



https://img.deviant.com/deviant/rant/r_307878_RGc57.jpg

III Funktionsweise eines Rechners

1 Aufbau eines Computersystems

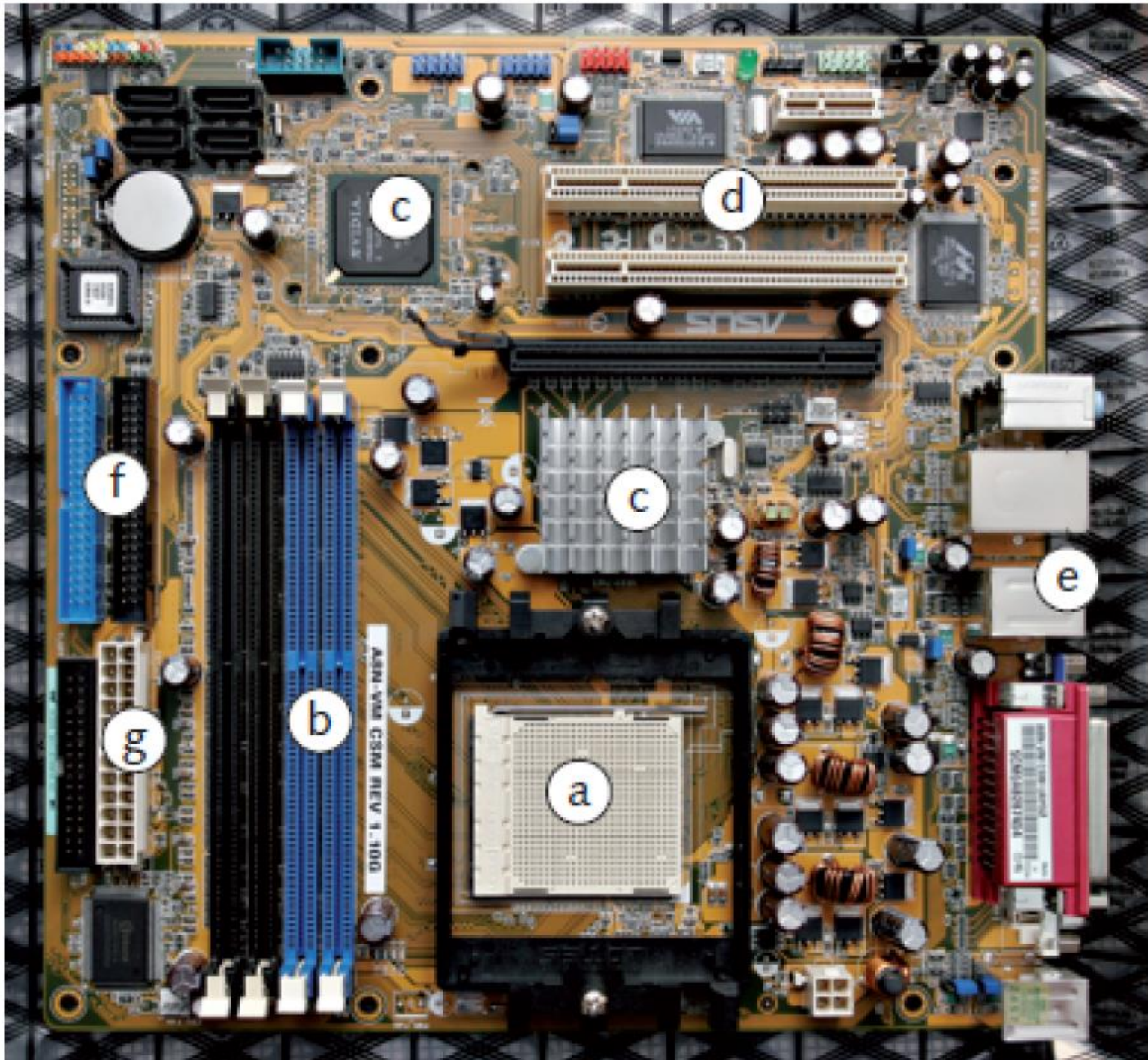
Aus welchen Geräten besteht ein Computersystem?

