

Einführung in die Informatik
Ausarbeitung Übung 2

Julian Bertol

November 6, 2023

1 Aufgabe 1

1 Umrechnung zwischen Zahlensystemen

Üben Sie die Umrechnung von verschiedenen Ausgangs-Zahlensystemen in verschiedene Basissysteme:

Folgende Zahlen sollen umgerechnet werden:

19210,0C16,7648,011111102

Berechnen Sie für jede Zahl die Darstellung in den anderen hier verwendeten Zahlensystemen.

Bestimmen Sie für die nachfolgenden aufeinander aufbauenden Festlegungen jeweils den unteren und oberen darstellbaren Wert im Dezimalsystem und im Hexadezimalsystem.

1.1 Problem

Folgende Zahlen müssen umgerechnet werden:

19210,0C16,7648,011111102

1.2 Lösungskonzept

Verstehen wie die Umrechnung funktioniert mit Hilfe von Google

1.3 konkrete L"osung

192₁₀ ist eine Dezimalzahl, da es sich um das 10er Zahlensystem handelt.

Umrechnung in Binär

$$192 / 2 = 96 \text{ Rest } 0$$

$$96 / 2 = 48 \text{ Rest } 0$$

$$48 / 2 = 24 \text{ Rest } 0$$

$$24 / 2 = 12 \text{ Rest } 0$$

$$12 / 2 = 6 \text{ Rest } 0$$

$$6 / 2 = 3 \text{ Rest } 0$$

$$3 / 2 = 1 \text{ Rest } 1$$

$$1 / 2 = 0 \text{ Rest } 1$$

192 in Binär ist = 11000000

192 in Hexadezimal

$$192 / 16 = 12 \text{ Rest } 0$$

$$12 / 16 = 0 \text{ Rest } 12$$

192 in Hexadezimal = C0

192 in Oktalzahlsystem

$$192 / 8 = 24 \text{ Rest } 0$$

$$24 / 8 = 3 \text{ Rest } 0$$

$$3 / 8 = 0 \text{ Rest } 3$$

192 in Oktal = 300

*oC*₁₆ in Dezimal:

$$= 12 * 16 + 0 * 16^1$$

$$= 12$$

*oC*₁₆ in Binär

$$12 / 2 = 6 \text{ Rest } 0$$

$$6 / 2 = 3 \text{ Rest } 0$$

$$3 / 2 = 1 \text{ Rest } 1$$

$$1 / 2 = 0 \text{ Rest } 1$$

*oC*₁₆ in Binär = 1100

*oC*₁₆ in Oktal

$$12 / 8 = 1 \text{ Rest } 4$$

$$1 / 8 = 0 \text{ Rest } 1$$

*oC*₁₆ in Oktal = 14

764₈ Oktal in Dezimal
= 4 * 8 + 6 * 8¹ + 7 * 8²
= 500

764₈ in Hexadezimal
500 / 16 = 31 Rest 4
31 / 16 = 1 Rest 15
1 / 16 = 0 Rest 1
764₈ in Hexa = 1F4

764₈ in Binär
500 / 2 = 250 Rest 0
250 / 2 = 125 Rest 0
125 / 2 = 62 Rest 1
62 / 2 = 31 Rest 0
31 / 2 = 15 Rest 1
15 / 2 = 7 Rest 1
7 / 2 = 3 Rest 1
3 / 2 = 1 Rest 1
1 / 2 = 0 Rest 1
764₈ in Binär = 111110100

01111110₂ von Binär in Dezimal
= 2¹+2²+2³+2+2+2
= 126

01111110₂ in Hexadezimal
126 / 16 = 7 Rest 14
7 / 16 = 0 Rest 7
01111110₂ in Hexa = 7E

01111110₂ in Oktal
126 / 8 = 15 Rest 6
15 / 8 = 1 Rest 7
1 / 8 = 0 Rest 1
01111110₂ in Oktal = 176

HexadezimalSystem erklärt:
Zahlen von 0-9 wie auch im normalen DezimalSystem.
Buchstaben von A bis F
A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

Oktalsystem erklärt:

Oktal bedeutet 8.

Man nimmt also jeden Koeffizienten und multipliziert ihn mit 8 und eine Hochzahl

Dabei beginnt man von rechts nach links und erhöht die Zahl hochzahl immer um 1. Man beginnt mit 0.

Am Ende werden alle Zahlen addiert.

