1 Vorbetrachtungen

Wie könnte eine Codierung von Zeichen im Computer realisiert werden? Der Computer arbeitet mit elektrischem Strom, d. h. er kann lediglich zwischen den beiden Zuständen „Strom an“ und „Strom aus“ unterscheiden. Mit Hilfe einer Leitung können also lediglich zwei unterschiedliche Zeichen „codiert“ werden:
1. Leitung hat Strom (1)
2. Leitung hat keinen Strom (0)
Man nennt die Information, die durch den Strom in einer Leitung codiert ist, ein **Bit** („Binary Digit“). Allerdings kann man mit einem Bit eben nur zwei unterschiedliche Zeichen codieren. Frage, wie viel Zeichen könnte man mit zwei Leitungen codieren? Schreiben wir uns die Möglichkeiten der Leitungen abgekürzt mit 0 und 1 auf:

Leitung 1	**Leitung 2**
0	0
0	1
1	0
1	1

Insgesamt ergeben sich damit 4 Möglichkeiten der Codierung. Dies ist allerdings auch noch nicht genug. Wie viele Möglichkeiten erhält man nun bei drei Leitungen? .... Richtig, Acht Kombinationsmöglichkeiten, also doppelt so viele wie bei zwei Leitungen:

Leitung 1	**Leitung 2**	**Leitung 3**
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

In der Informatik ist es üblich, acht Leitungen zur Speicherung von Informationen zusammenzufassen. Insgesamt lassen sich mit 8 Leitungen also $2^8 = 256$ verschiedene Zeichen codieren. Man spricht bei dieser Bündelung von acht Leitungen von einem **Byte**. Diesen Begriff hast du mit Sicherheit schon einmal im Zusammenhang mit Diskettengrößen (1.44 **Megabyte** = ca. 1.44 · 1.000.000 Byte) oder Festplattengrößen
(1 **Gigabyte** = ca. 1000 · 1 Megabyte = ca. 1.000.000.000 Byte) gehört. Früher rechnete man noch in **Kilobyte**, was ca. 1.000 Bytes entspricht. Das „ca.“ kommt daher, dass hier das Wort „Kilo“ nicht wie üblich die Vervielfachung um 1.000 sondern genau genommen um 1.024 angibt. Genauso ist ein Megabyte = 1.024 Kilobyte usw.

2 Das binäre Zahlensystem

Wollen wir nun anstelle von Zeichen natürliche Zahlen codieren, so müssen wir uns ein Schema überlegen, welche Codierung eines Bytes z. B. der Zahl 17 entspricht. In der Unterstufe hast du das sogenannte **Binärsystem** oder **Dualsystem** oder aber auch **Zweiersystem** kennen gelernt. Dabei handelt es sich wie bei unserem bekannten Dezimalsystem um ein Stellenwertsystem:

Dezimalsystem					**Binärsystem**							
T 10^3 1000	H 10^2 100	Z 10^1 10	E 10^0 1	Stellenwert	128er 2^7 128	64er 2^6 64	32er 2^5 32	16er 2^4 16	8er 2^3 8	4er 2^2 4	2er 2^1 2	1er 2^0 1
4	**1**	**0**	**6**		**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
$= \mathbf{4} \cdot 1000$ $+ \mathbf{1} \cdot 100$ $+ \mathbf{0} \cdot 10$ $+ \mathbf{6} \cdot 1$ $= 4106$					$= \mathbf{1} \cdot 128$ $+ \mathbf{0} \cdot 64$ $+ \mathbf{1} \cdot 32$ $+ \mathbf{1} \cdot 16$ $+ \mathbf{0} \cdot 8$ $+ \mathbf{0} \cdot 4$ $+ \mathbf{1} \cdot 2$ $+ \mathbf{1} \cdot 1$ $= 179$							

Um kenntlich zu machen, in welchem Zahlensystem die Zahl zu interpretieren ist, schreibt man dieses als kleine Zahl unten dran: $1011\ 0011_2 = 179_{10}$. Außerdem schreibt man der Übersicht halber Dualzahlen stets in Viererblöcken, so wie man auch schon mal große Dezimalzahlen in Dreierblöcken aufschreibt. Im Dezimalsystem heißt ein solcher Block Tausenderblock, im Dualsystem heißt ein solcher Viererblock ein **Nibble**.

