

Tutorium 17, #6

Max Göckel– *uzkns@student.kit.edu*

Institut für Theoretische Informatik - Grundbegriffe der Informatik

Die MIMA: Aufgabe

Beschreibt die folgenden Befehle.

JMP a
LDV a
HALT
ADD a

Beschreibt die folgenden Befehle.

JMP a: Springt zum Befehl an Stelle a (oder in GBI: Mit Markierung a)

LDV a: Lädt den Wert an Adresse a in den Akku

HALT: Stoppt die MIMA

ADD a: Addiert den Wert an Adresse a auf den Akku drauf (Ergebnis landet wieder im Akku)

Wir kennen schon:

- $x - y$
- $x \text{ div } 2$

Erinnerung: div

■ $\text{adr1} = \text{adr1} \text{ div } 2$:

```
LDV    EINS
NOT
AND     adr1
RAR
STV     adr1
HALT
```

■ $\text{adr3} = \text{adr1} - \text{adr2}$:

```
LDV    adr2
NOT
ADD     EINS
ADD     adr1
STV     adr3
HALT
```

Was macht folgendes Programm?

Zeilen	Code
1	LDC 4
2	STV adr0
3	ADD EINS
4	STV adr1
5	ADD EINS
6	STV adr2
7	LDC 0
8	STIV adr0
9	ADD EINS
10	STIV adr1
11	<i>m1</i> : LDIV adr1
12	STV adr3
13	LDIV adr0
14	ADD adr3
15	STIV adr2
16	LDV adr0
17	ADD EINS
18	STV adr0
19	ADD EINS
20	STV adr1
21	ADD EINS
22	STV adr2
23	JMP m1

Der Speicher hat folgende Form:

Adresse	Wert
0 aka adr0	?
1 aka adr1	?
2 aka adr2	?
3 aka adr3	?
4	?
5	?
6	?
:	:

Schreibe ein Programm dass $z = x \bmod y$ ausführt.

Dabei:

- $x = \text{Wert an Adresse adr1}$
- $y = \text{Wert an Adresse adr2}$
- $z = \text{Wert an Adresse adr3}$
- $x \text{ aus } \mathbb{N}_0$
- $y \text{ aus } \mathbb{N}_+$

Schreibe ein Programm dass $z = x \bmod y$ ausführt.

```
LDV adr1
STV adr3
m1: LDV adr2
NOT
ADD EINS
ADD adr3
JMN m0
STV adr3
JMP m1
m0: HALT
```

Schreibe ein Programm dass $z = x \text{ div } y$ ausführt.

Dabei:

- $x = \text{Wert an Adresse adr1}$
- $y = \text{Wert an Adresse adr2}$
- $z = \text{Wert an Adresse adr3}$
- $x \text{ aus } \mathbb{N}_0$
- $y \text{ aus } \mathbb{N}_+$

Schreibe ein Programm dass $z = x \text{ div } y$ ausführt.

```
LDC 0
STV counter
LDV adr1
STV rest
m1: LDV adr2
    NOT
    ADD EINS
    ADD rest
    JMN m2
    STV rest
    LDV counter
    ADD EINS
    STV counter
    JMP m1
m2: LDV counter
    STV adr3
    HALT
```

An Speicherstelle `adr1` steht ein Wert $x \in \mathbb{N}_+$.

Schreibe ein Programm, dass den Wert an `adr2` x -mal multipliziert
($\text{adr2} \cdot x$)

Die MIMA: Aufgabe

Schreibe ein Programm, dass den Wert an adr2 x -mal multipliziert:

```
LDC 0
NOT
STV adr3 ; -1 an adr3
LDV adr2 ; Wert von adr2 abspeichern
STV adr4
m1: LDV adr1
    ADD adr3
    STV adr1 ; adr1 - 1 rechnen und speichern
    LDV adr2
    ADD adr4
    STV adr2 ; adr2 + adr4 rechnen und speichern
    LDV adr3
    JMN m2
    JMP m1
m2: HALT
```

LDC c - lädt Konstante c in den Akku

LDV a - Lädt Wert an Stelle a in den Akku

STV a - speicher Wert vom Akku in Stelle a

ADD a - Addiert Wert an Stelle a auf den Wert im Akku drauf (Ergebn. → Akku)

AND a - VerUNDet Wert v.St. a und Akku → Akku

OR a - VerODERt Wert v.St. a und Akku → Akku

XOR a - VerXORt Wert v.St. a und Akku → Akku

EQL a - Falls Akku = Wert a.St. a → Akku -1 in Akku, sonst 0

JMP a - Springt zum Befehl mit Marker a

JMN a - Springt zum Befehl a wenn Akku < 0

HALT - Endet die Ausführung der MIMA

NOT - Invertiert den Akku ($0 \rightarrow 1$, $1 \rightarrow 0$)

RAR - Rotiert den Akku um eins nach rechts

Der Wert des Akku wird immer überschreiben wenn man neue Werte lädt, also Werte die man mehrfach braucht zwischenspeichern (STV).