

Alexander Halles



... working with boundless passion

Zahlensysteme

Stand: 26.01.2004

- Inhalt -

1. Die verschiedenen Zahlensysteme und Umwandlungen zwischen diesen	3
1.1 Dezimalzahlensystem	3
1.2 Das Dualzahlensystem	4
1.2.1 Umwandlung einer Dezimalzahl in eine Dualzahl	4
1.2.2 Umwandlung einer Dualzahl in eine Dezimalzahl	5
1.3 Das Oktalsystem	5
1.3.1 Umwandlung einer Dezimalzahl in eine Oktalzahl	5
1.3.2 Umwandlung einer Oktalzahl in eine Dezimalzahl	6
1.4 Das Hexadezimalsystem (Sedezimalsystem)	6
1.4.1 Umwandlung einer Dezimalzahl in eine Hexadezimalzahl	6
1.4.2 Umwandlung einer Hexadezimalzahl in eine Dezimalzahl	7
1.5 Umwandlung einer Dezimalzahl in ein beliebiges Zahlensystem	7
1.6 Umwandlung einer Zahl eines beliebigen Zahlensystems in eine Dezimalzahl	7
1.7 Umwandlung von beliebigen Zahlensystemen in beliebige Zahlensysteme	7
2. Genauere Betrachtung des Dualzahlensystems	8
2.1 Grundrechenarten im Dualzahlensystem	8
2.1.1 Addition	8
2.1.2 Subtraktion	9
2.1.3 Multiplikation	10
2.1.4 Division	11
2.2 Darstellung der Nachkommastellen im Binärsystem	12
2.2.1 Umwandlung der Nachkommastellen einer Dezimalzahl in eine Dualzahl	12
2.2.2 Umwandlung von dualen Nachkommastellen in Dezimalnachkommastellen	14
2.3 Binäre Codierung (BCD, BCO, BCH)	15
2.3.1 Binär Codierte Dezimalzahl (BCD)	15
2.3.1.1 Dezimalzahl in eine BCD konvertieren	15
2.3.1.2 BCD in eine Dezimalzahl konvertieren	15
2.3.2 Binär codierte Oktalzahl (BCO)	16
2.3.3 Binär codierte Hexadezimalzahl (BCH)	16
2.3.3.1 Hexadezimalzahl in eine BCH überführen	16
2.3.3.2 Binär codierte Hexadezimalzahl in eine Hexadezimalzahl konvertieren	17
2.4 Darstellung negativer Binärzahlen und das Rechnen damit	17
2.4.1 Vorzeichenbit - Betragsvorzeichenform	17
2.4.2 Rechnen mit der Komplementdarstellung	18
2.4.2.1 Das Einer-Komplement	18
2.4.2.2 Das Zweier-Komplement	19
2.4.2.3 Addition und Subtraktion mit dem Zweier-Komplement	20
Anhang A : Ziffernumwandlung (BCD, BCO, BCH)	23
Anhang B : Stellenwertsystem	23

1. Die verschiedenen Zahlensysteme und Umwandlungen zwischen diesen

Am Anfang ein paar Definitionen:

Ziffer: Eine Ziffer ist ein Element einer Menge von Symbolen zur Darstellung von Zahlen.

Zahl: Eine Zahl ist eine gewisse Anordnung von Ziffern, die in einem vereinbarten Zahlensystem einen bestimmten Wert darstellen.

Kennzeichen eines Zahlensystems sind:

- Ein Zeichenvorrat, der die Menge der zulässigen Ziffern darstellt (z. B. in unserem gebräuchlichen Dezimalsystem: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9).
- Ein Bildungsgesetz, das die Anordnung für die Ziffern und die Ermittlung des Wertes der Zahl genau festlegt.

Bildungsgesetze können z. B. auf einem Ziffernwertsystem basieren, bei dem der Wert der Zahl ausschließlich durch die Ziffer bestimmt wird, wie bei den römischen Zahlen, oder sie können auf einem Stellenwertsystem basieren, bei dem die Stelle einer Ziffer innerhalb der Zahl den Wert der Zahl ausdrückt, wie bei unserem Dezimalsystem.

Die Darstellung von Zahlen durch Ziffern wird Codierung genannt.

1.1 Dezimalzahlensystem

Im Alltag verwenden wir, ohne es konkret zu registrieren, das Dezimalsystem. Dieses Zahlensystem beruht auf der Basis 10. Nehmen wir als Beispiel die Dezimalzahl „1358“. Man kann diese Zahl nun wie folgt zerlegen:

$$1 \cdot 10^3 + 3 \cdot 10^2 + 5 \cdot 10^1 + 8 \cdot 10^0$$

Berechnet man diese Summe mit einem Taschenrechner, so zeigt dieser als Ergebnis 1358 an, was zeigt, dass unsere Zerlegung in Potenzen richtig war.

Man erkennt also, dass jede Stelle eine bestimmte Wertigkeit besitzt. Man beginnt bei der äußersten rechten Zahl mit der Potenz 10^0 , die folgende Stelle erhält die Potenz 10^1 , usw. Jede Stelle wird also, von rechts nach links ansteigend, mit einer Potenz bewertet und diese Stelle wird dann mit dem Vorfaktor multipliziert.

In der Schule lernt man das Dezimalzahlensystem nicht mit Potenzen kennen, sondern man bezeichnet die Wertigkeiten der Stellen mit Worten (Einsertelle, Zehnerstelle, Hunderterstelle, Tausenderstelle).

Für eine genauere, schematische Darstellung des Stellenwertsystems vgl. [Anhang B](#).

Für den Vorfaktor sind zehn verschiedene Werte möglich (0 - 9). Man bezeichnet diese möglichen Werte auch als den Zeichenvorrat des Systems (vgl. [Abschnitt 1](#)).

Der Wert vor der Potenz zeigt an, wie oft man den Stellenwert multiplizieren muss.

Unsere alltägliche Schreibweise ist also eine Kurzform der Summenformel (vgl. [Anhang B](#)).

Um Verwechslungen mit anderen Zahlensystemen zu vermeiden, setzt man als Index oft das verwendete Zahlensystem.

Beispiel 1:

1234_{10}

Das bedeutet, dass die Zahlenfolge 1234 mit den Potenzen des Dezimalzahlensystems bewertet werden soll, also $1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0$.

1.2 Das Dualzahlensystem

Das Dualzahlensystem, oder auch Binärsystem genannt, findet Verwendung in der elektronischen Signal- und Datenverarbeitung. Auch der PC arbeitet grundsätzlich nur mit Binärdaten.

