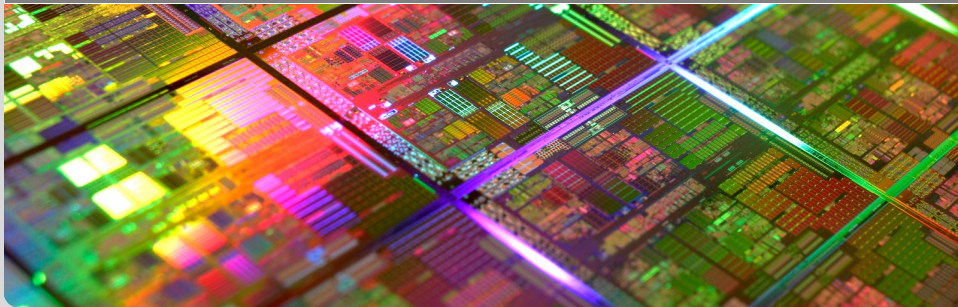


RO-Tutorien 3 / 6 / 12

Tutorien zur Vorlesung "Rechnerorganisation"

Christian A. Mandery

WOCHE 3 AM 13./14.05.2013



- Gleitkommazahlen
- BCD-Code
- Übungsaufgaben

- Das Komma wird an eine feste Stelle verfügt, mit der entsprechenden Wertigkeit der Stellen
- Beispiel für Format mit drei Nachkommazahlen:

Stelle	Wertigkeit
0. Bit (LSB)	$1/8$
1. Bit	$1/4$
2. Bit	$1/2$
3. Bit	1
4. Bit	2
...	...

- Festkomma-Darstellung hat Vorteile ...
 - Assoziativgesetz gilt
 - Keine Auslöschung
- ... aber auch Nachteile:
 - Sehr große und sehr kleine Zahlen nicht darstellbar
 - Differenz von aufeinanderfolgenden Zahlen ist über den ganzen Zahlenbereich gleich groß
- → Gleitkommazahlen

- Eine Gleitkommazahl wird durch drei Werte beschrieben:
 - **Vorzeichen-Bit**
 - **Mantisse**: Ziffern der Gleitkommazahl (als Nachkommastellen)
 - **Exponent**: Verschiebung der Mantisse, um die Zahl zu erhalten
- Der Exponent wird in der Offset-Darstellung (Exzess-Code) gespeichert und dann als Charakteristik bezeichnet
- Berechnung des Werts der Gleitkommazahl:
 - $Zahl = (-1)^{\text{Vorzeichen-}Bit} * (0, \text{Mantisse}) * b^{\text{Exponent}}$
 - Achtung: Bei normierter Darstellung steht vor dem Komma eine Eins!
 - Die Basis b ist meistens 2

- Eine Gleitkommazahl wird durch drei Werte beschrieben:
 - **Vorzeichen-Bit**
 - **Mantisse**: Ziffern der Gleitkommazahl (als Nachkommastellen)
 - **Exponent**: Verschiebung der Mantisse, um die Zahl zu erhalten
- Der Exponent wird in der Offset-Darstellung (Exzess-Code) gespeichert und dann als Charakteristik bezeichnet
- Berechnung des Werts der Gleitkommazahl:
 - $Zahl = (-1)^{Vorzeichen-Bit} * (0, Mantisse) * b^{Exponent}$
 - Achtung: Bei normierter Darstellung steht vor dem Komma eine Eins!
 - Die Basis b ist meistens 2

- Eine Gleitkommazahl wird durch drei Werte beschrieben:
 - **Vorzeichen-Bit**
 - **Mantisse**: Ziffern der Gleitkommazahl (als Nachkommastellen)
 - **Exponent**: Verschiebung der Mantisse, um die Zahl zu erhalten
- Der Exponent wird in der Offset-Darstellung (Exzess-Code) gespeichert und dann als Charakteristik bezeichnet
- Berechnung des Werts der Gleitkommazahl:
 - $Zahl = (-1)^{\text{Vorzeichen-}Bit} * (0, \text{Mantisse}) * b^{\text{Exponent}}$
 - Achtung: Bei normierter Darstellung steht vor dem Komma eine Eins!
 - Die Basis b ist meistens 2

- Jeder Binärzahl außer der Null enthält irgendwo eine Eins
- Wenn die Zahl nicht zu klein ist, kann man die erste Eins mit dem Exponent immer vor das Komma schieben
- → Normalisierte Darstellung
- Durch eine implizite Eins statt Null vor dem Komma gewinnt man ein Bit Genauigkeit
- Aber: Man braucht dann eine spezielle Darstellung für die Null (und ggf. sehr kleine Zahlen)

- Jeder Binärzahl außer der Null enthält irgendwo eine Eins
- Wenn die Zahl nicht zu klein ist, kann man die erste Eins mit dem Exponent immer vor das Komma schieben
- → Normalisierte Darstellung
- Durch eine implizite Eins statt Null vor dem Komma gewinnt man ein Bit Genauigkeit
- Aber: Man braucht dann eine spezielle Darstellung für die Null (und ggf. sehr kleine Zahlen)

