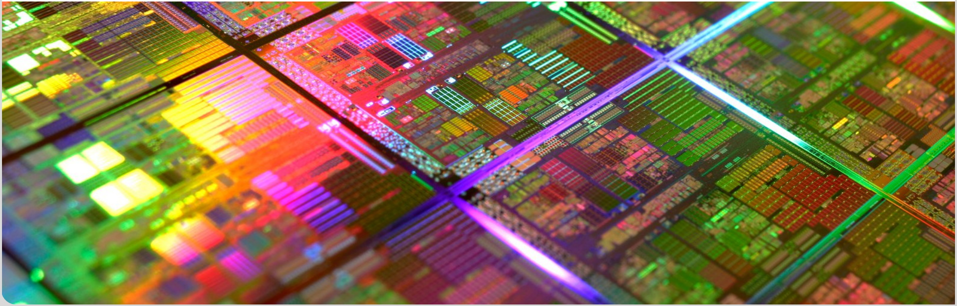
Karlsruher Institut für Technologie

DuE-Tutorien 4 und 6

Tutorien zur Vorlesung “Digitaltechnik und Entwurfsverfahren”

Christian A. Mandery

WOCHE 2 AM 30.10.2012



Heute

- Vorstellung / Organisatorisches
- Zahlensysteme und Umrechnung $*$
- Darstellung vorzeichenbehafteter Zahlen $*$
- Codes (BCD-, Gray-, Aiken- und Stibitz-Code)
- Hammingcode
- Gleitkomma-Darstellung / IEEE 754
- Gesetze der Schaltalgebra / Boolesche Funktionen
- Übungsaufgaben

$*$ Wiederholung des Stoffs der VL “Rechnerorganisation” (TI-1)

Vorstellung

- Name: Christian Mandery
- Studiengang: Diplom-Informatik im 11. Semester
- Zum 8. Mal Tutor in TI (Rechnerorganisation/Digitaltechnik)
- Mail: mail@chrismandery.de
- Homepage des Tutoriums: http://tutorium.chrismandery.de/

Vorstellungsrunde

Und jetzt seid ihr dran!

Tutorium

Karlsruher Institut für Technologie

- 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:
  - Dienstags, 14.00 Uhr, SR 148, Geb. 50.20 (Tut. 4)
  - Dienstags, 17.30 Uhr, SR 148, Geb. 50.20 (Tut. 6)
- Ihr könnt den Termin besuchen, den ihr bevorzugt, egal in welches der beiden Tutorien ihr eingeteilt wurdet.

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:
  - Dienstags, 14.00 Uhr, SR 148, Geb. 50.20 (Tut. 4)
  - Dienstags, 17.30 Uhr, SR 148, Geb. 50.20 (Tut. 6)
- 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://tutorium.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://tutorium.chrismandery.de
  - Tutoriumsfolien zum Download
  - Organisatorisches zum Tutorium
  - Links zu wichtigen Seiten

Übungsblätter

- Die Übungsblätter erscheinen immer am Anfang der Woche auf der Vorlesungs-Homepage (keine physische Ausgabe)
- Ausarbeitungen bis zum darauffolgenden Montag, 13.15 Uhr
  - Einwurfkasten im Keller des Geb. 50.43 (“Infobau”)
- Eine Abgabe in Lerngruppen ist nicht gestattet

Übungsschein: Allgemeines

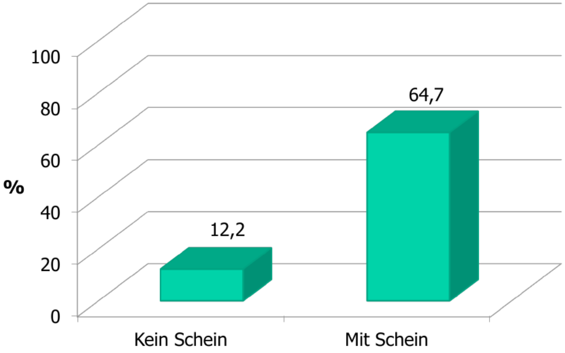
- Das Erzielen des Übungsscheins ist freiwillig, d.h. dass auch die Bearbeitung der Übungsblätter freiwillig ist
- Bei erreichtem Übungsschein wird ein Bonuspunkt auf eine **bestandene** Klausur angerechnet
- Nicht nur deshalb ist es äußerst empfehlenswert, sich mit den Übungsblätter zu beschäftigen!

