1.3 Zahlen-Darstellungen-Arithmetik

Stellenwertsysteme/Zahlensysteme/Konvertierung

Komplement-Zahldarstellungen

ganze pos. & neg. Zahlen (INTEGER, ...)

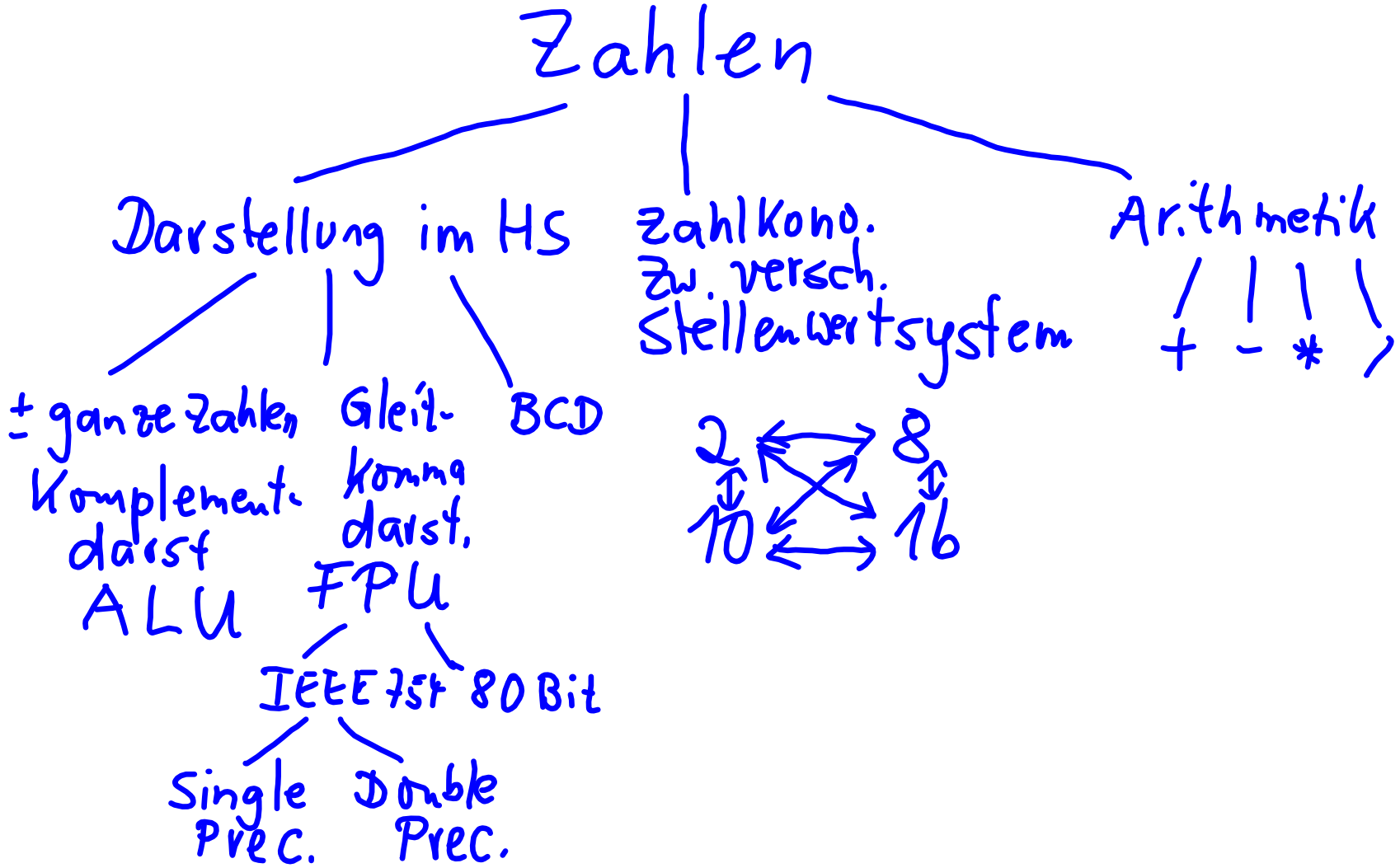
Gleitkomma-Zahldarstellung (IEEE 754)

$Z = v \cdot m\, B^e$ m = Mantisse, e = Exponent, v = +/-

Alternative Zahlendarstellungen

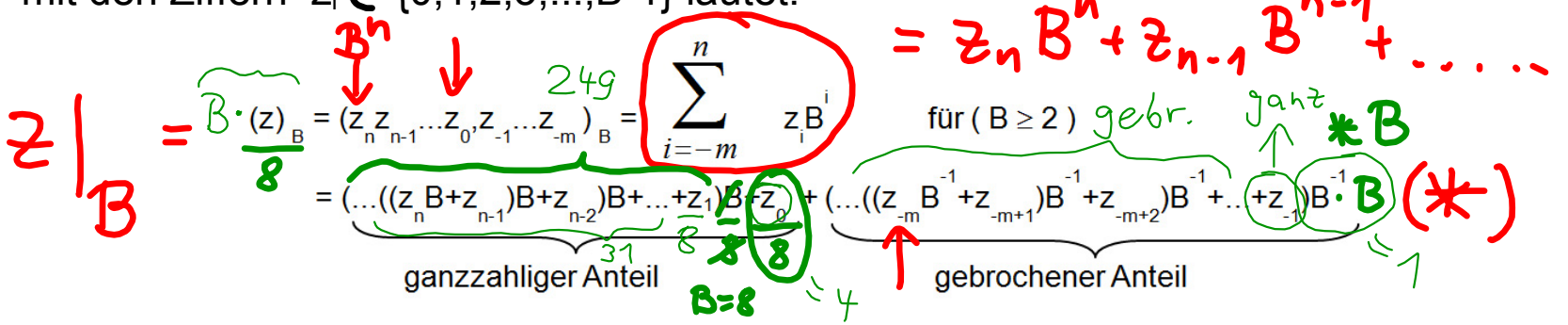
BCD-Zahldarstellung (Binary Coded Decimals)