- Um ein einheitliches Format zu verwenden, legt man fest, welche Bits der Gleitkommazahl Vorzeichen, Mantisse und Exponent spezifizieren
- Eine solche Festlegung macht der IEEE 754-Standard (“IEEE Standard for Binary Floating-Point Arithmetic for Microprocessor Systems”)

- Bestimmte Gleitkommazahlen, mit denen man eine Gleitkomma-Darstellung charakterisieren kann, haben einen speziellen Wert.
 - **Maxreal**: Größte normalisierte Zahl
 - **Minreal**: Kleinste normalisierte Zahl
 - **Smallreal**: Kleinste Zahl x mit $1 + x \neq 1$

- Single Precision (“float“):
 - 32 Bit = 1 Bit Vorzeichen + 8 Bit Charakteristik + 23 Bit Mantisse
 - Offset für Charakteristik: 127
- Double Precision (“double“):
 - 64 Bit = 1 Bit Vorzeichen + 11 Bit Charakteristik + 52 Bit Mantisse
 - Offset für Charakteristik: 1023

- Single Precision (“float“):
 - 32 Bit = 1 Bit Vorzeichen + 8 Bit Charakteristik + 23 Bit Mantisse
 - Offset für Charakteristik: 127
- Double Precision (“double“):
 - 64 Bit = 1 Bit Vorzeichen + 11 Bit Charakteristik + 52 Bit Mantisse
 - Offset für Charakteristik: 1023

- $1 \leq \textit{Charakteristik} \leq 254$:
 - $\textit{Exponent} = \textit{Charakteristik} - 127$
 - Normalisierung
- $\textit{Charakteristik} = 0$:
 - Kleinstmöglicher Exponent (-126)
 - Keine Normalisierung (nur so ist die Null darstellbar!)
- $\textit{Charakteristik} = 255$:
 - $\textit{Mantisse} = 0$: Überlauf (“unendlich”)
 - $\textit{Mantisse} \neq 0$: NaN (“Not a Number”)

- $1 \leq \textit{Charakteristik} \leq 254$:
 - $\textit{Exponent} = \textit{Charakteristik} - 127$
 - Normalisierung
- $\textit{Charakteristik} = 0$:
 - Kleinstmöglicher Exponent (-126)
 - Keine Normalisierung (nur so ist die Null darstellbar!)
- $\textit{Charakteristik} = 255$:
 - $\textit{Mantisse} = 0$: Überlauf ("unendlich")
 - $\textit{Mantisse} \neq 0$: NaN ("Not a Number")

- $1 \leq \textit{Charakteristik} \leq 254$:
 - $\textit{Exponent} = \textit{Charakteristik} - 127$
 - Normalisierung
- $\textit{Charakteristik} = 0$:
 - Kleinstmöglicher Exponent (-126)
 - Keine Normalisierung (nur so ist die Null darstellbar!)
- $\textit{Charakteristik} = 255$:
 - $\textit{Mantisse} = 0$: Überlauf (“unendlich”)
 - $\textit{Mantisse} \neq 0$: NaN (“Not a Number”)

- **B**inary **C**oded **D**ecimal
- Je vier Bits werden zu einer dezimalen Stelle zusammengefasst und als Tetrade bezeichnet
- Die hexadezimalen Ziffern, die keine dezimale Ziffer darstellen (A-F) werden als Pseudotetraden bezeichnet und sind ungültig
- Kommadarstellung z.B. durch Festkomma-Darstellung
- Beispiel: $1001010100010111_{BCD} = 9517_{10}$

Wert	Codewort	Wert	Codewort
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

- Vor- und Nachteile des BCD-Code?

Wert	Codewort	Wert	Codewort
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

- Vor- und Nachteile des BCD-Code?

- 1 Wie werden Gleitkommazahlen üblicherweise dargestellt?
- 2 Wie wird diese Darstellung auf eine Binärdarstellung abgebildet?
- 3 Wie sehen die entsprechenden Größen bei einer 32-Bit Gleitkommazahl nach dem IEEE-754-Standard aus?

- 1 Wie werden Gleitkommazahlen üblicherweise dargestellt?
- 2 Wie wird diese Darstellung auf eine Binärdarstellung abgebildet?
- 3 Wie sehen die entsprechenden Größen bei einer 32-Bit Gleitkommazahl nach dem IEEE-754-Standard aus?

- 1 Wie werden Gleitkommazahlen üblicherweise dargestellt?
- 2 Wie wird diese Darstellung auf eine Binärdarstellung abgebildet?
- 3 Wie sehen die entsprechenden Größen bei einer 32-Bit Gleitkommazahl nach dem IEEE-754-Standard aus?