Übungsschein: Motivation

Übungsschein: Motivation

Klausur WS 2011

Teilnehmer, die **bestanden** haben ...



Übungsschein: Bedingungen

- Den Übungsschein erhält, wer:
  - zu mindestens 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
- Maßgeblich hierzu ist das “Merkblatt zu den Übungen”, welches auf der Vorlesungshomepage heruntergeladen werden kann

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 $\Rightarrow$ 2 Bonuspunkte
  - Note 2 $\Rightarrow$ 1,5 Bonuspunkte
  - Note 3 $\Rightarrow$ 1 Bonuspunkt
  - Note 4 $\Rightarrow$ 0,5 Bonuspunkte
  - Note 5 $\Rightarrow$ 0 Bonuspunkte

Klausur

- Termin: 3. April 2012, 14 Uhr
- Weitere Informationen und Tipps folgen über das ganze Semester
- Wichtig: Aufgabentypen anschauen und verstehen, aber nicht ausschließlich auf Schemata lernen!

Wozu Zahlensysteme?

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

Zahlensysteme

- Wichtige Zahlensysteme:
  - Binäres Zahlensystem (2-adische Darstellung)
  - Oktales Zahlensystem (8-adische Darstellung)
  - Dezimales Zahlensystem (10-adische Darstellung)
  - Hexadezimales Zahlensystem (16-adische Darstellung)
- Zur Umrechnung gibt es verschiedene (generische und spezielle) Verfahren

Umrechnungsverfahren

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

Euklidischer Algorithmus

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

Horner-Schema

Getrennte Behandlung von

- Vorkomma-Teil
- Nachkomma-Teil

Horner-Schema (Vorkomma-Teil)

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

Horner-Schema (Nachkomma-Teil)

1. Setze $Zahl := Nachkomma - Teil$
2. Multipliziere Zahl mit Basis, schreibe Vorkomma-Teil des Ergebnisses als eine Nachkomma-Ziffer der Zahl im Zielsystem (beginnend bei der höchstwertigen)
3. Setze $Zahl := 0, Ergebnis\ der\ Multiplikation$ (verwende nur Nachkomma-Teil)
4. Gehe zu Schritt 2, falls $Zahl \neq 0$ und keine Periodizität festgestellt

Wie kann man vorzeichenbehaftete Zahlen darstellen?

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

Vorzeichen-Bit-Darstellung

- Höchstwertiges Bit (MSB) gibt das Vorzeichen der danach folgenden Zahl an (0 = positiv, 1 = negativ)
- Nachteil: Darstellung der Null nicht eindeutig (kann als +0 und -0 dargestellt werden)
- Beim Rechnen müssen die Vorzeichen separat betrachtet werden ($\rightarrow$ 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: Null kann auf zwei Arten dargestellt werden
- Vorteil: Addition ohne gesonderte Betrachtung des Vorzeichens

Zweierkomplement-Darstellung

Karlsruher Institut für Technologie

- 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
- Keine Redundanz mehr: Null ist positive Zahl
- Vorteil: Addition ohne gesonderte Betrachtung des Vorzeichens (wie bei Einerkomplement-Darstellung)
- Beispiel:
  - Gegeben: 8 Bit-Zahlen
  - $2^8 = 256$ Werte sind darstellbar
  - Zahlenbereich im ZK: -128 bis $+127$
  - Allgemein?

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
- Keine Redundanz mehr: Null ist positive Zahl
- Vorteil: Addition ohne gesonderte Betrachtung des Vorzeichens (wie bei Einerkomplement-Darstellung)
- Beispiel:
  - Gegeben: 8 Bit-Zahlen
  - $2^8 = 256$ Werte sind darstellbar
  - Zahlenbereich im ZK: -128 bis $+127$
  - Allgemein?

Warum funktionieren die Komplement-Darstellungen?

- Man rechnet auf dem Restklassenring von mod 2^n, der auch grafisch als Kreis dargestellt werden kann
- Addition bzw. Subtraktion entspricht einfach Bewegung auf diesem Kreis im bzw. gegen den Uhrzeigersinn
- Anschauliche Darstellung verschiedener Sachverhalte, z.B. “Größte Zahl + 1 = kleinste Zahl”
- Woran erkennt man bei einer komplement-dargestellten Zahl das Vorzeichen?

