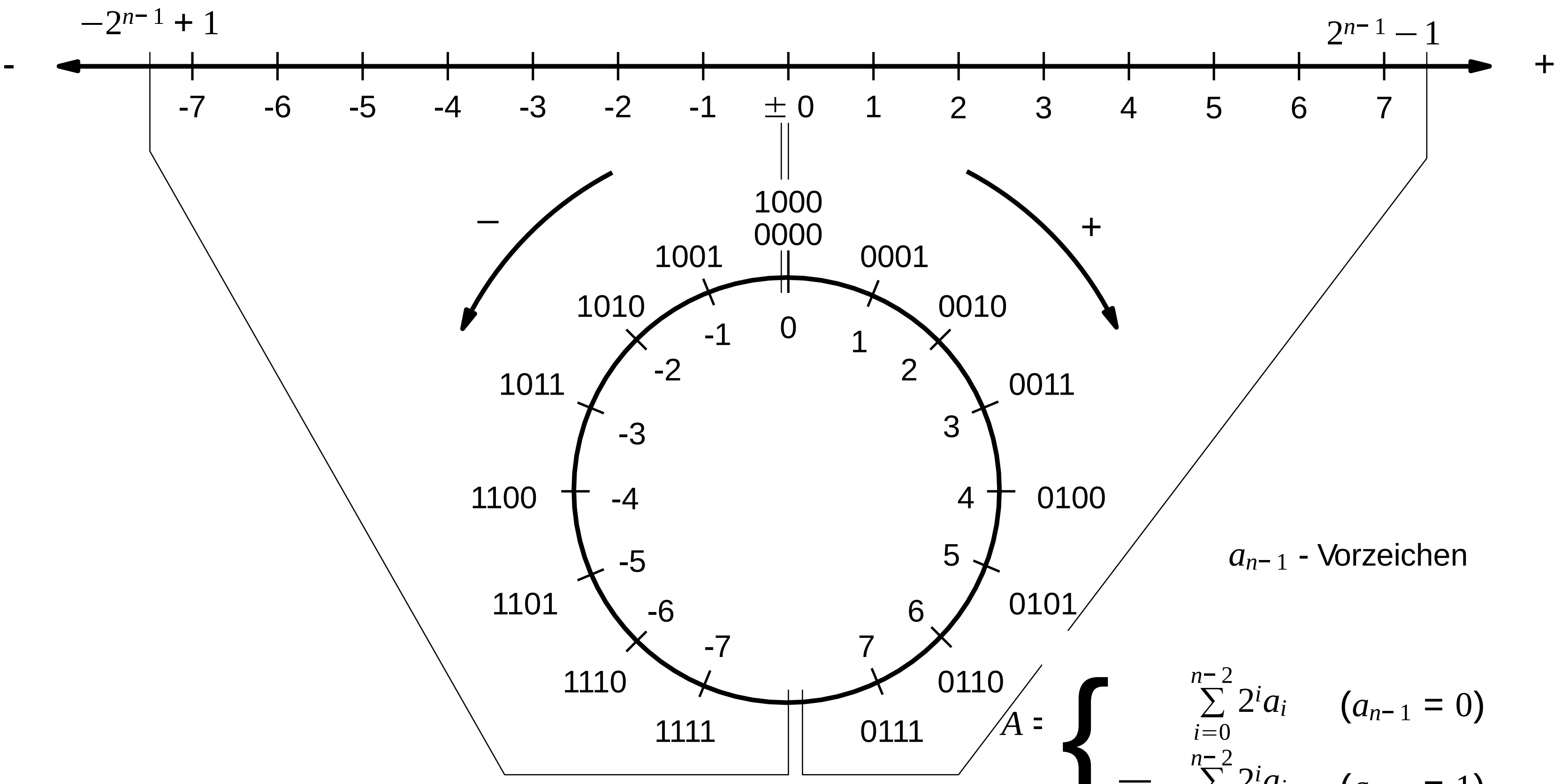
Wertebereich: $-2^{n-1}+1 \leq x \leq 2^{n-1}-1$