Beantworten Sie folgende Fragen zur BCD-Kodierung:

- 1 Wie viele Bits benötigt man mindestens, um alle möglichen Ziffern einer Dezimalzahl binär kodieren zu können?
- 2 Wie wird diese kleinste Einheit an Bits auch bezeichnet?
- 3 Sind alle möglichen Kodierungen gültige Dezimalzahlen?
- 4 Was bedeutet: Eine Dezimalzahl ist BCD-kodiert?

Beantworten Sie folgende Fragen zur BCD-Kodierung:

- 1 Wie viele Bits benötigt man mindestens, um alle möglichen Ziffern einer Dezimalzahl binär kodieren zu können?
- 2 Wie wird diese kleinste Einheit an Bits auch bezeichnet?
- 3 Sind alle möglichen Kodierungen gültige Dezimalzahlen?
- 4 Was bedeutet: Eine Dezimalzahl ist BCD-kodiert?

Beantworten Sie folgende Fragen zur BCD-Kodierung:

- 1 Wie viele Bits benötigt man mindestens, um alle möglichen Ziffern einer Dezimalzahl binär kodieren zu können?
- 2 Wie wird diese kleinste Einheit an Bits auch bezeichnet?
- 3 Sind alle möglichen Kodierungen gültige Dezimalzahlen?
- 4 Was bedeutet: Eine Dezimalzahl ist BCD-kodiert?

Beantworten Sie folgende Fragen zur BCD-Kodierung:

- 1 Wie viele Bits benötigt man mindestens, um alle möglichen Ziffern einer Dezimalzahl binär kodieren zu können?
- 2 Wie wird diese kleinste Einheit an Bits auch bezeichnet?
- 3 Sind alle möglichen Kodierungen gültige Dezimalzahlen?
- 4 Was bedeutet: Eine Dezimalzahl ist BCD-kodiert?

Übungsaufgabe 3

Wandeln Sie nun die folgenden Zahlen in ihre entsprechende BCD-Darstellung um:

- 1 12345
- 2 101010
- 3 8442219

Übungsaufgabe 3

Wandeln Sie nun die folgenden Zahlen in ihre entsprechende BCD-Darstellung um:

- 1 12345
- 2 101010
- 3 8442219

Übungsaufgabe 3

Wandeln Sie nun die folgenden Zahlen in ihre entsprechende BCD-Darstellung um:

- 1 12345
- 2 101010
- 3 8442219

Übungsaufgabe 4

Sind folgende BCD-kodierten Zahlen gültige Dezimalzahlen? Falls ja, welchen Wert stellen sie dar?

- 1 0101 0010 1001
- 2 1001 1000 0111
- 3 0100 1001 1010
- 4 0010 1000 0001 1100

Übungsaufgabe 4

Sind folgende BCD-kodierten Zahlen gültige Dezimalzahlen? Falls ja, welchen Wert stellen sie dar?

1 0101 0010 1001

2 1001 1000 0111

3 0100 1001 1010

4 0010 1000 0001 1100

Übungsaufgabe 4

Sind folgende BCD-kodierten Zahlen gültige Dezimalzahlen? Falls ja, welchen Wert stellen sie dar?

- 1 0101 0010 1001
- 2 1001 1000 0111
- 3 0100 1001 1010
- 4 0010 1000 0001 1100

Übungsaufgabe 4

Sind folgende BCD-kodierten Zahlen gültige Dezimalzahlen? Falls ja, welchen Wert stellen sie dar?

- 1 0101 0010 1001
- 2 1001 1000 0111
- 3 0100 1001 1010
- 4 0010 1000 0001 1100

Übungsaufgabe 5

Gegeben sei folgender C-Code, der auf einer 64-Bit-Plattform ausgeführt wird. Geben Sie die Ausgabe an.

```
#include <stdio.h>

int main()
{
    printf("sizeof(int) = %lu\n", sizeof(int));
    printf("sizeof(long) = %lu\n", sizeof(long));

    int int_array[] = {1, 2, 3, 4, 5, 6};
    long long_array[] = {1, 2, 3, 4, 5, 6};

    int* int_pointer = int_array;
    long* long_pointer = long_array;

    printf("\nint_pointer = %d\n", (int)int_pointer);
    printf("long_pointer = %d\n", (int)long_pointer);

    int_pointer++;
    long_pointer++;

    /* Fortsetzung von links */
    printf("\nint_pointer = %d\n", (int)int_pointer);
    printf("long_pointer = %d\n", (int)long_pointer);

    int_pointer += 2;
    long_pointer += 3;

    printf("\n*int_pointer = %d\n", *int_pointer);
    printf("*long_pointer = %ld\n", *long_pointer);

    printf("\n*int_pointer = %d\n", *(int_pointer + 2));
    printf("*long_pointer = %ld\n", *(long_pointer + 2));

    return 0;
}
```

Welcome to text-only Counterstrike.

You are in a dark, outdoor map.

> GO NORTH

You have been pwned by a grue.

Quelle: <http://xkcd.com/91/>