

Zahlencodierung I

Wenn wir an Zahlen denken, handelt es sich im Normalfall um Zahlen im Zehnersystem. (Lohn: 5'000 SFr, 15 km bis zum Ziel usw.) Das sind ganze Zahlen, welche in einem **Stellenwertsystem** notiert sind und aus Ziffern bestehen. Die verwendeten Ziffern sind: 0,1,2,3,4,5,6,7,8,9

Je nachdem wo eine Ziffer steht, ändert dies den Wert der Zahl:

$$5000 > 0050$$

10er System (Dezimalsystem)

Wenn wir obig erwähnte Dezimalzahlen addieren, fangen wir bei der tiefsten Stelle (10^0) an, und arbeiten uns nach links durch bis zur höchsten Stelle (10^3).

Begriffe: Basis ^{Exponent} Bsp: 4^2

$$\begin{array}{cccccccc} 2 & & 4 & & 8 & & 1 & \\ 2 * 10^3 & + & 4 * 10^2 & + & 8 * 10^1 & + & 1 * 10^0 & \\ 2'000 & + & 400 & + & 80 & + & 1 & = 2'481_{10} \end{array}$$

Schreibweise: 2481_{10} für die Angabe, dass es sich um das 10er bzw. Dezimalsystem handelt. Es wird als Basis die 10 verwendet. Der Exponent richtet sich nach der Stelle der Ziffer in der Darstellung und fängt immer rechts mit der 0 an. Pro Stelle wird er um 1 erhöht.

Mit diesem Schema können Sie Zahlen aus den unterschiedlichsten Zahlensystemen *in das Dezimalsystem* umrechnen. →1

Es gibt aber auch andere Systeme als Stellenwertsysteme:
Denken Sie an die römischen Zahlen!

Ein Film aus dem Jahre MCMXCVIII stammt also von?



2er System (Binärsystem) S. 79

"Es gibt nur 10 Arten von Menschen. Die, die das Binärsystem verstehen und die, die es nicht verstehen." (Autor unbekannt)

Auch das Binärsystem ist ein Stellenwertsystem. Es ist deshalb so wichtig, weil es bei digitalen Schaltungen verwendet wird: Spannung oder keine Spannung bzw. ein oder aus.

Alle Informationen im Arbeitsspeicher, auf der Festplatte, Netzwerkkarte usw. werden digital, d.h. binär gespeichert oder übertragen.

Für die Umrechnung vom Binär- ins Dezimalsystem können wir das obige Schema in der Tabelle abändern (Basis =2) und verwenden, wie können wir aber in die andere Richtung rechnen?

$1999_{10} = ?_2$

Dazu schauen wir zuerst an, wie sich das Binärsystem dezimal darstellt:

Durch die Basis 2 verdoppelt sich der Wert im Dezimalsystem bei jedem Schritt. (Erhöhung des Exponents um +1)

00000001_2	2^0	1_{10}
00000010_2	2^1	2_{10}
00000100_2	2^2	4_{10}
00001000_2	2^3	8_{10}
00010000_2	2^4	16_{10}
00100000_2	2^5	32_{10}
01000000_2	2^6	64_{10}
10000000_2	2^7	128_{10}

Das Beispiel rechts zeigt die korrekte Schreibweise für 8 Bit im Binärsystem. Es werden lediglich die Ziffern 0 und 1 verwendet.

Alle Kombinationen innerhalb der 8 Bit sind möglich, daraus ergeben sich 2^8 verschiedene Werte, bzw. die Zahlen 0-255₁₀.

Hinweis: Die Zahlenreihe 2^0 bis 2^{16} sollten Sie **auswendig** können, das erleichtert viele Umrechnungen enorm.

Umrechnung vom Dezimal- ins Binärsystem

Die Umrechnung erfolgt nach einem vorgegebenen Schema: Bsp: $1999_{10} = ?_2$

Dividend	Quotient	Rest
1999	999	1
999	499	1
499	249	1
249	124	1
124	62	0
62	31	0
31	15	1
15	7	1
7	3	1
3	1	1
1	0	1

Stelle (n)	2^n	
0	1	1
1	2	2
2	4	4
3	8	8
4	16	0
5	32	0
6	64	64
7	128	128
8	256	256
9	512	512
10	1024	1024
Summe		1999

Die Startzahl wird durch 2 dividiert, es handelt sich um eine Ganzzahldivision (**DIV** genannt), das Resultat ist in der Spalte *Quotient* zu sehen.

