

Rechnerorganisation im WS 2017/18

### 3. Übungsblatt

Abgabetermin: 27. November, 13:15 Uhr

Prof. Dr. Wolfgang Karl  
Haid-und-Neu-Str. 7

Dr.-Ing. Ömer Terlemez  
Adenauerring 2, Geb. 50.20

Email: [ti@ira.uka.de](mailto:ti@ira.uka.de)  
Web: <http://ti.ira.uka.de>

#### Aufgabe 1

(3 Punkte)

Die MIMA-Architektur soll um einen weiteren Maschinenbefehl erweitert werden. Der TRI-Befehl soll den Inhalt des Akku-Registers mit 3 multiplizieren und das Ergebnis wieder im Akku speichern:

TRI :  $3 * \text{Akku} \rightarrow \text{Akku}$

Schreiben Sie ein Mikroprogramm, das die Ausführungsphase des TRI-Befehls realisiert. Geben Sie dieses Mikroprogramm in Register-Transfer-Schreibweise an.

#### Aufgabe 2

(6 Punkte)

Welche MIMA-Befehle werden durch die folgenden in Register-Transfer-Schreibweise angegebenen Mikroprogramme realisiert?

Bei den Mikroprogrammen handelt sich dabei immer um die Ausführungs-Phase, die in Takt 7 beginnt.

1.
  7. Takt: IR  $\rightarrow$  SAR; R = 1
  8. Takt: Akku  $\rightarrow$  X; R = 1
  9. Takt: R = 1
  10. Takt SDR  $\rightarrow$  Y
  11. Takt ALU auf 111
  12. Takt Z  $\rightarrow$  Akku
2.
  7. Takt: IR  $\rightarrow$  IAR
3.
  7. Takt: IR  $\rightarrow$  SAR; R = 1
  8. Takt: R = 1
  9. Takt: R = 1
  10. Takt: SDR  $\rightarrow$  Akku
4.
  7. Takt: IR  $\rightarrow$  Akku
5.
  7. Takt: Akku  $\rightarrow$  SDR
  8. Takt: IR  $\rightarrow$  SAR; W = 1
  9. Takt: W = 1
  10. Takt: W = 1

Aufgabe 3

(4 Punkte)

Welche Funktion erfüllt die folgende Sequenz von MIMA-Befehlen?

LDV 0x2014  
NOT  
STV 0x2015  
LDC 0x0001  
ADD 0x2015  
STV 0x2016

Geben Sie für die ersten drei Maschinenbefehle das jeweilige Befehlswort (siehe Beiblatt) hexadezimal und binär an.