Arithmetik (Add./Sub./Mult. im 2/8/16er-Zahlensystem)


Zahlen
Darstellung im HS
Zahlkonv. zw. verschiedenen Stellenwertsystemen
Arithmetik
+ - * /
± ganze Zahlen
Komplement-darst
ALU
Gleit-komma darst.
FPU
BCD
IEEE 754 80 Bit
Single Prec.
Double Prec.
2
8
10
16

Stellenwertsysteme/Zahlensysteme

Die allgemeine Zahldarstellung einer Zahl z in einem Stellenwertsystem zur Basis B mit den Ziffern $z_i \in \{0,1,2,3,...,B-1\}$ lautet:



spezielle Zahlsysteme:			
	Dualsystem	B=2	$z_i \in \{0, 1\}$
	Oktalsystem	B=8	$z_i \in \{0,\ldots,7\}$
	Dezimalsystem	B=10	$z_i \in \{0,\ldots,9\}$
	Hexadezimalsystem	B=16	$z_i \in \{0,\ldots,9,A,\ldots,F\}$

Hornerschema zur Berechnung des Wertes von z:

Zahlenkonvertierung

Konvertierung: ⇄ ganzer Anteil / gebrochener Anteil

Umwandlung der Zahldarstellung in Stellenwertsysteme verschiedener Basen

a) $10 \rightleftarrows \begin{matrix}2\\8\\16\end{matrix}$ (ganzer Anteil, Divisionsmethode)

Beim Dividieren einer ganzen Zahl z durch die Zielbasis B ergeben die Divisionsreste nacheinander gelesen die von rechts gelesene ganze Zahl im Zielstellenwertesystem zur Basis B.

$z = (1996)_{10} = (3714)_8$

gebr. Teil	Rest
0,125	1
0,25	2 → $\frac{2}{8} = \frac{1}{4}$
0,375	3
→ 0,5	4
0,625	5
0,75	6
0,875	7
0	0

Rest ↓

$1996 : 8 = 249,5 = 249\frac{4}{8} = 249 \cdot 8 + 4$

$249 : 8 = 31,125 = 31 \cdot 8 + 1$

$31 : 8 = 3$ Rest 7

$3 : 8 = 0$ Rest 3

b) 

Nach obigem Algorithmus müßten die Divisionen im Stellenwertsystem B durchgeführt werden.

Besser: Berechne den dezimalen Wert mit Hilfe des Hornerschemas.

Bsp: $A5DH = 0xa5d = (2653)_{10}$

	10	5	13
16	↓	160	2640
	10	165	2653

Bsp: $(0,11010011)_2 = (\frac{105}{128})_{10}$

	1	0	0	1	0	1	1	
1/2	↓	1/2	1/4	1/8	9/16	9/32	41/64	105/128
reziproke Basis	1	1/2	1/4	9/8	9/16	41/32	105/64	
							↓	
							1	

c) gebr. Dezimalzahlen ⇄ 2, 8, 16 Multiplikationsmethode

Beim Multiplizieren einer echt gebrochenen Dezimalzahl mit der Zielbasis B ergeben die Überläufe über 1 nacheinander gelesen die Ziffernfolge der gebrochenen Zahl im Stellenwertsystem zur Zielbasis B.

Bsp: $z = (0,6)_{10} = (0,\overline{4631})_8 = 0,463146314631\ldots$

$0,6 \cdot 8 = 4,8$ ↑ z_{-1} Rest = ( ) $B^{-1} \cdot B$

$0,8 \cdot 8 = 6,4$ $z_{-2} = 6$

$0,4 \cdot 8 = 3,2$ $z_{-3} = 3$

$0,2 \cdot 8 = 1,6$

$0,6 \cdot 8 = 4,8$

Bsp: $(0,\overline{3})_{10} = (0,1)_3$

d)

Ziffern ⇄ Aufspaltung / Zusammenfassung

Dualziffern → Triple = Oktalziffer

Dualziffern → Tetrade = Hexadezimalziffer

Bsp: $ABCH = (1010\,1011\,1100)_2$

$(5\;2\;7\;4)_8$

Bsp: $(0,\overline{4631})_8 = (0,\overline{100110011001})_2$

$= (0,\overline{999})_{16} = (0,\overline{9})_{16}$

Komplement-Zahldarstellungen

B-Komplementdarstellung (echtes Komplement):

$$z = \begin{cases} (Z)_B & z \geq 0 \text{ und } z < B^{n-1} \\ & \text{falls} \\ (B^n - |z|)_B & z < 0 \text{ und } |z| \leq B^{n-1} \end{cases}$$

