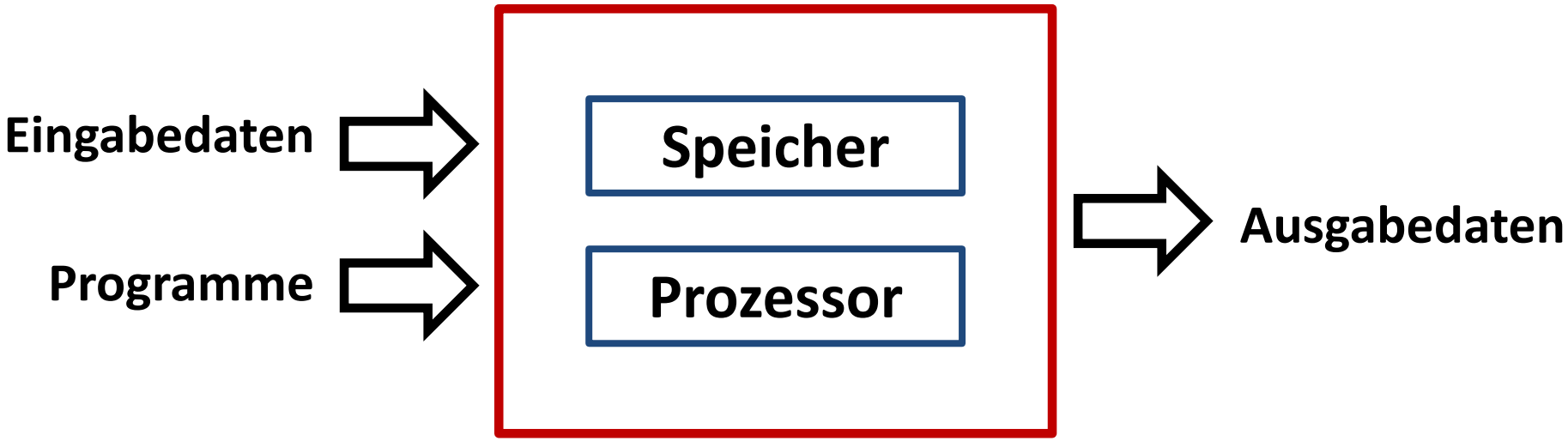
Kapitel 2

Zahlensysteme, Darstellung von Informationen

2. Zahlensysteme, Darstellung von Informationen

Ein Computer speichert und verarbeitet mehr oder weniger große Informationsmengen, je nach Anwendung und Leistungsfähigkeit.



Die Programmierung derartiger Rechner setzt Kenntnisse über die Darstellung der Informationen voraus!

Wegen der in Computern verwendeten Bauelemente werden alle Informationen binär (dual) dargestellt.

Informationselement	Realisierung
Schalter	offen / geschlossen
Festplatte	Magnetfeld „ein“ / „aus“
Speicher (RAM)	Spannung ein / aus
CD-ROM, DVD	Reflexion / keine Refl.

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Kapitel 2 – Zahlensysteme

Zahlen werden im täglichen Leben im **Dezimalsystem** dargestellt. Das gilt erst recht für ganze Zahlen. Das Dezimalsystem ist ein **Stellenwertsystem** (auch „**polyadisches Zahlensystem**“).

Beispiel:

$$1532 = 2\cdot 1 + 3\cdot 10 + 5\cdot 100 + 1\cdot 1000$$

$$= 2\cdot 10^0 + 3\cdot 10^1 + 5\cdot 10^2 + 1\cdot 10^3$$

$2 \cdot 10^0$

$+ 3 \cdot 10^1$

$+ 5 \cdot 10^2$

$+ 1 \cdot 10^3$

Basis = 10

Für die Darstellung ganzer Zahlen in Stellenwertsystemen gilt ganz allgemein (z. B. im Dezimalsystem mit der Basis b = 10):

$$X = a_0 \cdot b^0 + a_1 \cdot b^1 + a_2 \cdot b^2 + \ldots + a_n \cdot b^n$$

Oder kürzer:

$$X = \sum a_i \cdot b^i \qquad i = 0, 1, \ldots$$

$$a_i \in \{0, 1, 2, \ldots, b-1\}$$

Zur Erinnerung: $b^0 = 1$

2.1. Darstellung positiver ganzer Zahlen

Im Zusammenhang mit Digitalrechnern sind drei Stellenwert-systeme besonders relevant:

- **Das Dezimalsystem**
- **Das Dualsystem (Binärsystem)**
- **Das Hexadezimalsystem**

Die Bedeutungen der ersten beiden Systeme liegen auf der Hand. Das Hexadezimalsystem wird oft für eine verkürzte Darstellung von Binärzahlen genutzt, weil sich jeweils vier benachbarte Binärziffern zu einer einzelnen Hexadezimalziffer zusammenfassen lassen.