Hauptkomponenten (Zentraleinheit):

- Prozessor (Central Processing Unit = CPU)
- Arbeitsspeicher (Random Access Memory = RAM)
(temporäres Speichern von Programmen und Daten)

Peripheriegeräte:

- Eingabegeräte: Tastatur, Maus, Scanner, Mikrofon, ...
- Ausgabegeräte: Monitor, Drucker, Lautsprecher, ...
- Datenspeichergeräte: Festplatte, CD, DVD, USB-Stick, ...
(Hintergrundspeicher)
- Kommunikationsgeräte: Netzwerkkarte, WLAN-Karte, ...



2 Von-Neumann-Rechner

ca. 1945: fest vorgegebene Programme

John von Neumann (1903-1957):

Programme befinden sich wie die zu bearbeitenden Daten im Speicher des Rechners.

➔ Universalrechner: von-Neumann-Rechner

Der von-Neumann-Rechner besteht aus **5** Grundeinheiten:

- **Rechenwerk** (Arithmetic Logic Unit = ALU)
führt einfache arithmetische und logische Operationen auf den Daten durch.
- **Steuerwerk** (Leitwerk, Control Unit)
steuert den Ablauf der Befehle eines Programms.
- **Speicherwerk** (Arbeitsspeicher, Hauptspeicher)
speichert sowohl Programme, als auch die zu bearbeitenden Daten.
- **Eingabewerk**
steuert Eingabe von Programmen und Daten in den Speicher.
- **Ausgabewerk**
steuert Ausgabe von Daten aus dem Speicher nach außen.