Der Zeichenvorrat des Dualsystems beinhaltet nur die beiden Ziffern Null und Eins, jede Stelle wird mit einer Potenz zur Basis Zwei bewertet, wieder von rechts nach links stellenweise ansteigend.

Das Dualzahlensystem wird in der Informationstechnologie daher verwendet, da mit den beiden Ziffern Null und Eins die Zustände „Spannung vorhanden“ und „keine Spannung vorhanden“ repräsentiert und abgefragt werden können.

Auch hier wird wieder ein **Index** zur Unterscheidung von anderen Zahlensystemen verwendet:

1010_2

Man muss hier nur beachten, dass diese Zahl nicht die Zahl „Eintausendzehn“, sondern die Zahl „Eins Null Eins Null“ ist, also einzeln gesprochen, von links nach rechts gelesen. Durch Angewöhnung dieser Sprechweise wird, wie bei der Verwendung des Index, eine Verwechslung der Zahlensysteme vermieden.

1.2.1 Umwandlung einer Dezimalzahl in eine Dualzahl

Eine Umwandlungsmethode ist die sogenannte „fortgesetzte Divisionsmethode“, oder auch „Resteverfahren“ genannt.

Bei dieser Methode wird die Dezimalzahl durch die Basis des Dualsystems, also Zwei, geteilt und der Rest notiert, auch wenn der Rest null sein sollte. Das Ergebnis dieser Division wird wiederum durch Zwei geteilt und der Rest wiederum notiert. Die Division wird so lange fortgesetzt, bis das Ergebnis null ist. Die Reste, die ja nur null und eins sein können, ergeben dann die Dualzahl.

Beispiel 2:

Wir möchten die Zahl 85_{10} in eine Dualzahl umwandeln. Wir teilen also die Zahl 85_{10} durch Zwei:

85	:	2	=	42	Rest	1
42	:	2	=	21	Rest	0
21	:	2	=	10	Rest	1
10	:	2	=	5	Rest	0
5	:	2	=	2	Rest	1
2	:	2	=	1	Rest	0
1	:	2	=	0	Rest	1

$85_{10} = 1010101_2$

Es muss also noch beachtet werden, dass der **Rest der letzten Division** die erste (linke) Ziffer der Dualzahl bezeichnet.

1.2.2 Umwandlung einer Dualzahl in eine Dezimalzahl

Haben wir im obigen Beispiel 2 richtig gerechnet? Machen wir also die Probe. Wir bewerten jede Stelle der Dualzahl mit einer Potenz zur Basis Zwei, wie wir es auch im **Abschnitt 1.1** mit der Dezimalzahl getan haben, nur eben nicht zur Basis Zehn, sondern zur Basis Zwei.

$$1010101_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Man muss nun einfach diese mit den Potenzen bewerteten Stellen **aufsummieren** und wir gelangen wieder zur uns vertrauten Dezimalzahl, wobei die bewerteten Stellen, die mit Null multipliziert werden, natürlich sofort wegfallen.

$$1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^0 = 64 + 16 + 4 + 1 = \underline{85}_{10}$$

Wir haben nun also durch Nachrechnen geprüft, ob wir im Beispiel aus **Abschnitt 1.2.1** richtig gerechnet haben.

1.3 Das Oktalsystem

Das Oktalsystem findet hauptsächlich Anwendung in der Mikroprozessortechnik.

Dieses System beinhaltet den Zeichenvorrat von acht Ziffern (0, 1, 2, 3, 4, 5, 6, 7). Alle Stellenwerte sind Potenzen mit der Basis Acht.

Deshalb wird bei der Schreibweise auch der Index Acht verwendet (2043_8), um eine Oktalzahl zu kennzeichnen.

1.3.1 Umwandlung einer Dezimalzahl in eine Oktalzahl

Die Umwandlung einer Dezimalzahl in eine Oktalzahl funktioniert analog zur Umwandlung einer Dezimalzahl in eine Dualzahl. Der Unterschied ist lediglich, dass die fortgesetzte Division nun durch Acht erfolgt.

Beispiel 3:

Umwandlung der Dezimalzahl 1059_{10} in die entsprechende Oktalzahl. Wir dividieren also 1059 fortgesetzt durch Acht:

1059	:	8	=	132	Rest	3	
132	:	8	=	16	Rest	4	
16	:	8	=	2	Rest	0	
2	:	8	=	0	Rest	2	

$1059_{10} = \underline{2043}_8$

Wie bei der Umwandlung der Dezimalzahl in eine Dualzahl ist auch hier zu beachten, dass der Rest der letzten Division die erste Ziffer der Zahl im neuen System bildet.

1.3.2 Umwandlung einer Oktalzahl in eine Dezimalzahl

Nun werden wir wieder den Beweis für die Richtigkeit unserer Rechnung aus Beispiel 3 antreten und die Oktalzahl 2043_8 in eine Dezimalzahl umwandeln.

Dazu bewerten wir wieder jede Stelle mit einer Potenz, diesmal zur Basis Acht, von der rechten Ziffer beginnend (8^0) bis zur linken Ziffer (8^3). Die rechte Ziffer stellt immer die niederwertigste Ziffer dar und wird mit $(\text{Basis})^0$ bewertet.

$$2043_8 = 2 \cdot 8^3 + 0 \cdot 8^2 + 4 \cdot 8^1 + 3 \cdot 8^0 = 1024 + 0 + 32 + 3 = \underline{1059}_{10}$$

Wir sehen, dass wir im das Beispiel 3 richtig gerechnet haben.

1.4 Das Hexadezimalsystem (Sedezimalsystem)

Das Hexadezimalsystem, auch Sedezimalsystem genannt, findet, wie die Oktal-Codierung, hauptsächlich Anwendung in der Mikroprozessortechnik.

Es ist auf der Basis 16 aufgebaut und beinhaltet daher einen Zeichenvorrat von 16 Zeichen. Unser Zahlensystem besitzt jedoch nur zehn Ziffern (0-9), deshalb werden die fehlenden sechs Ziffern durch die Buchstaben A - F aufgefüllt. Es gelten also für die letzten sechs Ziffern des Hexadezimalsystems:

10	à	A		13	à	D
11	à	B		14	à	E
12	à	C		15	à	F

(Tabelle 1.4a)

Für diejenigen, die sich jetzt fragen, warum bei einem 16-stelligen System die letzte Zahl 15 ist, sei nochmals erwähnt, dass immer mit der Null als erste Ziffer begonnen wird!

