

# Bemerkungen zum Tutorium vom 1.12.

Lukas Bach

11. Dezember 2016

Befehlssyntax	Formel	Bedeutung
<i>LDC const</i>	$Akku \leftarrow const$	Lade eine Konstante <i>const</i> in den Akku
<i>LDV adr</i>	$Akku \leftarrow M(adr)$	Lade einen Wert vom Speicher bei Adresse <i>adr</i> in den Akku
<i>STV adr</i>	$M(adr) \leftarrow Akku$	Lade Speichere den Wert aus dem Akku im Speicher bei Adresse <i>adr</i>
<i>LDIV adr</i>	$Akku \leftarrow M(M(adr))$	Lade einen Wert vom Speicher bei der Adresse, die bei <i>adr</i> gespeichert ist, und lade den Wert in den Akku
<i>STIV adr</i>	$M(M(adr)) \leftarrow Akku$	Speichere den Wert im Akku bei der Adresse, die in <i>adr</i> gespeichert ist.
<i>ADD adr</i>	$Akku \leftarrow Akku + M(adr)$	Addiere den Wert bei <i>adr</i> zum Akku dazu.
<i>"OP" adr</i>	$Akku \text{ "OP" } M(adr)$	Wende bitweise Operation auf Akku mit Wert bei <i>adr</i> an. $Op \in \{AND, OR, XOR\}$ .
<i>NOT</i>		Bitweise Invertierung aller Bits des Akku-Datenwortes
<i>RAR</i>		Rotiere alle Akku-Bits eins nach rechts
<i>EQL adr</i>		Setze Akku auf 11...11, falls Wert bei <i>adr</i> gleich Akku-Wert, setze Akku auf 00...00 sonst.
<i>JMP adr</i>		Springe zu Befehlsadresse <i>adr</i>
<i>JMN adr</i>		Springe zu Befehlsadresse <i>adr</i> , falls Akku negativ (also erstes Bit = 1), sonst fahre normal fort.