

Rechnerorganisation im WS 2020/21

2. Übungsblatt

Abgabetermin: 30. November, 13:15 Uhr

Prof. Dr. Jörg Henkel
Dr.-Ing. Lars Bauer
Roman Lehmann, M. Sc.
Haid-und-Neu-Str. 7,
Geb. 07.21 (Technologiefabrik)
Email: roman.lehmann@kit.edu

Aufgabe 1

(4 Punkte)

1. Was besagt das Moore'sche Gesetz der Mikroelektronik?
2. Aus welchen Komponenten besteht ein Mikroprozessor?
3. Was versteht man unter einem Ein-Chip-Mikrocomputer?
4. Was bedeutet „Ein Prozessor ist ein 64-Bit-Prozessor“?

1 P.

1 P.

1 P.

1 P.

Aufgabe 2

(4 Punkte)

Beantworten Sie folgende Fragen zur Funktionsweise einer arithmetisch-logischen Einheit (*Arithmetic Logic Unit*, ALU):

1. In welchem Register legen die meisten ALUs Informationen über das Ergebnis der letzten Operation ab? Nennen und erläutern Sie drei Flags, die auf einer gängigen Prozessorarchitektur aus diesem Register ausgelesen werden können.
2. Viele ALUs bieten als Operation sowohl logisches als auch arithmetisches Rechtsschieben an. Erklären Sie den Unterschied zwischen beiden Operationen. Welcher mathematischen Operation entspricht das arithmetische Rechtsschieben.

2 P.

2 P.

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

(7 Punkte)

Die MIMA sei eine mikroprogrammierte Minimalmaschine, die nach dem Von-Neumann-Prinzip aufgebaut ist, d. h. Maschinenbefehle werden sequentiell abgearbeitet (siehe Tutoriumsblatt: Architektur der MIMA). In der Lese-Phase (*fetch phase*) wird ein über IAR adressierter Befehl aus dem Speicher gelesen und im IR abgelegt. In der anschließenden Ausführungsphase (*execute phase*) wird der Befehl dekodiert und ausgeführt. Dann folgt ein Zugriff auf den nächsten Befehl.

Die Befehlsbearbeitung wird vom Steuerwerk SW gesteuert, welches über 10 Meldesignale als Eingänge und 18 Steuersignale als Ausgänge verfügt. Außerdem enthält das SW zumindest einen Mikroprogrammspeicher für maximal 256 Mikrobefehle und ein Register für den ausgelesenen Mikrobefehl. Nehmen Sie an, dass ein Hauptspeicherzugriff (Lesen und Schreiben) drei Takte dauert und währenddessen $R = 1$ (bzw. $W = 1$) sein muss. Eine ALU-Operation sei nach einem Takt abgeschlossen.

Das Mikroprogramm für die Lese-Phase (*fetch phase*) besteht aus fünf Mikrobefehlen:

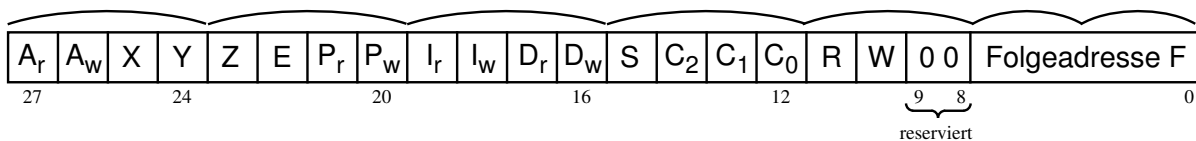
- | | |
|--|---------------|
| 1. Takt: $IAR \rightarrow SAR;$ $IAR \rightarrow X;$ $R = 1$ | } Fetch-Phase |
| 2. Takt: $Eins \rightarrow Y;$ $R = 1$ | |
| 3. Takt: ALU auf Addieren; $R = 1$ | |
| 4. Takt: $Z \rightarrow IAR$ | |
| 5. Takt: $SDR \rightarrow IR$ | |

1. Erklären Sie, was innerhalb dieser 5 Takte passiert.

2 P.

2. Kodieren Sie das oben angegebene Mikroprogramm für die Lese-Phase (*fetch phase*). Das Mikroprogramm soll bei der 8-Bit-Adresse 0x00 beginnen. Verwenden Sie das folgende 28-Bit-Mikrobefehlsformat:

5 P.



Beispiel: 0x77:

7	0	0	0	0	7	9
---	---	---	---	---	---	---

 A_w = X = Y = 1 (Akku -> X; Akku -> Y)
 0x78: Adresse des nächsten Befehls ist 0x79
 0x79:

Aufgabe 5

(6 Punkte)

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

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

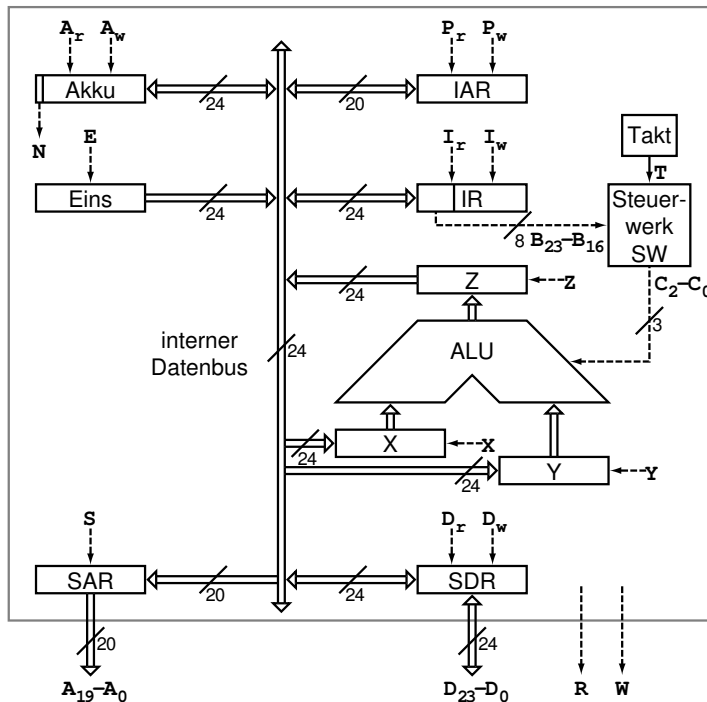
1. 7. Takt: IR \rightarrow SAR; R = 1
 8. Takt: Akku \rightarrow X; R = 1
 9. Takt: R = 1
 10. Takt SDR \rightarrow Y
 11. Takt ALU auf 111
 12. Takt Z \rightarrow Akku
2. 7. Takt: IR \rightarrow IAR
3. 7. Takt: IR \rightarrow SAR; R = 1
 8. Takt: R = 1
 9. Takt: R = 1
 10. Takt: SDR \rightarrow Akku
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 6

(2 Punkte)

Füllen Sie folgende Tabelle aus. Richtige Antworten werden mit 0.5 Punkten bewertet, falsche mit −0.5 Punkten. Nicht ausgefüllte Felder werden nicht bewertet.

	wahr	falsch
Alle Maschinenbefehle der MIMA-Architektur haben die Länge von einem Datenwort, sind also 24 Bit lang.		
Die Lese- und Dekodierphase des Befehlszyklus laufen unabhängig vom auszuführenden Befehl immer identisch ab.		
Um -1 als Konstante in das Akku-Register zu laden, kann der LDC-Befehl (load constant) der MIMA-Architektur verwendet werden.		
Die MIMA-Architektur kann 2^{24} Speicherzellen adressieren, da das Speicher-AdressRegister (SAR) eine Breite von 24 Bit besitzt		

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

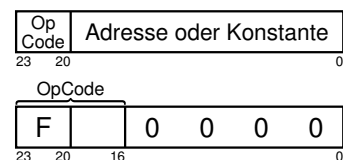
R : Leseanforderung
 W : Schreibanforderung

Meldesignale zum SW

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

$c_2 c_1 c_0$	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 $0 \rightarrow Z$

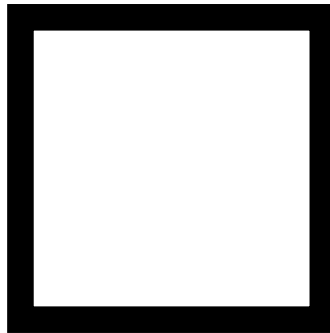
OpCode	Mnemonic	Beschreibung
0	LDC c	$c \rightarrow \text{Akku}$
1	LDV a	$\langle a \rangle \rightarrow \text{Akku}$
2	STV a	$\text{Akku} \rightarrow \langle a \rangle$
3	ADD a	$\text{Akku} + \langle a \rangle \rightarrow \text{Akku}$
4	AND a	$\text{Akku AND } \langle a \rangle \rightarrow \text{Akku}$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

Befehlsformate

Vorlesung Rechnerorganisation Wintersemester 2020/21

- Übungsblatt 2 -

Tutoriumsnummer



Name, Vorname: _____

Matrikelnummer: _____

Studiengang: _____

Name des Tutors: _____

/27 Punkte