Leibniz Universität IT Services

Leibniz Universität Hannover

C++
Binäre Zahlen

Zahlensysteme

- Das Dezimalsystem basiert auf der Zahl 10. Es werden die Ziffern 0 bis 9 genutzt.
- Das Binärsystem nutzt die Zahlen 0 und 1. Das System basiert auf der Zahl 2.
- Das Hexadezimalsystem basiert auf der Zahl 16. Es werden die Ziffern 0 bis 9 und die Buchstaben A bis F genutzt. Zahlen im Hexadezimalsystem werden durch 0x gekennzeichnet. Farbcodierungen werden häufig im Hexadezimalsystem dargestellt.

Darstellung der Zahlen 0 … 15

Deziml	Binär	Hexadezimal
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	9
10	1010	A

Darstellung der Zahlen 10 … 15

Deziml	Binär	Hexadezimal
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Binäre Zahlen

- Nutzung der Ziffern 0 und 1.
- Darstellung der Zustände An und Aus.

Umrechnung von Binär- in Dezimalzahlen

1		1		0		1
$1 * 2^3$	+	$1 * 2^2$	+	$0 * 2^1$	+	$1 * 2^0$
8	+	4	+	0	+	1
13						

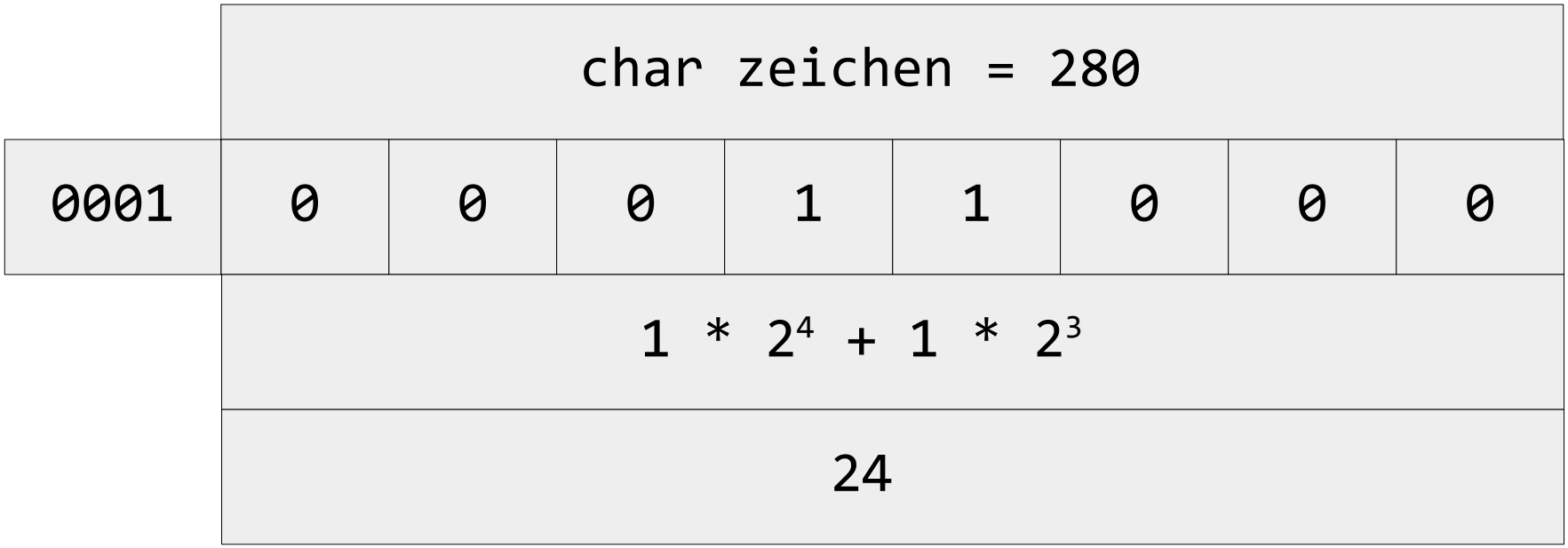
Nutzung in der EDV

- Ein Bit hat den Wert 0 oder 1. Ein Bit ist entweder an oder aus.
- 8 Bits werden zu einem Byte zusammengefasst. Mit einem Byte kann man maximal 256 Zustände darstellen.