Aufgabe 1: Überführe die folgenden Binärzahlen ins Dezimalsystem:
$0010\ 0111_2$ $1111\ 1111_2$ $0101\ 1010_2$

Kommen wir zur Umwandlung einer Dezimalzahl in eine Binärzahl. Hierbei muss man ausgehend vom höchsten Stellenwert des Zweiersystems probieren, ob der Stellenwert in der Dezimalzahl enthalten ist, oder nicht. Ist er enthalten, so steht im Zweiersystem an der entsprechenden Stelle eine 1 und die Dezimalzahl wir um den Stellenwert erniedrigt.
Beispiel: $119_{10} = ???_2$

Der erste Stellenwert (128) ist in 119 nicht enthalten,	also 1. Stelle 0	
Der zweite Stellenwert (64) ist in 119 enthalten,	also 2. Stelle 1	119 – 64 = 55
Der dritte Stellenwert (32) ist in 55 enthalten,	also 3. Stelle 1	55 – 32 = 23
Der vierte Stellenwert (16) ist in 23 enthalten,	also 4. Stelle 1	23 – 16 = 7
Der fünfte Stellenwert (8) ist in 7 nicht enthalten,	also 5. Stelle 0	

.... usw. .... (probiere es selbst zu Ende)
Am Schluss solltest du herausbekommen haben: $119_{10} = 0111\ 0111_2$. Wenn nicht, dann übe es an folgender Aufgabe:

Aufgabe 2: Überführe die folgenden Dezimalsystem ins Binärzahlen. Schreibe die Binärzahlen in Nibble-Block-Schreibweise.
37_{10} 127_{10} 90_{10}

3 Rechnen mit binären Zahlen

Du kannst dir vorstellen, dass es nicht ausreicht, lediglich Zahlen vom Binärsystem ins Dezimalsystem und zurück umzuwandeln. Wir wollen mit diesen Zahlen auch rechnen können – genauso, wie es der Computer tut.

3.1 Die Addition von Binärzahlen

Die Addition von Binärzahlen funktioniert prinzipiell genauso, wie die Addition von Dezimalzahlen. Allerdings muss der Übertrag nicht erst bei Überschreitung der Ziffer 9 sondern bei Überschreitung der Ziffer 2 erfolgen. Schauen wir uns ein Beispiel an:

Dezimal	**Binär**
75	0100 1011
+ 43	+ 0010 1011
	1 1 1
= 118	= 0111 0110

Werden mehrere Binärzahlen addiert, so kann der Übertrag auch größer als 1 werden:

Dezimal	**Binär**
75	0100 1011
+ 43	+ 0010 1011
+ 43	+ 0010 1011
	1 1(10) (10)1
= 161	= 1010 0001

Aufgabe 3: Berechne folgende Aufgaben, indem du alle Summanden ins Binärsystem überführst, dann im Binärsystem addierst und das Ergebnis schließlich ins Dezimalsystem zurück überführst. Kontrolliere Dein Ergebnis mit der dezimalen Rechnung.

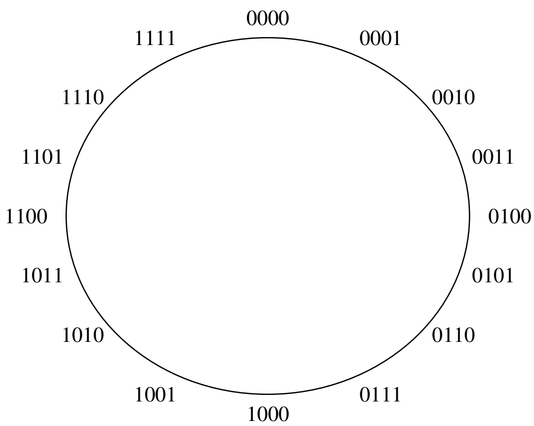
35 + 17　　119 + 31　　63 + 63 + 1

3.2 Negative Zahlen im Binärsystem

Schauen wir uns zuerst einmal vierstellige Binärzahlen (Nibbles) an. Insgesamt können mit einem Nibble 16 verschiedene Zahlen dargestellt werden: 0, 1, 2, ... , 15. Was passiert nun bei fortlaufender Addition von +1 ausgehend von der Zahl 0?