Mit der Darstellung von Binär- und Hexadezimalzahlen muss der Programmierer ebenso vertraut sein, wie mit der Umrechnung von Zahlen zwischen diesen Stellenwertsystemen.

Das **Dualsystem (Binärsystem)** besitzt folgende Eigenschaften:

Basis	2
Menge der Ziffern	{0, 1}
Stellenwerte	Potenzen von 2 2^0 2^1 2^2 2^3 2^4 … 1_{10} 2_{10} 4_{10} 8_{10} 16_{10} …

Beispiel: $1011_2 = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3$

$= 1_{10} + 2_{10} + 0_{10} + 8_{10}$

$= 11_{10}$

2.1. Darstellung positiver ganzer Zahlen

Da in nur einem Bit keine sinnvolle Informationsmenge gespeichert werden kann, arbeitet man mit Gruppen von mehreren/vielen Bits:

- Eine Gruppe von 8 Bits nennt man ein **Byte**
- Eine Gruppe von 2 Bytes nennt man ein **Wort (Word)**
- Eine Gruppe von 4 Bytes nennt man ein **Doppelwort (Longword)**
- 1024 Bytes nennt man ein **Kilobyte** (1 KB, mit großem „K“)
- 1024 Kilobytes nennt man ein **Megabyte**
- 1024 Megabyte nennt man ein **Gigabyte**
- 1024 Gigabyte nennt man ein **Terabyte**

- 1 KB = 2^{10} Bytes = 1 024 Bytes
- 1 MB = 2^{20} Bytes = 1 048 576 Bytes
- 1 GB = 2^{30} Bytes = 1 073 741 824 Bytes
- 1 TB = 2^{40} Bytes = 1 099 511 627 776 Bytes

2.1. Darstellung positiver ganzer Zahlen

Mit **8 Bits** (oder **einem Byte**), von denen jedes Bit zwei Zustände (1/0) annehmen kann, lassen sich **256 verschiedene Kombinationen** bilden:

```
0000 0000
0000 0001
0000 0010
0000 0011
0000 0100
0000 0101
0000 0110
   . . .
1111 1110
1111 1111
```

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Beispiele:

Der neue Office PC:
Office 5200 V.2
Miditower silber/schwarz
Intel DualCore E5200 (2,5GHz),
2048MB DDR2, 160GB SATA,
18x DVDRW MultiBrenner,
VGA/DVI, LAN /USB/Sound
- mit Parallelport
- neue Grafik: X4500 DVI/VGA
359,-

Der Business PC:
DATEC Business E7404:
Silent Miditower silber/schwarz
Intel Core2Duo E7400 (2,8GHz)
4GB DDR2, 320GB SATA HD,
DVDRW Brenner, CardReader
Intel X3500 VGA, analog+DVI
455,-

Monitor-Sondermodell:
Benq G2410HD
24" TFT, Full-HD,
1920*1080, 16:9,
Kontrast: 40.000:1,
2ms, 300cd/m²,
Analog / DVI,
glossy-black
199,-

Notebooks:
ASUS K50IJ-SX009C
15,6" 16:9 TFT (1366*768),
Intel T4200 CPU, 250GB HD
2GB DDR2, 8x DVD Brenner
Intel 4500M VGA, WLAN-N,
Gigabit-LAN, Webcam
MS Vista Home Premium
519,-

Beamer:
BenQ LED-Beamer Joybee GP1
DLP-Technologie mit LED Lampe
mit bis zu 20.000h Lebensdauer,
SVGA (858*600), 100ANSI,
2000:1, USB Reader,
nur 0,64kg!!
499,-

Multimedia:
Verbatim MediaStation HD DVR 500GB
Video aufnehmen, wiedergeben via LAN,
WLAN, USB, SD-Card, HDMI, etc.
MediaStation HD DVR Multimedia
Wireless Network Recorder 500 GB
NEU!
249,-

Das **Hexadezimalsystem** besitzt folgende Eigenschaften:

Basis	16
Menge der Ziffern	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F} A bis F besitzen die dezimalen Nennwerte zehn bis fünfzehn!
Stellenwerte	Potenzen von 16 16^0 16^1 16^2 16^3 … 1_{10} 16_{10} 256_{10} 4096_{10} …

Beispiel: $AFFE_{16} = 14 \cdot 16^0 + 15 \cdot 16^1 + 15 \cdot 16^2 + 10 \cdot 16^3$