1.4.1 Umwandlung einer Dezimalzahl in eine Hexadezimalzahl

Für diese Umwandlung kann wieder das Verfahren der fortgesetzten Division angewandt werden. Dieses Mal wird durch die Zahl 16 dividiert. Die Reste ergeben wieder die Ziffern der Zahl im neuen Zahlensystem, also hier der Hexadezimalzahl.

Beispiel 4:

Wir wollen nun die Zahl 6398_{10} des Dezimalsystems in die entsprechende Hexadezimalzahl konvertieren. Wir beginnen also mit der fortgesetzten Division durch Sechzehn:

6398	:	16	=	399	Rest	14	→
399	:	16	=	24	Rest	15	→
24	:	16	=	1	Rest	8	→
1	:	16	=	0	Rest	1	→

$6398_{10} = \underline{1\ 8\ F\ E}_{16}$

Genauso wie bei Dezimal-Dual- und Dezimal-Oktal-Umwandlung wird wieder der letzte Rest die erste Ziffer der Zahl im neuen Zahlensystem. So komisch es auch anmuten mag, aber „18FE“ ist eine Zahl. Zwar eine Zahl in einem uns nicht sehr geläufigen System, aber immerhin eine gültige Zahl. Hier wird nochmals deutlich, wie wichtig der Index hinter den jeweiligen Zahlen ist.

Des weiteren sieht man in diesem Beispiel die Verwendung der Buchstaben. Die Zahl 14 wird einfach durch den Buchstaben E ersetzt und die Zahl 15 durch den Buchstaben F (vgl. [Tabelle 1.4a](#))

1.4.2 Umwandlung einer Hexadezimalzahl in eine Dezimalzahl

Wie zu erwarten war, werden wir jetzt wieder unsere Berechnung überprüfen. Die Umrechnung einer Hexadezimalzahl funktioniert wieder nach dem gleichen Schema. Stellen mit Potenzen zur Basis 16 bewerten, mit dem Vorfaktor multiplizieren und aufsummieren.

$$18FE_{16} = 1 \cdot 16^3 + 8 \cdot 16^2 + 15 \cdot 16^1 + 14 \cdot 16^0 = 4096 + 2048 + 240 + 14 = \underline{6398}_{10}$$

Wieder stimmt unsere Umrechnung. Beachtet werden muss, dass nun der umgekehrte Fall der Ersetzung vorgenommen werden muss. Jeder Buchstabe der Hexadezimalzahl muss mit dem zugehörigen Wert vertauscht werden, vgl. [Tabelle 1.4a](#).

1.5 Umwandlung einer Dezimalzahl in ein beliebiges Zahlensystem

Das in den [Abschnitten 1.2.1, 1.3.1 und 1.4.1](#) vorgestellte Verfahren der fortgesetzten Division kann auf jedes beliebige Zahlensystem angewandt werden. Die Dezimalzahl wird fortlaufend durch die Basis des gewählten Ziel-Zahlensystems dividiert und die dabei entstehenden Reste notiert. Die Reste werden von der letzten Division bis zur ersten Division, also quasi rückwärts, aufgeschrieben und ergeben so die äquivalente Zahl im neuen Zahlensystem.

Beachtet werden muss, dass das letzte Divisionsergebnis immer die Null sein muss. Man neigt dazu schon vorher abzubrechen, wenn z. B. „2 : 2 = 1“ auf dem Papier steht und vergisst sozusagen eine Ziffer der umgerechneten Zahl. Die Division muss fortgesetzt werden, bis als [Divisionsergebnis Null](#) steht!!

1.6 Umwandlung einer Zahl eines beliebigen Zahlensystems in eine Dezimalzahl

Genauso wie die fortgesetzte Division besitzt auch die Summation der Stellenwerte Allgemeingültigkeit. Die Basis der Potenz ist immer das Zahlensystem aus dem umgewandelt wird. Die Stellenbewertung erfolgt immer von rechts nach links, mit $(\text{Basis})^0$ beginnend.

Die Summe aus den Produkten von Vorfaktor und jeweiliger Potenz ergibt dann die gesuchte Dezimalzahl, vgl. [Beispiele 2-4](#).

1.7 Umwandlung von beliebigen Zahlensystemen in beliebige Zahlensysteme

Ist das Start- und das Zielsystem beliebig, also nicht das Dezimalsystem, so empfiehlt sich die Umwandlung in zwei Schritten durchzuführen und das Dezimalsystem als Zwischenschritt zu benutzen.

Die Zahl des Startsystems wird per Summationsregel (vgl. **Abschnitt 1.6**) in das Dezimalsystem umgewandelt.

Die entsprechende Dezimalzahl wird nun mit Hilfe der fortgesetzten Division (vgl. **Abschnitt 1.5**) in das Ziel-Zahlensystem umgewandelt.

2. Genauere Betrachtung des Dualzahlensystems

Wie in **Abschnitt 1.2** erwähnt ist das Dualzahlensystem besonders wichtig in der Informationstechnik und verdient es deshalb eingehender betrachtet zu werden.

2.1 Grundrechenarten im Dualzahlensystem

2.1.1 Addition

Binäre Zahlen werden nach folgenden Rechenregeln addiert:

$$\begin{array}{l} 0 + 0 = 0 \\ 0 + 1 = 1 \\ 1 + 0 = 1 \\ 1 + 1 = 0 \Rightarrow \text{Übertrag der 1 zur nächsten Stelle} (\Rightarrow 1+1 = 10) \end{array}$$

Dieser Übertrag wird in der Literatur auch oft als „Carry“ bezeichnet

Beispiel 5:

Addition der beiden binären Zahlen 1011_2 und 111_2

E	D	C	B	A	
↓	↓	↓	↓	↓	
	1	0	1	1	
		1	1	1	
1	1	1	1		←Carry
1	0	0	1	0	

Das Beispiel zeigt, wie der Übertrag funktioniert. Stehen zwei Einsen übereinander (Spalte A), so steht als Ergebnis in dieser Spalte eine Null und dafür wird eine Eins als Übertrag in die nächste Spalte (Spalte B) geschrieben („Eins gemerkt“).

Jetzt müssen in der Spalte B die beiden Einsen addiert werden, die von den binären Zahlen stammen, was als Ergebnis eine Null und einen Übertrag von Eins zur Folge hätte, wenn da nicht noch die eine Eins wäre, die vom Übertrag aus Spalte A herrührt, die noch zum Ergebnis addiert werden muss. Deshalb steht im Ergebnis von Spalte B eine Eins und es erfolgt ein Übertrag in Spalte C.

