

## Zahlensysteme

---

Will man verstehen wie ein Computer rechnet, muß man sich zwangsläufig mit Zahlen und Zahlensystemen auseinander setzen.

Ein im Umgang mit Computerprogrammierung geübter Mensch erkennt viele Zusammenhänge besser, wenn Angaben NICHT im dezimalen System, sondern im dualen (binären) oder hexadezimalen System gemacht sind.

Hier soll der Zusammenhang von dualen (binären), hexadezimalen und dezimalen Zahlen aufgezeigt werden.

Das dem Mensch geläufige dezimale Zahlensystem dürfte sich wohl deshalb entwickelt haben, weil der Mensch 10 Finger hat. In Analogie dessen, hat ein Computer derer nur 2 Finger.

Eine erste Fehlerquelle beim Umgang mit Zahlen wäre die Frage, wie weit ein Mensch mit den Fingern zählen kann und die gängige Antwort ist häufig : Zehn. Und diese Antwort ist falsch, denn ein Zählvorgang startet nicht mit dem Wert 1 sondern mit dem Wert 0. Würde jeder Finger von 0 an beschriftet werden, so würde der letzte beschriftete Finger den Ziffernwert 9 erhalten.

Von 0 angefangen zu zählen endet unser Ziffernvorrat an Fingern mit dem Wert 9, aber dennoch sind es eben 10 Finger.

Der Ziffernvorrat der 10 Finger beträgt also einen Umfang von 0 bis 9. Beim binären / dualen Zahlensystem beträgt der Umfang der 2 verfügbare Ziffern: 0 und 1.

An dieser Stelle sei auch der Unterschied zwischen Ziffern und Zahlen aufgezeigt:

Zahlen BESTEHEN aus Ziffern genau so, wie Wörter aus Buchstaben bestehen. Hier muß man unterscheiden, ob es sich um eine einstellige Zahl oder um eine Ziffer (als Bestandteil einer Zahl) handelt. Analog hierzu wäre ein "I" im Englischen auch nicht eindeutig zugewiesen, es könnte schlichtweg ein einzelner Buchstabe sein, oder für "ich" als die Übersetzung gelten. "I" könnte also entweder ein Buchstabe oder ein Wort sein.

Auch wenn wir Zahlenangaben und das Zählen im dezimalen Zahlensystem im Schlaf beherrschen, sollte man sich wieder einmal vor Augen führen, wie das Zählen denn eigentlich funktioniert. Diese Vorgehensweise des Zählens kann in jedes x-beliebige Zahlensystem übertragen zu werden. Auch wenn das Folgende wie eine Exkursion in die Anfänge der Grundschule anmutet: Den ganzen Text zu lesen wird damit belohnt, das binäre und hexadezimale Zahlensystem zu verstehen.

### **Was passiert denn beim Zählen genau ?**

Fängt man bei 0 an zu zählen, erhöht man die Ziffer so lange um 1, bis der Ziffernvorrat zu neige geht, im dezimalen Zahlensystem ist das die 9. Danach erhöht man die nächst höhere Stelle um 1 und fängt wieder (im wahrsten Sinn des Wortes) bei 0 an zu zählen. Eben so lange, bis der Ziffernvorrat zu Ende geht. Dann erhöht man wieder bis zur größten Ziffer, erhöht die nächst höhere Stelle um 1 und fängt bei 0 wieder an.

Dieses Vorgehen ist in jedem Zahlensystem gleich !

dezimales zählen		duales zählen	
00		0	
01		1	Ziffernvorrat zu Ende
02		10	nächste Stelle +1, von 0 anfangen
03		11	Ziffernvorrat zu Ende
04		100	nächste Stelle +1, von 0 anfangen
05		101	Ziffernvorrat zu Ende
06		110	nächste Stelle +1, von 0 anfangen
07		111	Ziffernvorrat zu Ende
08		1000	nächste Stelle +1, von 0 anfangen
09	Ziffernvorrat zu Ende	1001	Ziffernvorrat zu Ende
10	nächste Stelle +1, von 0 anfangen	1010	nächste Stelle +1, von 0 anfangen
11		1011	Ziffernvorrat zu Ende
12		1100	nächste Stelle +1, von 0 anfangen
13		1101	Ziffernvorrat zu Ende
14		1110	nächste Stelle +1, von 0 anfangen
15		1111	Ziffernvorrat zu Ende