Es gibt zwei Darstellungen für die Zahl 0: $+0$ und -0

Beispiel für Vorzeichen-Wert-Darstellung ($n = 8$):

dezimal		dual	dezimal		dual
$+0$	–	$0000\ 0000$	-0	–	$1000\ 0000$
$+3$	–	$0000\ 0011$	-3	–	$1000\ 0011$
$+127$	–	$0111\ 1111$	-127	–	$1111\ 1111$

Zahlenkreis für 4-Bit-Zahlen in Vorzeichen-Wert-Darstellung



a_{n-1} - Vorzeichen

$$A = \begin{cases} \sum_{i=0}^{n-2} 2^i a_i & (a_{n-1} = 0) \\ -\sum_{i=0}^{n-2} 2^i a_i & (a_{n-1} = 1) \end{cases}$$

Basiswert-Darstellung (biased)

Darstellung vorzeichenbehafteter ganzer Zahlen x als positive Dualzahl d mit einer festen Basiswertverschiebung B.

$$x = d - B \quad \rightarrow \quad d = x + B \geq 0$$

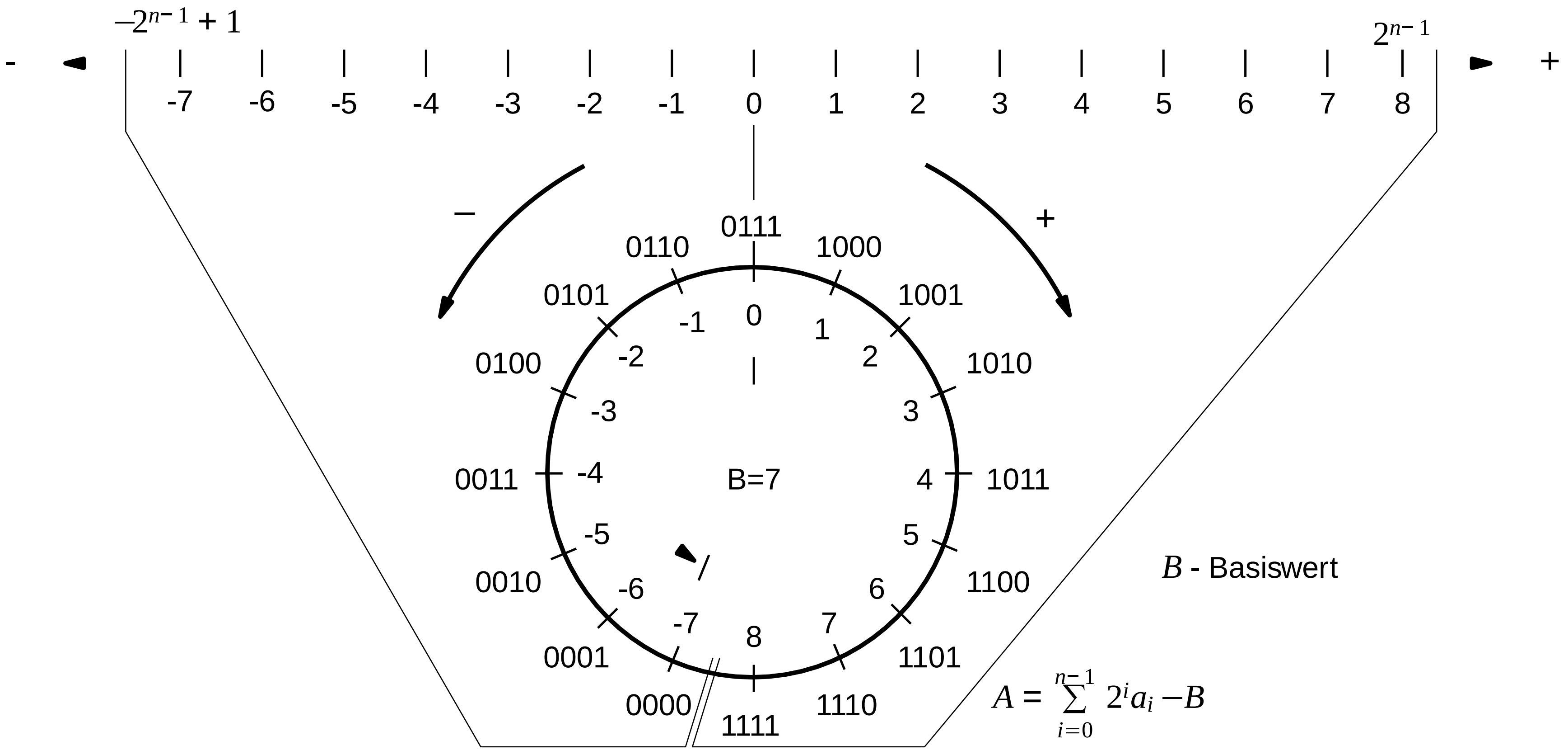
Eindeutige Darstellungen der 0

Wertebereich: $-B \leq x \leq 2^n - 1 - B$ (symmetrisch für $B = 2^{n-1} - 1$)

Beispiel für Vorzeichen-Wert-Darstellung ($n = 8$, $B = 127$):

dezimal		dual	dezimal		dual
$+0$	–	0111 1111	-0	–	–