Exzesscode

- $0...0_2$ entspricht der kleinsten, $1...1_2$ der größten darstellbaren Zahl
- D.h. verschobener Nullpunkt der Zahlengerade (“Offset-Darstellung”)
- Wird z.B. für die Darstellung der Charakteristik bei IEEE 754-Gleitkommazahlen verwendet

BCD-Code: Einführung

- **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: $1001010100010111_{BCD} = 9517_{10}$

BCD-Code: Codetabelle

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?

BCD-Code: Codetabelle

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?

Gray-Code

- Code, bei dem sich benachbarte Codewörter nur in einer Ziffer unterscheiden
- Auch das niedrigste und höchste Codewort sind als benachbart zu betrachten
- Gray-Code für 3 Bit:

Wert	Codewort	Wert	Codewort
0	000	4	110
1	001	5	111
2	011	6	101
3	010	7	100

- Vor- und Nachteile?

Gray-Code

Karlsruher Institut für Technologie

- Code, bei dem sich benachbarte Codewörter nur in einer Ziffer unterscheiden
- Auch das niedrigste und höchste Codewort sind als benachbart zu betrachten
- Gray-Code für 3 Bit:

Wert	Codewort	Wert	Codewort
0	000	4	110
1	001	5	111
2	011	6	101
3	010	7	100

- Vor- und Nachteile?

Aiken-Code

- Ähnlich dem BCD-Code, aber mit den Wertigkeiten 2-4-2-1
- Pseudotetraden sind hier 5 bis A
  - Aiken(n) = not Aiken(9 - n)
  - Berechnung von 9 - n (Neunerkomplement) ist sehr einfach
  - Was nützt einem das?
- Aiken-Code für 4 Bit:

Wert	Codewort	Wert	Codewort
0	0000	5	1011
1	0001	6	1100
2	0010	7	1101
3	0011	8	1110
4	0100	9	1111

- Vor- und Nachteile?

Aiken-Code

- Ähnlich dem BCD-Code, aber mit den Wertigkeiten 2-4-2-1
- Pseudotetraden sind hier 5 bis A
  - Aiken(n) = not Aiken(9 - n)
  - Berechnung von 9 - n (Neunerkomplement) ist sehr einfach
  - Was nützt einem das?
- Aiken-Code für 4 Bit:

Wert	Codewort	Wert	Codewort
0	0000	5	1011
1	0001	6	1100
2	0010	7	1101
3	0011	8	1110
4	0100	9	1111

- Vor- und Nachteile?

Stibitz-Code

- Entspricht dem BCD-Code, wenn man zu jeder Ziffer 3 addiert
  - Pseudotetraden sind 0 bis 2 und D bis F
  - Bereich der gültigen Tetraden ist “zentriert”
  - Daher wieder einfache Berechnung von Neunerkomplement durch Negation (wie Aiken-Code)
- Codetabelle:

Wert	Codewort	Wert	Codewort
0	0011	5	1000
1	0100	6	1001
2	0101	7	1010
3	0110	8	1011
4	0111	9	1100

- Vor- und Nachteile?

Stibitz-Code

- Entspricht dem BCD-Code, wenn man zu jeder Ziffer 3 addiert
  - Pseudotetraden sind 0 bis 2 und D bis F
  - Bereich der gültigen Tetraden ist “zentriert”
  - Daher wieder einfache Berechnung von Neunerkomplement durch Negation (wie Aiken-Code)
- Codetabelle:

Wert	Codewort	Wert	Codewort
0	0011	5	1000
1	0100	6	1001
2	0101	7	1010
3	0110	8	1011
4	0111	9	1100

- Vor- und Nachteile?

Hammingcode

- Fehlerkorrigierender linearer Blockcode, d.h.
  - Abbildung von Datenwörtern fester Länge auf Codewörter (anderer) fester Länge
  - Summe zweier Codewörter ist wieder ein Codewort
- Alle folgenden Folien: Hammingcode-Konstruktion der Vorlesung (Hamming-Abstand 3)

Eigenschaften

- Hamming-Distanz ist drei
- Ein-Bit-Fehler können erkannt und korrigiert werden
- Zwei-Bit-Fehler können nur erkannt werden
- **Problem**: Erkennung “Wie viele Bitfehler liegen vor?”