(B-1)-Komplementdarstellung (unechtes Komplement):

$$z = \begin{cases} (z)_B & z \geq 0 \text{ und } z < B^{n-1} \\ & \text{falls} \\ (B^n - |z| - 1)_B & z < 0 \text{ und } |z| < B^{n-1} \end{cases}$$

Ganze Zahlen in **Zweier-Komplementdarstellung** zur Basis 2 mit n-Bitstellen

$$z = \begin{cases} (z)_2 & z \geq 0 \text{ und } z < 2^{n-1} \\ & \text{falls} \\ (2^n - |z|)_2 & z < 0 \text{ und } |z| \leq 2^{n-1} \text{, d.h. } z \geq -2^{n-1} \end{cases}$$

Ganze Zahlen im **Einerkomplement** (n-Bitdarstellung)

$$z = \begin{cases} (z)_2 & z \geq 0 \text{ und } z < 2^{n-1} \\ & \text{falls} \\ (2^n - |z| - 1)_2 & z < 0 \text{ und } |z| < 2^{n-1} \text{, d.h. } z > -2^{n-1} \end{cases}$$

Komplement-Zahldarstellungen

für Ganze Zahlen

B-komplement-Darstellung ganzer Zahlen (echtes komplement)

$$z'' = \begin{cases} z|_B & \text{falls } z \geq 0,\ z < B^{n-1} \\ (B^n - |z|)_B & \text{falls } z < 0,\ |z| \leq B^{n-1} \end{cases}$$

(B-1)-komplement-Darstellung ganzer Zahlen (unechtes komplement)

$$z' = \begin{cases} z|_B & \text{falls } z \geq 0,\ z < B^{n-1} \\ (B^n - |z| - 1)_B & \text{falls } z < 0,\ |z| < B^{n-1} \end{cases}$$

speziell für B=2:

Zweier-komplement

$$z'' = \begin{cases} z|_2 & \text{falls } z \geq 0,\ z < 2^{n-1} \\ (2^n - |z|)_2 & \text{falls } z < 0,\ |z| \leq 2^{n-1} \end{cases}$$

(NOT |z| + 1)

Einer-komplement

$$z' = \begin{cases} z|_2 & \text{falls } z \geq 0,\ z < 2^{n-1} \\ (2^n - |z| - 1)_2 & \text{falls } z < 0,\ |z| < 2^{n-1} \end{cases}$$

(NOT |z|)

Motivation:

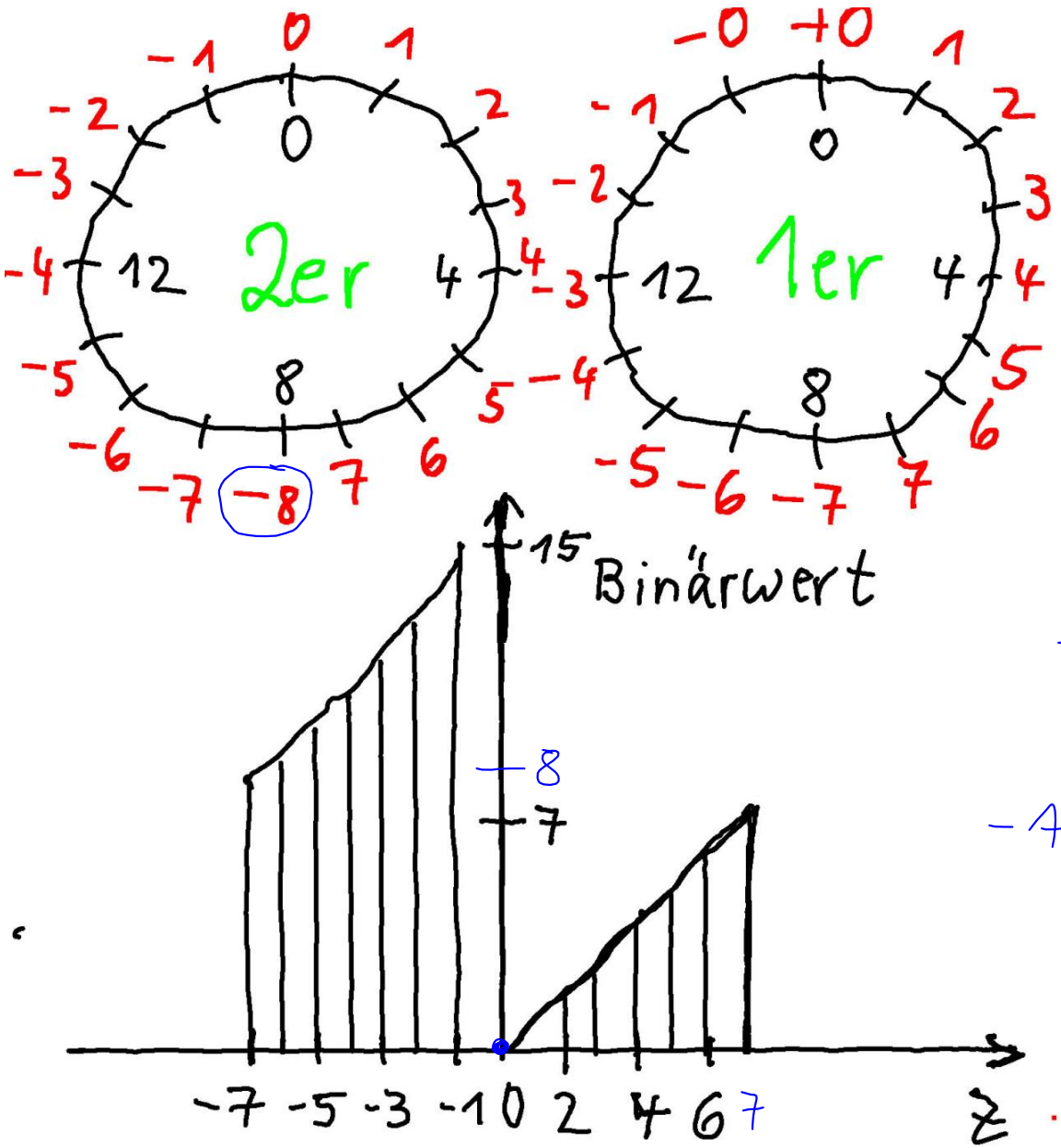
n = Anzahl von Bits

z.B. nur 8 Bit zur Speicherung einer ganzen Zahl → 256 ganze Z.

Subtraktion: $d = a - b + 2^n - 2^n$

$= a + (2^n - b) = a + (-b)$

Komplement



Tetraden

$n = 4$

↓

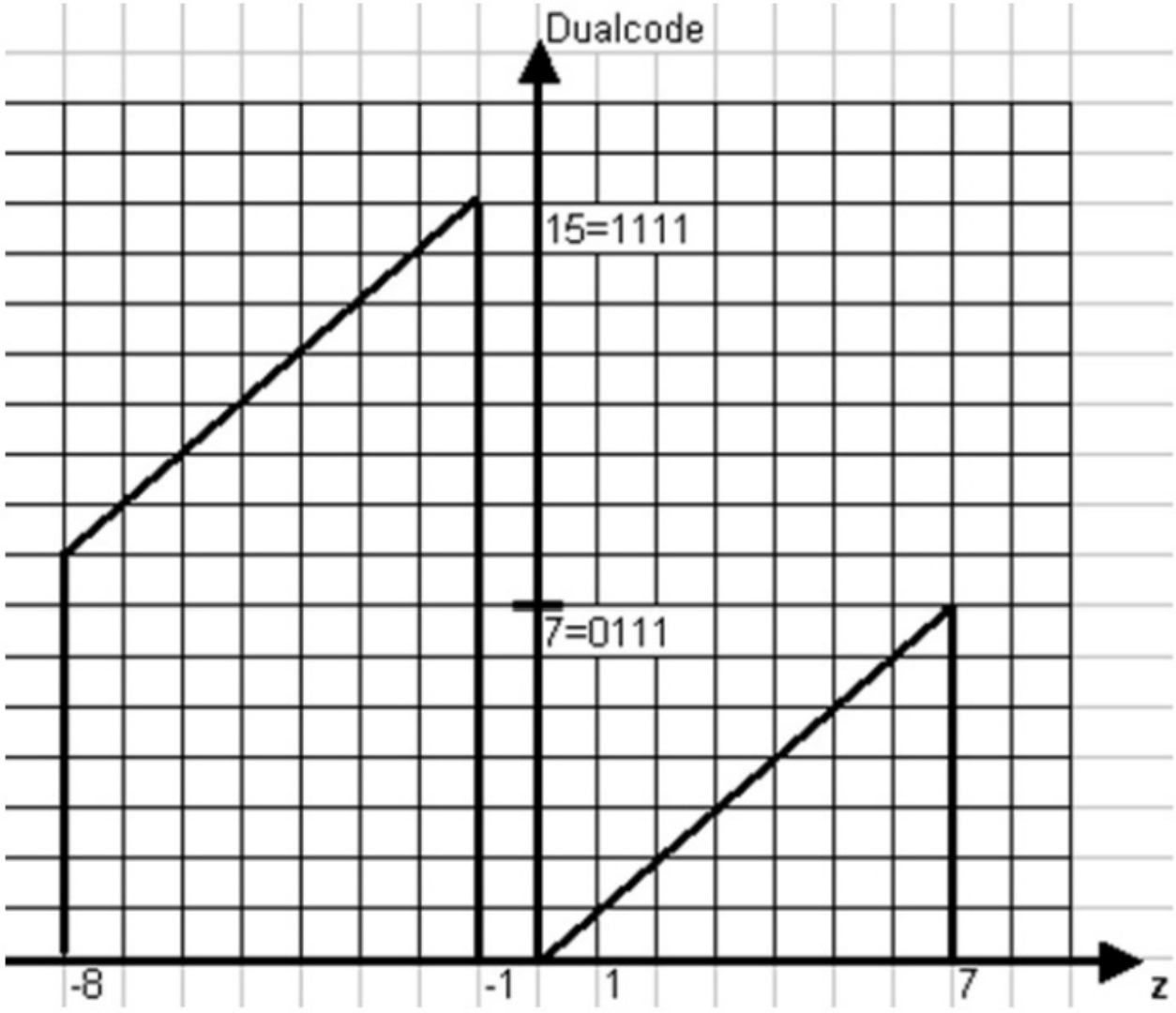
16 Zahlen

$-7 \ldots -1 \; {}^{+0}_{-0} \ldots 7$

$-1 \rightarrow 8$

n=4 Bits

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	-7
A	1010	-6
B	1011	-5
C	1100	-4
D	1101	-3
E	1110	-2
F	1111	-1