CPU

Rechenwerk

Steuerwerk



Bus-System

Adressbus

Datenbus

Steuerbus



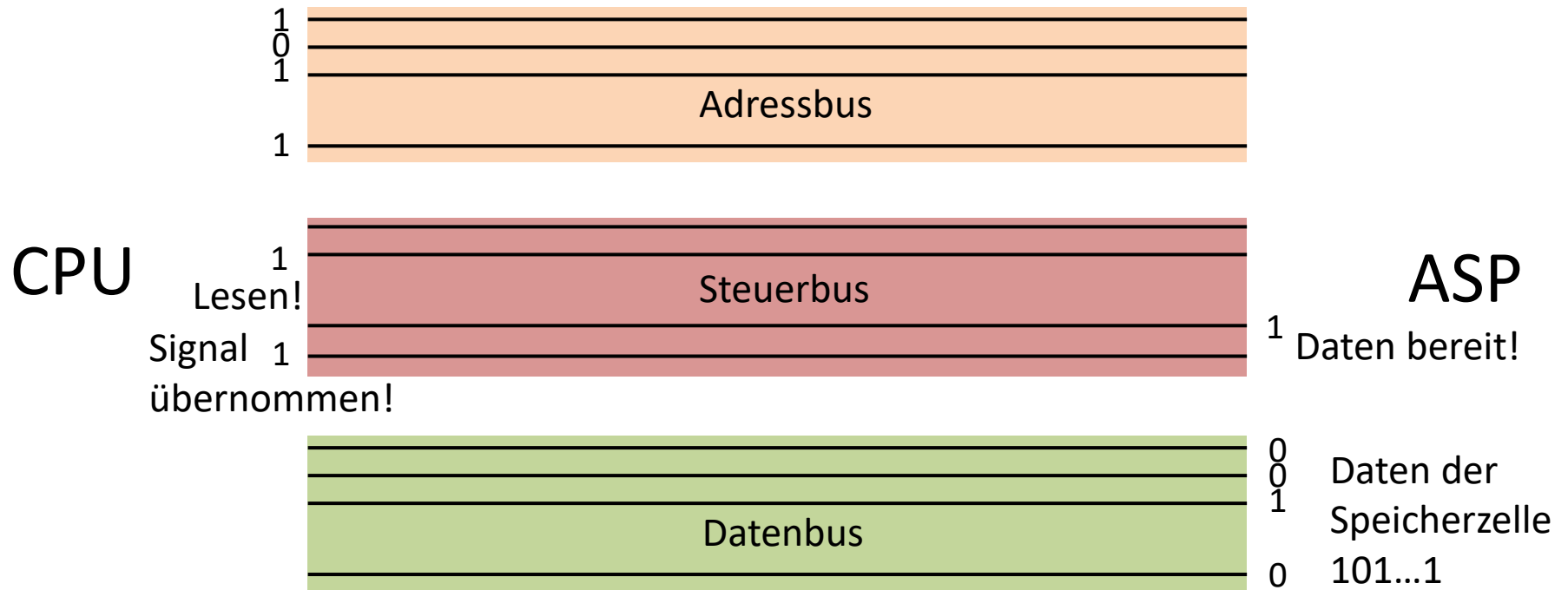
Ein-/Ausgabewerk

Speicherwerk

Bus-System:

- paralleler Bus:
viele Einzelleitungen (z.B. Adressbus)
- serieller Bus:
aufeinanderfolgende Signale (z.B. USB)

Beispiel: Kommunikation CPU – Arbeitsspeicher



früher:

Rechen- und Steuerwerk langsam,
d.h. Datenbereitstellung durch Bus unproblematisch

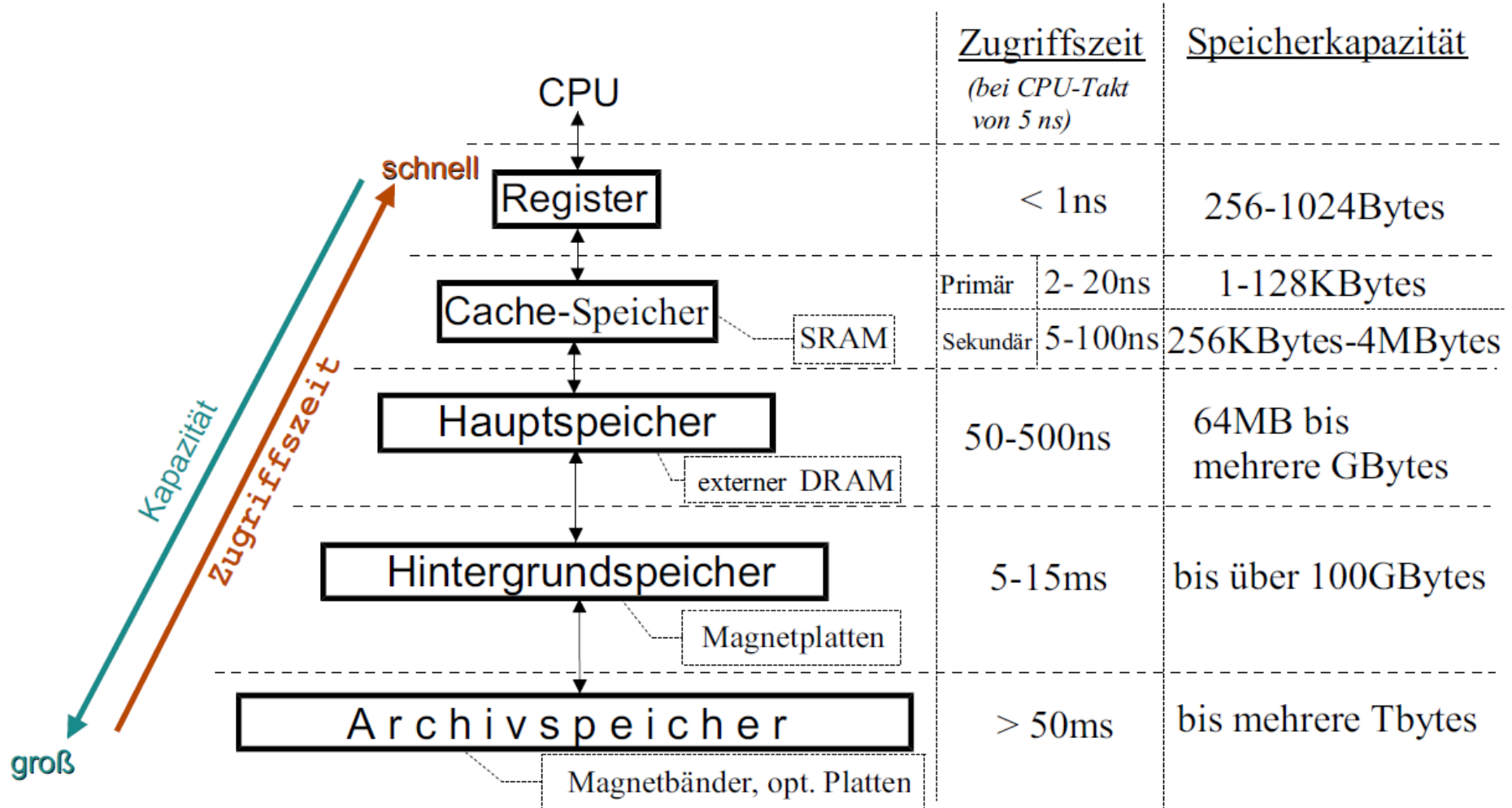
schnellere Prozessoren:

Verbindungssystem wird zum Engpass
(von Neumannscher Flaschenhals)

Lösung des Problems:

- schneller zugreifbarer Speicher (Cache)
- Parallelverarbeitung







Speicherhierarchie des „von-Neumann-Rechners“



AB: Binärsystem und Codierung

1.1 Kein Stellenwertsystem:

- römische Zahlen: I, II, III, IV,...
- Hieroglyphen

					
Einer 1	Zehner 10	Hunderter 100	Tausender 1.000	Zehntausender 10.000	Hunderttausender 100.000

1.2 Hexadezimalsystem

- Basis: 16
- Ziffern: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (=10), B (= 11), C (= 12), D (= 13), E (= 14), F (= 15)

$$2.1 \text{ a) } 1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$$

$$= 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 8 + 2 + 1 = 11$$

$$\text{b) } 110110_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 =$$

$$= 32 + 16 + 4 + 2 = 54$$

$$\text{c) } 100110 = 32 + 4 + 2 = 38$$

$$\text{d) } 11111111 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

$$2.2.1 \text{ a) } 21 = 16 + 4 + 1 = 2^4 + 2^2 + 2^0 =$$

$$= 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 10101_2$$

$$\text{b) } 65 = 1000001_2$$

$$\text{c) } 135 = 10000111_2$$

$$\text{d) } 543 = 1000011111_2$$

2.2.2 Resteverfahren

$$135 : 2 = 67 \text{ R}\mathbf{1}$$

$$67 : 2 = 33 \text{ R}\mathbf{1}$$

$$33 : 2 = 16 \text{ R}\mathbf{1}$$

$$16 : 2 = 8 \text{ R}\mathbf{0}$$

$$8 : 2 = 4 \text{ R}\mathbf{0}$$

$$4 : 2 = 2 \text{ R}\mathbf{0}$$

$$2 : 2 = 1 \text{ R}\mathbf{0}$$

$$1 : 2 = 0 \text{ R}\mathbf{1}$$



$$135 = 10000111_2$$

3.1

Arbeitsauftrag

$$2 \cdot 2 \cdot 2 = 2^3 = 8$$

Mögliche Kombinationen:

000; 001; 010; 100; 011; 101; 110; 111

3.2 a) 2 Byte = 16 bit: $2^{16} = 65536$

4 Byte = 32 bit: $2^{32} = 4294967296$

8 Byte = 64 bit: $2^{64} \approx 1,8 \cdot 10^{19}$

b) 2^n

4.1 Morsecode

Arbeitsauftrag

Signal in zwei Längen:

- kurz .
- lang –

z.B. A = . –



Pausen verschiedener Länge zwischen Signalen, Zeichen, Worten.