Konstruktion

- Codewortbits werden beginnend mit Eins gezählt
- Die Prüfbits stehen an den Zweierpotenzen-Stellen
  - Die Zahl k der Prüfbits hängt von der Zahl m der Datenbits ab
  - Es muss gelten: $2^k \geq k + m + 1$
- Prüfbit prüft die Stellen mit der Nummer, die jeweilige Zweierpotenz enthalten
  - Beispiel: Prüfbit 2 prüft Codewortbits 3, 6, 7, 10, 11, ...
  - $\rightarrow$ Kein Prüfbit prüft ein anderes Prüfbit (Zweierpotenzen!)

Konstruktion

- Codewortbits werden beginnend mit Eins gezählt
- Die Prüfbits stehen an den Zweierpotenzen-Stellen
  - Die Zahl k der Prüfbits hängt von der Zahl m der Datenbits ab
  - Es muss gelten: $2^k \geq k + m + 1$
- Prüfbit prüft die Stellen mit der Nummer, die jeweilige Zweierpotenz enthalten
  - Beispiel: Prüfbit 2 prüft Codewortbits 3, 6, 7, 10, 11, ...
  - $\rightarrow$ Kein Prüfbit prüft ein anderes Prüfbit (Zweierpotenzen!)

Konstruktion

- Codewortbits werden beginnend mit Eins gezählt
- Die Prüfbits stehen an den Zweierpotenzen-Stellen
  - Die Zahl k der Prüfbits hängt von der Zahl m der Datenbits ab
  - Es muss gelten: $2^k \geq k + m + 1$
- Prüfbit prüft die Stellen mit der Nummer, die jeweilige Zweierpotenz enthalten
  - Beispiel: Prüfbit 2 prüft Codewortbits 3, 6, 7, 10, 11, ...
  - $\rightarrow$ Kein Prüfbit prüft ein anderes Prüfbit (Zweierpotenzen!)

Codierung und Decodierung

- Codierung:
  - Man berechnet die Prüfbits durch Antivalenz (XOR) zwischen entsprechenden Datenbits
- Decodierung:
  - Man berechnet für jedes Prüfbit die Antivalenz aller anderer Datenbits zusammen mit dem entsprechenden Prüfbits
  - Alle Werte sind 0 $\rightarrow$ Kein Ein-/Zwei-Bit-Fehler
  - Mindestens ein Wert $\neq 0 \rightarrow$ Falls Ein-Bit-Fehler vorliegt, liefern die Werte als Binärzahl die Stelle des Fehlers
- Trick für Antivalenz: Zahl der Einsen zählen (gerade Zahl $\rightarrow$ 0)

Codierung und Decodierung

- Codierung:
  - Man berechnet die Prüfbits durch Antivalenz (XOR) zwischen entsprechenden Datenbits
- Decodierung:
  - Man berechnet für jedes Prüfbit die Antivalenz aller anderer Datenbits zusammen mit dem entsprechenden Prüfbits
  - Alle Werte sind 0 $\rightarrow$ Kein Ein-/Zwei-Bit-Fehler
  - Mindestens ein Wert $\neq 0 \rightarrow$ Falls Ein-Bit-Fehler vorliegt, liefern die Werte als Binärzahl die Stelle des Fehlers
- Trick für Antivalenz: Zahl der Einsen zählen (gerade Zahl $\rightarrow$ 0)

Codierung und Decodierung

- Codierung:
  - Man berechnet die Prüfbits durch Antivalenz (XOR) zwischen entsprechenden Datenbits
- Decodierung:
  - Man berechnet für jedes Prüfbit die Antivalenz aller anderer Datenbits zusammen mit dem entsprechenden Prüfbits
  - Alle Werte sind 0 $\rightarrow$ Kein Ein-/Zwei-Bit-Fehler
  - Mindestens ein Wert $\neq 0 \rightarrow$ Falls Ein-Bit-Fehler vorliegt, liefern die Werte als Binärzahl die Stelle des Fehlers
- Trick für Antivalenz: Zahl der Einsen zählen (gerade Zahl $\rightarrow$ 0)

Bisher: Festkomma-Darstellung

- 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
...	...

