Digitaltechnik & Rechnersysteme

Zahlenkodierung

Martin Kumm

Hochschule Fulda
University of Applied Sciences

AI | Angewandte Informatik

WiSe 2022/2023

Was bisher geschah...

Angewandte Informatik

- Die Macht der Abstraktion
- Was ist Information?
- Codierung mit ...
  - fester Länge
  - variabler Länge, optimale Huffmann-Codierung

Inhalte

Angewandte Informatik

Stellenwertsysteme

Angewandte Informatik

Bei Stellenwertsystemen (sog. polyadischen Zahlensystemen) wird jedem Symbol eine Wertigkeit in Abhängigkeit der Stelle zugewiesen.

Wertigkeit der i-ten Ziffer x_i entspricht R^i

R wird als Basis oder auch Radix genannt.

In einem kanonischen Zahlensystem besteht der Ziffernvorrat aus den Ziffern $\{0, 1, 2, \ldots, R-1\}$

Der Wert einer n-stelligen Zahl lautet

$$X = x_0 R^0 + x_1 R^1 + x_2 R^2 + \ldots = \sum_{i=0}^{n-1} x_i R^i$$

Der Wertebereich einer n-stelligen Zahl lautet $0 \ldots R^n - 1$

Stellenwertsysteme

Angewandte Informatik

Beispiel: Dezimalsystem

- Das Dezimalsystem hat Radix $R = 10$ und die Ziffern $\{0, 1, 2, \ldots, 9\}$
- Die Dezimalzahl 1234_{10} ist eine verkürzte Schreibweise für

$$1234 = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0$$

Beispiel: Binärsystem (Dualsystem)

- Das Binärsystem hat Radix $R = 2$ und die Ziffern $\{0, 1\}$
- Die Binärzahl 0101_2 hat den Wert

$$\begin{aligned} 0101_2 &= 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 1 \cdot 2^2 + 1 \cdot 2^0 \\ &= 4 + 1 = 5 \end{aligned}$$

Konvertierung I

Angewandte Informatik

Algorithmus: Konvertierung in anderes Stellenwertsystem

1. Zahl durch R ganzzahlig mit Rest teilen
2. Der Rest entspricht der gesuchten Ziffer, beginnend mit der niedrigsten Stelle
3. Solange Divisionsergebnis ungleich Null ist, mit dem Divisionsergebnis die Schritte 1-2 wiederholen

Konvertierung II

Beispiel: Konvertierung von 100_{10} ins Binärsystem

$$
\begin{aligned}
100 : 2 &= 50 \text{ Rest } 0 \\
50 : 2 &= 25 \text{ Rest } 0 \\
25 : 2 &= 12 \text{ Rest } 1 \\
12 : 2 &= 6 \text{ Rest } 0 \\
6 : 2 &= 3 \text{ Rest } 0 \\
3 : 2 &= 1 \text{ Rest } 1 \\
1 : 2 &= 0 \text{ Rest } 1 \uparrow \text{ In dieser Richtung Ablesen}
\end{aligned}
$$

$\Rightarrow 100_{10} = 1100100_2$

Vorlesungsaufgabe: Konvertieren Sie die Zahl 42_{10} ins Binärsystem!

Konvertierung III

Angewandte Informatik

Beispiel: Konvertierung von 42_{10} ins Binärsystem

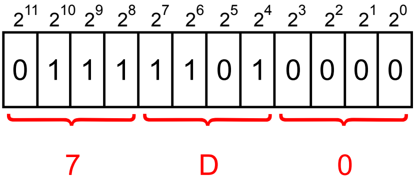
Hexadezimale Zahlen

Angewandte Informatik

- Hexadezimale Zahlen ($R = 16$) werden häufig zur kompakten Darstellung von Binärzahlen verwendet.
- Es werden die Ziffern $\{0, 1, 2, \ldots, 9, A, B, C, D, E, F\}$ verwendet (A bis F repräsentieren die Wertigkeit 10 bis 15)
- Jedes Hexadezimale Digit kann mit genau 4 Bit dargestellt werden:

Hexadezimal - Basis 16

0000 – 0	1000 – 8
0001 – 1	1001 – 9
0010 – 2	1010 – A
0011 – 3	1011 – B
0100 – 4	1100 – C
0101 – 5	1101 – D
0110 – 6	1110 – E
0111 – 7	1111 – F



0b011111010000 = 0x7D0

Konvertierung IV

Angewandte Informatik

Beispiel: Konvertierung von 100_{10} ins Hexadezimalsystem

$$100 : 16 = 6 \text{ Rest } 4$$
$$6 : 16 = 0 \text{ Rest } 6 \uparrow \text{ In dieser Richtung Ablesen}$$

$\Rightarrow 100_{10} = 64_{16}$

Vorlesungsaufgabe: Konvertieren Sie die Zahl 42_{10} ins Hexadezimalsystem!

Lösung: Konvertierung von 42_{10} ins Hexadezimalsystem

Darstellung vorzeichenbehafteter Zahlen

Bisher unterstützt das Stellenwertsystem erst mal nur positive Zahlen

Negative bzw. allgemein vorzeichenbehaftete Zahlen können durch folgende Codierungen dargestellt werden:

- Vorzeichen-Betrag-Darstellung
- Einerkomplement-Darstellung
- Zweierkomplement-Darstellung

Vorzeichen-Betrag Darstellung ($R = 2$)

Angewandte Informatik

Das Bit mit höchster Wertigkeit (*most significant bit*, MSB) (x_{n-1}) wird für das Vorzeichen verwendet:

$$X = \begin{cases} \sum_{i=0}^{n-2} x_i 2^i & \text{for } x_{n-1} = 0 \\ -\sum_{i=0}^{n-2} x_i 2^i & \text{for } x_{n-1} = 1 \end{cases}$$

Der Wertebereich ist $-(2^{n-1} - 1) \leq X \leq 2^{n-1} - 1$

Problem 1: Die Zahl 0 ist redundant (doppelt kodiert): $0 = -0$

Problem 2: Das Addieren/Subtrahieren erfordert Fallunterscheidung je nach Vorzeichen!

Vorzeichen-Betrag Darstellung

Angewandte Informatik

Beispiel: Darstellung der Zahl -6 mit $n = 4$ Bit

$6_{10} = 110_2$

Erweiterung auf 4 Bit:

$6_{10} = 0110_2$

Vorzeichen-Betrag (Setzen des Vorzeichenbits, da negativ):

$-6_{10} = 1110_{\text{VB}}$

Komplement-Darstellung

Die vorzeichenbehaftete ganze Zahl X wird auf eine positive Zahl X_R abgebildet, welche als Binärzahl dargestellt wird

Die Abbildung erfolgt über

$$X_R = \begin{cases} X & \text{für } X \geq 0 \\ C - |X| & \text{für } X < 0 \end{cases}$$

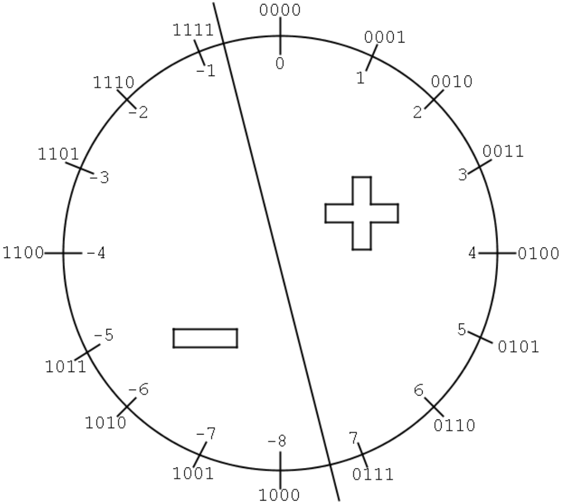
Die Umkehrfunktion lautet

$$X = \begin{cases} X_R & \text{für } X_R < C/2 \\ X_R - C & \text{für } X_R \geq C/2 \end{cases}$$

Für C gilt:

- $C = 2^n - 1$ definiert die Einerkomplement-Darstellung
- $C = 2^n$ definiert die Zweierkomplement-Darstellung

Zweierkomplement Zahlenkreis ($n = 4$)



Vorzeichenbit Einer-/Zweierkomplement

Vorzeichenerkennung:

- X ist negativ wenn $X_R \geq 2^{n-1}$
- Dies entspricht exakt der Wertigkeit des MSB x_{n-1}

$\Rightarrow$ Das MSB X_{n-1} definiert auch hier das Vorzeichen!