$+3$	–	1000 0010	-3	–	0111 1100
$+128$	–	1111 1111	-127	–	0000 0000

Zahlenkreis für 4-Bit-Zahlen in Basiswert-Darstellung



B - Basiswert

$$A = \sum_{i=0}^{n-1} 2^i a_i - B$$

1-Komplement-Darstellung (1)

Darstellung negativer ganzer Zahlen x durch das 1-Komplement ${}^{(1)}x$.

$${}^{(1)}x = (2^n - 1) - x \quad \rightarrow \quad {}^{(1)}x + x = (2^n - 1) = 0$$

Wertebereich: $-2^{n-1} + 1 \leq x \leq 2^{n-1} - 1$

Es gibt zwei Darstellungen für die Zahl 0: $+0$ und -0
Negative Zahlen sind durch eine 1 im höchstwertigen Bit (MSB) gekennzeichnet.

Beispiel für 1-Komplement-Darstellung ($n = 8$):

dezimal		dual	dezimal		dual
$+0$	–	$0000\ 0000$	-0	–	$1111\ 1111$
$+3$	–	$0000\ 0011$	-3	–	$1111\ 1100$
$+127$	–	$0111\ 1111$	-127	–	$1000\ 0000$

1-Komplement-Darstellung (2)

Bildungsvorschrift für das 1-Komplement (Stellenkomplement):

Stelleweise Negation von x: ${}^{(1)}x \quad \rightarrow \quad (\overline{x}_\nu$ für $\nu = 1\dots n-1)$

