



Vorlesung „Grundlagen der Informatik und Programmierung 1“

DATENTYPEN UND IHRE DARSTELLUNG

Negative ganze Zahlen, Bits und Bytes

Prof. Dr. Tom Vierjahn

Visual Computing (<https://vc.w-hs.de>)

Fachbereich Wirtschaft und Informationstechnik – Campus Bocholt



Wintersemester 2020/21



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

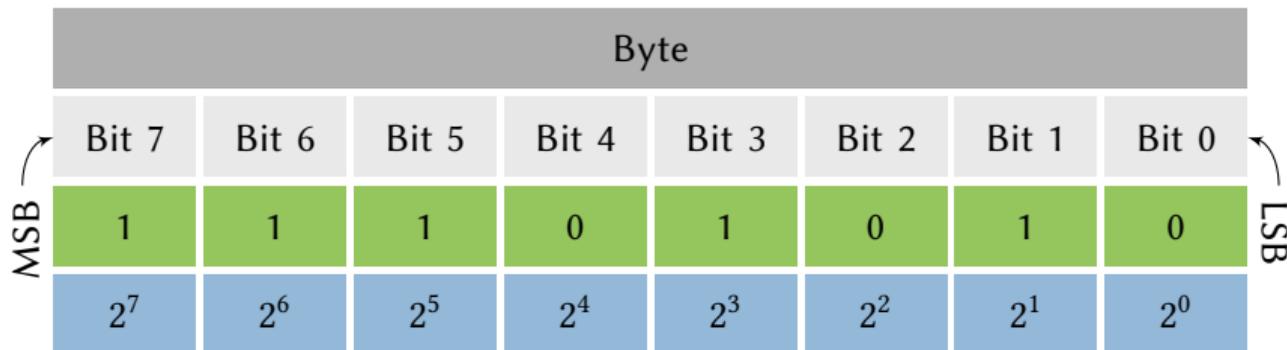
Definition: Bit

Die kleinste Informationseinheit auf einem Digitalrechner wird als **Bit** (Binary digit) bezeichnet. Ein Bit kann die Werte 0 (gelöscht) und 1 (gesetzt) annehmen.

Definition: Byte

8 Bits bilden ein **Byte**. Als positive Dualzahl interpretiert kann ein Byte Zahlen von 0 bis 255 (0x00 – 0xff) darstellen.

Example: Dezimalzahl 234



Weitere Größenordnungen

gemessen in Bytes

Wie in der Physik gibt es **Präfixe** für die Maßeinheiten Bit (bit) und Byte (B):

- ▶ Physik: Faktor $10^3 = 1000$
- ▶ Informatik: Faktor $2^{10} = 1024$

Physik (SI ¹ -Präfixe)			Informatik (Binärpräfixe)			rel. Abweichung
kilo	k	10^3	kibi	Ki	2^{10}	2,40 %
mega	M	10^6	mebi	Mi	2^{20}	4,86 %
giga	G	10^9	gibi	Gi	2^{30}	7,37 %
tera	T	10^{12}	tebi	Ti	2^{40}	9,95 %
peta	P	10^{15}	pebi	Pi	2^{50}	12,59 %
exa	E	10^{18}	exbi	Ei	2^{60}	15,29 %

Oft verwendet man in der Informatik die SI-Präfixe,
meint aber die Faktoren der Binärpräfixe.

¹ SI = Système international d'unités

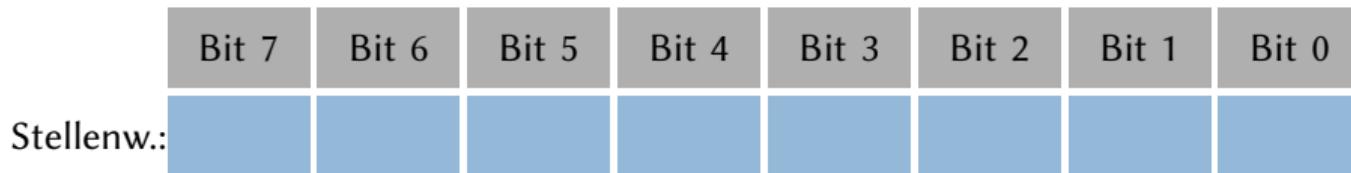
Negative Zahlen

Ganze Zahlen im Computer verhalten sich wie ein Zählwerk mit fester Stellenzahl.

