LU01h - Binär codierte Ganzahlen - Einführung

Siehe http://www.ulthryvasse.de/index.html

Bei der binären Codierung von Zahlen werden drei Ziele verfolgt:

- Kompakte Speicherung der Information
- Effiziente Verarbeitung von Berechnungen
- Präzise Speicherung der Zahlen

Je nach gewähltem Datentyp werden diese Ziele unterschiedlich gewichtet.

Ganzzahlen

Positive Zahlen

Bei positiven Ganzzahlen wird der Wert im binären Zahlensystem mit '0' und '1' gespeichert. Das erste Bit ist dabei immer '0'. Beim Programmieren bestimmt der Datentyp die Anzahl Bits die verwendet werden.

Beispiel

```
short zahl = 5;
```

- `short` verwendet 16 Bit
- Resultat: $0000\ 0000\ 0000\ 0101_2$

Negative Zahlen

Bei negativen Zahlen wird das sogenannte Zweierkomplement gespeichert:

1. Addieren Sie 1 zur Zahl: $-13_{10} + 1_{10} = -12_{10}$
2. Entfernen Sie das Vorzeichen der Zahl: $-12_{10} \Rightarrow 12_{10}$
3. Übertragen Sie die Zahl ins Binärsystem: $12_{10} = 0000\ 0000\ 0000\ 1100_2$
4. Invertieren Sie alle Bits: $0000\ 1100_2 \Rightarrow \mathbf{1111\ 1111\ 1111\ 0011_2}$

Negative Zahlen haben immer Bit '1' an erster Stelle.

Binary Coded Decimals (BCD)

Bei dieser Codierung wird jede Ziffer einzeln binär codiert. Üblicherweise werden 4 Bits verwendet, um eine dezimale Ziffer zu codieren. Falls negative Zahlen möglich sind, wird das Vorzeichen ebenfalls einzeln codiert.

Beispiel

Die Zahl +157 soll als BCD codiert werden.

```
Dezimal:        +    1    5    7
BCD 8-4-2-1:  1010 0001 0101 0111
```

In diesem Beispiel gilt:

- Jede Ziffer wird mit 4 binären Stellen codiert.
- Vorzeichen: + = 1010_2 / - = 1011_2.
- Die binären Stellen haben (von links nach rechts) die Werte 8-4-2-1.

Weitere BCD-Codierungen

Es existieren weitere Möglichkeiten zur Codierung von Dezimalzahlen. Dabei muss festgelegt sein:

- Welche Stelle im Binärcode entspricht welchem Werte (8-4-2-1 oder 2-4-2-1).
- Wie viele Stellen vor und nach dem Komma werden gespeichert.

Fliesskommazahlen

Eine Fliesskommazahl wird als eine Multiplikation mit einer 10er-Potenz ausgedrückt. Anstelle von $+2.345 * 10^5$ schreiben wir `+2.345E5`. Zum Beispiel:

```
+2.345 E 5.
^  ^     ^
|  |     +-- Exponent
|  +-------- Mantisse
+----------- Vorzeichen
```

Ganzzahlige Mantisse

Für die binäre Codierung müssen wir zunächst die Mantisse ganzzahlig machen. Dazu verschieben wir den Dezimalpunkt nach links oder rechts.

```
+2345E+2
```

Hier wurde der Dezimalpunkt drei Stellen nach rechts verschoben. Deshalb wurde der Exponent um 3 reduziert.

Codierung

In der binären Codierung werden die verschiedenen Informationen einzelne codiert:

1. Vorzeichen: + = 0_2 / - = 1_2
2. Mantisse: Wird als positive binäre Ganzzahl codiert (Vorzeichen wird ignoriert).
3. Exponent: Ein Korrekturwert (Bias) wird addiert und das Resultat als positive binäre Ganzzahl codiert.

In der Regel werden Fliesskommazahlen entweder mit 32 Bit (`short`-Format) oder 64 Bit (`long`-Format) codiert. Da die Regeln für beide gleich sind, beschränken wir uns auf das `short`-Format.

Bias

Der Korrekturwert oder Bias dient dazu, dass der codierte Exponent immer positiv ist. Dadurch entfällt eine separate Logik für das Vorzeichen des Exponenten.

Im `short`-Format können die Exponenten -127 bis +128 sein. Daher beträgt der Bias **127**, wodurch wir einen korrigierten Exponenten von +0 bis +256 erhalten.

Anzahl Stellen

Die Anzahl der binären Stellen wird vom Datentyp bestimmt. Bei `short` gilt:

- Vorzeichen: 1 Bit
- Exponent: 8 Bit (Bias 127)
- Mantisse: 23 Bit

---

m114-A1G, m114-A1F

Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m114/learningunits/lu01/binaercodiert

Last update: **2026/01/28 21:12**