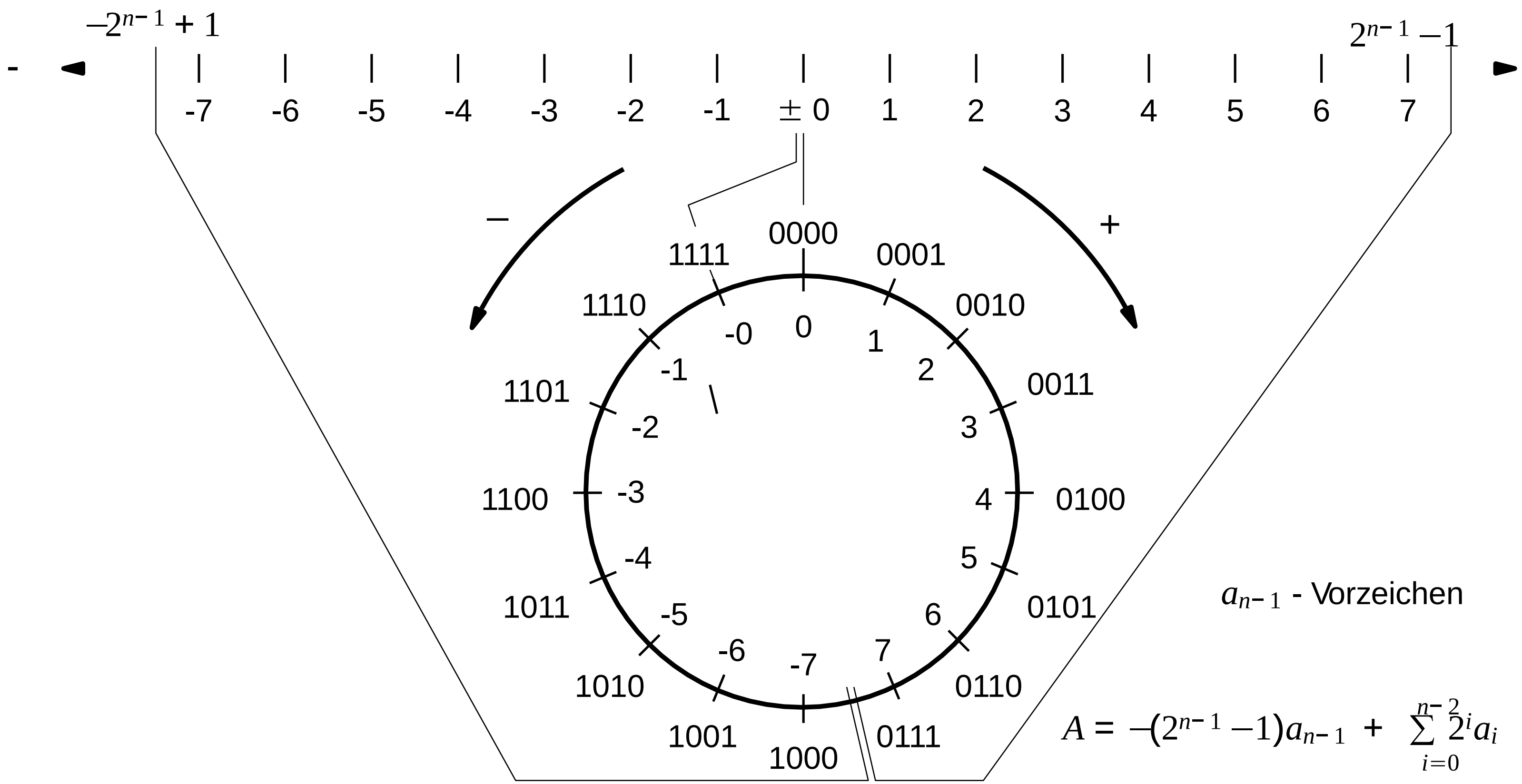
Daraus folgt:

$$\begin{aligned} {}^{(1)}x + x &= x_{n-1}x_{n-2}\dots x_0 + \overline{x}_{n-1}\overline{x}_{n-2}\dots\overline{x}_0 \\ &= 1_{n-1}1_{n-2}\dots 1_0 = 2^n - 1 \\ x_\nu + \overline{x}_\nu &= 1 \\ \\ {}^{(1)}({}^{(1)}x) &= x \quad \rightarrow -(-x) = x \end{aligned}$$

Hin- und Rücktransformation (positive Zahl $\longleftrightarrow$ negative Zahl) sind identisch.

Beim Rechnen mit dem 1-Komplement kann ein Fehler auftreten.
Ist bei $s = a + b$ der auslaufende Übertrag 1 so wird $s = a + b + 1$ korrigiert.