Umrechnung Einerkomplement

Für eine n Bit Zahl ist die Operation $2^n - 1 - X$ identisch mit dem bitweisen Komplement von X:

$$2^n - 1 - X = \overline{X}$$

Die Abbildung für das Einerkomplement erfolgt über

$$X_R = \begin{cases} X & \text{für } X \geq 0 \\ 2^n - 1 - |X| = \overline{|X|} & \text{für } X < 0 \end{cases}$$

Die Umkehrfunktion lautet

$$X = \begin{cases} X_R & \text{für } X_R < C/2 \\ X_R - 2^n + 1 = -(2^n - 1 - X_R) = -\overline{X_R} & \text{für } X_R \geq C/2 \end{cases}$$

$\Rightarrow$ Die Umrechnung in das Einerkomplement und wieder zurück erfolgt durch bitweise Invertierung

Umrechnung Zweierkomplement

Angewandte Informatik

Für eine n Bit Zahl ist die Operation $2^n - 1 - X$ identisch mit dem bitweisen Komplement von X:

$$2^n - 1 - X = \overline{X}$$

Die Abbildung für das Zweierkomplement erfolgt über

$$X_R = \begin{cases} X & \text{für } X \geq 0 \\ 2^n - |X| = \overline{|X|} + 1 & \text{für } X < 0 \end{cases}$$

Die Umkehrfunktion lautet

$$X = \begin{cases} X_R & \text{für } X_R < C/2 \\ X_R - 2^n = -(2^n - X_R) = -(\overline{X_R} + 1) & \text{für } X_R \geq C/2 \end{cases}$$

$\Rightarrow$ Die Umrechnung in das Zweierkomplement und wieder zurück erfolgt durch bitweise Invertierung und anschließender Addition $+1$

Einer- vs. Zweierkomplement

Angewandte Informatik

Eigenschaften Einerkomplement

- Der Wertebereich ist $-(2^{n-1}-1) \leq X \leq 2^{n-1}-1$
- Nachteil 1: Die Zahl 0 nach wie vor ist redundant: $0 = -0$
- Nachteil 2: Das Addieren/Subtrahieren erfordert eine anschließende Addition mit Vorzeichen!

Eigenschaften Zweierkomplement

- Der Wertebereich ist $-2^{n-1} \leq X \leq 2^{n-1}-1$
- Die Zahl 0 ist eindeutig: $000\ldots000_2$
- Das Addieren/Subtrahieren erfordert keine weitere Behandlung!
- Daher rechnen heute fast alle Rechner im Zweierkomplement!

Konvertierung Komplementdarstellung I

Angewandte Informatik

Beispiel: Darstellung von –6 im 4 Bit Einerkomplement

$6_{10} = 0110_2$ (erweitert auf 4 Bit)

Einerkomplement-Darstellung (Invertierung aller Bits, da negativ):
$-6_{10} = 1001_{1K}$

Beispiel: Darstellung von –6 im 4 Bit Zweierkomplement

Invertierung aller Bits $+ 1$ (da negativ):

$$\begin{array}{rr} & 1001 \\ + & 1 \\ \hline & 1010 \end{array} \qquad \Rightarrow -6_{10} = 1010_{2K}$$

Achtung: Positive Zahlen bleiben wie sie sind:
$6_{10} = 110_2 = 0110_{VB} = 0110_{1K} = 0110_{2K}$

Konvertierung Komplementdarstellung II

Angewandte Informatik

Beispiel: Rückkonvertierung Einerkomplement

$X = 1001_{1K} \Rightarrow$ Da MSB=1, alle Bits invertieren:

$|X| = 0110_2 = 6_{10}, \Rightarrow X = -6_{10}$

Beispiel: Rückkonvertierung Zweierkomplement

$X = 1010_{2K} \Rightarrow$ Da MSB=1, alle Bits invertieren und + 1:

$$\begin{array}{rr} & 0101 \\ + & 1 \\ \hline & 0110 \end{array}$$

$|X| = 0110_2 = 6_{10}, \Rightarrow X = -6_{10}$

Vorlesungsaufgabe

Kodieren Sie -42 mit 8 Bit im Zweierkomplement.

