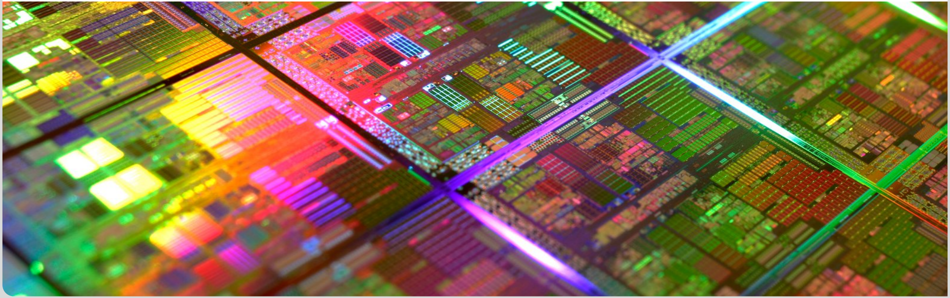
Tutorium Rechnerorganisation

Woche 1

Tutorien 3 und 4 zur Vorlesung Rechnerorganisation



Heute

- Vorstellung des Tutors
- Vorstellungsrunde
- Organisatorisches
- Darstellung von Zahlen in verschiedenen Zahlensystemen und Umwandlung zwischen verschiedenen Zahlensystemen
- BCD-Code
- Repräsentation vorzeichenbehafteter Zahlen im Rechner
- Übungsaufgaben

Vorstellung des Tutors

- Name: Christian Mandery
- Studiengang: Diplom-Informatik im 6. Semester
- Kontakt über:
  - Mail: mail@chrismandery.de
  - XMPP bzw. Google Talk: chrism2711@googlemail.com
  - Facebook: http://facebook.com/chrismandery
  - Twitter: http://twitter.com/chrismandery

Vorstellungsrunde

Und jetzt seid ihr dran!

Tutorium

- Das Tutorium ersetzt nicht die Vorlesung/Übung, sondern ergänzt sie sinnvoll. Der Besuch der Vorlesung und Übung wird empfohlen.
- Bei Fragen im Tutorium sofort fragen. Es gibt keine dummen Fragen!
- Es gibt zwei Tutorien von mir:
  - Montag, 14.00 Uhr, SR -120
  - Donnerstag, 15.45 Uhr, SR -119
- Ihr könnt den Termin besuchen, den ihr bevorzugt, egal in welches der beiden Tutorien ihr eingeteilt wurdet.

Homepages

- Homepage der Vorlesung: http://ti.itec.uka.de
  - Vorlesungs- und Übungsfolien
  - Übungsblätter und Lösungen
  - Tutoriumsblätter (Aufgaben, die wir im Tutorium rechnen)
  - Links und Literaturangaben
  - Organisatorisches
- Homepage des Tutoriums: http://ro-tut.chrismandery.de
  - Tutoriumsfolien zum Download
  - Organisatorisches zum Tutorium
  - Links zu wichtigen Seiten

Homepages

- Homepage der Vorlesung: http://ti.itec.uka.de
  - Vorlesungs- und Übungsfolien
  - Übungsblätter und Lösungen
  - Tutoriumsblätter (Aufgaben, die wir im Tutorium rechnen)
  - Links und Literaturangaben
  - Organisatorisches
- Homepage des Tutoriums: http://ro-tut.chrismandery.de
  - Tutoriumsfolien zum Download
  - Organisatorisches zum Tutorium
  - Links zu wichtigen Seiten

Übungsblätter

- Die Übungsblätter erscheinen immer (spätestens) montags, beginnend diese Woche, auf der Vorlesungs-Homepage
- Keine physische Ausgabe der Übungsblätter
- Ausarbeitungen können bis zum darauffolgenden Montag, 13.15 Uhr, in den Einwurfkasten im Keller des Geb. 30.43 (“Infobau”) eingeworfen werden
- Eine Abgabe in Lerngruppen ist nicht gestattet (Ausnahme: MIPS-Programme, hierzu später mehr)

