

Gerlach, Luisa	gerlaclu	599244
Heine, Tom Martin	heinetom	597978
Kühne, Marc Sebastian	kuehnese	599833
Seegert, Noah-Joël	segertno	596234

Aufgabe 1

(a) 0 001 000010 000100

MOV R2, R4

(b) 0 110 010111 000011 — 0 000 000000 010001

ADD R23, R3, 17

(c) 0 000 110101 010010

PUSH R18

Aufgabe 2

Aufbauen eines Befehlsformats:

16 Befehle	=> 4 Bit
2 - Adressmaschine	=> (2 Adressfelder)
16 Register	=> 4 Bit
16 MWorte Hauptspeicher	=> 24 Bit
Speicher Wort-/Byteadressiert	=> 1 Bit
5 Adressarten	=> 3 Bit
16 Bit Konstanten	=> 16 Bit
16 Bit Index	=> 16 Bit
Datentransfer	=> 1 Bit

Befehlsformat:

1.Wort:	Opcode 4 Bit	1.Operand (reg.direkt) 4 Bit	Adressart 3 Bit	Richtung 1 Bit	2.Operand 4 Bit
2.Wort:	Konstante / Index / 1.Teil Speicheradresse 16 Bit				

Aufgabe 3

Adresse	Mikrobefehl	Kommentar
0	IP_out, MAR_in, RD, A_in, $F = A+1$, NA[1]	Befehl wird aus dem Hauptspeicher in die ALU geladen
1	ALU_out, IP_in, WRDY, NA[2]	wartet, bis Speicherzugriff beendet wurde
2	MDR_out, IR_in, DECODE[jpz: 3, sub: 8]	Befehl wird gelesen und dekodiert
3	IP_out, MAR_in, RD, A_in, $F = A+1$, NA[4]	Sprungdistanz wird aus dem Hauptspeicher in die ALU gelesen
4	ALU_out, IP_in, WRDY, NA[5]	wartet, bis Speicherzugriff beendet wurde
5	IFZ[6]	prüft, ob die Flag 0 ist
6	MDR_out, B_in, $F=A+B$, NA[7]	addiert Sprungdistanz in ALU zu IP
7	ALU_out, IP_in, NA[0]	schreibt IP zurück
8	SR=ROP1, R_out, MAR_in, RD, NA[9]	erster Operand wird adressiert
9	SR=ROP2, R_out, A_in, NA[10]	zweiter Operand wird in ALU geladen
10	IRop_out, B_in, $F=A+B$, WRDY, NA[11]	Adresse des zweiten Operanden wird berechnet, wartet bis Speicherzugriff beendet
11	MDR_out, Y_in, NA[12]	erster Operand wird gespeichert
12	ALU_out, MAR_in, RD, NA[13]	zweiter Operand wird adressiert
13	Y_out, B_in, WRDY, NA[14]	erster Operand wird in die ALU geladen
14	MDR_out, A_in, $F=B-A$, FLE, NA[15]	zweiter Operand wird ALU geladen und Subtraktion wird ausgeführt
15	ALU_out, MDR_in, WR, WRDY, NA[0]	Ergebnis wird in den Speicher geschrieben und es wird gewartet, bis der Speicherzugriff beendet wurde