



**Aufgabe 1:**

Welchen gravierenden Nachteil hat der *circuit switched* Datentransfer? Schildern Sie diesen Nachteil in einer Fallstudie. Welche Alternativen gibt es?

**Aufgabe 2:**

Was versteht man unter *Little Endian Byte Ordering* und *Big Endian Byte Ordering*?

Welche Probleme existieren bei der Benutzung von (Byte für Byte abgelegten) Binärdateien etwa in heterogenen Computernetzwerken? Wie lassen sich diese Probleme lösen?

**Aufgabe 3:**

Welche ganzzahligen numerischen Datentypen kennt eine typische CPU?

Wie groß sollten Integer-Register gewählt werden, um alle diese Datentypen verarbeiten zu können?

Welche Probleme ergeben sich bei der Verarbeitung z.B. von Daten des Typs Short Integer (16 Bit) in Integer-Registern (32 Bit)?

**Aufgabe 4:**

Warum ist es bei der Codierung von Integer Daten (z.B. Byte-Integer:  $-128, \dots, +127$ ) sinnvoll, die obere Hälfte des Zahlenbereichs der zugehörigen Ordinalzahlen (Byte-Ordinal:  $0, \dots, 255$ ) für die negativen Zahlen zu verwenden?

**Aufgabe 5:**

Die relative Adressierung läßt im allgemeinen nur Sprünge innerhalb eines kleinen Zielbereichs zu. Beim 6502-Prozessor ist dieser Bereich beispielsweise  $-128, \dots, +127$ . Bedingte Sprünge lassen sich dabei nur in relativer Adressierungsweise verwenden.

Beispiel:

```
LDA AFFFH (Lade Akkumulator mit Inhalt von Speicherstelle AFFFH)
SBC #05H (Subtrahiere 05H)
BEQ 0FH (Überspringe die folgenden 15 Bytes, falls Ergebnis = 0)
...
```

Durch welches kleine Programmkonstrukt lassen sich auch bedingte Sprünge zu beliebigen Speicherstellen realisieren?

*Tip:* Verwenden Sie eine Kombination aus bedingten und absoluten Sprüngen.

***Ende der Bearbeitungszeit:*** 24. Mai 2002