Zahlenkreis für 4-Bit-Zahlen in 1-Komplement-Darstellung



a_{n-1} - Vorzeichen

$$A = -(2^{n-1}-1)a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

2-Komplement-Darstellung (1)

Darstellung negativer ganzer Zahlen x durch das 2-Komlement ${}^{(2)}x$.

$$ {}^{(2)}x = (2^n) - x \quad \rightarrow \quad {}^{(2)}x + x = 2^n = 0 $$

Wertebereich: $-2^{n-1} \leq x \leq 2^{n-1} - 1$

Eindeutige Darstellungen für die Zahl 0 (2^n ausserhalb des Darstellungsbereiches)
Negative Zahlen sind durch eine 1 im höchstwertigen Bit (MSB) gekennzeichnet.

Beispiel für 2-Komplement-Darstellung ($n = 8$):

dezimal		dual	dezimal		dual
$+0$	–	0000 0000	-0	–	–
$+3$	–	0000 0011	-3	–	1111 1101
$+127$	–	0111 1111	-128	–	1000 0000

2-Komplement-Darstellung (2)

Bildungsvorschrift für das 2-Komplement (echtes Komplement):

Stelleweise Negation von x (1-Komplement) und anschliessende Inkrementierung (1-Addition): ${}^{(2)}x = {}^{(1)}x + 1$

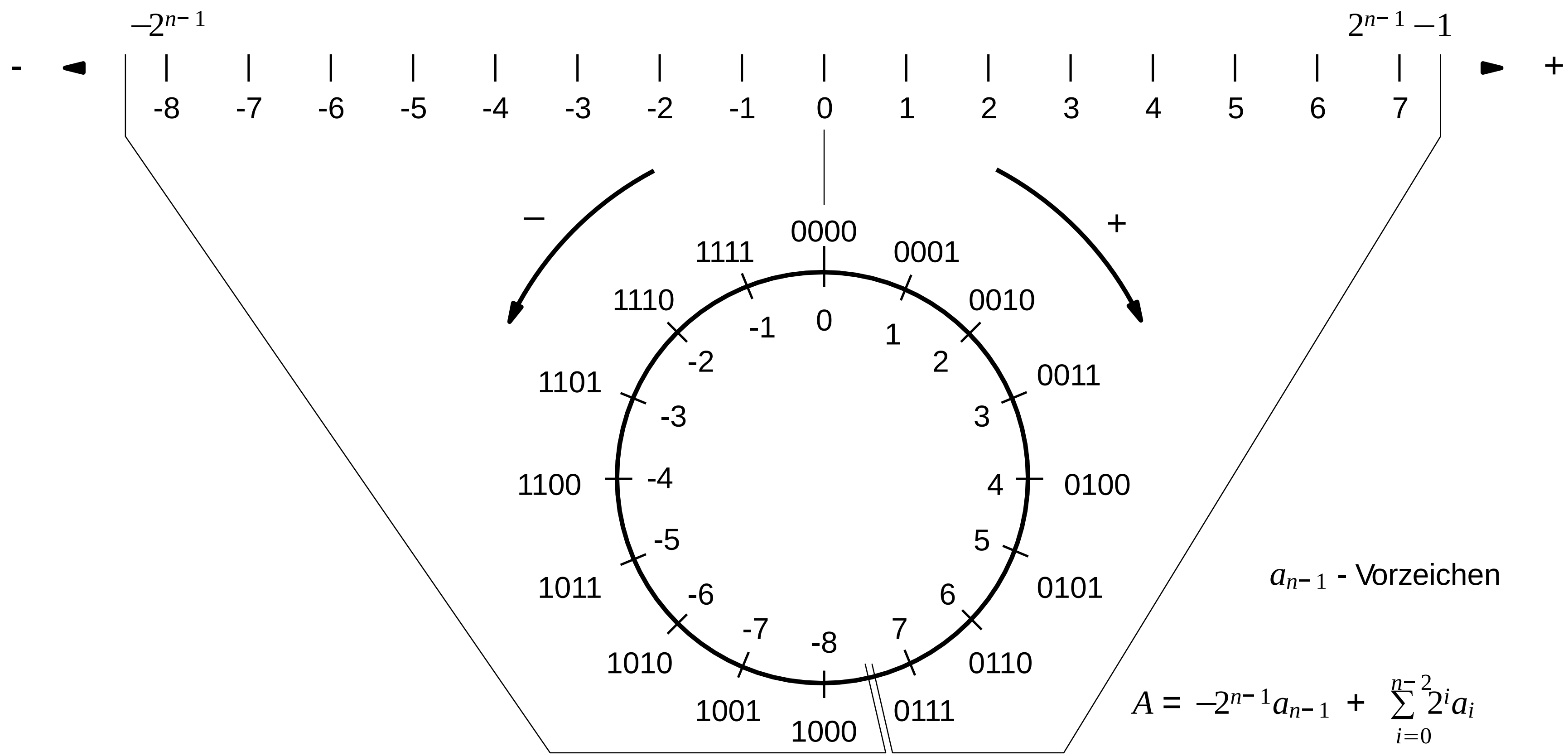
Daraus folgt:

$$
\begin{aligned}
{}^{(2)}x + x &= x_{n-1}x_{n-2}\ldots x_0 + \bar{x}_{n-1}\bar{x}_{n-2}\ldots\bar{x}_0 + 1 \\
&= 1_{n-1}1_{n-2}\ldots 1_0 + 1 = 2^n \\
x_\nu + \bar{x}_\nu &= 1 \\
{}^{(2)}({}^{(2)}x) &= x \quad \rightarrow -(-x) = x
\end{aligned}
$$

Hin- und Rücktransformation (positive Zahl $\longleftrightarrow$ negative Zahl) sind identisch.

Die 2-Komplementdarstellung für negative ganze Zahlen wird in der Computertechnik am häufigsten angewendet.

Zahlenkreis für 4-Bit-Zahlen in 2-Komplement-Darstellung



a_{n-1} - Vorzeichen

$$A = -2^{n-1}a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i$$

2-Komplement-Darstellung (3)

Addition und Subtraktion

- Die Darstellung negativer Dualzahlen erfolgt durch das 2-Komplement.
- Die Konvertierung positiver Dualzahlen in negative und umgekehrt erfolgt am einfachsten durch bitweise Negation der Dualzahl und anschließender Inkrementierung (1-Addition).
- Negative Zahlen sind durch die 1 im MSB gekennzeichnet.
- Die Subtraktion entspricht einer Addition mit einer negativen Zahl.

$$a - b \quad = \quad a + (-b)$$

- Die Addition wird analog zur Addition vorzeichenloser ganzer Zahlen durchgeführt (bitweise modulo-2 Addition mit Übertrag).
- Ein Überlauf (Wertebereichsüberschreitung) liegt vor, wenn beim höchstwertigsten Bit (MSB) der einlaufende und der auslaufende Übertrag unterschiedlich sind.

2-Komplement-Darstellung (3) Beispiele

dezimal ($n = 2$)

$$\begin{array}{rcc} & 1 & 6_{10} \\ + & 3 & 2_{10} \\ \hline = & 4^0 & 8_{10} \end{array}$$

$$\begin{array}{rccc} & & 8 & 5_{10} \\ + & & 4 & 3_{10} \\ \hline = & 1^1 & 2^0 & 8_{10} \end{array}$$

$$\begin{array}{rcc} & 6 & 5_{10} \\ - & & 3_{10} \\ \hline = & 6^0 & 2_{10} \end{array}$$

dual ($n = 8$)

$$\begin{array}{rc|cccccccc} & & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0_2 \\ + & & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0_2 \\ \hline = & 0^0 & 0^0 & 0^0 & 1^0 & 1^0 & 0^0 & 0^0 & 0^0 & 0_2 \end{array}$$

$$\begin{array}{rc|cccccccc} & & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1_2 \\ + & & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1_2 \\ \hline = & 0^0 & 1^1 & 0^1 & 0^1 & 0^1 & 0^1 & 0^1 & 0^1 & 0_2 \end{array} \quad \text{OV}$$

$$\begin{array}{rc|cccccccc} & & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1_2 \\ + & & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1_2 \\ \hline = & 1^1 & 0^1 & 0^0 & 1^0 & 1^0 & 1^0 & 1^0 & 1^1 & 0_2 \end{array}$$