In Spalte C müssen nun die beiden Ziffern Null und Eins der Binärzahlen addiert werden, was eine Eins ergibt. Addiert man jedoch zu dieser Eins noch den Übertrag aus Spalte B, so steht im Ergebnis von Spalte C eine Null und es erfolgt nach der Rechenregel ($1+1=0 \rightarrow$ Übertrag 1) ein Übertrag in Spalte D.

In Spalte D muss nun die Ziffer 1 der ersten Dualzahl und der Übertrag aus C addiert werden, was eine Null im Ergebnis und wiederum einen Übertrag zur Folge hat.

In Spalte E steht letztlich nur der Übertrag. Man kann jedoch in Gedanken beliebig viele führende Nullen hinzufügen. Man addiert in Spalte E letztlich $0+0+1$. Als Ergebnis dieser Addition steht in Spalte E eine Eins.

Überprüfen wir nun unsere Rechnung, in dem wir die Zahlen ins Dezimalsystem konvertieren.

$$1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = \underline{11}_{10}$$

$$111_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 4 + 2 + 1 = \underline{7}_{10}$$

$$10010_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 16 + 2 = \underline{18}_{10}$$

Die Addition, die wir binär durchgeführt haben, stimmt also ($11_{10} + 7_{10} = 18_{10}$).

2.1.2 Subtraktion

Binäre Zahlen werden nach folgenden Rechenregeln subtrahiert:

$$0 - 0 = 0$$

$$0 - 1 = 1 \Rightarrow \text{von der nächsten Stelle nimmt man eine 1 } (\Rightarrow 10 - 1 = 1)$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Der Übertrag bei der Subtraktion wird auch oft als „Borrow“ bezeichnet.

Der Vorgang [$0 - 1 = 1 \Rightarrow$ Wegnahme eine 1 von der höheren Stelle] ist vergleichbar mit dem schriftlichen Subtrahieren bei Dezimalzahlen.

Betrachten wir den einfachen Fall, dass wir im Dezimalsystem die Zahl 268_{10} von der Zahl 734_{10} abziehen wollen.

Schriftlich sieht das dann folgendermaßen aus:

$$\begin{array}{r} \text{C} \quad \text{B} \quad \text{A} \\ \downarrow \quad \downarrow \quad \downarrow \\ 7 \quad 3 \quad 4 \\ 2 \quad 6 \quad 8 \\ \hline 1 \quad 1 \\ 4 \quad 6 \quad 6 \end{array}$$

Wenn wir in Spalte A ($4 - 8$) rechnen wollen stehen wir vor dem Problem, dass diese Rechnung zu einer negativen Zahl führen würde. Deshalb „borgen“ wir uns von der nächsten Stelle eine Eins und rechnen stattdessen ($14 - 8$), was zum Ergebnis 6 führt. Die Eins, die wir uns „geborgt“ haben, müssen wir, wie alles im Leben, wieder zurück zahlen, deshalb schreiben wir sie als „gemerkt“ in die Zwischenzeile der Spalte B. Das heißt, wir müssen jetzt in Spalte B eins mehr abziehen.

Nun stehen wir in Spalte wieder vor dem Problem, dass ($3 - (6 + 1)$), also ($3 - 7$), eine negative Zahl ergibt. Deshalb „borgen“ wir uns wieder von der nächsten Stelle eine Eins und rechnen ($13 - 7 = 6$); die „geborgte“ Eins zahlen wir auch gleich wieder zurück und schreiben deshalb eine Eins in die Zwischenzeile der Spalte C.

In der Spalte C müssen wir nun $(2 + 1)$ von (7) abziehen, was 4 ergibt und somit lautet das Endergebnis 466.

Dies, was wohl den meisten bekannt sein dürfte, wurde an dieser Stelle so genau ausgeführt, weil das selbe Prinzip bei der Subtraktion von Binärzahlen gilt.

Beispiel 6:

Rechnen wir nun $1110_2 - 1001_2$:

$$\begin{array}{rcccc}
 & D & C & B & A \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
 & 1 & 1 & 1 & 0 \\
 & 1 & 0 & 0 & 1 \\
 \hline
 & & & 1 & \\
 & 0 & 1 & 0 & 1 \quad \leftarrow \text{Borrow}
 \end{array}$$

In Spalte A stehen wir gleich vor dem Problem $(0-1)$. Wir „borgen“ uns also von der Spalte B eine Eins und rechnen $(10 - 1 = 1)$. Die „geborgte“ Eins zahlen wir in Spalte B gleich wieder zurück und müssen statt $(1-0)$ nun $(1-1)$ rechnen, was nach der Rechenregel Null ergibt. Spalte C und D funktionieren nun genau nach Rechenregel.

Überprüfen wir wieder das Ergebnis, indem wir die Zahlen in das Dezimalsystem konvertieren¹:

$$\begin{array}{l}
 1110_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8 + 4 + 2 + 0 = \underline{14}_{10} \\
 1001_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 0 + 1 = \underline{9}_{10} \\
 101_2 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 0 + 1 = \underline{5}_{10}
 \end{array}
 \left. \vphantom{\begin{array}{l} 1110_2 \\ 1001_2 \\ 101_2 \end{array}} \right\} 14_{10} - 9_{10} = 5_{10}$$

2.1.3 Multiplikation

Binäre Zahlen werden nach folgenden Rechenregeln multipliziert:

$$\begin{array}{l}
 0 \cdot 0 = 0 \\
 0 \cdot 1 = 0 \\
 1 \cdot 0 = 0 \\
 1 \cdot 1 = 1
 \end{array}$$

¹ 101_2 : genauso, wie man führende Nullen beliebig hinzufügen kann, ohne den Sinn (den Wert) der Zahl zu verändern, kann man führende Nullen auch weglassen ohne Veränderungen hervorzurufen (aber nur führende Nullen!!)

Beispiel 7:

Wir wollen die beiden Binärzahlen 111_2 und 110_2 miteinander multiplizieren

$$\begin{array}{r} 111 \cdot 110 \\ \hline 111 \\ 1110 \\ 0000 \\ \hline 101010 \end{array}$$

Die Multiplikation an sich erfolgt genauso wie beim schriftlichen Multiplizieren von Dezimalzahlen. Dann erfolgt wieder die binäre Addition.

Wer hier, z. B. bei den Überträgen, Schwierigkeiten hat oder sich Unsicher fühlt, der möge nochmals **Abschnitt 2.1.1** durcharbeiten.

2.1.4 Division

Binäre Zahlen werden nach folgenden Rechenregeln dividiert:

1	:	1	=	1
0	:	1	=	0

Die Division von Dualzahlen wird auf die Subtraktion zurückgeführt (analog zum schriftlichen Dividieren im Dezimalzahlensystem).