4.2 ASCII-Code

7-Bit-Zeichencodierung (128 Zeichen)

Tabelle: z.B. A = 65 = 1000001_2

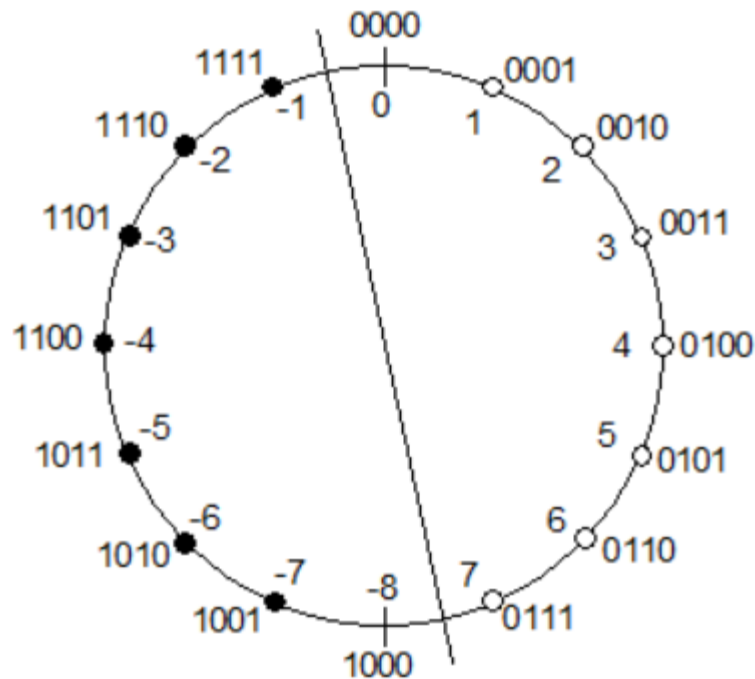
4.3 Codierung ganzer Zahlen

- Versuch: 1 Byte, 1. Bit für Vorzeichen (+ \rightarrow 0, - \rightarrow 1):
+5 = 0000 0101₂
-5 = 1000 0101₂
+5 - 5 = 1000 1010₂ = -10 
- Gesucht: Binärzahl, die zu 0000 0101₂ addiert 0 ergibt
+5 = 0000 0101₂
-5 = 1111 1011₂
+5 - 5 = 0000 0000₂ = 0 
- Gesucht: Binärdarstellung von -19
19 = 0001 0011₂
Komplement von 19: 1110 1100₂
Addiere 1 zum Komplement: -19 = 1110 1101₂

Welcher Zahlbereich kann mit 1 Byte bei der Codierung ganzer Zahlen abgedeckt werden?

Größte positive Zahl: $0111\ 1111_2 = 127$

Kleinste negative Zahl: $10000000_2 = -128$



Arbeitsauftrag:

Überprüfe die Abbildung:

Stelle die Zahlen von 0 bis 7 binär mit 4 Bit dar und ermittle jeweils das Zweierkomplement, also die Binärdarstellung von -0, ..., -7.

Darstellung von Gleitkommazahlen:

<http://de.wikipedia.org/wiki/Gleitkommazahl>

S. 99 / 3a)

$$153 = 128 + 16 + 8 + 1 = 2^7 + 2^4 + 2^3 + 2^0 = 10011001_2$$

$$87 = 64 + 16 + 4 + 2 + 1 = 2^6 + 2^4 + 2^2 + 2^1 + 2^0 = 1010111_2$$

3b)

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

3c) $11110000_2 = 128 + 64 + 32 + 16 = 240$

3d)

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

$$\begin{aligned} 153 + 111 &= 264 \\ &= 256 + 8 \end{aligned}$$

Benutze im folgenden jeweils eine 8-Bit-Darstellung:

- Stelle drei Dezimalzahlen als Binärzahlen dar und umgekehrt.
Überprüfe dein Ergebnis mit dem Taschenrechner.
Verwende auch das Resteverfahren.
- Stelle -12, -98, -101 binär dar (mit Zweierkomplement).
- Führe jeweils drei binäre Additionen und Subtraktionen durch.
Vergleiche mit dem Taschenrechner.