OV - Überlauf (Wertebereichsüberschreitung, overflow)

2-Komplement-Darstellung (4) Beispiele

dezimal ($n = 2$)

$$\begin{array}{lccc} & -1 & 1 & 5_{10} \\ - & & 1 & 6_{10} \\ \hline = & -1^{0} & 3^{1} & 1_{10} \end{array}$$

$$\begin{array}{lccc} & -1 & 1 & 5_{10} \\ - & & 1 & 3_{10} \\ \hline = & -1^{0} & 2^{1} & 8_{10} \end{array}$$

$$\begin{array}{lccc} & & 1 & 6_{10} \\ - & & 3 & 2_{10} \\ \hline = & - & 1^{1} & 6_{10} \end{array}$$

dual ($n = 8$)

$$\begin{array}{ll|cccccccc} & & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1_2 \\ + & & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0_2 \\ \hline = & 1^{1} & 0^{0} & 1^{0} & 1^{0} & 1^{0} & 1^{0} & 1^{0} & 0^{0} & 1_2 \end{array} \quad \text{OV}$$

$$\begin{array}{ll|cccccccc} & & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1_2 \\ + & & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1_2 \\ \hline = & 1^{1} & 1^{1} & 0^{1} & 0^{1} & 0^{1} & 0^{1} & 0^{1} & 0^{1} & 0_2 \end{array}$$

$$\begin{array}{ll|cccccccc} & & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0_2 \\ + & & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0_2 \\ \hline = & 0^{0} & 1^{0} & 1^{0} & 1^{0} & 1^{0} & 0^{0} & 0^{0} & 0^{0} & 0_2 \end{array}$$

OV - Überlauf (Wertebereichsüberschreitung, overflow)

2-Komplement-Darstellung (5)

Multiplikation und Division (Rückführung auf Vorzeichen-Wert-Darstellung)

1. Überführung der 2-Komplement-Darstellung negativer Zahlen in die Vorzeichen-Wert-Darstellung.
2. Multiplikation bzw. Division der vorzeichenlosen Beträge analog zu vorzeichenlosen ganzen Zahlen.
3. Gesonderte Bestimmung des Vorzeichens (gleiche Vorzeichen - positives Ergebnis, unterschiedliche Vorzeichen - negatives Ergebnis).
4. Überführung der Vorzeichen-Wert-Darstellung negtiver Ergebnisse in die 2-Komplement-Darstellung.

Booth-Recording: Direkte Verarbeitung von Zweierkomplement-Darstellungen bei der Multiplikation.

B-Komplement ganzer Zahlen im Stellenwertsystem (1)

Das B-Komplement ${}^{(B)}Z_B$ einer ganzen Zahl Z_B im Stellenwertsystem der Basis B ist definiert durch (n Anzahl der Ziffern):

$$Z_B + {}^{(B)}Z_B = B^n \quad \text{mit} \quad Z_B = \sum_{i=0}^{n-1} z_i B^i \quad \text{und} \quad B^n = 1 + \sum_{i=0}^{n-1} (B-1) B^i$$

Der Wert dieses Komplements berechnet sich zu:

$${}^{(B)}Z_B = B^n - Z_B = 1 + \sum_{i=0}^{n-1} (B-1) B^i - \sum_{i=0}^{n-1} z_i B^i = 1 + \sum_{i=0}^{n-1} (B-1-z_i) B^i$$

