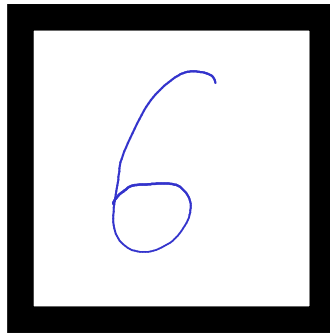


- Übungsblatt 2 -

Tutoriumsnummer



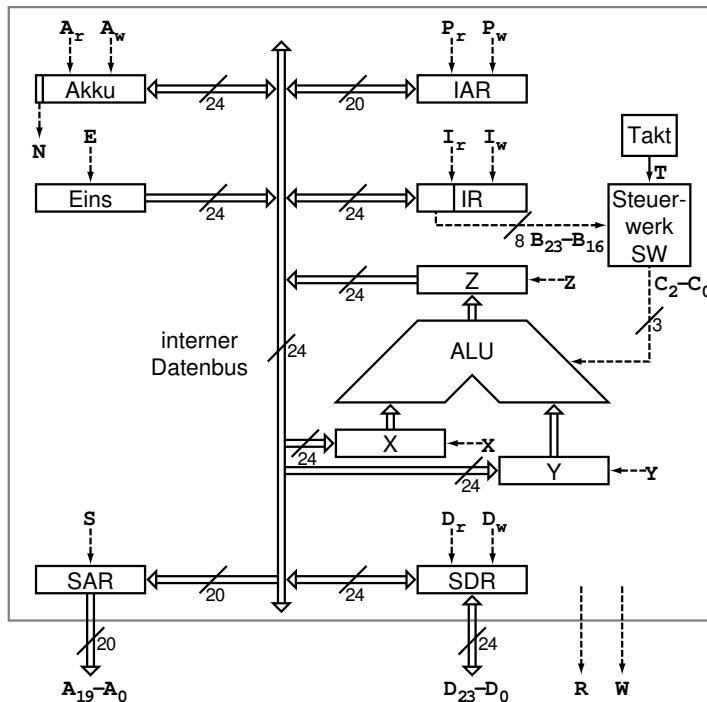
Name, Vorname: Slavov, Velislav

Matrikelnummer: 2385786

Studiengang: Informatik BSc

Name des Tutors: Jonas Heinle

/27 Punkte

Architektur der MIMA**Register**

Akku: Akkumulator
 X: 1. ALU Operand
 Y: 2. ALU Operand
 Z: ALU Ergebnis
 Eins: Konstante 1
 IAR: Instruktionsadreßregister
 IR: Instruktionsregister
 SAR: Speicheradreßregister
 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

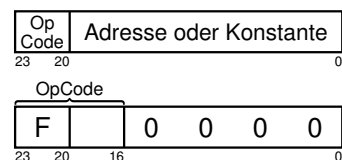
R : Leseanforderung
 W : Schreibanforderung

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	Mnemonik	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

A1)

1. Dass CPUs ungefähr jede 2 Jahren doppel so mächtig werden. (doppelt so viele Instruktionen wie vorher gleichzeitig ausführen können)
2. Registersatz, Steuerwerk, Rechenwerk, Adresswerk, Verbindungsnetzwerk (interner Datenbus)
3. Einer integrierter Schaltkreis voraus die CPU, Speicher und IO alle verfügbar sind.
4. Die Wortgröße ist 64 bit. D.h. Daten und Befehle sind 64 bit groß.

A2)

1. Den Statusregister
 - ZF = Nullbit : ob das Ergebnis Null ist
 - SF = Vorzeichenbit : ob das Ergebnis positiv oder negativ ist
 - EF = Even Flag : ob das Ergebnis gerade oder ungerade ist
 2. Beim logischen Rechtsschieben sind die neue Bits auf 0 gesetzt, beim arithmetischen - auf das Vorzeichen (MSB).
- Arithmetisches Rechtsschieben = division durch 2^n

A 3) Sie berechnet den Zweierkomplement

$$\begin{aligned}LDV\ 0x2014 &= \boxed{1}\ \boxed{0x2014} \\&= \boxed{0001}\ \boxed{0000}\ \boxed{0010}\ \boxed{0000}\ \boxed{0001}\ \boxed{0100}\end{aligned}$$

$$\begin{aligned}NOT &= \boxed{F1}\ \boxed{0x0000} \\&= \boxed{1111\ 0001}\ \boxed{0000\ 0000\ 0000\ 0000}\end{aligned}$$

$$\begin{aligned}STV\ 0x2015 &= \boxed{2}\ \boxed{0x2015} \\&= \boxed{0010}\ \boxed{0000}\ \boxed{0010}\ \boxed{0000}\ \boxed{0001}\ \boxed{0101}\end{aligned}$$

A 4.1)

Takt 1: Read bit wird gesetzt (CPU in lese Modus). Die Adresse der nächsten Instruktion wird gleichzeitig zum auslesen vorbereitet und an der ALU übergeben.

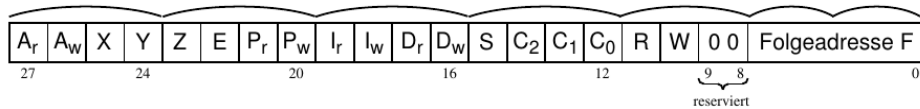
Takt 2: Der Wert 1 wird ebenso an der ALU übergeben

Takt 3: Die ALU wird auf "addieren" umgestellt und das Ergebnis wird im z-Register gespeichert

Takt 4: Die inkrementierte Adresse von Takt 3 wird nach dem IAR transportiert

Takt 5: Die Instruktion auf der Adresse von Takt 1 wird aus dem Hauptspeicher in dem Instruktionsregister gespeichert

A4.2)



Takt 1: 0010 0001 0060 1000 1000 0000 0001

Takt 2: 0001 0100 0000 0000 1000 0000 0010

Takt 3: 0000 0000 0000 0001 1000 0000 0011

Takt 4: 0000 1010 0000 0000 0000 0000 0100

Takt 5: 0000 0000 1001 0000 0000 0000 0101

A5)

1. EQL a
2. JMP a
3. LDV a
4. LDC a
5. STV a

A6)

1. F
2. W
3. F
4. F