Probleme der Festkomma-Darstellung

- 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
- $\rightarrow$ Gleitkommazahlen

Gleitkomma-Darstellung

- 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

Gleitkomma-Darstellung

- 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

Gleitkomma-Darstellung

- 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

Normalisierte Gleitkommazahlen

- 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
- $\rightarrow$ 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)

Normalisierte Gleitkommazahlen

- 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
- $\rightarrow$ 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)

Aufbau von Gleitkommazahlen

- 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”)

Spezielle Gleitkommazahlen

- 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 $1 + x \neq 1$

IEEE 754: Formate

- 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

IEEE 754: Formate

- 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

IEEE 754: Charakteristik

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

IEEE 754: Charakteristik

- $1 \leq \textit{Charakteristik} \leq 254$:
  - $\textit{Exponent} = \textit{Charakteristik} - 127$
  - 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”)

IEEE 754: Charakteristik

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

Boolesche Algebra

Menge V mit zwei zweistelligen **abgeschlossenen** Operationen, für die die vier Huntingtonschen Axiome gelten:

1. Kommutativgesetze:
   - $a \otimes b = b \otimes a$
   - $a \oplus b = b \oplus a$
2. Distributivgesetze:
   - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
   - $a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c)$
3. Neutrale Elemente: $\exists e, n \in V$
   - $a \otimes e = a$
   - $a \oplus n = a$
4. Inverse Element: $\forall a \in V : \exists \overline{a} \in V$
   - $a \otimes \overline{a} = n$
   - $a \oplus \overline{a} = e$

Boolesche Algebra

Karlsruher Institut für Technologie

Menge V mit zwei zweistelligen **abgeschlossenen** Operationen, für die die vier Huntingtonschen Axiome gelten:

1. Kommutativgesetze:
   - $a \otimes b = b \otimes a$
   - $a \oplus b = b \oplus a$
2. Distributivgesetze:
   - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
   - $a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c)$
3. Neutrale Elemente: $\exists e, n \in V$
   - $a \otimes e = a$
   - $a \oplus n = a$
4. Inverse Element: $\forall a \in V : \exists \overline{a} \in V$
   - $a \otimes \overline{a} = n$
   - $a \oplus \overline{a} = e$

Boolesche Algebra

Menge V mit zwei zweistelligen **abgeschlossenen** Operationen, für die die vier Huntingtonschen Axiome gelten:

1. Kommutativgesetze:
   - $a \otimes b = b \otimes a$
   - $a \oplus b = b \oplus a$
2. Distributivgesetze:
   - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
   - $a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c)$
3. Neutrale Elemente: $\exists e, n \in V$
   - $a \otimes e = a$
   - $a \oplus n = a$
4. Inverse Element: $\forall a \in V : \exists \overline{a} \in V$
   - $a \otimes \overline{a} = n$
   - $a \oplus \overline{a} = e$

Boolesche Algebra

Menge V mit zwei zweistelligen **abgeschlossenen** Operationen, für die die vier Huntingtonschen Axiome gelten:

1. Kommutativgesetze:
   - $a \otimes b = b \otimes a$
   - $a \oplus b = b \oplus a$
2. Distributivgesetze:
   - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
   - $a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c)$
3. Neutrale Elemente: $\exists e, n \in V$
   - $a \otimes e = a$
   - $a \oplus n = a$
4. Inverse Element: $\forall a \in V : \exists \overline{a} \in V$
   - $a \otimes \overline{a} = n$
   - $a \oplus \overline{a} = e$

Boolesche Algebra

Menge V mit zwei zweistelligen **abgeschlossenen** Operationen, für die die vier Huntingtonschen Axiome gelten:

1. Kommutativgesetze:
   - $a \otimes b = b \otimes a$
   - $a \oplus b = b \oplus a$
2. Distributivgesetze:
   - $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$
   - $a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c)$
3. Neutrale Elemente: $\exists e, n \in V$
   - $a \otimes e = a$
   - $a \oplus n = a$
4. Inverse Element: $\forall a \in V : \exists \overline{a} \in V$
   - $a \otimes \overline{a} = n$
   - $a \oplus \overline{a} = e$

