

3.3 Systemnahe Programmierung

Unsere heutigen Computer programmieren wir normalerweise in einer Hochsprache, wie z.B. Java, Python, C usw. Dabei helfen uns die Compiler und das Betriebssystem, die Ausführung dieser Befehle auf dem Prozessor zu steuern. In ihrem Ursprung musste der Prozessor des Computers allerdings „per Hand“ programmiert werden. Wir nennen eine Programmiersprache, die direkt auf dem Prozessor arbeitet **Assembler**.

Unser Simulator „Minimaschine“ versteht einen solchen Assembler Code. Jeder Befehl im Assembler besteht aus drei Bestandteilen:

Sprungmarke	Befehl	Adresse/Wert
z.B.		
Start:	LOAD	100

Unser Beispielcode lädt den Inhalt der Speicherzelle 100 in den Akkumulator (kurz Akku). Eine Liste aller Befehle findet sich auf der Webseite. Die Befehle gliedern sich grob in Transportbefehle, Arithmetische Befehle und Steuerbefehle.

Jedes Programm, das in Assembler geschrieben wird, wird anschließend durch Assemblieren in den Speicher der Registermaschine übernommen. Der Prozessor beginnt an der Speicherzelle auf die der Programmzähler (P.C.) zeigt und arbeitet taktweise das Programm nach dem Kreislauf in Abb.1 ab.

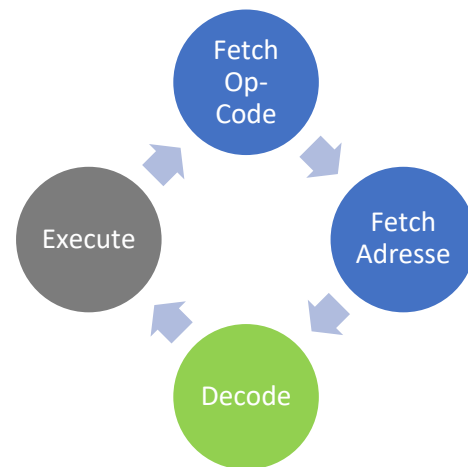


Abbildung 1 Takt-Kreislauf des Prozessors