Unterrichtsmaterialien für die Sekundarstufe I

André Hilbig und Johannes Pieper

# Binäre Zahlen

Binäre Zahlen

$$\begin{array}{rr} & 1010010 \\ + & 101011 \\ \hline & 1111101 \end{array}$$

## Autoren

Dieses Dokument ist ein sogenanntes Modul der Materialsammlung, die von der Fachgruppe Informatische Bildung in NRW der Gesellschaft für Informatik heraus gegeben wird. Verantwortliche Autoren für dieses Dokument: André Hilbig und Johannes Pieper

## Lizenz

Dieses Dokument steht unter der Creative Commons by-nc-sa-Lizenz. Folglich darf es beliebig kopiert und bearbeitet werden, sofern das Folgeprodukt wiederum unter dieser Lizenz vertrieben wird. Eine kommerzielle Nutzung ist nicht erlaubt. https://creativecommons.org/licenses/by-nc-sa/4.0/deed.de

## Verweise auf Material

Dieses Modul enthält Hinweise und Verweise (Links) auf Material.Wird eine Seite des Bezugsmaterials dargestellt, ist dieses immer die erste, auch wenn das entsprechende Material aus mehreren besteht. Die Weltkugel verweist dabei auf die entsprechende Webseite und das Dokumentensymbol hat einen relativen Pfad hinterlegt, der die Nutzung der Verzeichnisstruktur der Materialsammlung voraussetzt.

## Kontakt

Ihre Ansprechpartner für die vorliegenden Materialien sind Daniel Losch daniel.losch@gi.de und Johannes Pieper pieper@informatiktag-nrw.de

binZahlen.tex

git
**Version:** dd1fc2
**Stand:** 26. Juni 2026 18:18

git **Zuletzt bearbeitet von:** Johannes Pieper

# 1 Beschreibung

In der Sekundarstufe I kann die Arbeit mit binären Zahlen thematisiert werden. Mit diesen Arbeitsblättern werden die Schülerinnen und Schüler motiviert, sich mit binären Zahlen auseinander zu setzen. Dabei müssen sie sich auch Gedanken dazu machen, wie man bei binären Zahlen rechnet.

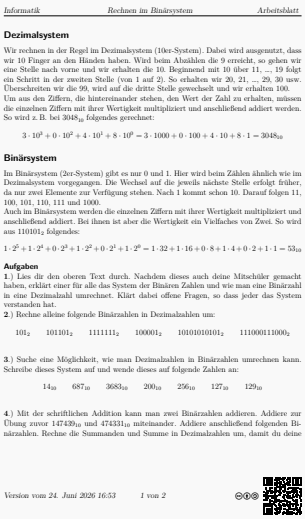
Es werden weitere Darstellungsformen wie Zweierkomplementzahlen, Gleitkommazahlen und die verwandte Hexdezimaldarstellung dabei thematisiert.

# 2 Material

Zu Beginn des Moduls liegt der Schwerpunkt auf der Umrechnung zwischen dem Dezimal- und dem Binärsystem. Dabei wird der Aufbau der Zahlen erklärt. Anhand der Addition und Subtraktion wird die Darstellung der Zahlen im Binärsystem bei einfachen Rechenoperationen genutzt.

**Material**

- ab_rechnen_im_binaersystem.pdf



*Informatik* *Rechnen im Binärsystem* *Arbeitsblatt*

**Dezimalsystem**

Wir rechnen in der Regel im Dezimalsystem (10er-System). Dabei wird ausgenutzt, dass wir 10 Finger an den Händen haben. Wird beim Abzählen die 9 erreicht, so gehen wir eine Stelle nach vorne und wir erhalten die 10. Beginnend mit 10 über 11, ..., 19 folgt ein Schritt in der zweiten Stelle (von 1 auf 2). So erhalten wir 20, 21, ..., 29, 30 usw. Überschreiten wir die 99, wird auf die dritte Stelle gewechselt und wir erhalten 100.
Um aus den Ziffern, die hintereinander stehen, den Wert der Zahl zu erhalten, müssen die einzelnen Ziffern mit ihrer Wertigkeit multipliziert und anschließend addiert werden. So wird z. B. bei $3048_{10}$ folgendes gerechnet:

$$3 \cdot 10^3 + 0 \cdot 10^2 + 4 \cdot 10^1 + 8 \cdot 10^0 = 3 \cdot 1000 + 0 \cdot 100 + 4 \cdot 10 + 8 \cdot 1 = 3048_{10}$$