Übungsschein

- Das Erzielen des Übungsscheins ist freiwillig, d.h. dass auch die Bearbeitung der Übungsblätter freiwillig ist
- Bei erreichtem Übungschein wird ein Bonuspunkt auf eine **bestandene** Klausur angerechnet
- Nicht nur deshalb ist es äußerst empfehlenswert, sich mit den Übungsblätter zu beschäftigen!
- Den Übungsschein erhält, wer:
  - zu 8 der 10 Übungsblätter eine Ausarbeitung abgegeben hat und
  - 50% der möglichen Punkte auf den Übungsblättern erzielt hat und
  - regelmäßig am Tutorium teilgenommen und erkennbare Bereitschaft zur aktiven Mitarbeit gezeigt hat
- Keinen Übungsschein erhält, wer zweimalig eine auf dem Übungsblatt gelöste Aufgabe im Tutorium nicht vorrechnen konnte

Probeklausur

- Freiwillige Probeklausur findet gegen Ende des Semesters statt
- Orientiert sich an echten Klausuren und kann daher als “Lernkontrolle” für euch dienen
- Maximal zwei weitere Bonuspunkte für eine **bestandene** Klausur erreichbar:
    - Note 1 => 2 Bonuspunkte
    - Note 2 => 1,5 Bonuspunkte
    - Note 3 => 1 Bonuspunkt
    - Note 4 => 0,5 Bonuspunkte
    - Note 5 => 0 Bonuspunkte

Wozu Zahlensysteme?

- Man kann jede Zahl im dezimalen System darstellen
- Wozu brauchen wir dann überhaupt diese ganzen verschiedenen Zahlensysteme?!

Gebräuchliche Zahlensysteme

- Binäres Zahlensystem (Basis 2)
- Oktales Zahlensystem (Basis 8)
- Dezimales Zahlensystem (Basis 10)
- Hexadezimales Zahlensystem (Basis 16)

Umrechnen

- Euklidischer Algorithmus
- Horner-Schema
- Spezialverfahren
  - z.B. wenn eine Basis eine Potenz der anderen ist- warum?

Euklidischer Algorithmus

1. Bestimme Anzahl n der notwendigen Ziffern vor dem Komma im Zielsystem
2. Teile Zahl durch b^n, schreibe Quotient als eine Ziffer der Zahl im Zielsystem und merke Rest der Division
3. Setze Zahl $:=$ Rest der Division, dekrementiere n
4. Gehe zu Schritt 2, falls $Rest \neq 0$ und gewünschte Genauigkeit noch nicht erreicht

Horner-Schema, Vorkomma-Teil

1. Setze Zahl $:=$ Vorkomma-Teil
2. Teile Zahl durch Basis, schreibe Rest als eine Ziffer der Zahl im Zielsystem (beginnend mit der niedrigstwertigen) und merke Ergebnis
3. Setze Zahl $:=$ Ergebnis der Division
4. Gehe zu Schritt 2, falls $Zahl \neq 0$

Horner-Schema, Nachkomma-Teil

1. Setze Zahl $:=$ Nachkomma-Teil
2. Multipliziere Zahl mit Basis, schreibe Vorkomma-Teil des Ergebnis als eine Nachkomma-Ziffer der Zahl im Zielsystem (beginnend mit der höchstwertigen) und merke Ergebnis
3. Setze Zahl $:=$ Ergebnis unter Entfernung des Vorkomma-Teils
4. Gehe zu Schritt 2, falls $Zahl \neq 0$ und keine Wiederholung festgestellt

BCD-Code

- Steht für "Binary Coded Decimal"
- Je vier Bits werden zu einer dezimalen Stelle zusammengefasst und als Tetraden bezeichnet
- Die hexadezimalen Ziffern, die keine dezimale Ziffer darstellen (also A-F) werden als Pseudotetraden bezeichnet und sind ungültig
- Kommadarstellung z.B. durch Festkomma möglich
- Beispiel: $1001010100010111_{BCD} = 9517_{10}$
- Was sind die Vor- und Nachteile dieser Darstellung?