Schaltalgebra

- Bestimmte boolesche Algebra mit $V = \{0, 1\}$ und den Operationen "Disjunktion" ($\vee$) und "Konjunktion" ($\wedge$, auch $\cdot$)
- Wahrheitswert eines Ausdrucks ergibt sich durch Angabe einer Variablenbelegung
- Tautologie: Ausdruck, der für alle möglichen Belegungen wahr ist

Gesetze der Schaltalgebra

Karlsruher Institut für Technologie

- Assoziativgesetze:
  - $a \wedge (b \wedge c) = (a \wedge b) \wedge c$
  - $a \vee (b \vee c) = (a \vee b) \vee c$
- Idempotenzgesetze:
  - $a \wedge a = a$
  - $a \vee a = a$
- Absorptionsgesetze:
  - $a \wedge (a \vee b) = a$
  - $a \vee (a \wedge b) = a$
- De Morgansche Gesetze:
  - $\overline{a \wedge b} = \overline{a} \vee \overline{b}$
  - $\overline{a \vee b} = \overline{a} \wedge \overline{b}$

Gesetze der Schaltalgebra

- Assoziativgesetze:
  - $a \wedge (b \wedge c) = (a \wedge b) \wedge c$
  - $a \vee (b \vee c) = (a \vee b) \vee c$
- Idempotenzgesetze:
  - $a \wedge a = a$
  - $a \vee a = a$
- Absorptionsgesetze:
  - $a \wedge (a \vee b) = a$
  - $a \vee (a \wedge b) = a$
- De Morgansche Gesetze:
  - $\overline{a \wedge b} = \overline{a} \vee \overline{b}$
  - $\overline{a \vee b} = \overline{a} \wedge \overline{b}$

Gesetze der Schaltalgebra

- Assoziativgesetze:
  - $a \wedge (b \wedge c) = (a \wedge b) \wedge c$
  - $a \vee (b \vee c) = (a \vee b) \vee c$
- Idempotenzgesetze:
  - $a \wedge a = a$
  - $a \vee a = a$
- Absorptionsgesetze:
  - $a \wedge (a \vee b) = a$
  - $a \vee (a \wedge b) = a$
- De Morgansche Gesetze:
  - $\overline{a \wedge b} = \overline{a} \vee \overline{b}$
  - $\overline{a \vee b} = \overline{a} \wedge \overline{b}$

Gesetze der Schaltalgebra

- Assoziativgesetze:
  - $a \wedge (b \wedge c) = (a \wedge b) \wedge c$
  - $a \vee (b \vee c) = (a \vee b) \vee c$
- Idempotenzgesetze:
  - $a \wedge a = a$
  - $a \vee a = a$
- Absorptionsgesetze:
  - $a \wedge (a \vee b) = a$
  - $a \vee (a \wedge b) = a$
- De Morgansche Gesetze:
  - $\overline{a \wedge b} = \overline{a} \vee \overline{b}$
  - $\overline{a \vee b} = \overline{a} \wedge \overline{b}$

Darstellung als Funktionstabelle

- Idee: Eine boolesche Funktion ist durch ihren Funktionswert unter allen möglichen Variablenbelegungen charakterisiert
- Angabe als Tabelle mit 2^n Zeilen bei n verschiedenen Variablen
- Wichtige Funktionen mit zwei Parametern:
  - Disjunktion
  - Konjunktion
  - Implikation
  - Äquivalenz
  - Antivalenz (XOR)
- Funktionstabellen für diese Funktionen?

Darstellung als Funktionstabelle

- Idee: Eine boolesche Funktion ist durch ihren Funktionswert unter allen möglichen Variablenbelegungen charakterisiert
- Angabe als Tabelle mit 2^n Zeilen bei n verschiedenen Variablen
- Wichtige Funktionen mit zwei Parametern:
  - Disjunktion
  - Konjunktion
  - Implikation
  - Äquivalenz
  - Antivalenz (XOR)
- Funktionstabellen für diese Funktionen?