In obiger Tabelle sollte man gut erkennen wie ein Zählvorgang funktioniert. Immer wenn in einem Zahlensystem die höchste Ziffer erreicht ist, wird beim nächsten Fortzählen die eigene Stelle zu 0 gesetzt und die nachfolgende erhöht. War bei der nachfolgenden (und allen nachfolgenden) die höchste Ziffer erreicht: 0 setzen, nächste erhöhen. Außerdem kann man auch gut sehen, welche dezimale Zahl welcher dualen Zahl entspricht, da in beiden Fällen nur jeweils um 1 erhöht wurde.

### Eine andere Art, mit Fingern zu zählen

Eine andere Art bspw. um mit den Fingern zu zählen könnte man derart in Betracht ziehen: Ein Mensch hat 2 Hände zu je 5 Fingern. Jetzt kann man festlegen, dass die rechte Hand die rechte Ziffer einer Zahl ist, die linke Hand dann die nächst höhere Ziffer. Mit der rechten Hand geht (von 0 beginnend) nach der Ziffer 5 der Ziffernvorrat zu Ende (kein Finger entspricht 0) und mit der nächsten Erhöhung wird an der linken Hand 1 weitergezählt, während an der rechten Hand von vorne (0 = kein Finger) begonnen wird:

Zählung	Anzahl Finger links	Anzahl Finger rechts	
0	0	0	
1	0	1	
2	0	2	
3	0	3	
4	0	4	
5	0	5	Ziffernvorrat zu Ende
6	1	0	nächste Stelle +1, von 0 anfangen
7	1	1	
8	1	2	
9	1	3	
10	1	4	
11	1	5	Ziffernvorrat zu Ende
12	2	0	nächste Stelle +1, von 0 anfangen

Nach jeder sechsten Zählung würde also ein Finger der linken Hand hinzukommen. Jeder einzelne Finger der linken Hand hätte damit die Wertigkeit 6, die Finger der rechten Hand die Wertigkeit 1 (weil mit jeder Zählung 1 Finger dazu kommt).

Welchen Zahlenwert hätten dann beispielsweise 3 Finger der linken und 4 Finger der rechten Hand ?

$$(3 * 6) + (4 * 1) = 22$$

Nach dieser Methode käme man mit 5 Fingern links und 5 Fingern rechts auf :  $(5 * 6) + (5 * 1) = 35$   
Hätten wir eine dritte Hand, so wäre die Wertigkeit der dritten Hand 36, weil der Ziffernvorrat beider Hände erschöpft wäre und nach jeder 36. Zählung an der dritten Hand ein Finger hinzukäme.

Zurück zum dezimalen Zahlensystem (und damit auch wieder ein bisschen zurück zur Grundschule). Mit dem dezimalen Zahlensystem verhält es sich genauso wie mit dem obigen Beispiel mit den Fingern, mit dem Unterschied, dass hier die Wertigkeiten 1 - 10 - 100 - 1000 und folgende sind (umgangssprachlich: einer, zehner, hunderter, tausender)

Hier kommen jetzt Exponentialzahlen ins Spiel und nein, es wird NICHT schwierig.

Woher kommt das denn mit den Einer, Zehner, Hunderter etc. ?

Es kommt daher dass eben jeweils 10 Vorgänger erhöht werden mußten (10 Ziffern), bis die nachfolgende Stelle erhöht wird. Für die "zehner" bedeutet dieses, es mußten 10 mal die "einer" gezählt werden, für die "hunderter" heißt das: 10 mal die "einer" mal 10 mal die "zehner" ergibt eben die "hunderter".

Die Wertigkeit der "einer" heißt ich mußte keinen Vorgänger hochzählen, es gibt keinen Vorgänger, also (etwas merkwürdig ausgedrückt): 10 Ziffern hoch keinen (0) Vorgänger.