$$0 \xrightarrow{+1} 1 \xrightarrow{+1} 2 \xrightarrow{+1} 3 \xrightarrow{+1} 4 \xrightarrow{+1} \ldots \xrightarrow{+1} 14 \xrightarrow{+1} 15 \xrightarrow{+1} ???$$

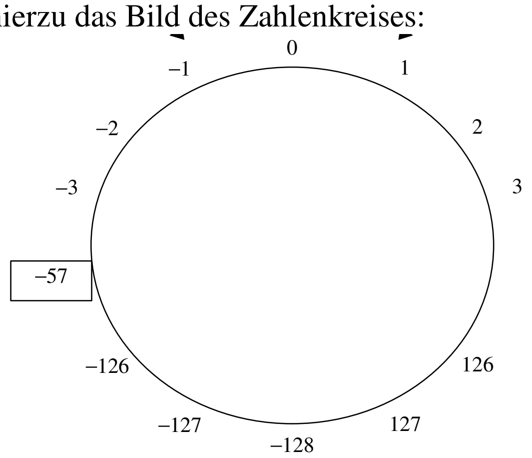
Addieren wir nun zur Zahl 1111_2 eine +1, so erhalten $(1)0000_2$, also die Zahl Null mit einem Überlauf. Man kommt also sozusagen wieder an den Anfang des Zahlenstrahls zurück. Man spricht deshalb von einem Zahlenkreis:



Vergleichen wir unser Zahlenmodell mit dem Zahlenstrahl: Um auf dem Zahlenstrahl die nächstgrößere Zahl zu erhalten (+1) gehen wir nach rechts, um die nächst kleinere Zahl zu erhalten (−1) nach links. Übertragen wir das Prinzip auf den Zahlenkreis, so müssten ausgehend von der 0 ganz oben die negativen Zahlen links, also entgegengesetzt dem Uhrzeigersinn angeordnet sein. Die -1_{10} entspräche demnach der Binärzahl 1111_2. Der -7_{10} demnach 1001_2.

Aufgabe 4: a) Addiere $-1_{10} + 1_{10}$ im Binärsystem.
b) Addiere $-7_{10} + 7_{10}$ im Binärsystem.
Stimmt unsere Vermutung?

Bis auf einen Überlauf, der bei der 4-Bit-Darstellung wegfällt, stimmt das Ergebnis. Also findet man die negative Zahl einer Binärzahl genau gegenüber auf dem Zahlenkreis. Die Frage die bleibt ist: Welcher Zahl entspricht die 1000_2? Es könnte −8 oder +8 bedeuten. Wenn du dir die negativen Zahlen anschaust, so stellst du fest, dass bei allen Zahlen das erste Bit gesetzt ist. Man nennt dieses Bit auch Vorzeichenbit. Sinnvollerweise sollte man dann die Zahl 1000_2 auch als negative Zahl, also als −8 interpretieren. Wir können also insgesamt mit einem Nibble Zahlen von −8 bis +7 darstellen. Was ist nun mit einer 8-Bit-Dualzahl? Zeichnen wir uns auch hierzu das Bild des Zahlenkreises:



Insgesamt kann man also mit einer 8-Bit-Dualzahl die Dezimalzahlen −128 bis +127 darstellen. Wie sieht nun aber die konkrete Codierung für die Zahl −57 aus? Eine Möglichkeit wäre, den Zahlenkreis komplett im Binärsystem aufzustellen und dann das Ergebnis abzulesen (gegenüber der Zahl +57). Dies wäre aber eine äußerst mühselige Angelegenheit. Eine andere Möglichkeit ist die folgende: Wir suchen zur folgenden Aufgabe den passenden zweiten Summanden:

Dezimal	**Binär**
57	0011 1001
+ (−57)	+
= 0	= (1) 0000 0000

Aufgabe 5: Versuche durch Probieren den zweiten Summanden herauszufinden.