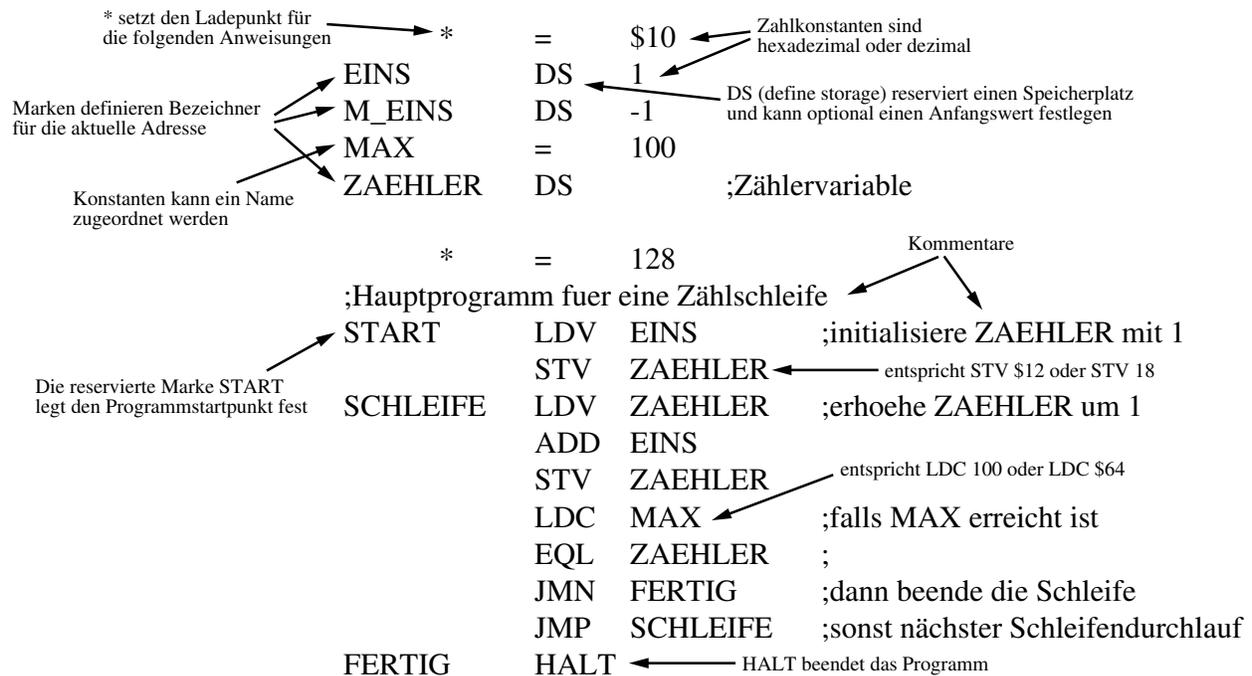
Aufgabe 4

(8 Punkte)

Beantworten Sie folgende Fragen zur MIMA-Architektur (siehe Aufgabe 4 auf Übungsblatt 2 und Beiblatt).

1. Wieso ist es wichtig, für den LDC-Befehl den Opcode 0x0 zu wählen? 2 P.  
Welche praktischen Probleme würde die Verwendung eines anderen Opcodes für LDC nach sich ziehen?
2. Wie viele Maschinenbefehle implementiert die MIMA in der Ihnen bekannten Version? 2 P.  
Wie viele verschiedene Maschinenbefehle könnte die MIMA aufgrund ihrer Architektur theoretisch in ihrem Befehlssatz zur Verfügung stellen?  
(Tipp: Betrachten Sie den Aufbau des MIMA-Befehlsformats!)
3. Wofür stehen die Abkürzungen der folgenden MIMA-Register und welche Aufgabe haben diese Register? 2 P.
  - IAR
  - IR
  - SAR
  - SDR
4. Drei der vier zuvor genannten Register werden nach der Lesephase (*fetch phase*) nicht zur Ausführung aller Maschinenbefehle benötigt. Welche Register sind dies und welcher Befehl benötigt diese beispielsweise nicht? 2 P.

Zur Erleichterung der MIMA-Programmierung wird die im folgenden Beispiel dargestellte Assemblernotation eingeführt:



### Aufgabe 5

(6 Punkte)

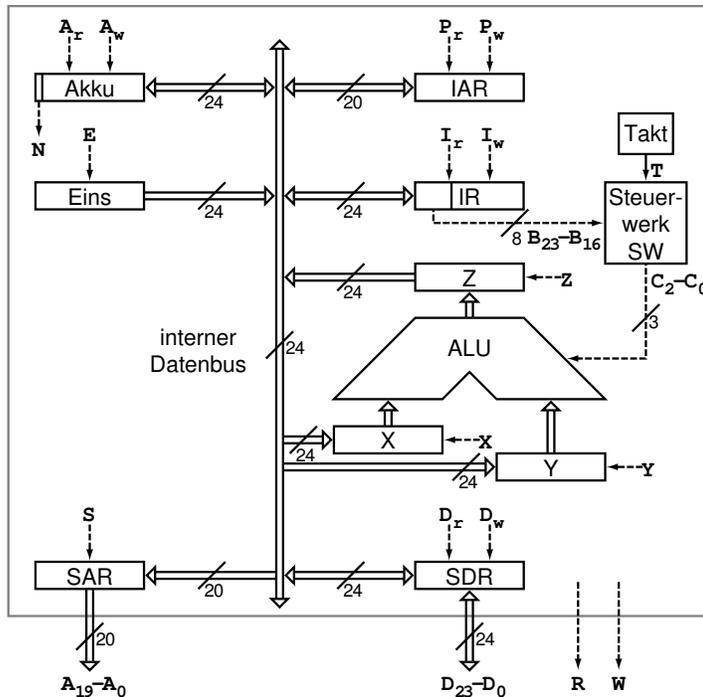
Schreiben Sie ein MIMA-Programm, das die Inhalte der Speicherzellen 0x00000 und 0x00001 addiert. Addieren Sie anschließend die Zahl 3 auf das Ergebnis, wenn dieses durch 2 teilbar ist und sonst die Zahl 4. Legen Sie das endgültige Ergebnis in Speicherzelle 0x00002 ab.