Mathematisch ausgedrückt:  $10^0 = 1$

Die "zehner" haben einen Vorgänger, also  $10^1 = 10$ . Die "hunderter" haben 2 Vorgänger, also  $10^2 = 100$  usw.

### Zerlegen einer Zahl in ihre Wertigkeit (mit einer willkürlichen Zahl 6938)

6938								
Ziffer	6		9		3		8	
Wertigkeit Ziffernposition	$10^3 = 1000$		$10^2 = 100$		$10^1 = 10$		$10^0 = 1$	
Rechnung	$(6 * 1000)$	+	$(9 * 100)$	+	$(3 * 10)$	+	$(8 * 1)$	= 6938

## Darstellung binärer Zahlen

Die Darstellung binärer / dualer Zahlen funktioniert exakt so, wie die vorherig beschriebene Darstellung dezimaler Zahlen mit einem Unterschied: Es gibt nicht mehr 10 Ziffern (0 bis 9), sondern derer nur noch 2 Ziffern (0 und 1).

Aus diesem Grund heißt es jetzt nicht mehr  $10^0$ ,  $10^1$ ,  $10^2$  sondern  $2^0$ ,  $2^1$ ,  $2^2$ ,  $2^3$  etc.

### Zerlegen einer binären Zahl in ihre Wertigkeit (mit einer willkürlichen binären Zahl 10010111)

Ziffer	1	0	0	1	0	1	1	1	
Wertigkeit	$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$	
	$1*128$	$0*64$	$0*32$	$1*16$	$0*8$	$1*4$	$1*2$	$1*1$	
Rechnung	128 +			16 +		4 +	2 +	1 =	151

Im Zusammenhang der Wertigkeit der Ziffernposition kann man hier auch von den "einer, zweier, vierer, achter" sprechen. Mit jeder weiteren Ziffernposition verdoppelt sich die Wertigkeit.

Die binäre größte Zahl, die mit 8 Stellen darstellbar ist, ist 1111.1111. Der Punkt zwischen den einzelnen Viererpaketen stellt kein Komma dar, er dient lediglich zum besseren Abzählen der Stellen einer Zahl.

$$1111.1111 \text{ entspricht } 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255.$$

Ziffernpositionen werden von 0 an gezählt, d. h. die niedrigste Ziffernposition einer 8-stelligen binären Zahl ist 0, die höchste Ziffernposition ist 7 (hier spricht man auch von Bit 0 und Bit 7).

Würde zum obigen Wert 1111.1111 eine 1 addiert werden, wäre der Ziffernvorrat an jeder Position erschöpft, jede einzelne Ziffer würde ihren Nachfolger um 1 erhöhen, so dass es zusätzlich eine Ziffernposition 8 benötigt. Diese Position hätte dann die Wertigkeit  $2^8 = 256$ .

Die höchste Zahl die mit 8 Stellen (8-Bit) dargestellt werden kann ist demnach:  $2^8 - 1 = 255$ .

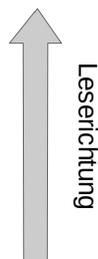
Grundsätzlich gilt:  $2^{\text{Anzahl Bits}} - 1 = \text{größte darstellbare Zahl!}$

### Umrechnung einer dezimalen Zahl in eine binär Zahl

Zur Umrechnung einer dezimalen Zahl in eine binäre Zahl wird die Modulo-Division verwendet. Ohne diesen Ausdruck zu kennen hat sicherlich jeder schon einmal ein solche Division durchgeführt. Sie bezeichnet den "Restanteil", der bei einer ganzzahligen Division entsteht. Beispiel:  $20 / 6 = 3 \text{ Rest } 2$

151		
Dividend	Divisor	Rest
$151 \div 2$	75	1
$75 \div 2$	37	1
$37 \div 2$	18	1
$18 \div 2$	9	0
$9 \div 2$	4	1
$4 \div 2$	2	0
$2 \div 2$	1	0
$1 \div 2$	0	1

Ergebnis:  $151 = 1001.0111$



Der "Rest" erklärt sich aus der Differenz von  $3 * 6$  zum originalen Dividenten:  $(3 * 6) + 2 = 20$

Bei der Umrechnung dividiert man so lange die umzurechnende Zahl und das daraus entstandene Ergebnis (Quotient) durch 2 bei Notierung des Restes. Dieses wird solange wiederholt, bis der Divisor = 0 ist.