Binärsystem erklärt:

Man hat eine Zahlenfolge von 0 und 1. Man beginnt von rechts nach links zu lesen. Ist die Zahl 0,

kann man diese überspringen. Ist die Zahl 1 rechnet man mit 2 und eine Hochzahl. Dabei wird die hochzahl von rechts nach links

immer um 1 erhöht, egal, ob die Binär Zahl 1 oder 0 ist. Am Ende addiert man alle Zahlen zusammen.

Beispiel IPV4 Adressen

- Das höchstwertige Bit muss 0 sein: Die Wertdarstellung geht von 0 bis 127:

$$01111111 = 127$$

- Jetzt muss das höchstwertige Bit immer 1 sein, das zweithöchste Bit muss 0 sein:

Die Wertdarstellung geht von: 128 bis 191

Niedrigste: $10000000 = 128$

Höchste: $10111111 = 191$

- Jetzt müssen das höchste und das zweithöchste Bit 1 gesetzt sein, das dritthöchste

Bit muss 0 sein: Die Wertdarstellung geht von 0 bis 10

Niedrigste: $11000000 = 192$

Höchste: $11011111 = 232$

1.4 Tests

Nach einen Umrechner im Internet suchen und die ergebnisse vergleichen.

2 Aufgabe 2

2.1 Problem

- zuerst muss man die GröÖte 4 Bit Zahl in Dezimal umrechnen
- Dann muss eine Tabelle ausgefüllt werden indem man in verschiedenen Systemen umrechnet

2.2 Lösungskonzept

Mit den vorher gelernten eigenschaften die Lösungen berechnen

Dualsystem	Oktalsystem	Hexadezimalsystem
101101.101	55.5	2D.A
10101011.11001101	253.632	AB.CD

h

2.3 Lösung

Die größte 4 bit zahl im Binärsystem ist:

$$1111 = 15$$

Die kommastellen werden mit $1 * 2^{-n}$ berechnet. Dabei ist n die Bitzahl

Berechnung der Kommastelle:

$$(1 * 2^{-1}) + (1 * 2^{-2}) + (1 * 2^{-3}) + (1 * 2^{-4})$$

Das ergibt dann 0.9375

Somit ist die größte darstellbare Zahl im Binärsystem mit 4 bit 15.9375

Die größte 4 bit Zahl im Hexadezimalsystem ist:

$$FFFF = 65535$$

Auch hier werden die Nachkomma zahl mit $15 * 16^{-n}$ berechnet

$$(15 * 16^{-1}) + (15 * 16^{-2}) + (15 * 16^{-3}) + (15 * 16^{-3})$$

somit ist das ergebniss : .9999847412109375

Die größte 4 bit Zahl im Hexadezimalsystem ist:

7777 = 4095 Die Zahlen nach dem Komma müssen hier dann als $7 * 8^{-n}$ geschrieben werden. Somit $(7 * 8^{-1}) + (7 * 8^{-2}) + (7 * 8^{-3}) + (7 * 8^{-4})$ was ca.999755859375 ergibt.

Somit ist die Gröste Zahl 4095.999755859375

3 Aufgabe 3

textbfBinäre Addition $125 + 199$

In Binär umwandeln:

01111101

11000111

=101000100

in Dezimal 324

$27+30$

11011

11110

=111001

in Dezimal: 57

$115 + 21$

1110011

0010101

=10001000

in Dezimal 136

Binäre Subtraktion 55 - 120

00110111

01111000-

Zuerst muss das unter dem Bruch in ein Zweierkomplement umgerechnet werden

Dabei wird die 1 zur 0 und die 0 zur 1. Am Ende wird noch +1 addiert

10000111

00000001+

= 10001000

Dieses Ergebniss wird dann vom ersten im Ursprünglichen Bruch abgezogen

00110111

10000111-

Das ergibt dann 10111111

42 - 12

Umrechnen in Binär:

00101010

00001100-

=00011110

Hier normale Subtraktion verwendet.

Wichtige Regeln:

1-1 = 0

0-1 = 1 Rest 1

1-0 = 1

0-0 = 0

18 - 105

Umrechnen in Binär:

00010010

01101001-

=10101001

4 Resumee zur dieser "Übungsaufgabe

Insgesamt habe ich c.a 4 Stunden mit dieser Aufgabe verwendet.

- Durchföhrung
- Dokumentation

Welche groöen Probleme waren zu lösen? Ich musste mir selber Wissen über die verschiedenn Systeme aneignen. Das war nicht immer einfach, da dass Internet dort nicht immer eine Hilfe sein konnte.

References

- [1] Walter Schmidt, Jörg Knappen, Hubert Partl, Irene Hyn: $\text{\LaTeX} 2_{\epsilon}$ -Kurzbeschreibung, Version 2.3, 2003