Darstellung in Operatorensystemen

- Darstellung als algebraischer Ausdruck
- Immer möglich, wenn das Operatorensystem vollständig ist ($\rightarrow$ jede boolesche Funktion ist darstellbar)
- Beispiele für vollständige Operatorensysteme:
  - Disjunktion und Negation
  - Konjunktion und Negation
  - NAND
  - NOR
- Beweis der Vollständigkeit: Alle Operatoren eines vollständigen Operatorensystems im Operatorensystem darstellen

Darstellung in Operatorensystemen

- Darstellung als algebraischer Ausdruck
- Immer möglich, wenn das Operatorensystem vollständig ist ($\rightarrow$ jede boolesche Funktion ist darstellbar)
- Beispiele für vollständige Operatorensysteme:
  - Disjunktion und Negation
  - Konjunktion und Negation
  - NAND
  - NOR
- Beweis der Vollständigkeit: Alle Operatoren eines vollständigen Operatorensystems im Operatorensystem darstellen

Darstellung in Operatorensystemen

- Darstellung als algebraischer Ausdruck
- Immer möglich, wenn das Operatorensystem vollständig ist ($\rightarrow$ jede boolesche Funktion ist darstellbar)
- Beispiele für vollständige Operatorensysteme:
  - Disjunktion und Negation
  - Konjunktion und Negation
  - NAND
  - NOR
- Beweis der Vollständigkeit: Alle Operatoren eines vollständigen Operatorensystems im Operatorensystem darstellen

Übungsaufgabe 1

Gegeben sei das folgende Maschinenformat für die Darstellung von Gleitkommazahlen:
Bit 31: Vorzeichen, Bits 30 - 24: Charakteristik, Bits 23 - 0: Mantisse

- Vorzeichen: VZ = 0 $\rightarrow$ positive Zahl, VZ = 1 $\rightarrow$ negative Zahl
- $\mathit{Charakteristik} = \mathit{Exponent} + 40_{16}$
- Basis 16

Die Mantisse liegt im Zahlenbereich $16^{-1} \leq \mathit{Mantisse} \leq (1 - 16^{-6})$.

1. Geben Sie in obigem Format die größte und die kleinste negative Zahl in normalisierter und nicht-normalisierter Maschinendarstellung an.
2. Was sind die Vor- und Nachteile, wenn man statt der Basis 16 die Basis 2 verwendet?
3. Was ändert sich, wenn man (im Fall der Basis 2) ein Bit der Mantisse aufgibt zugunsten eines Bits für die Charakterisik?

Übungsaufgabe 1

Gegeben sei das folgende Maschinenformat für die Darstellung von Gleitkommazahlen:
Bit 31: Vorzeichen, Bits 30 - 24: Charakteristik, Bits 23 - 0: Mantisse

- Vorzeichen: VZ = 0 $\rightarrow$ positive Zahl, VZ = 1 $\rightarrow$ negative Zahl
- $\mathit{Charakteristik} = \mathit{Exponent} + 40_{16}$
- Basis 16

Die Mantisse liegt im Zahlenbereich $16^{-1} \leq \mathit{Mantisse} \leq (1 - 16^{-6})$.

1. Geben Sie in obigem Format die größte und die kleinste negative Zahl in normalisierter und nicht-normalisierter Maschinendarstellung an.
2. Was sind die Vor- und Nachteile, wenn man statt der Basis 16 die Basis 2 verwendet?
3. Was ändert sich, wenn man (im Fall der Basis 2) ein Bit der Mantisse aufgibt zugunsten eines Bits für die Charakterisik?

Übungsaufgabe 1

Gegeben sei das folgende Maschinenformat für die Darstellung von Gleitkommazahlen:

Bit 31: Vorzeichen, Bits 30 - 24: Charakteristik, Bits 23 - 0: Mantisse

- Vorzeichen: VZ = 0 $\rightarrow$ positive Zahl, VZ = 1 $\rightarrow$ negative Zahl
- $\textit{Charakteristik} = \textit{Exponent} + 40_{16}$
- Basis 16

Die Mantisse liegt im Zahlenbereich $16^{-1} \leq \textit{Mantisse} \leq (1 - 16^{-6})$.

1. Geben Sie in obigem Format die größte und die kleinste negative Zahl in normalisierter und nicht-normalisierter Maschinendarstellung an.
2. Was sind die Vor- und Nachteile, wenn man statt der Basis 16 die Basis 2 verwendet?
3. Was ändert sich, wenn man (im Fall der Basis 2) ein Bit der Mantisse aufgibt zugunsten eines Bits für die Charakterisik?