$= 14_{10} + 240_{10} + 3840_{10} + 40960_{10}$

$= 45054_{10}$

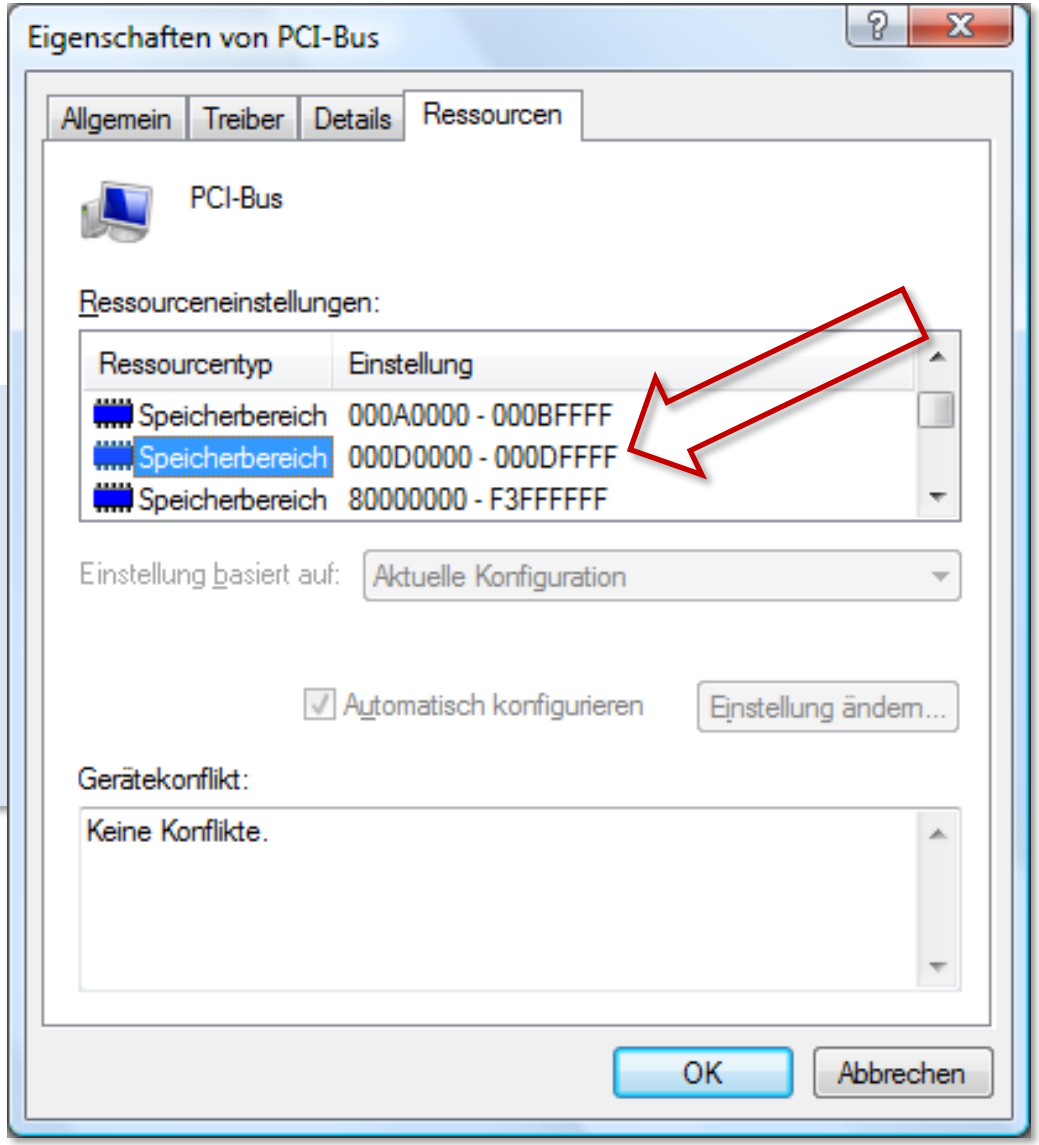
2.1. Darstellung positiver ganzer Zahlen

Beispiel:

Speicherbelegung am PCI-Bus bei Intel-PC, 32-Bit-Adresse

Windows Vista:

- **Rechtsklick auf „Computer“,**
- **Eigenschaften,**
- **Geräte-Manager,**
- **Systemgeräte,**
- **PCI-Bus**



$000D\ FFFF_{16}$ =

$0000\ 0000\ 0000\ 1101\ 1111\ 1111\ 1111\ 1111_2$

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Kapitel 2 – Zahlensysteme

2.2. Umrechnung zwischen Zahlensystemen

Zwischen den hier relevanten Stellenwertsystemen Dezimal, Binär und Hexadezimal gibt es **sechs Umrechnungsverfahren**, die auf den folgenden Folien vorgestellt werden:

a) Binär → Dezimal
b) Hexadezimal → Dezimal
c) Dezimal → Binär
d) Dezimal → Hexadezimal
e) Binär → Hexadezimal
f) Hexadezimal → Binär

2.2.1. Umrechnung Binär → Dezimal

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

<u>Methode</u>: Summe der Zweierpotenzen bilden, dabei am besten „von rechts“ beginnen