Hinweis: $42_{10} = 101010_2$

Lösung

Festkommazahlen

Angewandte Informatik

Bei Festkommazahlen wird die ganze Zahl um eine feste Anzahl Nachkommastellen erweitert:

$$X = (x_{n-1} x_{n-2} \dots x_1 x_0 \, , \, x_{-1} x_{-2} \dots x_{-f})$$

Der Wert der Zahl ist dann gegeben durch

$$X = \sum_{i=-f}^{n-1} x_i 2^i$$

Beispiel: $101,1101_2 = 2^2 + 2^0 + 2^{-1} + 2^{-2} + 2^{-4} = 5,8125_{10}$

Konvertierung in Festkommazahlen

Angewandte Informatik

Konvertierung einer (Dezimal-)Kommazahl in eine Festkommazahl:

- Multipliziere Zahl mit 2^f
- Runde das Ergebnis zur nächsten ganzen Zahl
- Konvertiere ganze Zahl ins Binärsystem
- Verschiebe die Zahl um f Bit nach rechts

Beispiel: Darstellung von $\pi = 3.14159265358979...$ mit 8 Nachkommastellen:

- $\pi \cdot 2^8 = 804.2477...$
- $\text{round}(804,2477) = 804_{10} = 1100100100_2$
- $\pi_{10} \approx 11,00100100_2 = 3,140625_{10}$

Gleitkommazahlen Idee

Der Dynamikbereich von Festkommazahlen ist stark eingeschränkt.

Sehr große oder sehr kleine Zahlen benötigen viele Bits, die Genauigkeit ist dabei unnötig hoch.

Lösung: Anpassung der Position des Kommas durch Gleitkomma- bzw. Fließkommazahlen

Das tun wir bereits im Umgang mit großen oder kleinen Zahlen.

Gleitkommazahlen Beispiel

Beispiel: Das Elektronenvolt (eV) hat eine Energie von $1,602176634 \cdot 10^{-19}$ Joule
$\Rightarrow$ Statt $0,0000000000000000001602176634$ Joule schreiben wir kompakt $1,602176634 \cdot 10^{-19}$ Joule

Genauso können wir statt der Festkommadarstellung

$$0,000000000000000000000000000000000000000000000000000000000000001011110100100110_2$$

die Gleitkommadarstellung $\underbrace{1,011110100100110_2}_{\text{Mantisse}} \cdot 2^{\overbrace{}^{}\underbrace{-63}_{\text{Exponent}}}$ verwenden.

Zur Kodierung dieser Zahl muss...

- dank Normalisierung der sog. Mantisse zu $1,011\ldots$ die führende 1 nicht gespeichert werden
- der Exponent (-63) extra gespeichert werden

Gleitkommazahl nach IEEE 754

Eine Gleitkommazahl nach IEEE 754 Standard ist folgendermaßen definiert:

s	Exponent e	Mantisse m
	B_e Bit	B_m Bit

- Das Sign-Bit s bestimmt das Vorzeichen ($0 \hat{=} +$, $1 \hat{=} -$)
- Der Exponent e definiert die Position des Kommas
- Mantisse m definiert normalisierte Zahl (ohne führende Eins)

Die Zahl berechnet sich hieraus zu:

$$X = (-1)^s \left(1 + \sum_{i=0}^{B_\mathrm{m}-1} m_i 2^{-i-1}\right) 2^{e-\mathrm{bias}} \quad \text{mit} \quad \mathrm{bias} = 2^{B_\mathrm{e}-1} - 1$$

Gleitkomma-Konvertierung

Beispiel: Reele Zahl → Gleitkommadarstellung

$$X = (-1)^s \; 1,m \; 2^{e-\text{bias}} \quad \text{mit} \quad \text{bias} = 2^{B_\text{e}-1} - 1$$

Beispiel: Darstellung der Zahl $x = 9,25_{10}$ mit einer Exponentenwortbreite von $B_\text{e} = 6$ Bit und einer Mantissenwortbreite von $B_\text{m} = 5$ Bit – (1,6,5)-Format:

1. Ermittlung der Festkommadarstellung: $x = 9,25_{10} = 1001,01_2$
2. Normalisierung: $1001,01_2 = 1,\underbrace{00101_2}_{=m} \cdot 2^3$
3. Ermittlung des Exponenten: $e - \text{bias} = 3 \rightarrow$
   $e = 3 + \text{bias} = 3 + 2^5 - 1 = 34 = 100010_2$

$$\Rightarrow \text{Codewort: } \underbrace{0}_{s}\,\underbrace{100010}_{e}\,\underbrace{00101}_{m}$$

Gleitkomma-Konvertierung

Beispiel: Fließkommazahl → reele Darstellung:

$$A = (-1)^s \; 1,m \; 2^{e-\text{bias}} \quad \text{mit} \quad \text{bias} = 2^{B_\text{e}-1} - 1$$

1. Bestimmung der Komponenten: $\underbrace{0}_{s}\,\underbrace{100010}_{e}\,\underbrace{00101}_{m}$
2. Erweiterung der Mantisse: $1,m = 1,00101_2$
3. Ermittlung Exponent $e = 100010_2 = 34_{10}$
4. Verschiebung und Konvertierung:
   $x = +1,00101_2 \cdot 2^{34-31} = +1001,01_2 = 9,25_{10}$

IEEE Standard

Der IEEE 754r Standard definiert folgende Gleitkommaformate:

	Half	Single	Double
Wortbreite	16 Bit	32 Bit	64 Bit
Mantisse (B_m)	10 Bit	23 Bit	52 Bit
Exponent (B_e)	5 Bit	8 Bit	11 Bit
Bias	15	127	1023
Wertebereich	$\pm 2^{16} \approx 6 \cdot 10^4$	$\pm 2^{128} \approx 10^{38}$	$\pm 2^{1024} \approx 10^{308}$

Einheitliches Format wichtig zum Austausch von Daten!

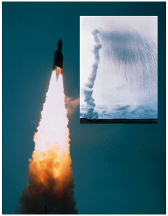
Zusätzlich spezielle Werte für

- 0 (lässt sich durch Normalisierung sonst nicht darstellen!)
- $\pm\infty$ (für Überläufe)
- NaN: Not-a-Number (keine Zahl), entsteht z.B. bei $\frac{0}{0}$, $0 \times \infty$

Wenn man Wertebereiche missachtet... I

Die Explosion der Ariane 5 (V88) 1996

- Die unbemannte Ariane 5 Rakete explodierte 40 Sekunden nach lift-off (Kosten: $500 Millionen)
- Der Fehler entstand aufgrund einer Konvertierung (cast) einer 64 Bit Gleitkommazahl in eine 16 Bit (vorzeichenbehaftete) ganze Zahl.
- Zahl repräsentierte die horizontale Ausrichtung
- Abzüglich Vorzeichenbit hat diese den max. Wert $2^{15} - 1 = 32.767$.
- Leider war die horizontale Ausrichtung größer...



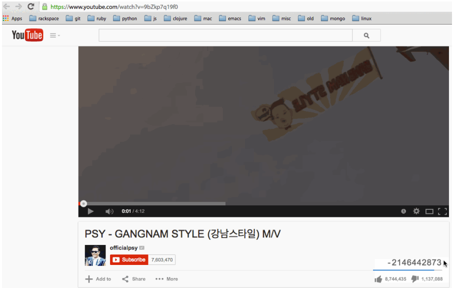
source: `http://www.math.umn.edu/~arnold/disasters/`

Wenn man Wertebereiche missachtet... II

Angewandte Informatik

Gangnam Style Video

- Als die Anzahl der Youtube views des »Gangnam Style« Video die 2.147.483.647 überstieg wurde diese negativ
- Warum? 2.147.483.647 = $2^{31} - 1$ is die größte positive Zahl die sich mit 32 Bit vorzeichenbehaftet darstellen lässt.
- Das nächste Codewort repräsentiert die Zahl $-2^{31} = -2.147.483.648$



source: `https://www.wired.com/2014/12/gangnam-style-youtube-math`