Bei 8 Bit ohne Vorzeichen:

▶ $255_{10} + 1_{10} = 11111111_2 + 00000001_2 =$

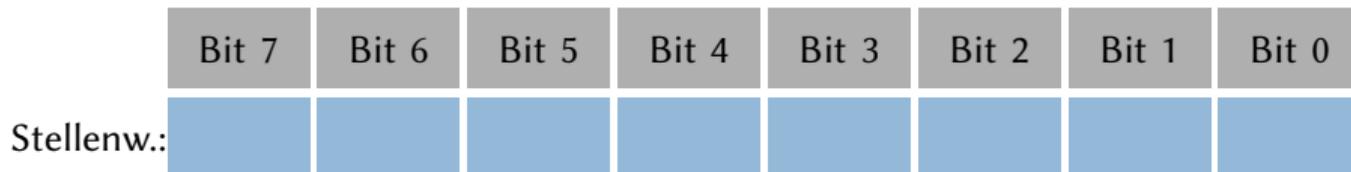
▶ $0_{10} - 1_{10} = 00000000_2 - 00000001_2 =$



Konsequenterweise, bei 8 Bit mit Vorzeichen:

▶ $0_{10} - 1_{10} = 00000000_2 - 00000001_2 =$

▶ $-1_{10} + 1_{10} = 11111111_2 + 00000001_2 =$



Zweier-Komplement

ermöglicht negative ganze Zahlen

Konvertierung von positiver Zahl zu negativer Zahl (und umgekehrt):

- ▶ schalte alle Bits um
- ▶ addiere 1

Datentyp	Größe (Byte)			
	mindestens	macOS apple-clang	CentOS gcc 4.8.5	Windows VS 2017
char	1	1	1	1
short	2	2	2	2
int	2	4	4	4
long	4	8	8	4
long long	8	8	8	8

Weiterhin muss der kleinere Datentyp in den größeren hineinpassen.

		Wertebereich	
		signed	unsigned
1 Byte	8 Bit	$-128 \dots 127$	$0 \dots 255$
2 Bytes	16 Bit	$-32\,768 \dots 32\,767$	$0 \dots 65\,534$
4 Bytes	32 Bit	$-2\,147\,483\,648 \dots 2\,147\,483\,647$	$0 \dots 4\,294\,967\,295$
8 Bytes	64 Bit	$-9,22337 \cdot 10^{18} \dots 9,22337 \cdot 10^{18}$	$0 \dots 1,84467 \cdot 10^{19}$
n Bytes	$n \cdot 8$ Bit	$-2^{n-1} \dots 2^{n-1} - 1$	$0 \dots 2^n - 1$

Das dicke oder das dünne Ende zuerst?

Zahlen liegen als Bitmuster codiert im Speicher:

$$1337_{10} = \underline{00000000} \underline{00000000} \underline{00000101} \underline{00111001}_2$$

Big Endian (z.B. Atmel AVR32)

- ▶ Das dicke Ende zuerst.
- ▶ Im Speicher:

Little Endian (z. B. Intel x86)

- ▶ Das dünne Ende zuerst.
- ▶ Im Speicher:

- ▶ Bits, Bytes, Präfixe
- ▶ Negative Zahlen, Zweierkomplement
- ▶ Speicherbedarf, Wertebereich
- ▶ Endianess

Prof. Dr. Tom Vierjahn

▶  tom.vierjahn@w-hs.de

Visual Computing

▶  <https://vc.w-hs.de>

▶  VisualComputingWH

▶  Visual Computing WH

▶  @VisComputingWH

Westfälische Hochschule

Fachbereich Wirtschaft und Informationstechnik

Campus Bocholt



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

WS 2020/21 – GIP1 (V): Datentypen und ihre Darstellung – Negative ganze Zahlen, Bits und Bytes –  – V 17.11.2020,

14:41 – Seite 10