Beispiel 1: $10111_2 = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4$

$= 1 + 2 + 4 + 16$

$= 23_{10}$

Beispiel 2: $110011_2 = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5$

$= 1 + 2 + 16 + 32$

$= 51_{10}$

2.2.1. Umrechnung Binär → Dezimal

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Hilfstabelle: Zweierpotenzen

n	2^n	n	2^n
0	1	11	2 048
1	2	12	4 096
2	4	13	8 192
3	8	14	16 384
4	16	15	32 768
5	32	16	65 536
6	64	17	131 072
7	128	18	262 144
8	256	20	1 048 576
9	512	24	16 777 216
10	1 024	32	4 294 967 296

2.2.2. Umrechnung Hexadezimal → Dezimal

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

<u>Methode</u>: Summe der Sechzehnerpotenzen bilden, dabei am besten „von rechts“ beginnen

Beispiel 1: $1E3_{16} = 3 \cdot 16^0 + 14 \cdot 16^1 + 1 \cdot 16^2$

$= 3 + 224 + 256$

$= 483_{10}$

Beispiel 2: $FFFF_{16} = 15 \cdot 16^0 + 15 \cdot 16^1 + 15 \cdot 16^2 + 15 \cdot 16^3$

$= 15 + 240 + 3840 + 61440$

$= 65535_{10}$

2.2.2. Umrechnung Hexadezimal → Dezimal

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Hilfstabelle: Sechzehnerpotenzen

n	16^n
0	1
1	16
2	256
3	4 096
4	65 536
5	1 048 576
6	16 777 216
7	268 435 456
8	4 294 967 296

2.2.3. Umrechnung Dezimal → Binär

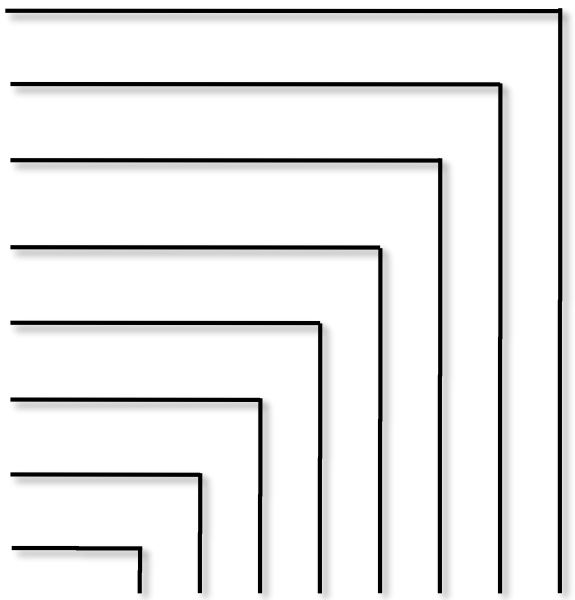
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Methode: Fortgesetzte Division durch zwei

Die umzuwandelnde Dezimalzahl wird fortlaufend durch zwei dividiert, bis null erreicht wird. Die dabei auftretenden Divisionsreste – in umgekehrter Reihenfolge – ergeben die gesuchte Binärzahl.

Beispiel:

223	: 2	=	111	Rest 1
111	: 2	=	55	Rest 1
55	: 2	=	27	Rest 1
27	: 2	=	13	Rest 1
13	: 2	=	6	Rest 1
6	: 2	=	3	Rest 0
3	: 2	=	1	Rest 1
1	: 2	=	0	Rest 1



Die entsprechende Binärzahl lautet: 1 1 0 1 1 1 1 1

2.2.4. Umrechnung Dezimal → Hexadezimal

Methode: Fortgesetzte Division durch 16

Die Dezimalzahl wird fortlaufend durch 16 dividiert, bis null erreicht wird. Die dabei auftretenden Divisionsreste – in umgekehrter Reihenfolge – ergeben die gesuchte Hexadezimalzahl.

Beispiel:

```
443 : 16 =  27   Rest 11
 27 : 16 =   1   Rest 11
  1 : 16 =   0   Rest 1
```

Die entsprechende Hexadezimalzahl lautet: 1 B B

Hilfstabelle: Ziffern A…F im Hexadezimalsystem

A	B	C	D	E	F
10	11	12	13	14	15

2.2.4. Umrechnung Dezimal → Hexadezimal

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Hilfstabelle: Vielfache von 16

n	n · 16
1	16
2	32
3	48
4	64
5	80
6	96
7	112
8	128

n	n · 16
9	144
10	160
11	176
12	192
13	208
14	224
15	240
16	256

2.2.5. Umrechnung Binär→ Hexadezimal

Methode: 4er-Gruppen von Binärzahlen bilden

Die umzuwandelnde Binärzahl wird von rechts nach links in 4er-Bündel von Binärziffern gruppiert. Anschließend werden die Bündel in die entsprechenden Hexadezimalziffern umgewandelt.

