A_DATEN_CODIEREN_NUM

Bit und Byte

Bit/Byte: Bit = **B**inary dig**it**
8 Bit = 1 Byte
16 Bit = 1 Word
Abkürzung für Bit = b
Abkürzung für Byte = B
LSB = "Least Significant Bit" oder das kleinstwertigste Bit
MSB = "Most Significant Bit" oder das höchstwertigste Bit
(*Die Beschriftung der LSB- bzw. MSB-Leitung ist z.B. bei Parallelverbindungen wichtig, damit ein Stecker nicht falsch herum angeschlossen wird*)

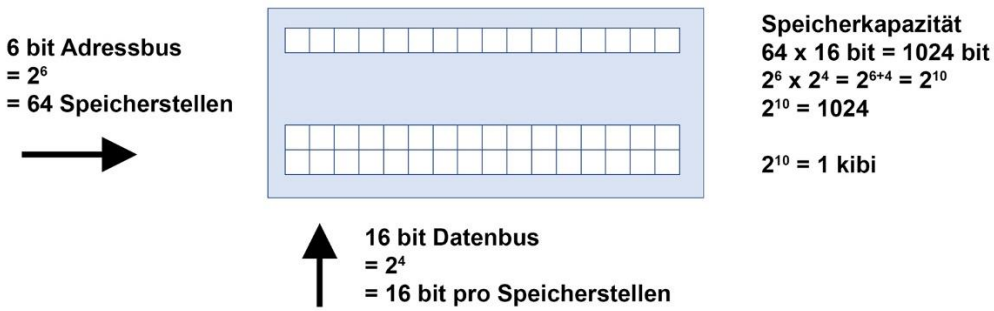
Massvorsätze

SI-Präfixe: [T] → Tera → 10^{12} → 1'000'000'000'000 → Billion
[G] → Giga → 10^{9} → 1'000'000'000 → Milliarde
[M] → Mega → 10^{6} → 1'000'000 → Million
[k] → kilo → 10^{3} → 1'000 → Tausend

IEC-Präfixe: [Ti] → Tebi → 2^{40} → 1'099'511'627'776
[Gi] → Gibi → 2^{30} → 1'073'741'824
[Mi] → Mebi → 2^{20} → 1'048'576
[Ki] → Kibi → 2^{10} → 1'024

Warum IEC-Präfixe?

Die IEC-Präfixe «International Electrotechnical Commission» werden für Kapazitätsangaben bei **Speichermedien** verwendet. Grund: Für Datenspeicher mit binärer Adressierung ergeben sich Speicherkapazitäten von 2^n Byte, d. h. Zweierpotenzen.



Zahlensysteme:

- BIN: **Binärsystem**, Zweiersystem, Dualsystem
  Basis: 2
  Zeichenvorrat: 0, 1
- OCT: **Oktalsystem**, Achtersystem
  Basis: 8
  Zeichenvorrat: 0, 1, 2, 3, 4, 5, 6, 7, 8
- DEZ: **Dezimalsystem**, Zehnersystem
  Basis: 10
  Zeichenvorrat: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- HEX: Hexadezimalsystem, Sechzehnersystem
  Basis: 16
  Zeichenvorrat: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14), F (15)
  (*Eine Hex-Ziffer = vierstelligen Dualzahl oder 4 Bit*)

Die Anzahl Kombinationen einer Bitabfolge lässt sich mit folgender Formel berechnen:

$$BitkombinationenAnzahl = 2^{BitstellenAnzahl}$$

Beispiel:
Q: 16 Bit ergeben wie viele Kombinationen?
A: $2^{16} = 65'536$

Die Umkehrfunktion lautet:

BitAnzahl = LOG BitkombinationenAnzahl / LOG 2
Das Ergebnis ist auf die nächsthöhere Ganzzahl aufzurunden!

LOG=Zehnerlogarithmus

Beispiel:
Q: Bei einer Distanzmessung sind 1000 unterscheidbare Kombinationen verlangt, von 0mm bis 999mm.
A: BitAnzahl = LOG(1000 / LOG2)
BitAnzahl = 3 / 0.301 = 9.966
BitAnzahl = 10
Kontrolle: 2^{10}=1024 (24 Kombinationen ergeben Redundanz. Ist aber unvermeidbar, weil 9 Bit nur 512 Kombinationen ergäben.)

Hier folgen Aufgaben zum Thema. Siehe separates Aufgabenblatt.