Als Beispiel soll die dezimale Zahl 151 in das binäre Zahlensystem umgerechnet werden.

Wie man relativ leicht feststellen kann, werden binäre Zahlen für den Menschen sehr schnell unhandlich. Schon recht kleine Zahlen benötigen eine hohe Zifferanzahl. Aus diesem Grund wurden sogenannte "binary coded decimals" (BCD) und das hexadezimale Zahlensystem eingeführt, die im nächsten Abschnitt erläutert werden.

## BCD - binary coded decimals

---

Möchte man eine dezimale Ziffer (keine Zahlen !) binär darstellen benötigt es insgesamt eine 4 stellige binäre Zahl. Mit einer 4 stelligen binären Zahl kann ein höchster Wert von  $2^{4-1} = 15$  dargestellt werden, was mehr ist, als zum Darstellen einer dezimalen Ziffer benötigt wird.

Bei einer BCD kodierten Zahl bilden ein vierstelliger Block aus 1 und 0 eine dezimale Ziffer ab. Ein solcher vierstelliger Block hat in der Informatik einen Namen: **Nibble**.

Hierzu wird die Wertigkeit einer binären Ziffernposition nicht kontinuierlich verdoppelt, sondern sie fängt nach jedem Block in ihrer Wertigkeit von neuem an.

Jeder dezimalen Ziffer werden demnach eine vierstellige Binärzahl, 4 Bit, zugeordnet. Es soll die Zahl 6938 aus dem Dezimalbeispiel in eine BCD Zahl zerlegt werden.

```
Ziffer 6 : ( 0 * 8 ) + ( 1 * 4 ) + ( 1 * 2 ) + ( 0 * 1 ) = 0110
Ziffer 9 : ( 1 * 8 ) + ( 0 * 4 ) + ( 0 * 2 ) + ( 1 * 1 ) = 1001
Ziffer 3 : ( 0 * 8 ) + ( 0 * 4 ) + ( 1 * 2 ) + ( 1 * 1 ) = 0011
Ziffer 8 : ( 1 * 8 ) + ( 0 * 4 ) + ( 0 * 2 ) + ( 0 * 1 ) = 1000
```

Hintereinander geschrieben stellt die binär kodierte Zahl 0110.1001.0011.1000 die Ziffern 6938 dar.

### ACHTUNG :

Der obige BCD - Code entspricht NICHT dem binären Zahlenwert von 6938, er stellt lediglich die Ziffern der Zahl 6938 dar, aber nicht diese Zahl.

### Warum ist das so?

Die dezimale Zahl 9 entspricht binär dargestellt 1001. Erhöht man diesen binären Zahlenwert um 1 (Ergebnis dann 10), entspricht dieses in binärer Darstellung 1010 (  $8+2 = 10$  ). in BCD kodierter Darstellung aber 0001.0000, weil jede Ziffer durch jeweils 4 Bit dargestellt werden.

Die höchste Zifferndarstellung die mit 16 Bit im BCD-Code darstellbar ist beträgt demnach:

```
1001.1001.1001.1001
  9      9      9      9
```

Mit einer 16 stelligen binären Zahl (16 Bit) kann jedoch  $2^{16}-1 = 65535$  als höchster Zahlenwert erfasst werden. Also eine Zahl die mehr als 6 mal so groß ist, als in BCD.

Um nun dennoch die vierer Blockbildung im vollem Umfang zur besseren Darstellung nutzen zu können, wird das **hexadezimale Zahlensystem** verwendet, was im nächsten Abschnitt beschrieben ist.

# Hexadezimals Zahlensystem

Wie beim BCD-Code werden 4 Bits zu einem Block zusammen gefasst (man erinnere, der Name ist Nibble). Mit 4 Bits können insgesamt 16 Ziffern dargestellt werden.