Beispiel:

1010	1111	1111	1110
A	F	F	E

Hilfstabelle: 4er-Bündel von Binärziffern und Hexadezimalziffern

0000	0001	0010	0011	0100	0101	0110	0111
0	1	2	3	4	5	6	7
1000	1001	1010	1011	1100	1101	1110	1111
8	9	A	B	C	D	E	F

2.2.6. Umrechnung Hexadezimal → Binär

Methode: Hexadezimal in 4er-Bündel von Binärziffern „auflösen“

Die umzuwandelnde Hexadezimalzahl wird Ziffer für Ziffer in 4er-Bündel von Binärziffern „aufgelöst“.

Beispiel:

3	7	A	F
0011	0111	1010	1111

Führende Nullen zu Beginn der Binärzahl können weggelassen werden:

$$37AF_{16} = 11\ 0111\ 1010\ 1111_2$$

Kapitel 2 – Zahlensysteme

Addition im Dezimal- und im Binärsystem

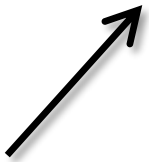
- Was muss man wissen, um eine Addition im Dezimalsystem durführen zu können?
- Welche Vorteile bietet das Binärsystem gegenüber dem Dezimalsystem bei der Durchführung von Berechnungen?

Addition im Dezimalsystem:

```
    1.234.567
+   2.345.678
      11 11
-------------
    3.580.245
```

Addition im Binärsystem:

```
  1100 0101 1100
+    1001 1011
       1 1
-----------------
  1100 1111 0111
```



Kontrolle: $3164_{10} + 155_{10} = 3319_{10}$ ✓

2.3. Rechnen im Dualsystem

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Additionstafel für Dezimalzahlen:

+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1		2	3	4	5	6	7	8	9	10
2			4	5	6	7	8	9	10	11
3				6	7	8	9	10	11	12
4					8	9	10	11	12	13
5						10	11	12	13	14
6							12	13	14	15
7								14	15	16
8									16	17
9										18

55 Regeln!

Dualzahlen:

+	0	1
0	0	1
1		10

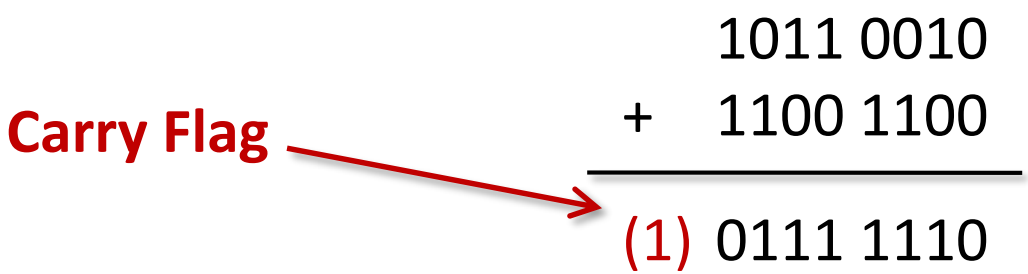
Nur 3 Regeln!

Überlauf und „Carry Flag“

Bei der Addition kann es zum **Überlauf** kommen, falls für das Ergebnis ein begrenzter Speicherplatz zur Verfügung steht!

Der Prozessor merkt sich dieses Ereignis durch Setzen eines speziellen Status-Bits („**Carry Flag**“). Dieses Bit kann im weiteren Programmverlauf abgefragt werden, falls auf den Überlauf reagiert werden muss.

Beispiel: **Addition von zwei 8-Bit-Zahlen, für das Ergebnis steht ebenfalls ein Speicherplatz von 8 Bit zur Verfügung**



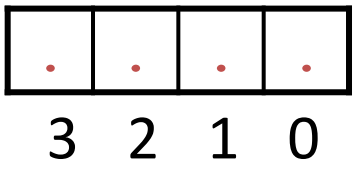
HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Kapitel 2 – Zahlensysteme

2.4. Darstellung negativer ganzer Zahlen

Mit den verfügbaren Bits (zum Beispiel einem Register im Prozessor) müssen neben dem **Betrag der Zahl** auch das **Vorzeichen** abgespeichert werden.

- Annahme: 4-Bit-Register



Erster Ansatz:

- Bit 3 ist das Vorzeichen-Bit: Bit 3 = 0 entspricht **+**
  Bit 3 = 1 entspricht **–**
- Bit 0 - Bit 2 speichern den Betrag der Zahl
- Beispiel: +2 | 0 | 0 | 1 | 0 |
  -2 | 1 | 0 | 1 | 0 |

Welche Nachteile besitzt dieses Verfahren?