Man subtrahiert den Divisor von einer Anzahl der Stellen des Dividenden, die so groß sein muss, dass das Ergebnis Eins wird.

Zum Ergebnis dieser Differenz zieht man die nächst niedrigere Stelle des Dividenden herab und subtrahiert wieder den Divisor von dieser (neu gewonnenen) Zahl.

Dieses Verfahren wird so lange angewendet, bis alle Stellen des Dividenden herabgezogen wurden.

Eine eventuell verbleibende Zahl, die kleiner ist als der Divisor wird entweder als Rest angehängt oder es wird mit zusätzlich gedachten Nullen als Dualbruch weiter gerechnet.

Beispiel 8:

Wir wollen nun die Binärzahl 101011111_2 durch 1101_2 dividieren.

$$\begin{array}{r}
 101011111 : 1101 = 11011 \\
 \underline{-1101} \\
 0100001 \\
 \underline{-1101} \\
 10011 \\
 \underline{-1101} \\
 1101 \\
 \underline{-1101} \\
 0000
 \end{array}$$

Wir überprüfen nun wieder das Ergebnis anhand der Konvertierung in Dezimalzahlen:

$$101011111_2 = 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = \\
 = 256 + 0 + 64 + 0 + 16 + 8 + 4 + 2 + 1 = \underline{351}_{10}$$

$$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 0 + 1 = \underline{13}_{10}$$

$$11011_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 8 + 2 + 1 = \underline{27}_{10}$$

2.2 Darstellung der Nachkommastellen im Binärsystem

Aus dem täglichen Leben kennen wir Zahlen, die sogenannte Nachkommastellen besitzen. Ein Sprinter läuft beispielsweise 100m in 9,85 Sekunden.

Jede Stelle hinter dem Komma wird, analog zu den Stellen vor dem Komma, mit einer bestimmten Potenz bewertet, die Ziffer an dieser Stelle ist wieder der Vorfaktor für den Stellenwert. Der Exponent dieser Potenz ist nun negativ.

2.2.1 Umwandlung der Nachkommastellen einer Dezimalzahl in eine Dualzahl

Die Umwandlung wird nach der sogenannten **Multiplikationsmethode** ausgeführt. Die Nachkommastellen werden notiert und mit einer Null vor dem Komma versehen und dann mit dem Faktor Zwei multipliziert. Die Vorkommastelle des Ergebnisses stellt die Ziffer der Dualzahl dar. Mit dem Nachkommateil wird nun die Multiplikation fortgesetzt. Dem Nachkommateil wird bei jeder neuen Multiplikation eine Null vor das Komma gesetzt, so dass die Rechnung **IMMER** $[0,xyz \cdot 2 = \dots]$ lautet. Die „abgezwickte“ Vorkommastelle der ersten Multiplikation ist auch die erste Ziffer der Nachkommastelle der Dualzahl.

Das Verfahren wird solange fortgesetzt, bis entweder der Rest der Multiplikation Null wird, oder die gewünschte Genauigkeit der Dualzahl erreicht ist.

BEACHTE:

Die Umwandlung der Nachkommastellen ist nicht immer eindeutig, sondern stellt in bestimmten Fällen nur eine Näherung dar (die jedoch beliebig genau sein kann)!!

Um dies zu vermeiden wird oft eine andere Form der Codierung verwendet, die sog. BCD-Codierung, vgl. hierzu **Abschnitt 2.3.1.**

Beispiel 9:

Wir möchten die Zahl $253,4375_{10}$ in das Binärsystem umwandeln.

Als erstes rechnen wir mit der fortgesetzten Division die Zahl 23_{10} ins Binärsystem um und anschließend benutzen wir die Multiplikationsmethode um die Nachkommastellen umzuwandeln.

253	:	2	=	126	Rest	1
126	:	2	=	63	Rest	0
63	:	2	=	31	Rest	1
31	:	2	=	15	Rest	1
15	:	2	=	7	Rest	1
7	:	2	=	3	Rest	1
3	:	2	=	1	Rest	1
1	:	2	=	0	Rest	1

$$\Rightarrow 253_{10} = \underline{\underline{11111101_2}}$$

0,4375	·	2	=	0,875	à	0
0,875	·	2	=	1,75	à	1
0,75	·	2	=	1,5	à	1
0,5	·	2	=	1,00	à	1
0						

$$\Rightarrow 0,4375_{10} = 0,0111_2$$

Zusammengefasst ergeben die Berechnungen: $253,4375_{10} = \underline{\underline{11111101,0111_2}}$

Beispiel 10:

Wir rechnen nun die Zeit, die unser Sprinter für 100m gebraucht hat (9,85), in eine Dualzahl um. Wie im Beispiel 9 wird die fortgesetzte Division für den Vorkommateil und die fortgesetzte Multiplikation für den Nachkommateil angewandt.

9	:	2	=	4	Rest	1
4	:	2	=	2	Rest	0
2	:	2	=	1	Rest	0
1	:	2	=	0	Rest	1

$$\Rightarrow 9_{10} = \underline{\underline{1001_2}}$$

0,85	·	2	=	1,7	à	1
0,7	·	2	=	1,4	à	1
0,4	·	2	=	0,8	à	0
0,8	·	2	=	1,6	à	1
0,6	·	2	=	1,2	à	1
0,2	·	2	=	0,4	à	0
0,4	·	2	=	0,8	à	0
0,8	·	2	=	1,6	à	1
0,6

⇒ $0,85_{10} \approx \underline{0,11011001_2}$

⇒ $9,85_{10} \approx \underline{1001,11011001_2}$

Wie man im Beispiel 10 sieht, ist die Umrechnung für $0,85_{10}$ nicht eindeutig. Sie kann zwar dem Wert beliebig genau angenähert werden, in dem man die Multiplikation fortgesetzt, doch kann man den Wert $0,85$ nie genau erreichen.

2.2.2 Umwandlung von dualen Nachkommastellen in Dezimalnachkommastellen

Die Umwandlung von dualen Nachkommastellen geschieht wieder durch die Bewertung der Stellenwerte mit Potenzen zur Basis Zwei.

Die Exponenten der Potenzen sind nun negativ und beginnen bei B^{-1} für die ganz linke Nachkommastelle. Die nächste Nachkommastelle erhält einen um eins höheren (betragsmäßig) Exponenten, also B^{-2} , usw.