Das Programm soll bei Adresse 0x00100 beginnen.

Schreiben Sie zu jeder Ihrer Programmzeilen einen Kommentar in Pseudo-Code, der beschreibt was Ihr Programm an dieser Stelle tut. Ohne Kommentar keine Punkte!

Beispiel: 0x00100: 087654 entspricht LDC 0x87654,  
lädt die Konstante 0x87654 in den Akku

**Architektur der MIMA**



**Register**

- Akku: Akkumulator
- X: 1. ALU Operand
- Y: 2. ALU Operand
- Z: ALU Ergebnis
- Eins: Konstante 1
- IAR: Instruktionsadreibregister
- IR: Instruktionsregister
- SAR: Speicheradreibregister
- SDR: Speicherdatenregister

**Steuersignale vom SW**

– für den internen Datenbus

- $A_r$ : Akku liest
- $A_w$ : Akku schreibt
- x: X-Register liest
- y: Y-Register liest
- z: Z-Register schreibt
- E: Eins-Register schreibt
- $P_r$ : IAR liest
- $P_w$ : IAR schreibt
- $I_r$ : IR liest
- $I_w$ : IR schreibt
- $D_r$ : SDR liest
- $D_w$ : SDR schreibt
- s: SAR liest

– für die ALU

$c_2-c_0$ : Operation auswählen

– für den Speicher

- R: Leseanforderung
- w: Schreibanforderung

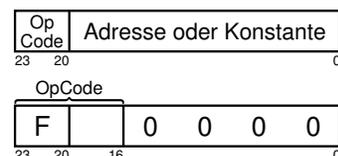
**Meldesignale zum SW**

- T: Takteingang
- N: Vorzeichen des Akku
- $B_{23}-B_{16}$ : OpCode-Feld im IR

$c_2c_1c_0$	ALU Operation
0 0 0	tue nichts ( d.h. $Z \rightarrow Z$ )
0 0 1	$X + Y \rightarrow Z$
0 1 0	rotiere X nach rechts $\rightarrow Z$
0 1 1	$X \text{ AND } Y \rightarrow Z$
1 0 0	$X \text{ OR } Y \rightarrow Z$
1 0 1	$X \text{ XOR } Y \rightarrow Z$
1 1 0	Eins-Komplement von X $\rightarrow Z$
1 1 1	falls $X = Y$ , $-1 \rightarrow Z$ , sonst $0 \rightarrow Z$

OpCode	Mnemonic	Beschreibung
0	LDC c	$c \rightarrow$ Akku
1	LDV a	$\langle a \rangle \rightarrow$ Akku
2	STV a	Akku $\rightarrow \langle a \rangle$
3	ADD a	Akku + $\langle a \rangle \rightarrow$ Akku
4	AND a	Akku AND $\langle a \rangle \rightarrow$ Akku
5	OR a	Akku OR $\langle a \rangle \rightarrow$ Akku
6	XOR a	Akku XOR $\langle a \rangle \rightarrow$ Akku
7	EQL a	falls Akku = $\langle a \rangle$ : $-1 \rightarrow$ Akku sonst: $0 \rightarrow$ Akku
8	JMP a	$a \rightarrow$ IAR
9	JMN a	falls Akku < 0 : $a \rightarrow$ IAR
F0	HALT	stoppt die MIMA
F1	NOT	bilde Eins-Komplement von Akku $\rightarrow$ Akku
F2	RAR	rotiere Akku eins nach rechts $\rightarrow$ Akku

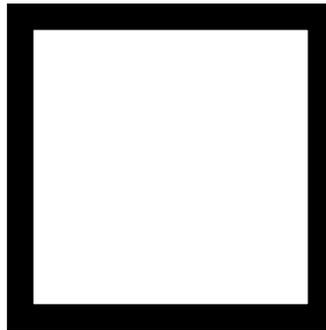
**Befehlsformate**



Vorlesung Rechnerorganisation Wintersemester 2017/18

- Übungsblatt 3 -

Tutoriumsnummer



Name, Vorname: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Studiengang: \_\_\_\_\_

Name des Tutors: \_\_\_\_\_

**/27 Punkte**