Vorzeichenbehaftete Dezimalzahlen in Binärschreibweise

(+0)	0000
(+1)	0001
(+2)	0010
(+3)	0011
(+4)	0100
(+5)	0101
(+6)	0110
(+7)	0111
(-8)	1000
(-7)	1001
(-6)	1010
(-5)	1011
(-4)	1100
(-3)	1101
(-2)	1110
(-1)	1111

Zweierkomplement bilden:

Variante 1
Negative Zahl erhält man:
a. Betrag der negativen Zahl
a. Betrag bitweise invertieren
b. Resultat um 1 addieren

Variante 2
Negative Zahl erhält man
durch Wertigkeit -8 / 4 / 2 / 1

Beispiele:

```
(-5)  1011
(-2) -1110
(-3)  1101

(-5)  1011
(+7) +0111
(+2)  0010

(+3)  0011
(+4) +0100
(+7)  0111

(-3)  1101
(+4) -0100
(-7)  1001
```

Binäres Rechnen und Datenüberlauf

```
0 1 0 0 1 1 1 0 +   78
1 0 1 0 0 1 1 1 =  167
1 1 1 1 0 1 0 1    245

1 1 0 0 1 0 0 0 +  200
1 0 1 1 0 1 0 0 =  180
0 1 1 1 1 1 0 0    124 -) DATA-OVERFLOW!
```

```
0 + 0 = 0
0 + 1 = 1
1 + 1 = 10       (Übertrag von 1 auf die nächsthöhere Stelle)
1 + 1 + 1 = 11   (Übertrag von 1 auf die nächsthöhere Stelle)
```

Der Wertebereich vom Datentyp INTEGER

Der Integer (int) ist aktuell eine 32 Bit-Ganzzahl. (Früher 16 Bit)
232 ergibt 4'294'967'296 Kombinationen.

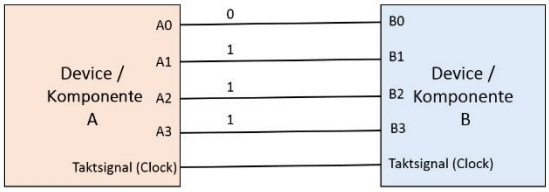
- Vorzeichenlos/unsigned: 0 bis 4'294'967'295
- Vorzeichenbehaftet/signed: -2'147'483'648 bis +2'147'483'647

Gleitkommazahlen

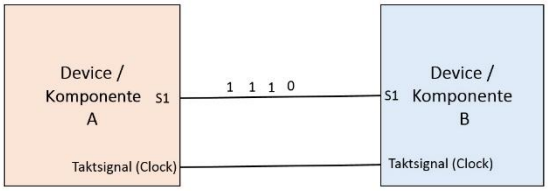
Die Norm IEEE 754 definiert Standarddarstellungen für binäre Gleitkommazahlen in Computern in unter anderem den beiden Grunddatenformate 32 Bit → Single Precision und 64 Bit → Double Precision. Eine Gleitkommazahl wird wie folgt dargestellt:

```
x = v * m * b e      v: Vorzeichen 1 Bit
                     m: Mantisse bei Single 23Bit, bei Double 52Bit
                     b: Basis bei normalisierten Gleitkommazahlen 2
                     e: Exponent bei Single 8Bit, bei Double 11Bit
```

Datenübertragung



Parallele Verbindung



Serielle Verbindung

- Eine **parallele Verbindung** zwischen zwei oder mehreren Komponenten nennt man Datenbus. Ein Codewort wird auf parallelen Leitungen auf einen Schlag bzw. Takt übertragen. Ist das Codewort z.B. 4 Bit breit, benötigt man 4 Leitungen.
  Datenbus auf dem Mainboard: Verbindet CPU, RAM und I/O.
  Adressbus auf dem Mainboard: Verbindet CPU, RAM und I/O.
  SCSI (Small Computer System Interface): Verbindung und Datenübertragung zwischen Peripheriegeräten und Computern.
  P-ATA (Parallel Advanced Technology Attachment): Paralleler Datentransfer zwischen Speichermedien bzw. Laufwerken und der entsprechenden Schnittstelle eines Computers.
- **Serielle Verbindung**: Um Leitungen einzusparen, kann ein Codewort auch seriell übertragen werden. Dann werden die Bit's nacheinander "auf den Weg geschickt". Der Takt ist jeweils der Startschuss für das "Loslaufen" des folgende Bit. Um die selbe Performance wie bei der parallelen Datenübertragung zu erreichen, muss die Elektronik entsprechend schneller sein. Um zum Beispiel die gleiche Datenmenge einer 4-Bit-Parallelverbindung zu erreichen, muss die serielle Verbindung 4x schneller liefern.
  S-ATA (Serial Advanced Technology Attachment): Serieller Datentransfer zwischen Speichermedien bzw. Laufwerken und der entsprechenden Schnittstelle eines Computers.
  USB (Universal Serial Bus) für Drucker, Speicher-Sticks etc.
  SAS (Serial Attached Small Computer System Interface): Die serielle Variante von SCSI.