Speicherung des Datentyps char

char zeichen = 'A'							
char zeichen = 65							
0	1	0	0	0	0	0	1

Überschreitung des Wertebereichs

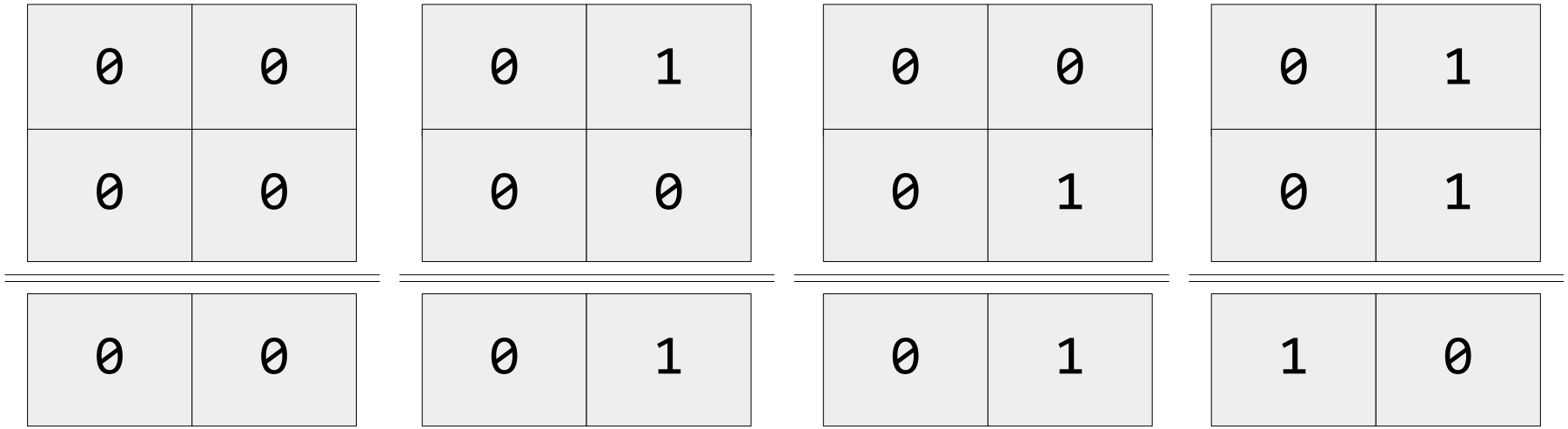


Addition

```
int ergebnis = 3 + 5;
```

3	0	0	0	0	0	0	1	1
+								
5	0	0	0	0	0	1	0	1
=								
8	0	0	0	0	1	0	0	0

Bitweise Addition



Überlauf

255	1	1	1	1	1	1	1	1
+								
1	0	0	0	0	0	0	0	1
=								
0	0	0	0	0	0	0	0	0

Erläuterung

- Mit Hilfe der Funktion `sizeof()` kann die Größe des Speicherplatzes eines Datentyps in Bytes ermittelt werden.
- In Abhängigkeit der Größe des Speicherplatzes kann die Variable einen Wert in einem bestimmten Wertebereich zugewiesen bekommen. Falls Ganzzahlen vorzeichenbehaftet sind, wird ein Bit für die Darstellung des Vorzeichens benötigt.
- Bei einem Überschreiten der oberen Grenze des Wertebereichs, wird der erste mögliche Wert anggezeigt.
- In diesem Beispiel haben Ganzzahlen einen Wertebereich von 1 Byte. Der Übertrag kann nicht gespeichert werden. Der Übertrag wird nicht berücksichtigt.

Subtraktion

```
int ergebnis = 5 - 3;
```

- Bildung des Einerkomplements vom Subtrahend (im Beispiel 3).
- Bildung des Zweierkomplements von dem vom Subtrahend: Einerkomplement + 1
- Addition des Zweierkomplements des Subtrahenden mit dem Minuend (in diesem Beispiel 5) .

Einerkomplement

3

0	0	0	0	0	0	1	1
1	1	1	1	1	1	0	0

Zweierkomplement

3	0	0	0	0	0	0	1	1
	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	1
	1	1	1	1	1	1	0	1

Addition

3	1	1	1	1	1	1	0	1
5	0	0	0	0	0	1	0	1
2	0	0	0	0	0	0	1	0

„Einfache“ Multiplikation

```
int ergebnis = 3 * 2;
ergebnis = 3 << 1;
```

- Die Multiplikation mit dem Faktor 2 entspricht einer Verschiebung der binären Zahlen nach links.
- Der Shift Left-Operator << verschiebt eine binäre Zahl um x Stellen nach links.

Beispiel

3	0	0	0	0	0	0	1	1
	0	0	0	0	0	1	1	0

- Hinweis: Das Byte wird mit Null aufgefüllt.

„Einfache“ Division

```
int ergebnis = 4 / 2;
ergebnis = 4 >> 1;
```

- Die Division mit dem Faktor 2 entspricht einer Verschiebung der binären Zahlen nach rechts.
- Der Shift Right-Operator >> verschiebt die binären Zahlen um x Stellen nach rechts.

Beispiel

4	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	1	0

- Hinweis: Das Byte wird mit Null aufgefüllt.