Bsp: $n = 3 \Rightarrow z \in \{-4 \ldots 0 \ldots 3\}$

$z = -1$

$z_{2er\text{-}kompl.} = 2^3 - |-1| = 7$

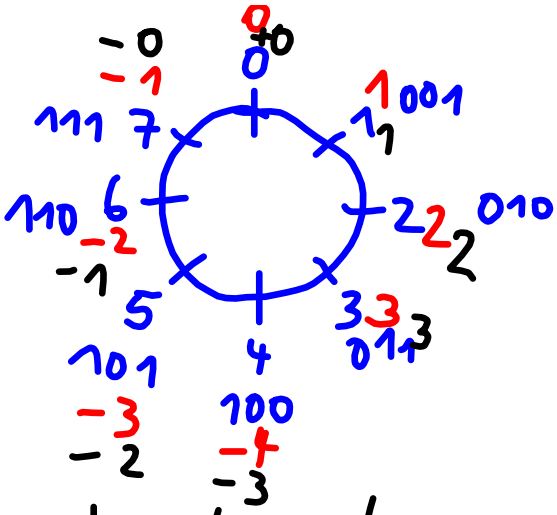
$z = -4$

$z_{2er\ kompl} = 2^3 - |-4| = 4$

$= NOT\ |z| + 1$

$= |-1001$

"011 + 001 = 100 = 4



Einerkomplement

$z = -3 \rightarrow z_{1er\text{-}k.} = NOT\ |-z|$

$= NOT\ |-011|$

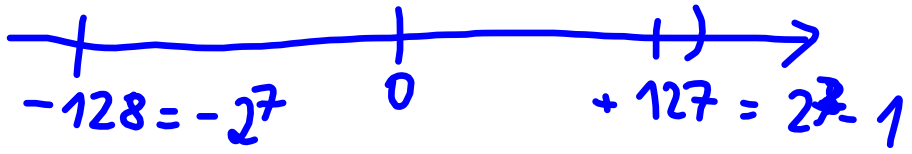
$= 100 = 4_{10}$

Zahlenbereichsüberschreitung / Overflow / Überläufe

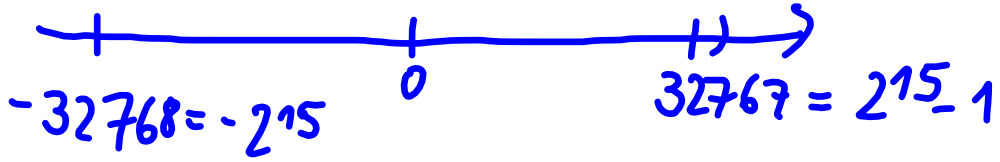
Add. $1 + 2 = \begin{array}{r} 001 \\ 010 \\ \hline 011 \end{array}$ $= 3$ aber $1 + 4 = \begin{array}{r} 001 \\ 100 \\ \hline 101 = 5 \end{array}$ $= -3$

$-2 - 3 = (-2) + (-3) = \begin{array}{r} 110 \\ +\ 101 \\ \hline \textcircled{1}\ 011 = 3 \end{array}$ $= +3$

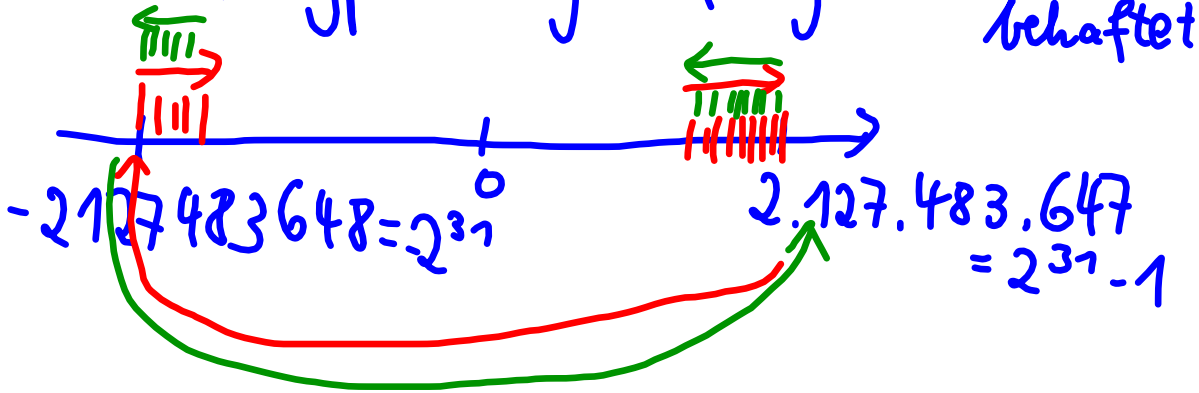
Bsp: $n = 8$ (Zweier-Komplement)



$n = 16$ Datentyp shortint ($65536 = 2^{16}$)



$n = 32$ (4 Bytes) Datentyp integer (signed = vorzeichenbehaftet)



Gleitkomma-Zahldarstellung

(nach IEEE 754 / 854)

Gleitkommazahl (Floating point) $z = \pm m \cdot B^e$ m = Mantisse (Genauigkeit), e = Exponent (Größenordnung)