**Binärsystem**

Im Binärsystem (2er-System) gibt es nur 0 und 1. Hier wird beim Zählen ähnlich wie im Dezimalsystem vorgegangen. Die Wechsel auf die jeweils nächste Stelle erfolgt früher, da nur zwei Elemente zur Verfügung stehen. Nach 1 kommt schon 10. Darauf folgen 11, 100, 101, 110, 111 und 1000.
Auch im Binärsystem werden die einzelnen Ziffern mit ihrer Wertigkeit multipliziert und anschließend addiert. Bei ihnen ist aber die Wertigkeit ein Vielfaches von Zwei. So wird aus $110101_2$ folgendes:

$$1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 32 + 1 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 53_{10}$$

**Aufgaben**

**1**.) Lies dir den oberen Text durch. Nachdem dieses auch deine Mitschüler gemacht haben, erklärt einer für alle das System der Binären Zahlen und wie man eine Binärzahl in eine Dezimalzahl umrechnet. Klärt dabei offene Fragen, so dass jeder das System verstanden hat.

**2**.) Rechne alleine folgende Binärzahlen in Dezimalzahlen um:

$101_2$ $101101_2$ $1111111_2$ $100001_2$ $10101010101_2$ $111000111000_2$

**3**.) Suche eine Möglichkeit, wie man Dezimalzahlen in Binärzahlen umrechnen kann. Schreibe dieses System auf und wende dieses auf folgende Zahlen an:

$14_{10}$ $687_{10}$ $3683_{10}$ $200_{10}$ $256_{10}$ $127_{10}$ $129_{10}$

**4**.) Mit der schriftlichen Addition kann man zwei Binärzahlen addieren. Addiere zur Übung zuvor $14743_{10}$ und $474331_{10}$ miteinander. Addiere anschließend folgenden Binärzahlen. Rechne die Summanden und Summe in Dezimalzahlen um, damit du deine

*Version vom 24. Juni 2026 16:53* *1 von 2*

Im zweiten Teil dieses Moduls liegt der Schwerpunkt auf weiteren Darstellungen innerhalb des Binärsystems. Da sind als erstes die Zweierkomplementzahlen, um auch negative Zahlen mit Nullen und Einsen darzustellen. Diese dienen als Vorwissen für den Aufbau von Gleitkommmazalen, mit denen sich Dezimalbrüche in binärer Form darstellen lassen.

**Material**

- ab_zweierkomplement.pdf



*Informatik* *Rechnen im Binärsystem – Zweierkomplement* *Arbeitsblatt*

**Zweierkomplement**

In Informatiksystemen ist es auch nötig, mit negativen Zahlen zu arbeiten. Auch diese werden natürlich als Binärzahlen gespeichert. Außerdem muss das Vorzeichen durch Null oder Eins dargestellt werden. Dieses geschieht mit Hilfe des Zweierkomplements.
Beim Zweierkomplement wird zuvor festgelegt, aus wie vielen Stellen eine Zahl überhaupt bestehen darf. Jede Null oder Eins wird dabei als Bit bezeichnet. Eine Zahl, die maximal aus acht Nullen und Einsen besteht, wird daher als 8-Bit Zahl bezeichnet. Bei ihr gibt man immer die führenden Nullen mit an, wenn dieses nötig ist. So wird die Zahl 6 als 8-Bit-Zahl wie folgt dargestellt: $00000110_2$. Das vorderste Bit gibt dabei an, welches Vorzeichen die Zahl hat. Bei einer Null handelt es sich um eine positive und bei einer Eins um eine negative Zahl.
Bei der Zahl $10000110_2$ handelt es sich aber nicht um $-6_{10}$ sondern um $-122$. Die Zahl $-6_{10}$ hat die Darstellung $11111010_2$. Das hat den Hintergrund, dass es auf diese Weise für den Rechner einfacher ist mit den entsprechenden Zahlen zu arbeiten. Führe dazu die folgenden Rechnungen mit schriftlicher Subtraktion der Binärzahlen durch. Beachte, dass du jeweils nur eine 8-Bit-Zahl hast. Solltest du das neunte Bit benötigen, wird dieses einfach ohne weitere Berücksichtigung gestrichen.