**Wie bitte, es gibt doch gar keine 16 Ziffern !?!  
DOCH, die gibt es !**

Hexa steht für sechs und Dezi für zehn, ergibt zusammengesetzt das Wort "hexadezimal", gleichbedeutend mit 16. Im "normalen" Sprachgebrauch gibt es keine 16 Ziffern, im Sprachgebrauch der Digital- und Computertechnik jedoch sehr wohl. Man erinnere sich daran: Man kann mit einer Anzahl x Ziffern einen Wert von 0 bis (x - 1) darstellen. Mit einem Ziffernvorrat von 16 Ziffern hat die höchste Ziffer die Wertigkeit 15.

Zifferntabelle

hexadezimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
dezimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
binär	0000	0001	0010	00110	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Nachdem mit dem dezimalen Zahlensystem der Ziffernvorrat mit 9 zu Ende gegangen ist, werden im hexadezimalen System die Ziffern mit den Buchstaben "A" bis "F" fortgeführt. Diese Ziffern "A" bis "F", die auch als **"alphanumerische Ziffern"** bezeichnet werden, haben die Wertigkeit 10 bis 15 (siehe obige Tabelle).

Der Ziffernvorrat im hexadezimalen Zahlensystem endet also mit der Ziffer "F" und NICHT bei 9 wie im dezimalen Zahlensystem. Zählt man also im hexadezimalen Zahlensystem, so ist der Nachfolger von 9 die Ziffer "A".

## Was ist der Nachfolger von "F" ?

Gemäß den Regeln der Eingangs beschriebenen Zahlensystemen geht mit der Ziffer "F" der Ziffernvorrat zu Ende: Die nächsthöhere Stelle wird um 1 erhöht und die Zählung beginnt wieder bei 0. Der Nachfolger von "F" ist demnach 10h (das kleine "h" hinter der 10 macht deutlich, dass es sich bei der Zahlenangabe um eine hexadezimale Zahl handelt).

Jede Vorgängerposition muß also 16 mal hochgezählt werden, damit diese Position um eins erhöht wird. Die Position 0 hat keine Vorgängerstelle und jeder Zählvorgang hat die Wertigkeit 1. Um die Position 1 um eins zu erhöhen mußte der Vorgänger 16 mal gezählt werden, also hat jede Ziffer der Position 1 die Wertigkeit mal 16. Die Position 2 mußte die 0te Position und 1te Position jeweils 16 mal zählen, also 16 mal 16 oder auch  $16^2$  und hat somit die Wertigkeit 256. Dieses setzt sich mit jeder weiteren Position so fort.

## Zerlegen einer hexadezimalen Zahl in ihre Wertigkeit (mit einer willkürlichen Zahl 3A4Bh)

3A4Bh						
Ziffer	3		A		4	B
binäre Zifferndarstellung	0011		1010		0100	1011
Ziffernwert (dezimal)	3		10		4	11
Wertigkeit Ziffernposition	$16^3 = 4096$		$16^2 = 256$		$16^1 = 16$	$16^0 = 1$
Rechnung	$( 3 * 4096 ) + ( 10 * 256 ) + ( 4 * 16 ) + ( 11 * 1 ) = 14923$					

Der Vorteil von hexadezimalen Zahlen besteht nun darin, dass jede einzelne Ziffer (0..9, A..F) relativ leicht einer binären 1-0 Kombination zuzuordnen ist. Eine 4 stellige binäre Zahl lässt sich noch relativ leicht in ihrer Wertigkeit bestimmen (siehe Zifferntabelle).

Theoretisch könnte man die Zusammenfassung von jeweils 4 Bit (man erinnere: 4 Bit = 1 Nibble) zu einer hexadezimalen Ziffer auch "binary coded hexadecimals" nennen, dieses ergibt jedoch keinen Sinn, weil die komplette 1-0 Folge direkt der Wertigkeit entspricht.