Datenspeicherung

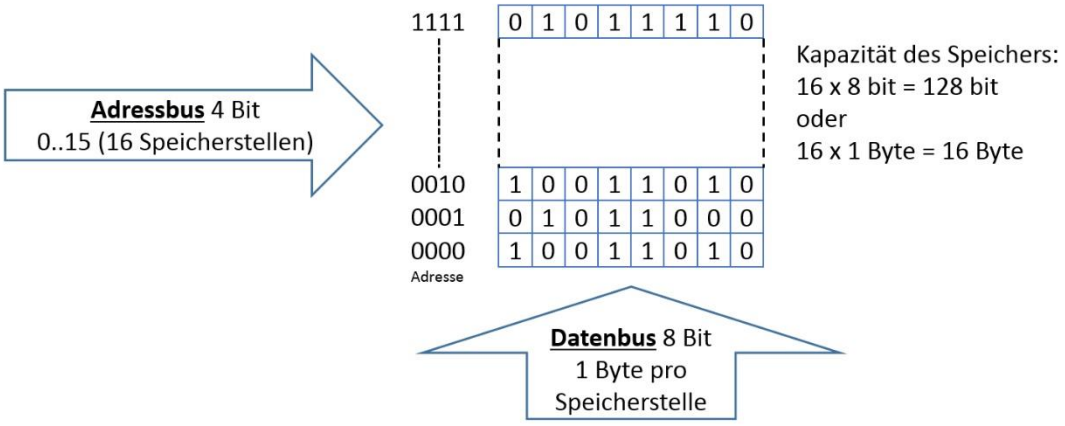
- **Nichtflüchtiger** oder permanenter **Speicher**: Dieser Speicher verliert seine Daten im stromlosen Zustand nicht.
  Typische Vertreter: Magnet-Harddisk, SSD, USB-Speicherstick.
  Man nennt solchen Speicher auch Sekundärspeicher.
- **Flüchtiger Speicher**: Dieser Speicher verliert seinen Inhalt, wenn er stromlos wird. Die Technologie solcher Speicher lässt wesentlich höhere Datenraten zu, als bei nichtflüchtigem Speicher.
  Typische Vertreter: Cache-Speicher in der CPU, RAM,
  Man nennt den RAM-Speicher auch Primärspeicher.
  Diese Speicher zeichnen sich darin aus, dass sie elektrisch bzw. verbindungstechnisch immer sehr nahe an der CPU liegen und von der CPU oft benötige Daten sehr schnell liefern bzw. zwischenspeichern können. (Effizienz, Performance)

Zugriff auf flüchtigen RAM-Speicher

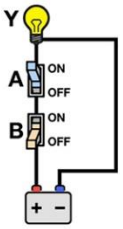
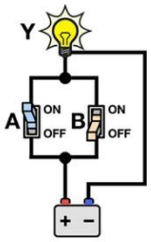
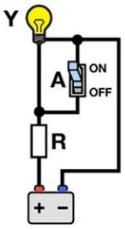
Eine Analogie aus der Bücherwelt: Das Bücherarchiv:
Möchte man gerne seine archivierten Bücher wieder finden, muss man sich bei deren Ablage merken, wo man sie hinlegt. So ist es zum Beispiel sicher keine schlechte Idee, sich Regal- und Tablarnummer zu merken. Vielleicht sind die Bücher dann ja auch noch durchnummeriert. Gemeint ist selbstverständlich nicht die 12 bändige Micky-Maus-Best-Of-Sammlung sondern eine fast unüberschaubare Büchersammlung wie sie z.B. eine Universität besitzt.

Wir unterscheiden also Ware (Daten) und Ablageort (Adresse).

Beim Computer ist die Problemstellung dieselbe: Die erzeugten und gespeicherten Daten wollen wieder gefunden werden. Dafür verwendet man einen Speicher-Chip, mit vielen "Speichernischen". Jede "Speichernische" wird über eine eindeutige Adresse erreicht:



Kombinatorik

Bezeichnung	UND/AND &&	ODER/OR ||	NICHT/NOT/INVERTER !
Schaltschema			
Wahrheits-tabelle 0=false 1=true	A B Y 0 0 0 0 1 0 1 0 0 1 1 1	A B Y 0 0 0 0 1 1 1 0 1 1 1 1	A Y 0 1 1 0

Byte-Reihenfolge Big/Little-Endian

Die Byte-Reihenfolge bezeichnet die Speicherorganisation für einfache Zahlenwerte (z.B. Integer) im Arbeitsspeicher.