Hast du es geschafft? Dann schaue dir einmal die beiden Zahlen 57 und −57 in ihrer Binärdarstellung an. Was kann man bezüglich ihrer einzelnen Bits sagen?
Genau diesen Zusammenhang kann man zur Berechnung der Darstellung einer negativen Zahl im Binärsystem verwenden.

Das Einerkomplement: Das Einerkomplement einer Binärzahl entsteht, indem jedes einzelne Bit „gekippt" wird, d. h. aus 0 wird 1 und aus 1 wird 0.

Beispiel: aus 0011 1001 entsteht das
Einerkomplement 1100 0110

Das Zweierkomplement: Das Zweierkomplement einer Binärzahl entsteht, indem zum Einerkomplement der Binärzahl die Zahl 1 hinzuaddiert wird.

Beispiel: aus 0011 1001 entsteht zuerst das
Einerkomplement 1100 0110
dann wird eine 1 hinzuaddiert:

$$\begin{array}{rr} & 1100\,0110 \\ + & \underline{0000\,0001} \\ = & 1100\,0111 \end{array}$$

Die mit dem Zweierkomplement erhaltene Zahl ist nun gerade die negative Zahl der Ursprungszahl. Konkret heißt das in diesem Beispiel: Will ich die Zahl –57 binär darstellen, so bilde ich das Zweierkomplement der Binärzahl zu +57.

Aufgabe 6: Teste, ob unsere Berechnung der Zahl –57 als Binärzahl mit Hilfe des Zweierkomplements in Ordnung ist.

Dezimal	**Binär**
57	0011 1001
+ (–57)	+ 1100 0111
= 0	=

Aufgabe 7: Berechne mit Hilfe des Zweierkomplements die Binärdarstellung der Zahlen
–17 –118 –126

In der anderen Richtung kann es allerdings auch mal interessant sein, zu einer gegebenen Binärzahl mit Vorzeichenbit die zugehörige Dezimalzahl zu berechnen. Nehmen wir uns dazu ein

Beispiel: Wir wollen zur Zahl **1011 1101**$_2$ die zugehörige vorzeichenbehaftete Dezimalzahl ermitteln. Da das erste Bit 1 ist, handelt es sich um eine negative Zahl.

Die einfachste Methode ist, die Gegenzahl zur Binärzahl, also das Zweierkomplement zu berechnen. Dadurch erhalten wir den Betrag der Binärzahl und können so auch die ursprüngliche Zahl angeben:
Das **Einer**komplement von $1011\,1101_2$ ist $0100\,0010_2$ und das **Zweier**komplement ist $0100\,0011_2$ (prüfe nach!), also die Zahl +67. Damit handelt es sich bei der Ausgangszahl $1011\,1101_2$ um die Dezimalzahl –67.

Aufgabe 8: Mache die Probe, indem du zur Zahl 67 als Binärzahl das Zweierkomplement bildest. du solltest die Ausgangszahl, also –67 als Binärzahl, erhalten.

3.3 Die Subtraktion von Binärzahlen

So, mit diesen Rechenregeln sind wir nun in der Lage, auch Subtraktionsaufgaben im Binärsystem zu lösen. Wollen wir z. B. die Aufgabe 127 − 19 im Binärsystem lösen, so ist die Aufgabe gleichzusetzen mit der Aufgabe 127 + (−19), d. h. wir berechnen zuerst die Binärdarstellungen von 127 und −19 und addieren schließlich beide Binärzahlen mit der bereits bekannten Methode. $127_{10} = 0111\ 1111_2$ und $-19_{10} = 1110\ 1101_2$ (prüfe nach!) und die Summe ergibt (1) $0110\ 1100_2$ mit Überlauf vor dem ersten Bit. $0110\ 1100_2$ ist aber gerade gleich 108_{10}.