Beispiel 11:

Wir rechnen die binäre Zahl $1111\ 1101,0111_2$ aus dem Beispiel 9 wieder in eine Dezimalzahl um, dabei trennen wir wieder Vor- und Nachkommateil, um uns die Arbeit ein wenig zu erleichtern.

$$1111\ 1101_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$$

$$= 128 + 64 + 32 + 16 + 8 + 4 + 0 + 1 = \underline{253}_{10}$$

$$0,0111_2 = 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} = 0,25 + 0,125 + 0,0625 = \underline{0,4375}_{10}$$

Beispiel 12:

Nun rechnen wir das Ergebnis unseres 100m-Sprinters $1001,11011001_2$ wieder in eine Dezimalzahl um. Wieder trennen wir Vor- und Nachkommateil der Binärzahl bei der Umrechnung.

$$1001_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 0 + 1 = \underline{9}_{10}$$

$$0,11011001_2 = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} + 0 \cdot 2^{-6} + 0 \cdot 2^{-7} + 1 \cdot 2^{-8} =$$

$$= 0,5 + 0,25 + 0 + 0,0625 + 0,03125 + 0 + 0 + 0,00390625 = \underline{0,848}_{10}$$

⇒ $1001,11011001_2 = \underline{9,848}_{10}$

Wir sehen also, dass durch die Umrechnung Dezimal-Binär-Dezimal beim Nachkommateil ein Genauigkeitsverlust aufgetreten ist, da die Umrechnung Dezimal-Binär im Nachkommateil nicht eindeutig ist. Hätten wir noch ein paar Stellen weiter gerechnet kämen wir zwar näher an unsere $9,85_{10}$ heran, würden sie aber nie genau erreichen.

Die Gefahr ist, dass wir uns bei der Umrechnung von binären Nachkommastellen nie sicher sein können, ob zuvor gerundet worden ist (wie bei $9,85_{10}$) oder ob die Umrechnung genau war (wie bei $253,4375_{10}$). Kennt man also nur die binäre Zahl mit Nachkommateil, muss man sich immer diese Ungewissheit vergegenwärtigen.

2.3 Binäre Codierung (BCD, BCO, BCH)

Hinweis: Die Konvertierung der einzelnen Ziffern sind im **Anhang A** zu finden.

Bei den BCD, BCO und BCH wird jeder Ziffer ein binärer Block zugewiesen. Doch darauf gehen wir nun im einzelnen ein.

2.3.1 Binär Codierte Dezimalzahl (BCD)

Dezimalzahlen sind u. a. dadurch gekennzeichnet, dass sie an jeder Stelle zehn verschiedene Ziffern aufweisen können (0-9). Für die Darstellung von zehn verschiedenen Ziffern sind binär vier Stellen notwendig, deshalb hat eine BCD je Stelle einer binäre Vierergruppe.

Eigentlich wäre eine binäre Darstellung der Zahlen von 0 – 15 möglich, jedoch bleiben die Darstellungsmöglichkeiten von 10 – 15 hier ungenutzt (vgl. hierzu **Abschnitt 2.3.3**).

2.3.1.1 Dezimalzahl in eine BCD konvertieren

Jede Ziffer der Dezimalzahl wird nach der Tabelle in **Anhang A** in einen dualen Viererblock konvertiert. Für das Endergebnis dürfen eventuell führende Nullen weggelassen werden.

Beachtet werden muss die ziffernweise Konvertierung, denn, wie wir oben schon erfahren haben, bleiben bei der BCD-Konvertierung die Darstellungsmöglichkeiten der Zahlen 10 – 15 ungenutzt!

Die Konvertierung des Nachkommateils verläuft dabei genauso wie die Konvertierung des Vorkommateils: ziffernweise in binäre Viererblöcke.

Beispiel 13:

Wir konvertieren nun die Dezimalzahl 317_{10} in BCD-Code.

$$\begin{array}{ccc} 3 & 1 & 7 \\ \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} \\ 0011 & 0001 & 0111 \\ \Rightarrow 317_{10} = 11\ 0001\ 0111_{\text{BCD}} \end{array}$$

2.3.1.2 BCD in eine Dezimalzahl konvertieren

Zur Transformierung einer BCD-Codierten Zahl beginnt man mit der niederwertigsten Vierergruppe, also der ganz rechten, und überträgt diese in die Dezimalziffer, dann überträgt man die nächste Vierergruppe, usw. Bei der letzten Gruppe muss man nur beachten, dass eventuell führende Nullen anzuhängen sind, um eine Vierergruppe zu erhalten.

Beispiel 14:

Wir wollen nun die BCD $101000101110000_{\text{BCD}}$ in eine Dezimalzahl übertragen.

Zuerst gruppieren wir die BCD zu vier Ziffern, von rechts her begonnen:

101 0001 0111 0000

Um die höchstwertige Gruppe zu vervollständigen, fügen wir eine führende Null hinzu:

0101 0001 0111 0000

Nun haben wir die Vorarbeiten geleistet, um die Konvertierung auszuführen:

0101	0001	0111	0000
5	1	7	0

⇒ 5170₁₀

2.3.2 Binär codierte Oktalzahl (BCO)

Die BCO arbeitet nach dem gleichen Prinzip, wie die BCD. Nur dass bei einer Oktalzahl nur acht Ziffern je Stelle möglich sind und daher nur binäre Dreiergruppen gebildet werden müssen.

Hin- und Rückkonvertierung erfolgen nach dem gleichen Prinzip wie bei der BCD.

Beispiel 15:

Wir konvertieren die Oktalzahl 634_8 in BCO.

6	3	4
⏟	⏟	⏟
110	011	100

⇒ $634_8 = \underline{\underline{110.011.100}}_{\text{BCO}}$

2.3.3 Binär codierte Hexadezimalzahl (BCH)

Eine hexadezimale Zahl kann nun an jeder Stelle sechzehn verschiedene Ziffern aufweisen, was zu binären Vierergruppen bei der BCH je Ziffer der Hexadezimalzahl führt. Auch die Umwandlung der Ziffern A - F findet der Leser im **Anhang A**.

Im Gegensatz zur BCD wird bei der BCH nun der volle binäre Viererblock ausgenutzt.

2.3.3.1 Hexadezimalzahl in eine BCH überführen

Die Transformation gelingt nach dem gleichen Prinzip wie bei der BCD-Codierung. Jede Ziffer der Hexadezimalzahl bildet einen binären Vierblock, wobei beim Endergebnis führende Nullen wieder weggelassen werden dürfen.

Beispiel 16:

Wir transformieren die Hexadezimalzahl $31AF_{16}$ in die BCH.