2.4. Darstellung negativer ganzer Zahlen

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Zweiter Ansatz:

Dezimalzahl	2er-Komplement	Dezimalzahl	2er-Komplement
0	0 0 0 0		
1	0 0 0 1	-1	1 1 1 1
2	0 0 1 0	-2	1 1 1 0
3	0 0 1 1	-3	1 1 0 1
4	0 1 0 0	-4	1 1 0 0
5	0 1 0 1	-5	1 0 1 1
6	0 1 1 0	-6	1 0 1 0
7	0 1 1 1	-7	1 0 0 1
		-8	1 0 0 0

Darstellbarer Zahlenbereich beim sog. **„2er-Komplement“**: -8 ... +7

Fast alle Prozessoren verwenden diese Darstellung – mit mehr als 4 Bits!

Eigenschaften der 2er-Komplementdarstellung

a) Positive Zahlen: höchstwertiges Bit = **0** hier Bit 3 (typisch Bit 7, 15...)
   Negative Zahlen: höchstwertiges Bit = **1**

b) Wie wird für eine positive Zahl die Darstellung der zugehörigen negativen Zahl berechnet? **Antwort: 2er-Komplement**
   1.) Bilde bitweises Komplement (0→1 und 1→0) – „Einerkomplement“
   2.) Addiere anschließend 1 zum Einerkomplement

c) Umkehrung: Wie wird aus einer negativen Zahl die zugehörige positive Zahl berechnet? **Antwort: Ebenfalls mit dem 2er-Komplement!**

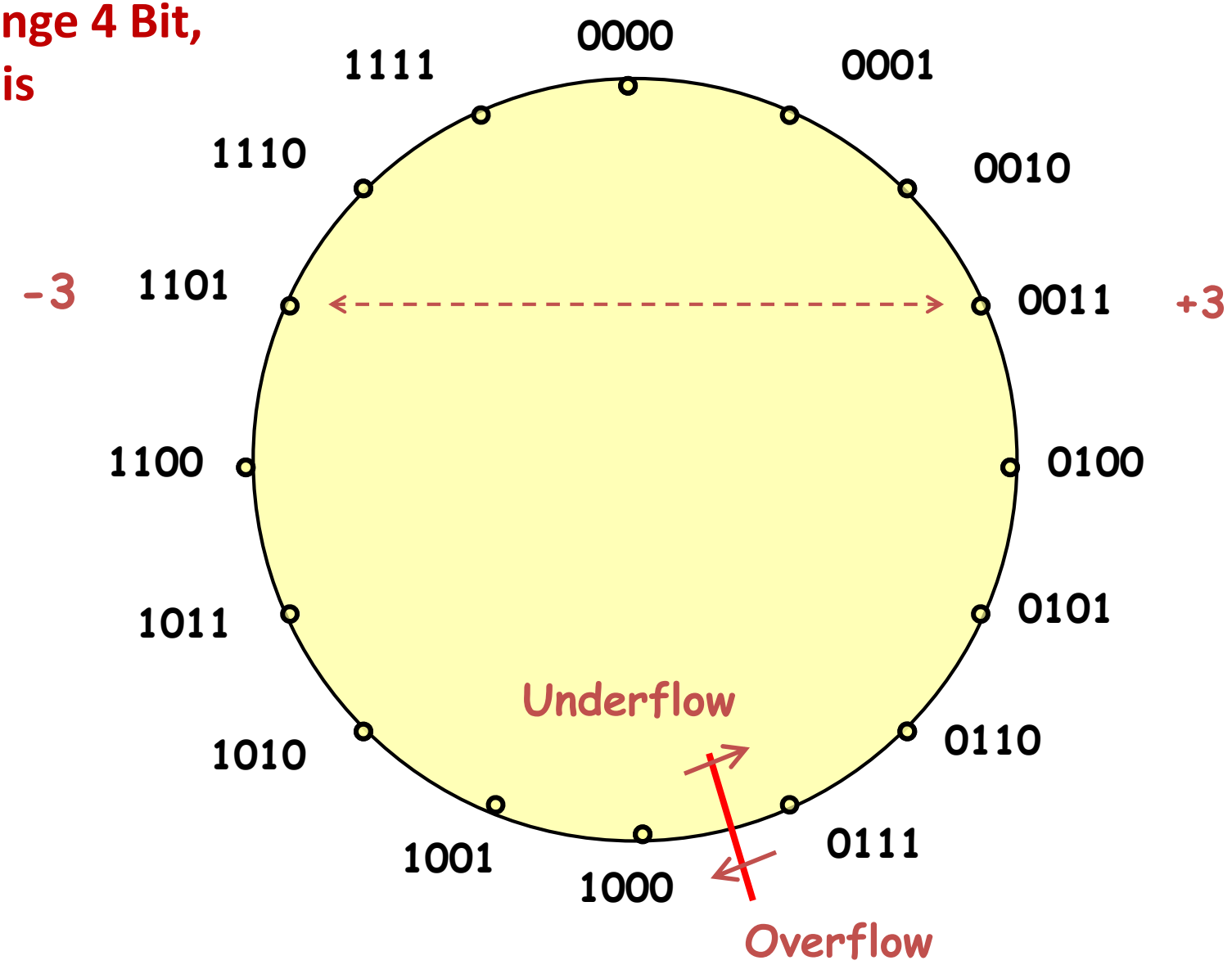
d) Bekannte Rechenregeln aus dem Dualsystem bleiben erhalten

e) z + (-z) = 0 – Übertrag (Carry Flag) wird ignoriert

f) Subtraktion wird auf Addition zurückgeführt – siehe d)