Aufgabe 9: Prüfe zuerst Durch Rechnung das obige Beispiel nach. Berechne dann die Aufgaben im Binärsystem und mache anschließend im Dezimalsystem die Probe:

115 – 48 77 – 76 98 – 33 – 65 16 – 29

4 Punktrechnung im Binärsystem

Bis jetzt hast du die Strichrechnungen im Binärsystem kennen gelernt und weißt, wie du solche Rechnungen durchzuführen hast. du hast gesehen, dass die Subtraktion im Prinzip nichts anderes ist, als eine Addition vom Minuenden mit dem Zweierkomplement des Subtrahenden.
Vielleicht weißt du noch aus der Grundschule, dass die Multiplikation eigentlich nichts anderes ist, als eine groooße Addition. und die Division nichts anderes ist, als eine grooooße Subtraktion. In diesem Kapitel wirst du nun beide Rechenarten im Binärsystem kennen lernen.

4.1 Die Multiplikation von Binärzahlen

Neben der Addition und Subtraktion von Binärzahlen spielt die Multiplikation von Binärzahlen eine wesentliche Rolle. Von den Dezimalzahlen her kennen wir ein Verfahren, welches wir direkt auf das Binärsystem anwenden können. Allerdings liegt im Binärsystem mehr der Schwerpunkt auf dem Addieren als auf dem Multiplizieren, wie das folgende Beispiel zeigt:

Beispiel: Berechne das Produkt von $0000\ 1001_2$ und $0010\ 0111_2$ als vorzeichenlose Binärzahl.

Dezimal		Binär
9 · 23		00001001 · 00010111
18		00001001
27		00001001
		00001001
		00001001
1	Übertrag	11
207		0000000011001111

Bei der Multiplikation entstehen so bis zu acht Summanden, die anschließend addiert werden müssen, dagegen ist die Multiplikation sehr einfach.

Im Ergebnis siehst du nun zwei Bytes aneinandergereiht. In diesem Fall haben wir Glück und das zweite Byte bleibt mit Nullen gefüllt, so dass unser Ergebnis in ein Byte hineinpasst. Es kann aber auch sein, dass das vordere Byte benötigt wird – nämlich genau dann, wenn das Ergebnis größer als 255 wird. In einem solchen Fall spricht man bei dem vorderen der beiden

Ergebnisbytes vom **Hi-Byte** (oder auch High-Byte bzw. höherwertiges Byte) beim hinteren der beiden Ergebnisbytes vom **Low-Byte** (oder auch niederwertiges Byte).

Bei der Addition haben wir gesehen, dass auch zweistellige (01, 10, 11) oder sogar dreistellige (100, 101, 110, 111) Überträge auftauchen können. Bei der Multiplikation ist dies sogar üblich. Schaue noch einmal bei der Addition nach, wenn du vergessen hast, wie diese Überträge gehandhabt werden und löse anschließend die folgende

Aufgabe 10: Berechne das Produkt von $17 \cdot 15$ ($7 \cdot 31$, $53 \cdot 37$) als vorzeichenlose 16-Bit-Binärzahl mit Hi-Byte und Low-Byte.

4.2 Die Division von Binärzahlen

Du hast eben gesehen, dass die Multiplikation im Binärsystem genauso funktioniert, wie im Dezimalsystem. Deshalb ist es auch nicht verwunderlich, dass auch die schriftliche Division des Dezimalsystems auf das Binärsystem übertragbar ist. Am besten, wir schauen uns zuerst ein Beispiel an:

Beispiel: Berechne den Quotienten aus 229_{10} und 5_{10} im 8-Bit-Binärsystem.
Zuerst wandeln wir die Zahlen ins Binärsystem um. Wir lassen allerdings in den Ergebnissen die führenden Nullen weg, da sie uns das Leben bei der Berechung vereinfachen:
$229_{10} = 11100101_2$ und $5_{10} = 101_2$
Nun führen wir die Rechnung durch, wie im Dezimalsystem auch:

Dezimal	**Binär**
229 : 5 = 45 Rest 4	11100101 : 101 = 101101 Rest 100
–20	–101
29	100
25	–000
4	1000
	– 101
	111
	–101
	100
	–000
	1001
	– 101
	100

Die Schwierigkeit dieses Verfahrens liegt darin, dass du immer wieder Subtraktionen durchführen musst. So muss im Beispiel die Subtraktion 1000 – 101 oder 111 – 101 oder 1001 – 101 durchgeführt werden. Das hast du allerdings schon im Kapitel 3.3 kennen gelernt und sollte kein Problem für dich sein. Erinnere dich dazu nochmals an das Zweierkomplement.