$$00000000_2 - 00000001_2 \quad 5_{10} - 7_{10} \quad 0_{10} - 63_{10}$$

**Aufgabe 1**
Erläutere schriftlich das System, wie sich die Zahlen -1, -2, -3, ...im Zweierkomplement darstellen lassen.
**Aufgabe 2**
Addiere schriftlich folgende Zahlen und gib jeweils auch die Dezimalzahl mit an.

$$00101101_2 + 10101001_2 \quad 10110101_2 + 11111101_2 \quad 11010110_2 + 00000011_2$$

**Aufgabe 3**
Gib an, welches die größte und welches die kleinste 4-, 8- bzw. 16-Bit-Zahl im Zweierkomplement ist. Schreibe dazu auch immer ihre binäre Darstellung.
**Aufgabe 4**
Überprüfe anhand von eigenen Beispielen, ob sich diese Rechnung auch auf die Multiplikation übertragen lässt.

*Version vom 24. Juni 2026 16:53* *1 von 1*

- ab_gleitkommazahl.pdf



*Informatik* *Gleitkommazahlen im Binärsystem* *Arbeitsblatt*

**Gleitkommazahlen**

Immer dann, wenn ganze Zahlen nicht ausreichen, wird heute im Computer mit Gleitkommazahlen gearbeitet. Sie werden so genannt, weil bei ihrer internen Darstellung das Komma immer passend verschoben wird, so dass sie der Norm $\pm m \cdot 10^e$ entspricht. Bei der Mantisse $m$ handelt es sich dabei um eine Kommazahl im Bereich zwischen 1 und $9,\overline{9}$. Die 10 bezeichnet man als Basis und $e$ als Exponent.
So wird z. B. die Lichtgeschwindigkeit (299 792 458 m/s) geschrieben als $2,997\,924\,58 \cdot 10^8$ m/s und ein Finger ist etwa $1,5 \cdot 10^{-2}$ m breit.
**Aufgabe 1**
Forme folgende Zahlen um, so dass sie der Norm mit der Basis 10 entsprechen und gib an, was genau der Exponent und die Mantisse sind:

1. 883,24
2. 0,038 2
3. -145 000 000
4. 39,302
5. -0,000 003 39

**Binäre Darstellung**

Der Computer rechnet in binärer Darstellung, bei der die Basis 2 ist. Folglich ist die erste Ziffer der Mantisse immer eine 1. Bei einer 32-Bit Gleitkommazahl wird für das Vorzeichen das erste Bit genutzt. Dann stehen $23 + 1$ Bit für die Mantisse zur Verfügung. Da immer das erste Bit eine 1 ist wird sie weggelassen. Es bleiben somit 8 Bit für den Exponenten.
Möchte man die Mantisse umrechnen in Dezimalzahlen, so wird folgendermaßen gerechnet:
$1 + \frac{m_1}{2} + \frac{m_2}{4} + \frac{m_3}{8} + \frac{m_4}{16} + \ldots$
Beim Exponenten wird nicht mit dem Zweierkomplement gerechnet, sondern der Biaswert von 127 vom Exponenten subtrahiert, so dass man mit einer Zahl zwischen 0 und 255 einen Exponenten von -127 bis 128 darstellt.