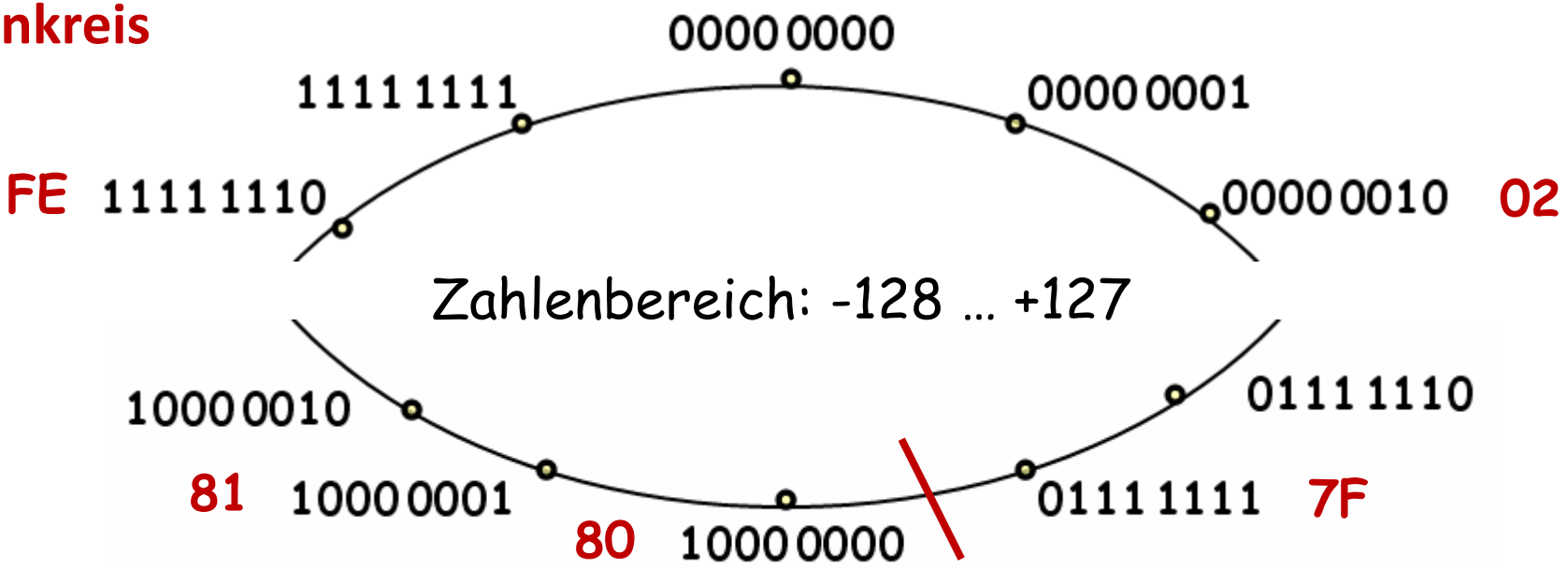
g) Mögliche Bereichsüberschreitung beachten!

Registerlänge 4 Bit, Zahlenkreis



HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

Registerlänge 8 Bit, Zahlenkreis



Allgemein (für eine Registerlänge von n Bit): $-2^{n-1} \ldots +2^{n-1} - 1$

n = 16:	-32 768	...	+32 767
n = 32:	-2 147 483 648	...	+2 147 483 647
n = 64:	$-9{,}22 * 10^{18}$	...	$+9{,}22 * 10^{18}$

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

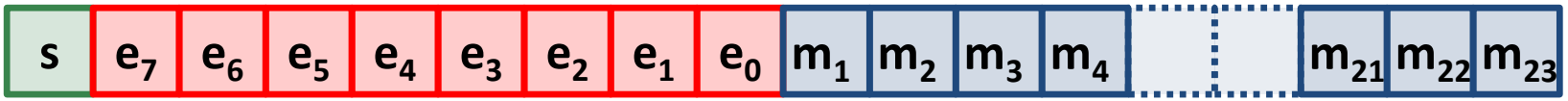
Kapitel 2 – Zahlensysteme

2.5. Darstellung gebrochener Zahlen

Zahlen mit Nachkommastellen werden in einem Rechner völlig anders dargestellt als ganze Zahlen und auch völlig anders verarbeitet! Die Zahlen werden zuerst in eine Standardform (halblogarithmisch) gebracht:

$$37{,}625 = +0{,}37625 \times 10^{2} = s \times m \times B^{ex}$$
$$(+100101{,}101)_2 = (+1{,}00101101 \times 2^{101})_2$$

Im Register werden **Vorzeichen**, **Mantisse** und **Exponent** abgespeichert. Die Basis ist stets 2 und wird deshalb nicht gespeichert.



