Die Darstellung von ganzen Zahlen

Ein Mikroprozessorsystem verarbeitet immer Bitmuster in Einheiten zu 8, 16, 32 oder mehr Bit. *Erst durch die Art der Verarbeitung wird diesem Bitmuster eine bestimmte Bedeutung zugewiesen.* Wende ich einen arithmetischen Maschinenbefehl auf ein Bitmuster an, so wird es als Zahl interpretiert, eine Ausgabe auf den Bildschirm interpretiert das gleiche Bitmuster dagegen als darstellbares Zeichen des aktuellen Zeichensatzes.

Beispiel: Ein Byte hat den Inhalt `01000011b = 43h = 67d`

Das kann unterschiedlich interpretiert werden.

- ASCII-Zeichen 'C'
- Vorzeichenlose oder vorzeichenbehaftete 8-Bit-Zahl: 67d = 43h
- als Maschinenbefehl
- Bitmuster um die Interrupts 0,1 und 6 freizugeben

Hier geht es um die Interpretation von Bitmustern ganze Zahlen betrachten.

Ein Mikroprozessor kann ein Bitmuster als Zahl interpretieren, dabei wird nach *ganzen Zahlen* mit und ohne Vorzeichen sowie *Fließkommazahlen* unterschieden.

Um die Darstellung der ganzen Zahlen zu verstehen, betrachten wir zunächst das uns geläufige Dezimalsystem , in dem zehn verschiedene Ziffern mit Potenzen der Zahl 10 gewichtet werden:

10^4	10^3	10^2	10^1	10^0
2	**3**	**0**	**8**	**9**

Ausgeschrieben bedeutet die Zahl: `2 * 10^4 + 3 * 10^3 + 0 * 10^2 + 8 * 10^1 + 9 * 10^0 = 2 * 10 000 + 3 * 1000 + 0 * 100 * 8 * 10 + 0 * 1`. Jede Stelle hat also einen bestimmten Wert: Die ganz rechts hat den Wert 1, dann kommt der Wert 10, 100 u.s.w. Man spricht von einem Stellenwertsystem.

Da ein Mikroprozessorsystem nur 2 Zustände kennt, 0 und 1, verwendet man hier als Basis die Zahl 2 anstatt der gewohnten 10:

2^4	2^3	2^2	2^1	2^0
1	**1**	**1**	**0**	**1**

From:
https://info-bw.de/ -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:techinf:assembler:zahlen:start?rev=1632761985

Last update: **27.09.2021 16:59**