Wie kann man vorzeichenbehaftete Zahlen darstellen?

- Vorzeichen-Bit-Darstellung
- Einerkomplement-Darstellung
- Zweierkomplement-Darstellung
- Offset-Binary-Darstellung (Exzesscode)

Vorzeichen-Bit-Darstellung

- Höchstwertiges Bit (MSB) gibt das Vorzeichen der danach folgenden Zahl an (0 = positiv, 1 = negativ)
- Nachteil: Darstellung der 0 nicht eindeutig (kann als +0 und -0 dargestellt werden)
- Beim Rechnen müssen die Vorzeichen separat betrachtet werden (=> Schaltungsaufwand steigt)

Einerkomplement-Darstellung

- Negative Zahlen werden durch bitweise Negierung der entsprechenden positiven Zahl dargestellt
- Beispiel: -3_{10} mit 4 Bit: $-(0011_2) = 1100_2(EK)$
- Ebenfalls redundant: 0 kann auf zwei Arten dargestellt werden
- Vorteil: Addition ohne gesonderte Betrachtung des Vorzeichens

Zweierkomplement-Darstellung

- Wie Einerkomplement, aber bei negativen Zahlen wird nach der Negation noch 1 addiert
- Deshalb kleinste darstellbare Zahl betragsmäßig eins größer als die größte darstellbare Zahl
- Beispiel:
  - Gegeben: 8 Bit-Zahlen
  - $2^8 = 256$ Werte sind darstellbar
  - Zahlenbereich im ZK: -128 bis +127
- Keine Redundanz mehr: 0 ist positive Zahl
- Vorteil: Addition ohne gesonderte Betrachtung des Vorzeichens (wie Einerkomplement)

Wie funktionieren die Komplement-Darstellungen?

- Man rechnet auf dem Restklassenring von mod 2^n
- Addition bzw. Subtraktion entspricht einfach Bewegung auf diesem Ring im bzw. gegen den Uhrzeigersinn
- Deswegen auch Überlauf: Größte darstellbare Zahl (0111...) + 1 = kleinste darstellbare Zahl (1000...)!
- Woran erkennt man bei einer komplement-dargestellten Zahl das Vorzeichen?

Offset-Binary-Darstellung

- 000... entspricht kleinster darstellbarer Zahl, 111... entspricht größter darstellbarer Zahl
- Wird z.B. bei der Darstellung der Charakteristik von Gleitkommazahlen im IEEE 754-Standard verwendet (kommt später)

Aufgabe 1

1. Wandeln Sie die Zahl $86,22_{10}$ in eine Zahl zur Basis 5 um.
2. Wandeln Sie die Zahl $435,317_8$ in eine Zahl zur Basis 16 um.
3. Wandeln Sie die Zahl -65_{10} in eine 16-Bit Zweierkomplement Zahl um.
4. Wandeln Sie die Zweierkomplement-Zahl $(1111111100111100)_{ZK}$ in eine dezimale Zahl um.

Aufgabe 2

Gegeben sei die folgende 32-Bit Folge
1001 1000 0000 0000 0000 0000 0000 0011
Was stellt diese Folge dar, wenn sie interpretiert wird als

1. Vorzeichenlose Dualzahl. Geben Sie den dezimalen Wert an.
2. Vorzeichenbehaftete Zahl in Betrag-Vorzeichen-Form. Geben Sie den dezimalen Wert an.
3. Vorzeichenbehaftete Zahl in Einerkomplement-Form. Geben Sie den dezimalen Wert an.
4. Vorzeichenbehaftete Zahl in Zweierkomplement-Form. Geben Sie den dezimalen Wert an.
5. BCD-Zahl.

Hinweis: Sie brauchen die Zweier-Potenzen nicht explizit auszurechnen.

Fertig!

$$\begin{bmatrix} \cos 90^\circ & \sin 90^\circ \\ -\sin 90^\circ & \cos 90^\circ \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} a_2 & a_1 \end{bmatrix}$$