Der Rest der Ganzzahldivision (**MOD** oder Modulo genannt) ist in der Spalte *Rest* eingetragen.

$1999 \text{ DIV } 2 = 999$

$1999 \text{ MOD } 2 = 1$

Der Quotient wird als neuer Startwert übernommen und die Berechnung solange wiederholt bis dieser 0 wird. Aus den Werten in der Spalte *Rest* ergibt sich die umgerechnete Binärzahl. In unserem Beispiel also 11111001111_2 .

Die Tabelle rechts ist nur zur Kontrolle, sie rechnet die Werte der Spalte *Rest* wieder in eine Dezimalzahl um. Mit diesem Schema können Sie nun Umrechnungen in beiden Richtungen durchführen. **→2 (erklären Sie sich beide Tabellen vorher)**

16er System (Hexadezimalsystem) S. 79

Das Hexadezimalsystem wird in der Informatik verwendet, da es die Zahlen viel kürzer darstellen kann. 8 Bit = 1 Byte = 2^8 Zustände = 16^2 Zustände

Ein Byte (0-255₁₀) kann also mit zwei Ziffern im Hexadezimalsystem abgebildet werden. Damit 16 Zeichen gut lesbar sind, werden folgende Ziffern verwendet:

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Bsp.: $188_{10} = 10111100_2 = BC_{16}$

Wir verwenden das Hexadezimalsystem zum Beispiel bei der Farbdefinition in HTML/CSS, bei den IP Adressen unter IPv6 oder den MAC Adressen der Netzwerkkarte.

Die Umrechnung des obigen Beispiels vom Binär- ins Hexadezimalsystem kann direkt erfolgen! Dazu trennen wir die 8 Bit Binärzahl in 2 **Nibbles** auf:

Linker Teil: $1011_2 = 8_{10} + 0_{10} + 2_{10} + 1_{10} = 11_{10} = B_{16}$

Rechter Teil: $1100_2 = 8_{10} + 4_{10} + 0_{10} + 0_{10} = 12_{10} = C_{16}$

So können wir riesige Zahlen schnell von einem in das andere System umrechnen, ohne Umweg über das Dezimalsystem. Der eigentliche Zweck, Binäre Zahlen kürzer darzustellen ist also erreicht. **→3**

Schreibweise: Das Hexadezimalsystem wird oft verwendet, da es zu kompliziert wäre die tiefgestellte 16 zu verwenden, wird eine andere Schreibweise angewandt:

Statt BC_{16} wird geschrieben 0xBC da dies ohne Sonderzeichen auskommt.

Selten trifft man auch die Schreibweise BCh an.

8er System (Oktalsystem)

Das Oktalsystem funktioniert genau gleich wie die anderen Stellenwertsysteme, es verwendet die Basis 8. Somit können 3 Bit mit einer Ziffer dargestellt werden.

Es gibt (wie erwartet) Übungen dazu **→4**

Umrechnungen zwischen den Systemen

Um Zahlen zwischen den unterschiedlichen Systemen umzurechnen, gibt es viele Möglichkeiten:

- Kopfrechnen
- Tabellen
- Taschenrechner (auch die im Betriebssystem)
- Vordefinierte Funktionen (je nach Programmiersprache, Tabellenkalkulation)

→5

Rechnen mit den unterschiedlichen Systemen (Addition, Subtraktion)

In jedem der Zahlensysteme können die Grundrechenarten ganz einfach angewendet werden. Als Beispiel werden wir ein paar Additionen durchführen **→6**

Logische Rechenoperationen mit Binärzahlen

Da die Prozessoren mit binären Zahlen rechnen, müssen auch wir die grundlegenden Operationen durchführen können. Es gibt 2 Grundoperationen mit jeweils 2 Inputparametern (grau hinterlegt):

AND	0	1
0	0	0
1	0	1

OR	0	1
0	0	1
1	1	1

Zudem gibt es noch eine Grundoperation mit nur einem Parameter:


NOT	
0	1
1	0

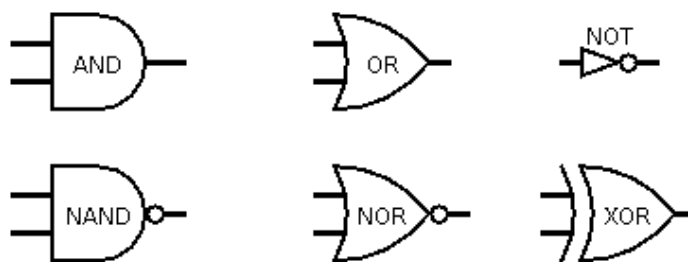
Aus Kombinationen dieser Grundoperationen können weitere Operationen erzeugt werden:

NAND	0	1
0	1	1
1	1	0

NOR	0	1
0	1	0
1	0	0

XOR	0	1
0	0	1
1	1	0

Für diese Operationen existieren Darstellungsformen ebenfalls in der Mengenlehre und in der digitalen Schaltarithmetik.  S. 122



➔7

5) Umrechnen zwischen den Systemen

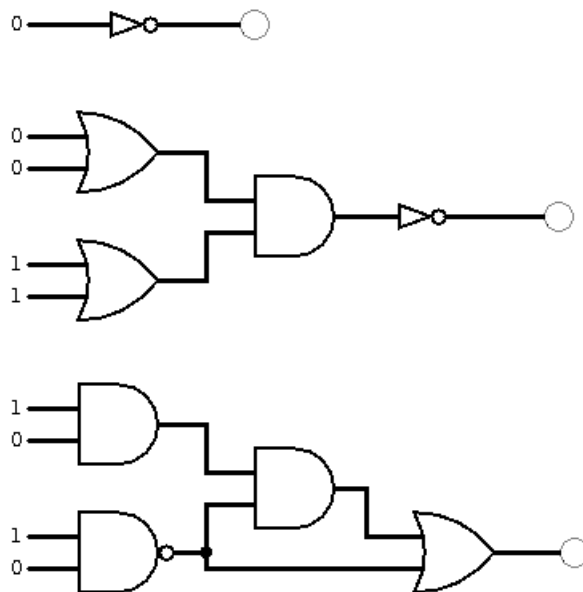
Erstellen Sie auf der bmLP1 mittels LibreOffice Calc eine Tabelle, welche in allen 4 Systemen die Zahlen zwischen 0 - 255 anzeigt.

Erstellen Sie zudem auf einem anderen Tabellenblatt eine "Umrechnungsmaschine", welche als Input Dezimalzahlen annimmt und diese in die Systeme 2/8/16 umrechnet. Verwenden Sie dazu die Formeln mit Quotient und Rest.

6) Rechnen in anderen Zahlensystemen

Binärsystem	Hexadezimalsystem
0 1 1 0 1 0 0 1 1	A 1 5 0 C D 3 2 F
+ 1 1 1 1 0 1 1 1 0	+ 1 B 7 3 0 C D 4 1
=	=

7) Bestimmen Sie bei den folgenden Beispielen die Resultate



Anschliessend laden Sie die Software "Logisim" auf die bmLP1 herunter.
 AB114-01 Dateien/logisim-generic-2.7.1.jar
 oder von <http://sourceforge.net/projects/circuit/>

Sie können das Programm von der Befehlszeile starten:

```
java -jar logisim-generic-2.7.1.jar
```

Bauen Sie nun mit Hilfe der Kurzanleitung (Menu: Hilfe) eine XOR Schaltung auf der Grundlage der Basisgatter AND OR NOT. Testen Sie Ihre Lösung!

Es gibt auf der Webseite

http://www.christophtutorials.de/beitrag.php?beitrag=Logisim_-_Grundlagen
sogar ein Tutorial in Videoform.

Benutzen Sie Logisim nun, um Ihre Resultate am Anfang dieser Übung zu validieren.

Weiterführende Themen für Fortgeschrittene:

8) Umrechnung selber geschrieben

Erstellen Sie in einer Ihnen bekannten Programmiersprache (C#, Java, Python, ...) eine Umrechnungsfunktion, welche zwischen Dezimal- und Binärsystem umrechnet. Sie benötigen dazu eine Schleife sowie die Befehle DIV und MOD.



9) Schaltalgebra

Studieren Sie die Ausführungen zur Schaltalgebra unter <http://www.sps-lehrgang.de/schaltalgebra/>
Erstellen Sie anschliessend einen "Volladdierer" mittels Logisim.