

Klasse:

Name:

Datum:

Programmierung

Eine Programmierung, in der Befehle und Zahlen BINÄR vom Programmierer eingegeben werden nennt man: Assembler, einfacher sind aber Hochsprachen, wie BASIC, Pascal, C++, etc.

Der Democomputer arbeitet ein Programm ab, das im ROM gespeichert ist. Ein ROM-Wort ist 4 Bit breit, das heißt es sind $2^4 = 16$ Zustände pro Wort möglich. Dies entspricht auch genau dem implementierten Befehlssatz, der aus 16 Befehlen besteht. Im Democomputer wird zwischen Einwort- und Zweiwort-befehlen unterschieden:

- **Einwortbefehl:** Benötigt keine Argumente.
- **Zweiwortbefehl:** Benötigt noch ein Argument (Datenwort oder Adresse), welches an der dem Befehl folgenden Adresse im ROM stehen muss

Mnemonic:
damit sich
Menschen die
Befehle
merken

Das ist der
„echte“ CPU
Befehl als
binäres Muster

aus wieviel
Worten der
Befehl besteht.

Anzahl der Taktzyklen, für
vollständige Abarbeitung.

Etwas Information
wie der Befehl
funktioniert.

Mnemonic	Code	Anz. Worte	CPU-Zyklen	Beschreibung
Datentransferbefehle:				
MVI R0	0100	2	8	unmittelbar folgendes Wort in Register 0
MVI R1	0101	2	8	Unmittelbar folgendes Wort in Register 1
STO R0	0000	2	10	Register 0 in RAM; 2. Wort RAM-Adresse
STO R1	0001	2	10	Register 1 in RAM; 2. Wort RAM-Adresse
LD R0	0010	2	10	RAM in Register 0; 2. Wort RAM-Adresse
LD R1	0011	2	10	RAM in Register 1; 2. Wort RAM-Adresse
MOV R1,R0	1001	1	5	Register 0 in Register 1 kopieren
MOV R0,R1	1010	1	5	Register 1 in Register 0 kopieren
Input-/Outputbefehle:				
IN	1000	1	7	Input-Port in Register 0
OUT	1011	1	7	Register 0 in Output-Port
Arithmetische Befehle:				
ADD R1	1101	1	5	Register 1 + Register 0 (ohne carry), Resultat in R0
Rotationsbefehle:				
ASL	1110	1	5	Register 0 links schieben, Überlauf in carry
RAR	1111	1	5	R0 rechts schieben, LSB in carry
Sprungbefehle:				
LMP	0110	2	8	Sprung zu ROM-Adresse in 2. Wort
JC	0111	2	8	Sprung zu ROM-Adresse in 2. Wort wenn carry = 1
JNC	1111	2	8	Sprung zu ROM-Adresse in 2. Wort wenn carry = 0

MITTE



Klasse:

Name:

Datum:

Adressierung Ram

Die RAM-Adressen im Democomputer sind 16-19. Diese Adressen lassen sich nicht mit 4 Bit abbilden und können folglich im ROM nicht als Argument übergeben werden. Deshalb gelten für RAM-Adressen im ROM folgende Regeln:

- 0000 - 0011 entsprechen den RAM-Adressen 16 - 19

Umrechnungstabelle von dualen Zahlen in dezimale Zahlen

4 Bit: 2^3 2^2 2^1 2^0
 1 1 0 1 = $8 + 4 + 0 + 1 = 13_{\text{dez}}$

Binärzahl	Umrechnung	Dezimal
0000		
0001		
0010		
0011		
0100		
0101		
0110	$0 + 4 + 2 + 0 = 6$	6
0111		
1000		
1001		
1010		
1011		
1100		
1101		
1110		
1111		

MITTE