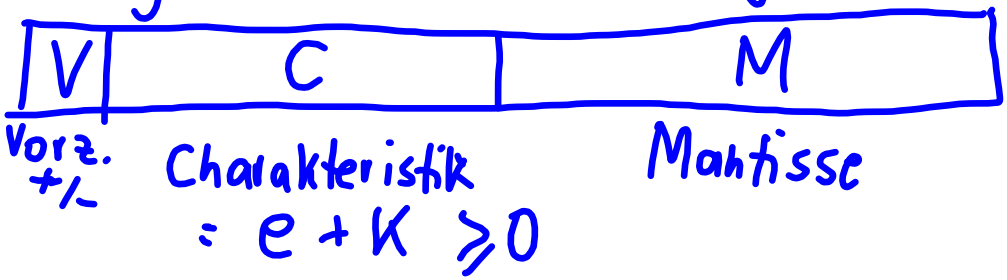
Festkommazahl $z = z_n \ldots z_0 , z_{-1} \ldots z_{-m}$

normierte Gleitkommazahl: mit $1 < |m| < B$ (außer Null)

→ B = 2 m = 1, ... (außer bei Null)

→ B = 10 m = 1, ..., 9, ...

allg. Zahldarstellung im HS (Codierung): VCM ≙ Bitanzahl



Gleitkomma-Zahldarstellung in FPU

(nach IEEE 754 / 854)

a) kurze Gleitkommazahlen SHORTREAL / Zahlen mit einfacher Genauigkeit
single precision

Gesamtlänge 4 Bytes = 32 Bits mit VCM = 1/8/23 Bits

mit $C \in \{0 \ldots 127\ 128 \ldots 255\}$

für $e \in \{-127 \ldots +128\}$, also $K = 127$

mit norm. Mantisse: $m_0, m_1 m_2 \ldots m_{23}$ $(m_0 = 1)$

Rundungsfehler $< 2^{-23} \sim \frac{1}{2} \cdot 2^{-23} = 2^{-24} \approx 2{,}4 \cdot 10^{-8}$ ← etwa in 8. Dez. Stelle

Datentyp in C: float ; in FORTRAN REAL*4

b) lange Gleitkommazahlen LONGREAL / Zahlen mit doppelter Genauigkeit
double precision

mit VCM = 1/11/52 also 8 Bytes = 64 Bits

$e \in \{-1023 \ldots +1024\}$

Char $\in \{0 \ldots 2047\} \Rightarrow K = 1023$,

Rundungsfehler $< 2^{-52} \approx 10^{-17}$ ← Fehler etwa in der 17. Dezimalstelle hinter dem Komma

Datentyp: in C double
in FORTRAN REAL*8

c) INTEL FPU (kein IEEE 754) mit 80 Bit-Register (10 Bytes)

VCM = 1/15/64 $\Rightarrow$ $k = 2^{15-1} - 1 = 16383$

Rundungsfehler $< 2^{-64} \sim 10^{-21}$

Datentyp in C: double extended

Bsp: $-38{,}625 = -100110{,}1\,0\,1 = -1{,}00110101 \cdot 2^5$ (0,5 | 0,25 | 0,125)

k = norm. Mantisse, e = charakteristik

charakteristik = $127 + 5 = 132 = 128 + 4 = 10000100$

1	10000100	00110101	00000000	00000000
	C2	1A	80	00

$= 0xC21A8000$

Gleitkommazahlen

Ausnahmen (exceptions)

(wird angezeigt durch Signalisierungsbit im Stausregister der FPU (floating point unit)
→ bewirken Behandlungsroutine

1) ungenaues Ergebnis → Rundung

2) Division durch Null d.h. x/0 mit x≠0
➔ Ergebnis = +∞

3) Unterlauf (underflow) → tiny numbers
(kleiner als kleinste darstellbare Zahl)

4) Überlauf (overflow)
Überschreiten des Zahlbereiches,
Ergebnis = +∞ bzw. Sonderzahl

5) unzulässige Operation/invalid operation
z.B. +∞-∞ oder 0* ∞ oder 0/0 oder ∞/∞
oder SQRT(-x)

Sonderzahlen

1) vorzeichenbehaftete Null (signed null) +0=-0

2) vorzeichenbehaftetes unendlich (signed infinity)

3) Nichtzahlen (not-a-numbers = NaN)

a) SNaN signaling NaNs
z.B. Anzeige nicht initialisierter Variablen
(anzeigende Nichtzahlen)