Übungsaufgabe 1

Gegeben sei das folgende Maschinenformat für die Darstellung von Gleitkommazahlen:

Bit 31: Vorzeichen, Bits 30 - 24: Charakteristik, Bits 23 - 0: Mantisse

- Vorzeichen: VZ = 0 $\rightarrow$ positive Zahl, VZ = 1 $\rightarrow$ negative Zahl
- $\mathit{Charakteristik} = \mathit{Exponent} + 40_{16}$
- Basis 16

Die Mantisse liegt im Zahlenbereich $16^{-1} \leq \mathit{Mantisse} \leq (1 - 16^{-6})$.

1. Geben Sie in obigem Format die größte und die kleinste negative Zahl in normalisierter und nicht-normalisierter Maschinendarstellung an.
2. Was sind die Vor- und Nachteile, wenn man statt der Basis 16 die Basis 2 verwendet?
3. Was ändert sich, wenn man (im Fall der Basis 2) ein Bit der Mantisse aufgibt zugunsten eines Bits für die Charakterisik?

Übungsaufgabe 2

Wandeln Sie die Dezimalzahl 21 in das 32-Bit-Format des IEEE-754-Standard um. Stellen Sie die Gleitkommazahl als hexadezimale Zahl dar.

Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
   1. 0
   2. 1
   3. a
   4. $\overline{a}$
   5. $\overline{0}$
   6. $\overline{b} \wedge \vee c$
   7. $a \vee b \wedge c$
   8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
   1. 0
   2. 1
   3. a
   4. $\overline{a}$
   5. $\overline{0}$
   6. $\overline{b} \wedge \vee c$
   7. $a \vee b \wedge c$
   8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
    1. 0
    2. 1
    3. a
    4. $\overline{a}$
    5. $\overline{0}$
    6. $\overline{b} \wedge \vee c$
    7. $a \vee b \wedge c$
    8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
   1. 0
   2. 1
   3. a
   4. $\overline{a}$
   5. $\overline{0}$
   6. $\overline{b} \wedge \vee c$
   7. $a \vee b \wedge c$
   8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
    1. 0
    2. 1
    3. a
    4. $\overline{a}$
    5. $\overline{0}$
    6. $\overline{b} \wedge \vee c$
    7. $a \vee b \wedge c$
    8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
   1. 0
   2. 1
   3. a
   4. $\overline{a}$
   5. $\overline{0}$
   6. $\overline{b} \wedge \vee c$
   7. $a \vee b \wedge c$
   8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
    1. 0
    2. 1
    3. a
    4. $\overline{a}$
    5. $\overline{0}$
    6. $\overline{b} \wedge \vee c$
    7. $a \vee b \wedge c$
    8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
   1. 0
   2. 1
   3. a
   4. $\overline{a}$
   5. $\overline{0}$
   6. $\overline{b} \wedge \vee c$
   7. $a \vee b \wedge c$
   8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

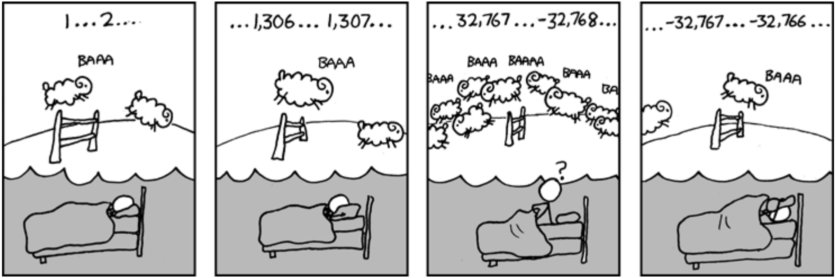
Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
    1. 0
    2. 1
    3. a
    4. $\overline{a}$
    5. $\overline{0}$
    6. $\overline{b} \wedge \vee c$
    7. $a \vee b \wedge c$
    8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

Übungsaufgabe 3

1. Was kennzeichnet eine Boolesche Algebra?
2. Welche der folgenden Booleschen Ausdrücke sind korrekt?
   1. 0
   2. 1
   3. a
   4. $\overline{a}$
   5. $\overline{0}$
   6. $\overline{b} \wedge \vee c$
   7. $a \vee b \wedge c$
   8. $(a \vee b) \wedge c$
3. Geben Sie für die letzten beiden Ausdrücke aus der vorherigen Teilaufgabe die Funktionstabelle an.

Fertig!



Quelle: http://xkcd.com/571/