

# LU01j - Binary coded Decimals (BCD)

Siehe <http://de.wikipedia.org/wiki/BCD-Code>

Der BCD-Code wird zur Speicherung von Dezimalzahlen genutzt. Dabei wird jede dezimale Stelle einzeln durch 4 Bits (Halbbyte) dargestellt. Durch Addieren der Werte der einzelnen binären Stellen, erhält man den Wert der Dezimalzahl.

## 8-4-2-1

Eine Variante des BCD-Codes weist den 4 Bits die Werte 8, 4, 2 und 1 zu.

| Dezimal | BCD               | Dezimal | BCD               |
|---------|-------------------|---------|-------------------|
| 0       | 0000 <sub>2</sub> | 5       | 0101 <sub>2</sub> |
| 1       | 0001 <sub>2</sub> | 6       | 0110 <sub>2</sub> |
| 2       | 0010 <sub>2</sub> | 7       | 0111 <sub>2</sub> |
| 3       | 0011 <sub>2</sub> | 8       | 1000 <sub>2</sub> |
| 4       | 0100 <sub>2</sub> | 9       | 1001 <sub>2</sub> |

### Beispiel

$$1001_2 = 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 9_{10}$$

## 2-4-2-1

Eine andere Variante weist den 4 Bits die Werte 2, 4, 2 und 1 zu.

| Dezimal | BCD               | Dezimal | BCD               |
|---------|-------------------|---------|-------------------|
| 0       | 0000 <sub>2</sub> | 5       | 1011 <sub>2</sub> |
| 1       | 0001 <sub>2</sub> | 6       | 1100 <sub>2</sub> |
| 2       | 0010 <sub>2</sub> | 7       | 1101 <sub>2</sub> |
| 3       | 0011 <sub>2</sub> | 8       | 1110 <sub>2</sub> |
| 4       | 0100 <sub>2</sub> | 9       | 1111 <sub>2</sub> |

### Beispiel

$$1110_2 = 1 \cdot 2 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 8_{10}$$

## Vorteile

Diese Codierung ist vor allem für Dezimalbrüche von Vorteil. Dabei werden die Rundungsfehler [binär codierter Fließkommazahlen](#) vermieden. Vor allem bei Grossbanken sind diese Rundungsfehler von

Bedeutung. Deshalb nutzen heute nur noch Grossrechner eine Form der BCD-Codierung. Zu meiner Zeit als Applikationsentwickler einer Grossbank galt: Jede Berechnung, die kaufmännisch relevant ist, muss mit BCD-codierten Variablen erfolgen.

## Nachteile

Der Nachteil des BCD-Codes ist der verschwenderische Umgang mit dem Speicher. Zum Beispiel können Sie bei der **8-4-2-1**-Codierung aus der oben stehenden Tabelle entnehmen, dass 6 von den 16 möglichen Bit-Kombinationen gar nicht verwendet werden:

- 1010
- 1011
- 1100
- 1101
- 1110
- 1111

Somit benötigt eine BCD-codierte Zahl 37.5% mehr Speicher als eine binär codierte Zahl.

## Vorzeichen

Für das Vorzeichen verwenden BCD-Codes entweder ein einzelnes Bit oder ein Halbbyte.

## Nachkommastellen

Eine Variable die im BCD-Format gespeichert wird, hat immer eine vorher festgelegte Anzahl an Stellen vor und nach dem Komma.

In PL/I bedeutet die Deklaration

```
dcl summe dec fixed(15,3);
```

- decimal fixed: Eine Dezimalzahl im BCD-Format.
- 15: Die Zahl hat insgesamt 15 Stellen.
- 3: Die Zahl hat 3 Stellen nach dem Komma (somit also 12 Stellen vor dem Komma).

In MySQL würde das gleiche Format so definiert:

```
summe DECIMAL(15,3)
```

---

[m114-A1G](#), [m114-A1F](#)

 Marcel Suter

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m114/learningunits/lu01/binarycodeddecimals>

Last update: **2026/01/30 11:06**