Aus unserer zerlegten hexadezimalen Zahl geht dadurch hervor:

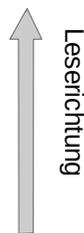
$$3A4Bh = 0011.1010.0100.1011 \text{ binär} = 14923 \text{ dezimal}$$

### Umrechnung einer dezimalen Zahl in eine hexadezimale Zahl

Zur Umrechnung einer dezimalen Zahl in eine hexadezimale Zahl wird die Modulo-Division verwendet. Ohne diesen Ausdruck zu kennen, hat sicherlich jeder schon einmal ein solche Division durchgeführt. Sie bezeichnet den "Restanteil", der bei einer ganzzahligen Division entsteht. Beispiel:  $20 / 6 = 3 \text{ Rest } 2$

14923			
Dividend	Divisor	Rest	Hexziffer
$14923 \div 16$	932	11	B
$932 \div 16$	58	4	4
$58 \div 16$	3	10	A
$3 \div 16$	0	3	3

Ergebnis:  $14923 = 3A4B$



Der "Rest" erklärt sich aus der Differenz von  $3 * 6$  zum originalen Dividenten:

$$(3 * 6) + 2 = 20$$

Bei der Umrechnung dividiert man so lange die umzurechnende Zahl und das daraus entstandene Ergebnis (Quotient) durch 16 bei Notierung des Restes. Dieses wird solange wiederholt, bis der Divisor = 0 ist.

Als Beispiel soll die dezimale Zahl 151 in das binäre Zahlensystem umgerechnet werden.

## Kenntlichmachung des Zahlensystems

Bisweilen kann es bei der Interpretation einer Zahl zu Mißverständnissen kommen, weil nicht klar ist, in welchem Zahlensystem eine Zahl angegeben ist. Zudem werden Zahlen, die nicht dem dezimalen Zahlensystem entsprechen, in verschiedenen Programmiersprachen unterschiedlich gekennzeichnet:

	allgemein	Assembler	C, C++	Pascal
dezimal	151, 151d, 151dez		151	151
hexadezimal	97h, 97hex	097h, 0x97	0x97	\$97
binär	1001b, 1001bin	unterschiedlich	-	%1001

## Sonderfall einer Zahlezuweisung

---

Ein, wenn auch seltener, Sonderfall einer Zuweisung einer Zahl zu dem Speicherort der Zahl soll hier gezeigt werden.

Es wird angenommen, dass dem Speicher eines Systems, Zahlen nur in dezimaler Schreibweise zugeordnet werden können (auch wenn, wie gelernt wurde, Zahlen in Computern nur mit 1 und 0 gespeichert sind). Weiterhin wird angenommen, dass es sich um ein 8-Bit System handelt, das heißt es kann jeder Speicherzelle auch nur ein 8-Bit Wert zugewiesen werden.

In den Speicher soll aber eine 16-Bit große Zahl (0 bis 65535) mittels 2 je 8-Bit großen Werten gespeichert werden.

Die 8-Bit mit dem größeren Stellenwert werden als High-Byte, die mit dem niedrigeren Stellenwert als Low-Byte bezeichnet.

Es soll die 16-Bit große dezimale Zahl 50000 dez im Speicher abgelegt werden.

Hierzu wird zuerst die dezimale Zahl in das hexadezimale System konvertiert (siehe Abschnitt **"Umrechnung einer dezimalen Zahl in eine hexadezimale Zahl"**):

$$50000 \text{ dez} = \text{C350 hex} \quad \left( = \begin{array}{cc} \text{High-Byte} & \text{Low-Byte} \\ 1100.0011 & . \quad 0101.0000 \text{ bin} \end{array} \right)$$

Die Ziffernfolge "C3" der hexadezimalen Zahl repräsentieren das High-Byte, die Ziffernfolge "50" das Low-Byte der hexadezimalen Zahl C350h.

Für eine Speicherung mit 2 \* 8-Bit muß jetzt C3h und 50h wiederum getrennt von einander ins dezimale System zurückkonvertiert werden:

$$\text{C3 hex} = 195 \text{ dez}; \quad 50 \text{ h} = 80 \text{ dez}$$

Für das High-Byte muß nun der dezimale Wert 195, für das Low-Byte der dezimale Wert 80 gespeichert werden.