Vorzeichen: Es ist nur ein Bit notwendig, 0 ↔ positiv und 1 ↔ negativ. Die verbleibenden Bits im Register müssen sich **Mantisse** und **Exponent** teilen. Ist die Mantisse zu lang, dann wird sie abgeschnitten.

Bei Zahlen kleiner als 1 möchte man negative Exponenten vermeiden.

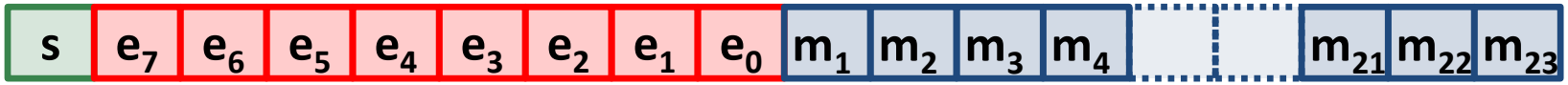
$$(+0{,}375)_{10} = (+0{,}011)_2 = (+1{,}1 \times 2^{-10})_2$$

Deshalb addiert man zum Exponenten stets eine feste Zahl („bias“).

2.5. Darstellung gebrochener Zahlen

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

$(+100101{,}101)_2 = (+1{,}00101101 \times 2^{101})_2$



32 Bit:

Exponent 8 Bits (bias = 127)
und **Mantisse 23 Bits**

101
\+ 0111 1111 (bias)
1000 0100

64 Bit: **Exponent 11 Bits** (bias = 1023) und **Mantisse 52 Bits**

Konsequenzen der Gleitkommadarstellung:

\+ großer Zahlenbereich darstellbar (sehr kleine/große Zahlen)

\- Zahlen werden meist nicht exakt dargestellt, da die Mantisse abgeschnitten wird (bei 32 Bit werden nur die ersten 7 führenden Dezimalstellen gespeichert, bei 64 Bit die ersten 15) → **Rundungsfehler!**

\- Rechenoperationen dauern länger als Operationen mit ganzen Zahlen.

2.5. Darstellung gebrochener Zahlen

Beim **Programmieren** wird durch den **Typ einer Variablen** festgelegt, ob eine Zahl als Gleitkommazahl, als ganze Zahl mit VZ oder als ganze Zahl ohne VZ gespeichert wird. Beispiele in der Programmiersprache C:

- Ganze Zahlen mit Vorzeichen

```
short n;          16-Bit  (Bereich:  -32768 … +32767)
int i;            32-Bit
long long j;      64-Bit
```

- Ganze Zahlen ohne Vorzeichen

```
unsigned short u;   16-Bit  (Bereich:  0 … +65535)
```

- Gleitkommazahlen

```
float x;          32-Bit  („einfache Genauigkeit“)
double y;         64-Bit  („doppelte Genauigkeit“)
```

Vergleichbare Variablentypen gibt es auch in anderen Programmier-sprachen wie Java oder Visual Basic.

Kapitel 2 – Zahlensysteme

2.6. Übungsaufgaben

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN · FH MÜNCHEN

1. Welchen Dezimalzahlen entsprechen die folgenden vier Zahlen
   10101100_2 $E3_{16}$ $F7_{16}$ 38_{16}
   wobei es sich jeweils um 2er-Komplementdarstellungen mit 8 Bit handelt? ($E3_{16}$ ist die Hexdarstellung einer Binärzahl: $E3_{16} = 1110\ 0011_2$)

2. Wie lautet die 2er-Komplementarstellung (dual, hex) der Dezimalzahlen -13 und -43 , wenn 8- bzw. 16-Bit-Register verwendet werden?

3. Eine Gleitkommazahl wird in der 32-Bit-Darstellung abgespeichert. Wie groß ist die betragsmäßig größte bzw. kleinste Gleitkommazahl, die dargestellt werden kann? (Alle Zahlen, die kleiner sind, werden als 0.0 gespeichert.)

4. Wie viele führende Dezimalstellen werden maximal abgespeichert? Überlegen Sie sich hierzu, wie die Zahl 1.0 abgespeichert wird. Wie lautet die kleinste Zahl größer als 1.0, die noch gespeichert werden kann? Hieraus ergibt sich die Anzahl der führenden Stellen, die mit 32 Bit gespeichert werden können.