Aufgabe 11: Berechne die folgenden Quotienten im Binärsystems und mache anschließend Probe im Dezimalsystem.
a) 119 : 3 **b)** 185 : 6 **c)** 225 : 15 **d)** 128 : 9

5 Andere Stellenwertsysteme

Nachdem du dich nun ausführlich mit dem Binärsystem und den Grundrechenarten in diesem Zahlensystem beschäftigt hast wird es nun Zeit, dass du sich auch noch mit anderen Zahlensystemen beschäftigst, die in der Informatik eine große Rolle spielen. Dazu gehört zweifelsohne das Hexadezimalsystem oder auch Sechszehnersystem genannt.

5.1 Das Hexadezimalsystem

Im Dezimalsystem stehen dir zehn Ziffern (0 bis 9) zur Verfügung, im Binärsystem stehen dir nur zwei Ziffern (0 und 1) zur Verfügung. Folglich gibt es im Sechszehnersystem insgesamt sechszehn Ziffern (0, 1, 2, 3, 4, 5, 6, 7, 8, 9 und A = 10, B = 11, C = 12, D = 13, E = 14, F = 15), welche für die Darstellung von Zahlen in diesem System verwendet werden können.

Wir schauen uns einmal die Stellenwerttafel für das Hexadezimalsystem an. Jeder Stellenwert des Hexadezimalsystems ist eine Potenz von 16. So wie im Dezimalsystem oder im Binärsystem wird nun einfach jede Ziffer mit dem entsprechenden Stellenwert multipliziert. Anschließend werden alle Ergebnisse zusammenaddiert.

Dezimalsystem						**Hexadezimalsystem**			
ZT 10^4 10000	T 10^3 1000	H 10^2 100	Z 10^1 10	E 10^0 1	Stellenwert	4096er 16^3 4096	256er 16^2 256	16er 16^1 16	1er 16^0 1
1	**4**	**9**	**4**	**0**		**3**	**A**	**5**	**C**
= **1** · 10000 + **4** · 1000 + **9** · 100 + **4** · 10 + **0** · 1 = 14940						= **3** · 4096 + **10** · 256 (A = 10) + **5** · 16 + **12** · 1 (C = 12) = 14940			

Eigentlich solltest du nun bereits in der Lage sein, eine Zahl aus dem Hexadezimalsystem in das Dezimalsystem umzurechnen.

Aufgabe 12: Überführe die folgenden Hexadezimalzahlen ins Dezimalsystem:
$1B9F_{16}$ $ABCD_{16}$ 1234_{16}

Kommen wir zur Umwandlung einer Dezimalzahl in eine Hexadezimalzahl. Hierbei muss man ausgehend vom höchsten Stellenwert des Hexadezimalsystems probieren, ob der Stellenwert in der Dezimalzahl enthalten ist, oder nicht. Ist er enthalten, so steht im Zweiersystem an der entsprechenden Stelle, wie oft der Stellenwert enthalten ist und die Dezimalzahl wir um das Vielfache des Stellenwerts erniedrigt. Das hört sich komplizierter an, als es ist. Schauen wir uns das Beispiel an:

Beispiel: $39722_{10} = ???_{16}$

- Der erste Stellenwert (4096) ist in 39722 insgesamt **9** Mal enthalten, also 1. Stelle **9**
  Jetzt rechnen wir: 39722 – **9** · 4096 = 2858.
- Der zweite Stellenwert (256) ist in 2858 insgesamt **11** Mal enthalten, also 2. Stelle **11** = B,
  Jetzt rechnen wir 2858 – **11** · 256 = 42
- Der dritte Stellenwert (16) ist in 42 insgesamt **2** Mal enthalten, also 3. Stelle **2**
  Jetzt rechnen wir 42 – **2** · 16 = 10
- Der vierte Stellenwert (1) ist in 10 insgesamt **10** Mal enthalten, also 4. Stelle **10** = A