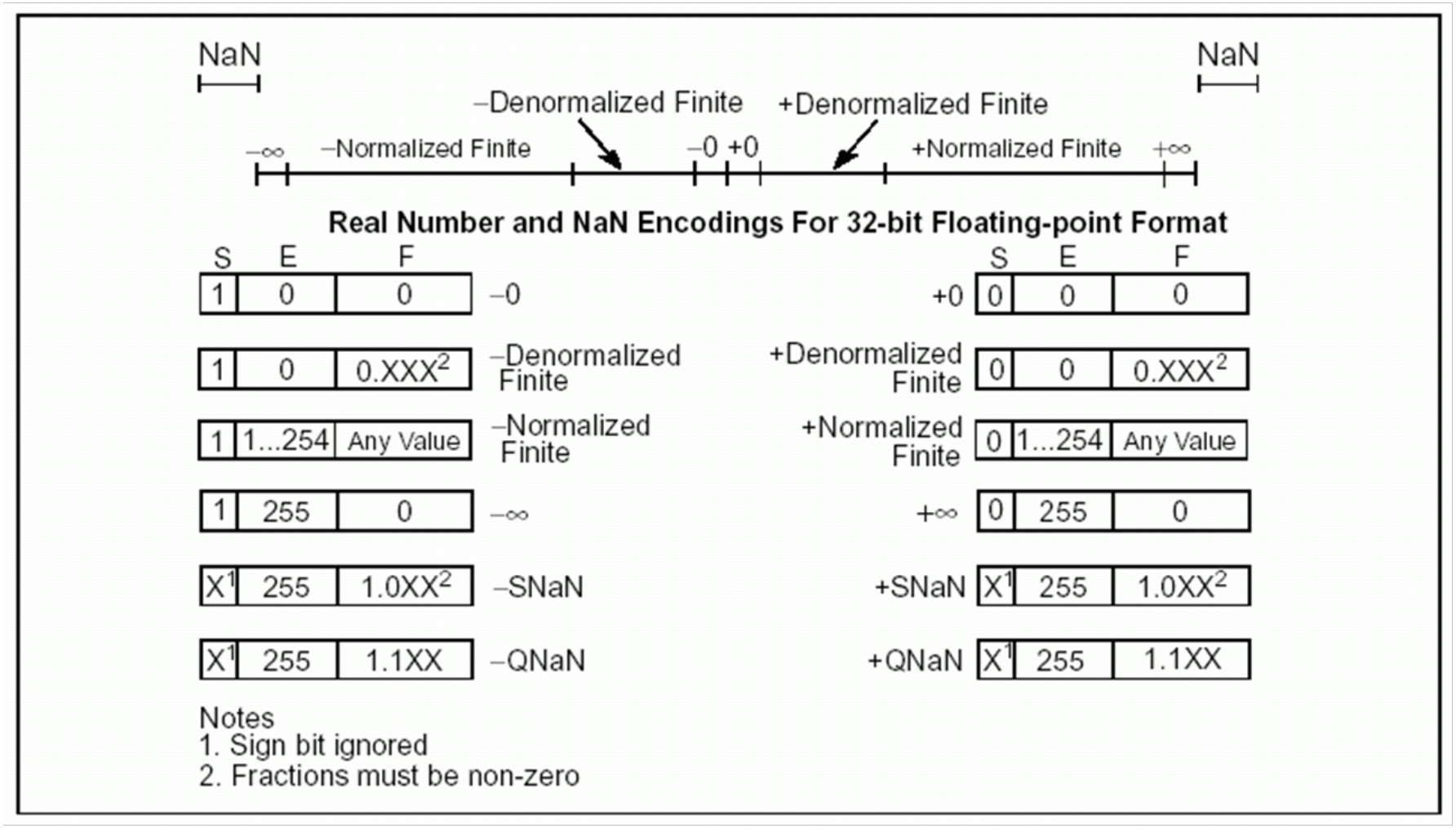
b) QNaN quiet NaNs bei ungültigen bzw.
nichtverfügbaren Operanden
(nichtanzeigende Nichtzahlen)

SPrec. DPrec.

Null: $z = 0{,}0*2^{0}$
unendlich: $z = \pm\ 0{,}0*2^{255} = \pm\ 0{,}0*2^{2047}$
normalisiert: $z = \pm\ 1{,}x*2^{e-127} = \pm\ 1{,}x*2^{e-1023}$
$(0<e<255)$ $(0<e<2047)$
unnormalisiert: $z = \pm\ 0{,}x*2^{e-126} = \pm\ 0{,}x\ 2^{e-1022}$

	Vorzeichen	Charakteristik	Mantisse
NaN	0	11.....11	11 ..... 11 ⋮ 00 01
pos./neg. normalisierte Zahlen	0/1	11 ..... 10 ⋮ 00 01	11 ..... 11 ⋮ 00 00
$+\infty$	0	11 11	00 00
± 0	0/1	00 00	00 00
$-\infty$	1	11 11	00 00
NaN	1	11.....11	SNaN 000 ..... 01 ⋮ QNaN 111 11

Gleitkommazahlen



alternative Zahldarstellungen

a) Zahlen als Strings/Zeichenkette

- Nullterminiert
  (String endet beim Zeichen 0x00 = \00)
- im ersten Byte wird eine Längenangabe L<256 codiert, danach folgen L-BCD-codierte Dezimalziffern

b) Bibliotheken für beliebig genaues Rechnen (bel. große Exponenten und Mantissen)

in PERL:

```
use Math:BigInt;  # lange Integerarithmetik
use Math:BigRat;  # Arithmetik mit langen rationalen Zahlen
```

c) Zahlen durch Intervalle ersetzen
(Intervallarithmetik zum Einschließen der Rechenergebnisse in Intervalle)

d) BCD-Darstellung / BCD-Arithmetik

BCD-Zahldarstellung (Binary Coded Decimals)

x87-BCD-Zahlenformat (10 Bytes)

Ziffernfolge - 18 BCD-codierte Dezimalziffern (9 Bytes)						1 Byte							
d_0 Einer-stelle	d_1 Zehner-stelle	d_2 Hunderter-stelle	d_3 Tausender-stelle	.....	d_{17} 10^{17}-stelle	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
						reserviert							Vorzeichen 0 = + 1 = -

