

# Vom Quellcode zum Prozessor

## 1. Compiler übersetzt Quellcode in Maschinensprache

-> Compiler muss wissen welche CPU weil unterschiedlich Sprache

-> z.B: 16 Bit CPU = Befehle von 16 Bit können überführt werden

<pre>void main (void){   int x = 2;   int y = 5;   int z = x + y; }</pre>	zu	00000000110000100000000010000110100
---	----	-------------------------------------

## 2. Maschinensprache in Einzelteile zerlegen

-> Wenn Nummer = 1, dann geht es um die Zahl, sonst um die Speicherzelle

-> die Befehle werden auf dem ACC durchgeführt

Reserve	Befehl	Nr.	Operant	Bedeutung		Nr.	Befehl	Bedeutung
---------	--------	-----	---------	-----------	--	-----	--------	-----------

				Laden Wert 2 in ACC Speichere ACC in Speicherzelle 13			
000000	001	1	000010	Laden Wert 5 in ACC Speichere ACC in Speicherzelle 14	000	NOOP	No Operation
000000	010	0	001101	Lade Speicherzelle 13 inACC Addiere Speicherzelle in 14 zu ACC Speichere ACC in Speicherzelle 15	001	LOAD	Was laden
000000	001	1	000101	Program beenden	010	STORE	Speicher Addieren
000000	010	0	001110		011	ADD	Minimieren
000000	001	0	001101		100	SUB	Vergleichen
000000	011	0	001110		101	EQUAL	Überspringe
000000	010	0	001111		110	JUMP	Beenden
000000	111	0	000000		111	HALT	

## Befehlsformat

Das **Befehlsformat** definiert für jeden einzelnen Befehl, wie dieser codiert ist:  
<Befehl><Num><Operand>

**Opcod** - Binäre Codierung, aus der sowohl der Befehl, als auch zusätzlich benötigte Steuerinformationen hervorgehen.

**Klassifizierungen** - Bei den Befehlsformaten werden verschiedene Klassifizierungen unterschieden:

- Einadressformat | Zweiadressformat | Dreiadressformat

Das **Einadressformat** entspricht dem Format Vom Quellcode zum Prozessor. Hier wird für die einzelnen Befehle nur der *Opcod* und die *Adresse eines Operanden* (deshalb *Einadressformat*) angegeben.

<Opcode><Operand1>

---

## Bedeutung des Akkumulators

Wird ein zweiter **Operand** benötigt, so wird vorausgesetzt, dass dieser **sich im Register Akkumulator** befindet. Das bei der Abarbeitung des Befehls berechnete *Ergebnis* wird per Definition wieder im *Akkumulator* gespeichert.

---

Revision #3

Created 18 August 2022 08:48:05 by Merith Holtmann

Updated 1 October 2022 08:32:24 by Merith Holtmann