Das Verfahren ist beendet und unsere zugehörige Hexadezimalzahl lautet 9B2A. Übe dieses Verfahren an der folgenden Aufgabe:

Aufgabe 13: Überführe die folgenden Dezimalsystem ins Hexadezimalsystem.
15114_{10} 4095_{10} 55555_{10}

5.2 Zusammenhang Binärsystem – Hexadezimalsystem

Vermutlich hast du bereits in der Stellenwerttafel gesehen, dass alle Stellenwerte des Hexadezimalsystems auch Stellenwerte des Binärsystems sind. Die Umwandlung vom Binärsystem in das Hexadezimalsystem (und auch umgekehrt) ist deshalb ganz einfach.

Immer 4 Bit im Binärsystem entsprechen einer Ziffer im Hexadezimalsystem: So ist z. B. die Zahl Binärzahl $1010\ 1101\ 1111\ 0010_2$ im Hexadezimalsystem gleich $ADF3_{16}$, denn schaut man sich immer vier Bit zusammen an, dann erkennt man:

Binärsystem:	1010	1101	1111	0010
Dezimalsystem:	= 10	= 13	= 15	= 3
Hexadezimalsystem:	A	D	F	3

Du siehst, dass man die Ziffern im Hexadezimalsystem ganz leicht ausrechnen kann, wenn die Zahl im Binärsystem vorliegt. Andersherum ist es aber genauso einfach möglich, die Binärzahl zu einer Hexadezimalzahl zu berechnen. So ist z. B. die Hexadezimalzahl $5B69_{16}$ im Binärsystem gleich der Zahl $0101\ 1011\ 0110\ 1001_2$, da auch hier wieder jeweils vier Bit einer Ziffer im Hexadezimalsystem entsprechen. Übe die Umwandlung an den folgenden Aufgaben:

Aufgabe 14: Wandle ins Hexadezimalsystem um.
$1011\ 0110\ 0111\ 1111_2$ $1010\ 0000\ 0001\ 1110_2$ $0101\ 1100\ 0011\ 1011_2$

Aufgabe 15: Wandle ins Binärsystem um.
$1B9F_{16}$ $ABCD_{16}$ 1234_{16}

5.3 Das Oktalsystem

Im Oktalsystem gibt es nur acht Ziffern (0, 1, 2, 3, 4, 5, 6 und 7). Die Stellenwerte der Stellenwerttafel entsprechen den 8er-Potenzen $8^0 = 1$, $8^1 = 8$,$8^2 = 64$, usw.
Schauen wir uns das wieder in der Stellenwerttafel an:

Dezimalsystem					**Oktalsystem**			
T 10^3 1000	H 10^2 100	Z 10^1 10	E 10^0 1	Stellenwert	4096er 8^3 512	256er 8^2 64	16er 8^1 8	1er 8^0 1
1	**6**	**3**	**9**		**3**	**1**	**4**	**7**
= **1** · 1000 + **6** · 100 + **3** · 10 + **9** · 1 = 1639					= **3** · 512 + **1** · 64 + **4** · 8 + **7** · 1 = 1639			

Da du nun schon viel Erfahrung mit Stellenwertsystemen hast solltest du nun problemlos die folgenden Aufgaben lösen:

Aufgabe 16: Wandle ins Dezimalsystem um.

7561_8 3256_8 7777_8

Aufgabe 17: Wandle ins Oktalsystem um.

7561_{10} 3256_{10} 7777_{10}

Aufgabe 18: Und nun wollen wir uns noch einmal alle Stellenwertsysteme anschauen.
Wandle dazu die folgenden Zahlen in alle dir bekannten Stellenwertsysteme um.

456_{10}

$A67C_{16}$

$0011\ 1111\ 1000\ 0011_2$

7465_8