- **Big-endian-Format** (Grossendig): Das höchstwertige Byte wird zuerst gespeichert, d. h. an der kleinsten Speicheradresse. Die höchstwertige Komponente wird zuerst genannt. Bsp. Uhrzeit → Stunde:Minute:Sekunde.
  Mikroprozessor: Das Motorola-Format steht für Big-Endian
  Serielle Übertragung: Big-Endian-Byte-Reihenfolge → Das höchstwertige Bit eines Bytes wird zuerst übertragen. Bsp.: I^2C
- **Little-endian-Format** (Kleinendig): Das kleinstwertige Byte wird an der Anfangsadresse gespeichert. Die kleinstwertige Komponente wird zuerst genannt. Bsp. Datum → Tag.Monat.Jahr.
  Mikroprozessor: Das Intel-Format steht für Little-Endian.
  Serielle Übertragung: Das niederwertigste Bit eines Bytes wird zuerst übertragen. Bsp.: RS-232

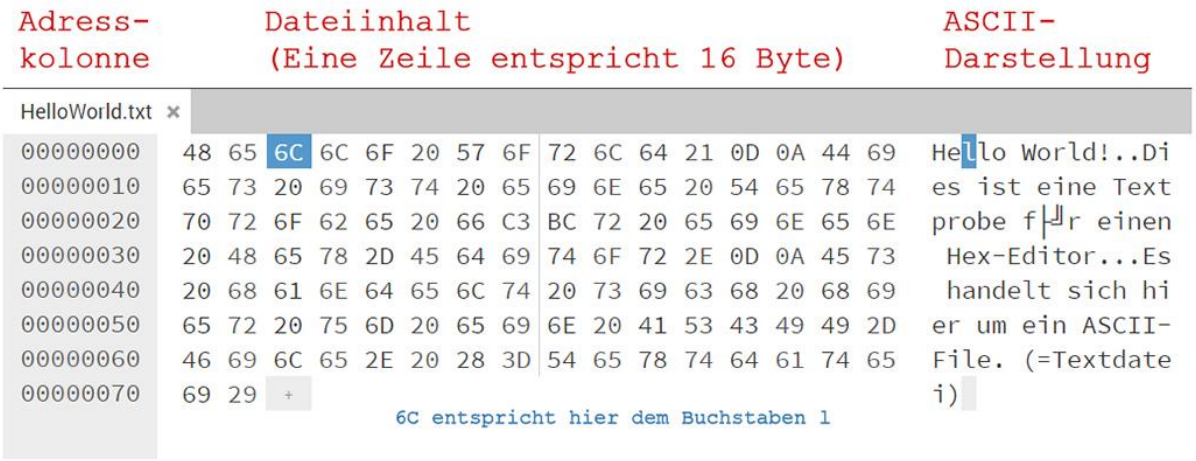
Hier folgen Aufgaben zum Thema. Siehe separates Aufgabenblatt.

HEX-Editor und Notepad++

Wichtige Editoren für die IT-Fachperson:

HEX-Editor HxD

Unter einem HEX-Editor versteht man ein Computerprogramm, mit dem sich die Bytes beliebiger Dateien als Folge von Hexadezimalzahlen darstellen und bearbeiten lassen. Der Hex-Editor stellt eine ausgewählte Datei so dar:



Adresskolonne:	Adresse des ersten Bytes der entsprechenden Zeile in hexadezimaler Darstellung.
Dateiinhalt:	16 Daten-Bytes. Pro Byte zwei Hex-Ziffern.
ASCII:	Den Versuch die 16 Bytes als ASCII-Character darzustellen.

Im Internet findet man einige Online-HEX-Editoren wie z.B. diesen: https://hexed.it/
Wer es gerne lokal als Applikation mag, findet z.B. die HEX-Editor-App HxD unter dem folgenden Link: https://mh-nexus.de/de/hxd/

Notepad++

Notepad++ ist ein freier Texteditor für Windows und kompatible Betriebssysteme und dem Standard-Texteditor von Windows eindeutig überlegen. Als Zeichensätze werden ASCII und verschiedene Unicode-Kodierungen unterstützt. Notepad++ findet man unter dem folgenden Link: https://notepad-plus-plus.org/

Hier folgen Aufgaben zum Thema. Siehe separates Aufgabenblatt.