Die beim B-Komplement auftretende Summe heißt $(B-1)$-Komplement:

$${}^{(B-1)}Z_B = \sum_{i=0}^{n-1} (B-1-z_i) B^i \quad \text{und das B-Komplement damit:} \quad {}^{(B)}Z_B = {}^{(B-1)}Z_B + 1$$

B-Komplement ganzer Zahlen im Stellenwertsystem (2)

Beispiele zum B-Komplement:

Zehn-Komplement von $Z_{10} = 51 \quad (n = 2)$: $\quad {}^{(10)}Z_{10} = 10^2 - 51 = 100 - 51 = 49$

Neun-Komplement von $Z_{10} = 51 \quad (n = 2)$: $\quad {}^{(9)}Z_{10} = 10^2 - 1 - 51 = 99 - 51 = 48$

Zwei-Komplement von $Z_2 = 0101 \; (n = 4)$: $\quad {}^{(2)}Z_2 = 10000 - 0101 = 1011$

(dezimal: $2^4 - 5 = 11$)

Eins-Komplement von $Z_2 = 0101 \; (n = 4)$: $\quad {}^{(1)}Z_2 = 10000 - 0001 - 0101 = 1111 - 0101 = 1010$

(dezimal: $2^4 - 5 - 1 = 10$)

Nummernsystem und Computer

Number System and Computers

- Some tips
  - Binary numbers often grouped in fours for easy reading
  - In computer programs (e.g. VHDL, C) by default decimal is assumed
  - To represent other number bases use

System	Representation	Example for 20
Hexadecimal	0x…	0x14
Binary	0b…	0b10100
Octal	0o… (zero and ‘O’)	0o24

Nummernsystem und Computer

Number System and Computers

- Addresses often written in Hex
  - Most compact representation
  - Easy to understand given their hardware structure
  - For a range 0x000 – 0xFFF, we can immediately see that 12 bits are needed, 4K locations
  - Tip: 10 bits = 1K

Zusammenfassung

- Binärzahlsystem ist am nützlichsten für Computerarithmetik
- Binary number system is most useful for computer arithmetic
- Oktal und Hex bieten eine praktische Kurzhandnotation
- Octal and Hex provide a convenient short-hand notation
- Zweierkomplement die häufigste Notation für vorzeichenbehaftete Zahlen
- 2‘s complement most common notation for Signed numbers

Anhang
Appendix

Binärzahlen

Binary Numbers

- How many distinct numbers can be represented by *n* bits?

No. of bits	Distinct nos.
1	2 {0,1}
2	4 {00, 01, 10, 11}
3	8 {000, 001, 010, 011, 100, 101, 110, 111}
n	2^n

- Number of permutations double with every extra bit
- 2^n *unique* numbers can be represented by *n* bits

Speicherorte

Memory Locations

- **Most** memories are byte-addressable
  - Each unique combination on the address line refers to a byte
- If we have 12-bit addresses
  - We can have 4096 unique combinations
  - We can address 4096 bytes of memory

Address	DATA
0x000	01010101
0x001	01010101
0x002	01010101
0x003	10101010
…	…
…	…
0xFFD	10101010
0xFFE	01001110
0xFFF	11110000

Size: 4K bytes

Speicherorte

Memory Locations

- Start address: 0x1000
- End address: 0x13FF
- How many bytes of memory is addressed?
  - Consider how many unique combinations are there in this range.
- How many bits are needed for the address?
  - Consider the size of address

Address	DATA
0x1000	01010101
0x1001	01010101
0x1002	01010101
0x1003	10101010
…	…
…	…
0x13FD	10101010
0x13FE	01001110
0x13FF	11110000

Size: ?? bytes

Quizzeit!!!

- Frage 2: The final address location if there are 9K locations starting from 0x3400 is
  A. 0x4499
  B. 0x5799
  C. 0x5800
  D. None of the above
- SMS ELE2020b <Response> to 77577

e.g. ELE2020b C