

8086 - Assemblycode

Der Computer besteht aus Elektronik. Um es nicht zu kompliziert zu machen, erklärt man das mit Transistoren: Es gibt im Mikroprozessor Millionen von Transistoren.

Eigentlich versteht der Computer kein Deutsch oder Englisch. Nein, er versteht nur „Einsen und Nullen“! Man sagt: Der Computer (eine Maschine!) versteht nur „Maschinensprache“.

Um nun eine „Programmiersprache“ für den Mikroprozessor zu haben, gibt es „Assembly“ – oder „Assemblycode“. Das ist ungefähr dasselbe wie „Maschinensprache“, nur leichter zu begreifen!

Assembly war so ungefähr die erste Programmiersprache überhaupt, nach Maschinensprache.

Ein erster Einstieg: 8086-Assembly

Die Sprache Assembly oder Assemblycode ist die Programmiersprache, die so ziemlich am nächsten an dem dran ist, was der Computer versteht.

Wir machen jetzt einmal Assembly:

Der Mikroprozessor mit dem Namen „8086“ besteht aus mehreren „Registern“. Das erste Register heißt AX (und dann folgen BX, CX und DX). Das AX-Register kann 16 Bits aufnehmen. Es kann aufgeteilt werden in AL und AH. AL und AH sind nur je 8 Bit breit. Man kann sich AL und AH nebeneinanderstehend vorstellen. So erhält man wieder das AX-Register.

Wenn in den Registern noch nichts drinsteht, dann nimmt man am besten den Befehl „MOV“ (das heißt: move, zu Deutsch: bewege).

Die Befehlszeile kann etwa heißen: MOV AX, 0001.

MOV AX, 0001 macht, dass der Wert 0001 (also eine Eins!) in das Register AX hineingeschrieben wird.

Nun machen wir das noch einmal, nur mit CX: MOV CX, 2222. Wenn dieser Befehl ausgeführt wurde, steht die Zahl 2222 im Register CX.

Wenn man mit dem 8086-Mikroprozessor addieren will, kann man das mit dem Befehl „ADD“ tun. ADD (to add) kommt auch aus dem Englischen und heißt: addiere. Zum Addieren könnte man AX und CX nehmen. Also heißt der ganze Befehl: ADD AX, CX.

ADD AX, CX macht das Folgende: Zuerst sieht der Prozessor nach, was in AX und was in CX steht. Diese beiden Werte in den beiden Registern werden addiert, das Ergebnis kommt in AX hinein. Folglich ist der Wert, der vorher in AX war, verloren gegangen.

Jetzt nimm mal bitte an, alle diese beschriebenen Befehle werden nacheinander ausgeführt! Was steht am Ende im Register AX?