3	1	A	F
⏟	⏟	⏟	⏟
0011	0001	1010	1111

Für das Endergebnis lassen wir die beiden führenden Nullen weg und erhalten:

$\Rightarrow 31AF_{16} = \underline{11\,0001\,1010\,1111}_{\text{BCH}}$

2.3.3.2 Binär codierte Hexadezimalzahl in eine Hexadezimalzahl konvertieren

Will man nun eine BCH wieder in eine Hexadezimal überführen, so muss man, wie bei der BCD-Dezimal-Konvertierung wieder binäre Vierergruppen bilden. So erhalten wir wieder pro Vierergruppe eine Hexadezimalziffer und somit die gesamte Hexadezimalzahl.

Beispiel 17:

Konvertieren wir nun die BCH $10100011110001_{\text{BCH}}$ in eine Hexadezimalzahl.

Als erstes bilden wir binäre Vierergruppen:

$10\,1000\,1111\,0001_{\text{BCH}}$

Nun fügen wir zwei führende Nullen an, um die erste Gruppe zu vervollständigen:

$0010\,1000\,1111\,0001_{\text{BCH}}$

Nun transformieren wir jede binäre Vierergruppe in eine Hexadezimalziffer:

0010	1000	1111	0001
⏟	⏟	⏟	⏟
2	8	F	1

$\Rightarrow 10100011110001_{\text{BCH}} = \underline{28F1}_{16}$

2.4 Darstellung negativer Binärzahlen und das Rechnen damit

2.4.1 Vorzeichenbit - Betragsvorzeichenform

Um eine negative Zahl binär zu codieren wäre es die einfachste Möglichkeit, einfach ein zusätzliches Bit zur Betragsdarstellung zu verwenden, die sogenannte Betrags-Vorzeichen-Form. Beispielsweise eine Null für ein positives Vorzeichen und eine Eins für ein negatives Vorzeichen.

Beispiel 18:

Wir konvertieren die Dezimalzahl 89_{10} in eine Binärzahl und versehen sie dann mit einem Vorzeichenbit:

89	:	2	=	44	Rest	1
44	:	2	=	22	Rest	0
22	:	2	=	11	Rest	0
11	:	2	=	5	Rest	1
5	:	2	=	2	Rest	1
2	:	2	=	1	Rest	0
1	:	2	=	0	Rest	1

$\Rightarrow 89_{10} = \underline{0101\ 1001}_2$

Im Ergebnis fügen wir noch eine führende Null hinzu um binäre Vierergruppen zu erreichen. Obwohl die Zahl KEINE BCD ist, trägt eine Vierergruppierung erheblich zur Anschaulichkeit bei und macht die Dualzahl verarbeitungsfreundlicher.

Nun fügen wir unser Vorzeichenbit hinzu und erhalten nun für 89_{10} und -89_{10} :

$89_{10} = 0\ 0101\ 1001_2$

$-89_{10} = 1\ 0101\ 1001_2$

Warum wird dieses Verfahren nun in der Praxis nicht gerne angewendet?

Bei Rechnungen, die mit Dualzahlen und Vorzeichenbit ausgeführt werden muss sowohl ein Addier-, als auch ein Subtrahierwerk vorhanden sein und es muss eine Logik geben, die zwischen Addition und Subtraktion unterscheidet.

2.4.2 Rechnen mit der Komplementdarstellung

2.4.2.1 Das Einer-Komplement

Beim Einer-Komplement werden alle Ziffern der Dualzahl invertiert (komplementiert), das heißt es werden alle Einsen mit Nullen ersetzt und alle vorhandenen Nullen mit Einsen.

Beispiel 19:

Wir nehmen wieder die Zahl 89_{10} , von der wir aus Beispiel 18 wissen, dass ihre binäre Darstellung 01011001_2 lautet. Nun bilden wir aus dieser Dualzahl durch Invertierung das Einer-Komplement:

$0101\ 1001_2 = 89_{10}$

$1010\ 0110_2 = -89_{10}$

Der gravierende Nachteil dieses Einer-Komplements ist, dass es in diesem System zwei Nullen gibt, nämlich $+0_{10} = 0000\ 0000_2$ und $-0_{10} = 1111\ 1111_2$, was wiederum unfreundlich für eine maschinelle Verarbeitung ist.

2.4.2.2 Das Zweier-Komplement

Das Zweier-Komplement, dessen Darstellung in Rechenmaschinen üblich ist, wird aus dem Einer-Komplement gebildet, in dem noch eine 1_2 addiert wird und der Überlauf nicht berücksichtigt wird.

Beispiel 20:

Wir bilden nun aus der Dezimalzahl 89_{10} das Zweierkomplement.

$$89_{10} = 0101\ 1001_2$$

$$\begin{array}{r} \text{Einer-Komplement: } 1010\ 0110_2 \\ \phantom{\text{Einer-Komplement: }} 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0 \\ \phantom{\text{Einer-Komplement: }} + \\ \hline \end{array}$$

$$\text{Zweier-Komplement: } \underline{1\ 0\ 1\ 0\ 0\ 1\ 1\ 1}$$

Das Zweier-Komplement lautet also $1010\ 0111_2$ und stellt die negative Zahl -89_{10} dar.

Jedoch haben wir nun das Problem, dass die Darstellung im Zweier-Komplement nicht eindeutig ist, was wir an Hand des nachfolgenden Beispiels sehen.

Beispiel 21:

Die Dezimalzahl 5_{10} umgerechnet in eine Binärzahl lautet:

$$5_{10} = 0101_2$$

Bildet man nun von der Zahl 5_{10} das Zweierkomplement, so lautet dies:

$$1010_2 + 1 = \underline{1011_2} = -5_{10}$$

Nun rechnen wir die Zahl 11_{10} in eine Binärzahl um und erhalten:

$$11_{10} = \underline{1011_2}$$

$$\underline{-5_{10}} = \underline{11_{10}} \text{ ??}$$

⇒ Mehrdeutigkeit der Komplementdarstellung

Da wir diese Feststellung so natürlich nicht stehen lassen können, müssen wir einen Weg finden, um eine Unterscheidung zwischen diesen Zahlen zu finden.

Bei der Komplement-Darstellung ist es wesentlich, dass man sich auf eine feste Stellenanzahl bezieht, also zwei binäre Vierergruppen, drei binäre Vierergruppen, usw.

Mit dieser Voraussetzung und der Einführung des MSB (most significant bit) können wir dieses Problem lösen.