**Umrechnung von Binär in Dezimal**

Umgerechnet werden soll die Zahl 1 10000101 01010100000000000000000. Das erste Bit stellt das Vorzeichen da: Es handelt sich um eine negative Zahl. Die nächsten 8 Bit sind der Exponent mit einem Biaswert von 127. Sie stellen die Zahl $1 \cdot 2^7 + 1 \cdot 2^2 + 1 \cdot 2^0 = 133$ da, was bedeutet der Exponent ist $133 - 127 = 6$.
Für die Mantisse muss gerechnet werden: $1 + \frac{0}{2} + \frac{1}{4} + \frac{0}{8} + \frac{1}{16} + \frac{0}{32} + \frac{1}{64} + \frac{0}{128} + \frac{0}{256} + \ldots = 1,328125$
Es ergibt sich daraus komplett: $-1,328125 \cdot 2^6 = -85 = -8,5 \cdot 10^1$

*Version vom 24. Juni 2026 16:53* *1 von 2*

Mit dem Arbeitsblatt zur hexadezimalen Darstellung von Zahlen kann das Modul abgeschlossen werden. Diese Darstellung wird häufig im Zusammenhang mit Programmen bzw. der Programmierung genutzt wird. Dieses ist der Fall, da sich 8-Bit-Werte zweistellig hexadezimal darstellen lassen. Den Schülerinnen und Schülern könnte es daher z. B. auch aus der Darstellung von Farbwerten bekannt sein.

## Material

- ab_hexadezimal.pdf

## Hexadezimale Zahlen

In vielen Bereichen des Computers wird mit 8, 16 oder mehr Bit gearbeitet. Damit werden dann unter anderem auch Speicherbereiche adressiert oder Befehle codiert. Mit 8 Bit, was einem Byte entspricht, hat man $2^8 = 256$ Möglichkeiten. Um diese einheitlich mit immer gleicher Anzahl Zeichen darzustellen, wird das hexadezimale System genutzt. Beim hexadezimalen System wird als Basis die 16 genutzt. Bei dem uns geläufigen dezimalen System ist 10 die Basis und beim binären System ist dieses die 2. Da $16^2 = 256$ ist, werden für die Darstellungen immer nur zwei Zeichen anstatt der 8 Zeichen in binärer Darstellung benötigt.
Um die Ziffern oberhalb der 9 darzustellen, wird auf die ersten Buchstaben des Alphabets zurückgegriffen. A steht dabei für 10, B für 11 und F für 15. Umgerechnet in das Dezimalsystem wird in ähnlicher Weise wie beim Binärsystem: Aus der Zahl 0x8B wird $8 \cdot 16^1 + 11 \cdot 16^0 = 8 \cdot 16 + 11 \cdot 1 = 128 + 11 = 139_{10}$.
Zur besseren Unterscheidung von Dezimalzahlen wird bei hexadezimalen Zahlen, wie im obigen Beispiel, ein 0x davor gesetzt. Dadurch wird verdeutlicht, dass z. B. mit 0x56 nicht die uns bekannte 56 gemeint ist. Diese Schreibweise wird unter anderem in Python verwendet. Ein h hinten anzufügen ist eine weitere, ebenso verbreitete Möglichkeit, hexadezimale Zahlen kenntlich zu machen. So gilt $0x56 = 56h = 86_{10}$.

**Aufgabe 1**
Rechne folgende Hexadezimalzahlen in Dezimalzahlen um:

$0xFF$ $0x7E$ $0xAFC3$ $0x42B1$ $0x1111$ $0x92EE$ $0xD6C3$

**Aufgabe 2**
Suche eine Möglichkeit, wie man Dezimalzahlen in Hexadezimalzahlen umrechnen kann. Schreibe diesen System auf und wende dieses auf folgende Zahlen an:

$114_{10}$ $687_{10}$ $3683_{10}$ $156200_{10}$ $256_{10}$ $127_{10}$ $129_{10}$