Einschub:
Binärcodierung elementarer Datentypen

Teile aus Kapitel 2 in Küchlin/Weber: Einführung in die Informatik

Binärcodierung elementarer Datentypen

- Unterscheide
  - Zahl-**Wert**
  - Zahl-**Bezeichner**
- Zu ein- und demselben Zahl-Wert kann es verschiedene Bezeichner geben
  - Fünf
  - 5
  - V
  - 101
- Verwendete Darstellung sollte das Rechnen unterstützen

Binärcodierung elementarer Datentypen

- Ein **Zahlsystem** besteht aus
  - endlich vielen **Ziffern** (digits) und
  - einer Vorschrift, wie Zeichenreihen, die aus diesen Ziffern gebildet wurden, als Zahl-Werte zu interpretieren sind

Binärcodierung elementarer Datentypen

- **Arabische Zahlsysteme** zur Basis β
  - Natürliche Zahl *z* wird geschrieben als Polynom

$$z = \sum_{i=0}^{n-1} z_i \beta^i$$

Dabei

$$0 \leq z_i < \beta$$

Binärcodierung elementarer Datentypen

- Basis β der Darstellung wird **Radix** genannt
- Namen einiger wichtiger Zahlsysteme
  - Radix 10: Dezimaldarstellung
  - Radix 2: Binärdarstellung
  - Radix 8: Oktaldarstellung
  - Radix 16: Hexadezimaldarstellung

Binärcodierung elementarer Datentypen

- Zur Kennzeichnung der Basis wird diese oftmals als Subscript angegeben
  - $7_{10} = 7_8 = 111_2$
  - $9_{10} = 11_8 = 1001_2$
  - $15_{10} = 17_8 = 1111_2$

Binärcodierung elementarer Datentypen

- Bei Hexadezimal werden zusätzlich die Buchstaben

  A, B, C, D, E, F

  als Bezeichner für die Ziffern, die die Zahlwerte

  10, 11, 12, 13, 14, 15

  bezeichnen, benutzt

Binärcodierung elementarer Datentypen

- Arithmetik im Dualsystem (Oktal, Hexadezimal, ...) analog wie im Dezimalsystem
- Überträge dann bei 2, 8, 16, ...
- Beispiel: $01_2 + 01_2 = 10_2$

Binärcodierung elementarer Datentypen

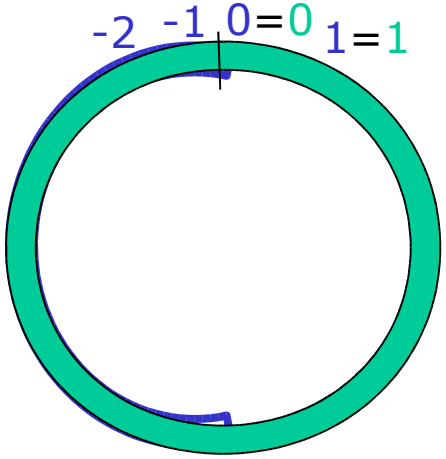
- Hardware eines Rechners realisiert nur Arithmetik für feste Zahllänge n
- Etwa n=32 oder n=64
- Mathematisch gesehen wird Arithmetik modulo 2^n realisiert:
  Überträge in die $n+1$–te Stelle fallen weg
- nur 2^n Dualzahlen 0 ... 2^n-1

Darstellung negativer Zahlen

- auch **negative Zahlen** werden benötigt
- Elegante Möglichkeit:

  Zweierkomplement-Darstellung
  - Positive Zahlen von 0 bis $2^{n-1}-1$
  - Negative Zahlen von -1 bis -2^{n-1}
  - Es gibt eine negative Zahl mehr als positive Zahlen

Zweierkomplement



Beispiel: 4-Bit Dualzahlen im Zweierkomplement

Dezimal	Zweierkomplement
+8	nicht darstellbar
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
0	0000
−1	1111
−2	1110
−3	1101
−4	1100
−5	1011
−6	1010
−7	1001
−8	1000

Zweierkomplement

- Zweierkomplement z'' von z kann einfach wie folgt erhalten werden
  - Bitweises Vertauschen von 0 und 1; $z \rightarrow z'$
  - Anschließend Addition von 1, $z'' = z' + 1$
  - Denn $z + z' = 2^n - 1$, also $z + (z' + 1) = 2^n$

Zweierkomplement

- Eigenschaften des Zweierkomplements
  - Addition negativer Zahlen passt sich in Addition modulo 2^n ein

$$x - y \equiv x + (2^n - y) \equiv x + \bar{y} \pmod{2^n}.$$

  - Eindeutige Darstellung der 0

$$\bar{0} = 2^n - 0 \text{ und damit } \bar{0} \equiv 0 \pmod{2^n}$$

  - Höchstes Bit zeigt an, ob Zahl positiv oder negativ ist

- **Zahlkonversion**

1. **Konversion dezimal → dual**. Die dezimale Zahl ist als Ziffernfolge gegeben, beim Einlesen z. B. als Folge von Zeichen.
   Wir nutzen nun folgende Darstellung von z im Horner-Schema:

$$\begin{aligned} z &= \sum_{i=0}^{n-1} z_i 10^i = z_{n-1}10^{n-1} + \ldots + z_2 10^2 + z_1 10^1 + z_0 \\ &= (\cdots((z_{n-1} \cdot 10 + z_{n-2}) \cdot 10 + z_{n-3}) \cdot 10 + \ldots + z_1) \cdot 10 + z_0 \end{aligned}$$

   Rein dual geschrieben, mit dualen Operatoren $+$ und $\cdot$, ergibt sich

$$z = (\cdots((d_{n-1} \cdot 1010_2 + d_{n-2}) \cdot 1010_2 + d_{n-3}) \cdot 1010_2 + \ldots + d_1) \cdot 1010_2 + d_0.$$

   Wir lesen die Zahl z ziffernweise von links nach rechts und wandeln jede Ziffer ins Dualsystem um. Zu Beginn der Konversion setzen wir $z = 0$, eine Dualzahl. Jedesmal, wenn wir eine weitere Ziffer z_i lesen, so setzen wir $z = z \cdot 1010_2 + d_i$, wobei d_i die Dualdarstellung der Dezimalziffer z_i ist. Da wir dual rechnen, ist z wieder eine Dualzahl. Per Induktion folgt, daß wir am Ende z als Dualzahl vorliegen haben.

- **Zahlkonversion**

2. **Konversion dual $\rightarrow$ dezimal**. Wir wenden wieder das Horner-Schema an, um aus der Dualzahl D die dezimale Ziffernfolge $\sum_{i=0}^{n-1} z_i 10^i$ zu bekommen. Wir erhalten d_0 als Rest der ganzzahligen Division von $D = D_0$ durch 10, also $d_0 = D_0 \% 1010_2$ (worauf wir d_0 in eine einzelne Dezimalziffer z_0 umwandeln). Setzen wir dann $D_1 = D_0/1010_2$ und fahren nach dieser Methode fort, so erhalten wir jeweils $z_i = D_i \% 1010_2$, $D_{i+1} = D_i/1010_2$. Wir brechen ab, sobald $D_i = 0$, da alle weiteren Dezimalziffern 0 bleiben. (Die so erhaltene Ziffernfolge muß zur Ausgabe am Bildschirm noch umgedreht werden.)