Ähnlich wie bei der Betragsvorzeichenform gibt die erste Ziffer an, ob es sich um eine positive oder um eine negative Zahl handelt.

führende Ziffer ist eine Null \Rightarrow Zahl ist positiv
führende Ziffer ist eine Eins \Rightarrow Zahl ist negativ

Es ist also stets darauf zu achten, dass man sich auf eine bestimmte Wortlänge beim Komplement bezieht.

2.4.2.3 Addition und Subtraktion mit dem Zweier-Komplement

Die Addition verläuft unverändert, wie im **Abschnitt 2.1.1** beschrieben. Es muss natürlich hierfür keine Komplementierung vorgenommen werden.

Die Subtraktion wird auf die Addition zurückgeführt, indem man, statt zu subtrahieren, das Zweier-Komplement addiert.

Ein eventuell auftretender **Übertrag** bei der Subtraktion **über die Wortlänge hinaus, wird ignoriert.**

Das Ergebnis dieser Addition muss nun noch mit dem MSB gedeutet werden:
- Ist das Ergebnis positiv, ist keine weitere Umwandlung mehr nötig
- Ist das Ergebnis negativ, muss aus dieser dualen Zahl wieder das Zweier-Komplement gebildet werden

Beispiel 22:

Wir führen nun die Operation $52_{10} - 70_{10}$ binär mit Hilfe des Zweier-Komplements durch. Wir beziehen uns auf acht Bits, d. h. auf zwei binäre Viererblöcke.

Als erstes wandeln wir die Zahl 52_{10} in eine Binärzahl um:

52	:	2	=	26	Rest	0
26	:	2	=	13	Rest	0
13	:	2	=	6	Rest	1
6	:	2	=	3	Rest	0
3	:	2	=	1	Rest	1
1	:	2	=	0	Rest	1

$\Rightarrow 52_{10} = 0011\ 0100_2$

Als nächstes wandeln wir die Zahl 70_{10} in eine Dualzahl um:

70	:	2	=	35	Rest	0
35	:	2	=	17	Rest	1
17	:	2	=	8	Rest	1
8	:	2	=	4	Rest	0
4	:	2	=	2	Rest	0
2	:	2	=	1	Rest	0
1	:	2	=	0	Rest	1

$\Rightarrow 70_{10} = 0100\ 0110_2$

Jetzt bilden wir aus $0100\ 0110_2$ das Einser- und anschließend das Zweier-Komplement:

1: $1011\ 1001_2$

2: $1011\ 1010_2 = -70_{10}$

Nun addieren wir 52_{10} und das Zweierkomplement von 70_{10} :

$$\begin{array}{r} 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0 \\ +\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0 \\ \hline 1\ 1\ 1\ 0\ 1\ 1\ 1\ 0 \end{array}$$

⇒ Ergebnis ist: $1110\ 1110_2$

⇒ **Beachte:** die führende Ziffer ist Eins ó die **Zahl ist negativ**

Um nun das richtige Ergebnis zu erhalten, müssen wir wieder komplementieren und Eins addieren:

1: $0001\ 0001_2$

2: $0001\ 0010_2$

Rechnen wir nun das Ergebnis 10010_2 um:

$$10010_2 = 1 \cdot 2^4 + 1 \cdot 2^1 = 16 + 2 = 18_{10}$$

Wird nun beachtet, dass uns das MSB im dualen Ergebnis eine negative Zahl gedeutet hat, ist das Endergebnis -18_{10}

Ein Nachrechnen im Dezimalzahlensystem bestätigt uns dieses Ergebnis ($52_{10} - 70_{10} = -18_{10}$).

Beispiel 23:

Wir möchten nun die Dezimalzahlen 25_{10} von 52_{10} binär mit Hilfe des Zweier-Komplements subtrahieren. Wir beziehen uns auch hier wieder auf 8 Bits, wie im Beispiel 22.

Die duale Darstellung von 52_{10} kennen wir bereits aus Beispiel 22:

$$52_{10} = 0011\ 0100_2$$

Nun konvertieren wir 25_{10} in eine Dualzahl:

25	:	2	=	12	Rest	1
12	:	2	=	6	Rest	0
6	:	2	=	3	Rest	0
3	:	2	=	1	Rest	1
1	:	2	=	0	Rest	1

$$\Rightarrow 25_{10} = 0001\ 1001_2$$

Nun bilden wir das Zweier-Komplement von $0001\ 1001_2$:

1: $1110\ 0110_2$

2: $1110\ 0111_2$

Nun erfolgt die Addition von $0011\ 0100_2$ und $1110\ 0111_2$:

$$\begin{array}{r} 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0 \\ +\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1 \\ \hline 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ (1)\ 0\ 0\ 1\ 1\ 0\ 1\ 1 \end{array}$$

⇒ Beachte: Der Überlauf über die gewählte Wortlänge hinaus wird ignoriert!

⇒ Die führende Ziffer ist Null ó die Zahl ist positiv, d. h. es ist auch keine Komplementierung wie in Beispiel 22 mehr notwendig; das Ergebnis kann direkt abgelesen werden!

Somit lautet das Additionsergebnis $\underline{11011}_2$

Eine Umrechnung des Ergebnisses in das Dezimalzahlensystem ergibt:

$$11011_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 8 + 2 + 1 = 27_{10}$$

Das MSB zeigt uns ein positives Ergebnis an, also ist $\underline{\pm 27}_{10}$ das Endergebnis, was Nachrechnen im dezimalen bestätigt ($52_{10} - 25_{10} = 27_{10}$).

Anhang A : Ziffernumwandlung (BCD, BCO, BCH)

Dual	Hexadezimal	Dezimal	Oktal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	8	-
1001	9	9	-
1010	A	-	-
1011	B	-	-
1100	C	-	-
1101	D	-	-
1110	E	-	-
1111	F	-	-

Anhang B : Stellenwertsystem

$$Z = \underbrace{a_{n-1} \cdot B^{n-1} + \dots + a_0 \cdot B^0}_{\text{Vorkommateil}} + \underbrace{a_{-1} \cdot B^{-1} + a_{-2} \cdot B^{-2} + \dots + a_{-m} \cdot B^{-m}}_{\text{Nachkommateil}} = \sum_{i=-m}^{n-1} (a_i \cdot B^i)$$

(ganzzahliger Teil mit n Stellen) (gebrochener Teil mit m Stellen)

Bitte besuchen Sie doch meine Homepage.

www.alexander-halles.de

Auch konstruktive Kritik oder Fragen werden gerne entgegengenommen unter:

mail@alexander-halles.de



Alexander Halles

... working with boundless passion

© Copyright 2004, alle Rechte vorbehalten.