BCD-Arithmetik:

18 Dez.ziffern

Tetraden

Bsp: 87 + 49 = 136

1000 0111

0100 1001

1101 0000

0110 0110 = 6

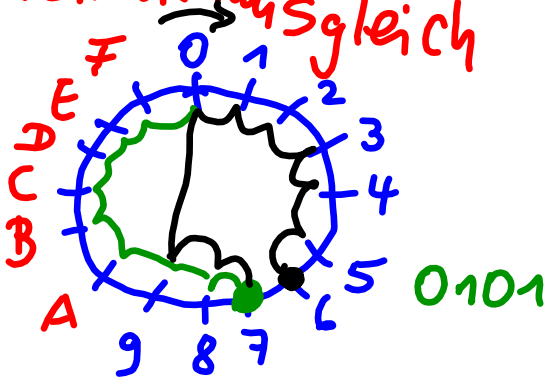
0011 0110

1 3 6

0000 ... 1001 1010 ... 1111 } keine Dez.ziffern Pseudotetraden

0 ... 9

Tetradenvergleich



Arithmetik

$1+1=2=10_2$

$1+1+1+1=4=100_2$

Addition im Dual-, Oktal- und Hexdezimalsystem

	Dez	Oktal	Hexadez.	Dual
Summand	$107,4$	357	$47,8$	$1011\,1,11$
+ Summand	$47,7$	$+\ 732$	$+\ 27,A$	$1\,110\,1,10$
	$^{1\,1}$	$^{1\,1}$	^{1}	$^{1\,1\,0\,1\,1\,1}$
Summe	$155,1$	1311	$6F,2$	$10\,1101,10$

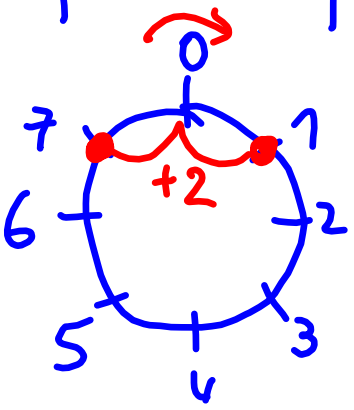
$11=1\cdot 8+3$

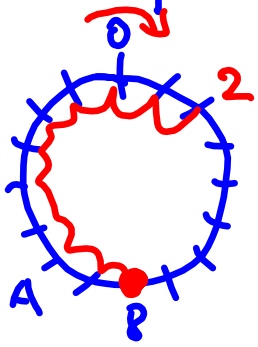
$7+2=9$

$=1\cdot 8+1$

$=11_8$

$8=10_8$





$8+A=18$

$=1\cdot 16+2$

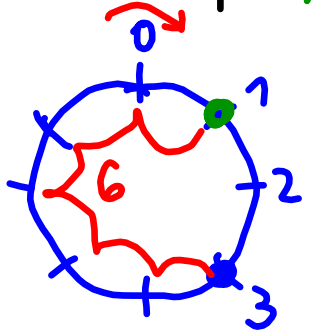
$=12H$

Arithmetik

Subtraktion im Dual-, Oktal- und Hexdezimalsystem

	Dez	Okt.	Hex.	Dual
Minuend	193	301	C1	2
- Subtrahend	- 57	- 103	- 43	!
Differenz	136	176	7E	

Borgen
3-7
13-7
Einer 1,
d.h. 10

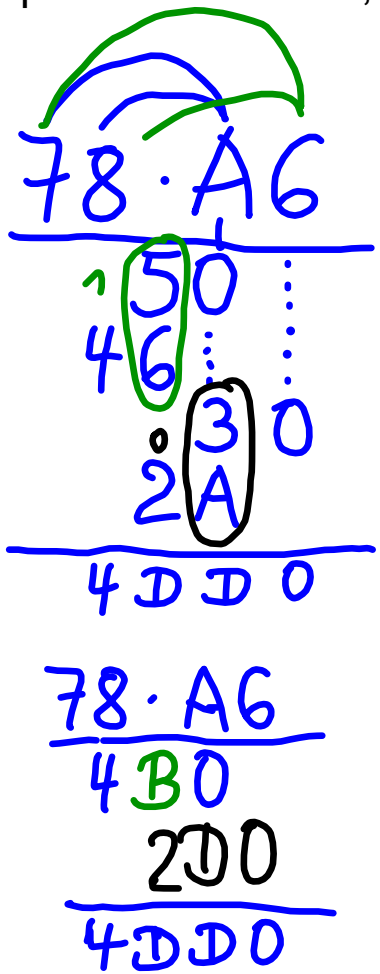


9.3 = 11_8 3 = 6

11H-3 = 17-3 = 14 = E
12-5 = 7
C
10_8-1=7

Arithmetik

Multiplikation im Dual-, Oktal- und Hexdezimalsystem



$A \cdot 8 = 80_{10}$
$= 5 \cdot 16 + 0$
$= 50H$
$A \cdot 7 = 70_{10}$
$= 4 \cdot 16 + 6$
$= 46H$
$6 \cdot 8 = 30H$
$6 \cdot 7 = 42 = 32 + 10$
$= 2A$

	0 1 … 8 9 A B C D E F
0	
1	
2	
⋮	30
8	(50)
9	
A	
B	
⋮	
F	

Duale kleine 1x1

	0	1
0	0	0
1	0	1