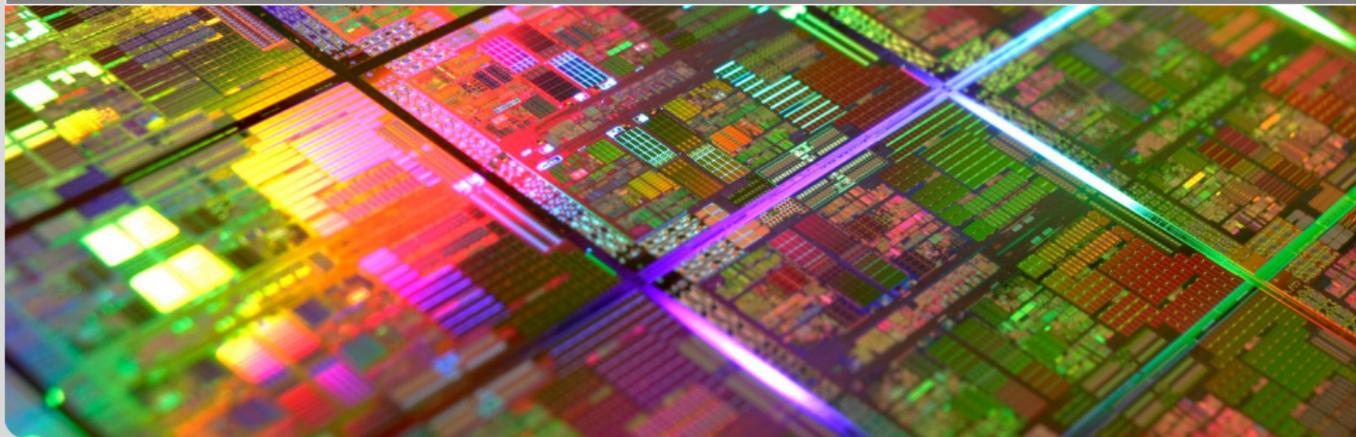


# RO-Tutorien 3 / 6 / 12

Tutorien zur Vorlesung "Rechnerorganisation"

Christian A. Mandery

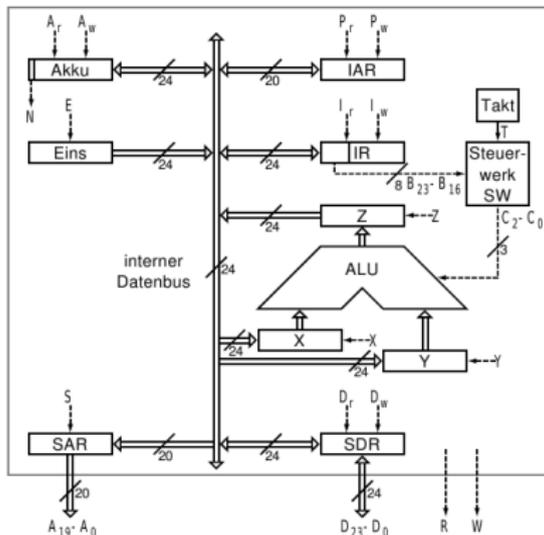
WOCHE 5 AM 27./28.05.2013



- Wiederholung der MIMA-Architektur
- Übungsaufgaben

# Die Architektur der MIMA

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<sub>r</sub>: Akku liest
- A<sub>w</sub>: Akku schreibt
- X: X-Register liest
- Y: Y-Register liest
- Z: Z-Register schreibt
- E: Eins-Register schreibt
- P<sub>r</sub>: IAR liest
- P<sub>w</sub>: IAR schreibt
- I<sub>r</sub>: IR liest
- I<sub>w</sub>: IR schreibt
- D<sub>r</sub>: SDR liest
- D<sub>w</sub>: SDR schreibt
- S: SAR liest

– für die ALU

C<sub>2</sub>, C<sub>0</sub>: Operation auswählen

– für den Speicher

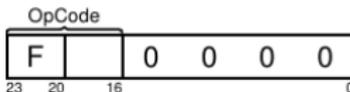
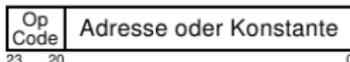
- R: Leseanforderung
- W: Schreib Anforderung

## Meldesignale zum SW

- T: Takteingang
- N: Vorzeichen des Akku
- B<sub>23</sub>, B<sub>16</sub>: OpCode-Feld im IR

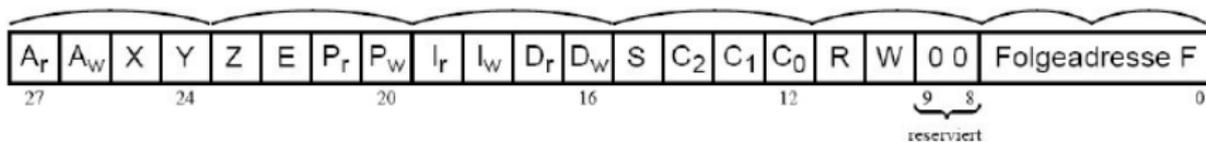
<i>OpCode</i>	<i>Mnemonic</i>	<i>Beschreibung</i>
0	LDC c	c -> Akku
1	LDV a	<a> -> Akku
2	STV a	Akku -> <a>
3	ADD a	Akku + <a> -> Akku
4	AND a	Akku AND <a> -> Akku
5	OR a	Akku OR <a> -> Akku
6	XOR a	Akku XOR <a> -> Akku
7	EQL a	falls Akku = <a>: -1 -> Akku sonst: 0 -> Akku
8	JMP a	a -> IAR
9	JMN a	falls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

## Befehlsformate



$C_2 C_1 C_0$	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 $\rightarrow Z$

# Mikrobefehlsformat der MIMA



# Übungsaufgabe 1.1: JMS-Befehl

- Jump Subroutine: Neuer Befehl mit dem Opcode C
- JMS a:  $IAR + 1 \rightarrow \langle a \rangle$ ;  $a + 1 \rightarrow IAR$
- Schreibt die Folgeadresse der aktuellen Adresse an die Speicherstelle a und springt zur Adresse, die auf a folgt
- Wie sieht das Mikroprogramm für die Ausführungsphase aus?

# Übungsaufgabe 1.1: JMS-Befehl

- Jump Subroutine: Neuer Befehl mit dem Opcode C
- JMS a:  $IAR + 1 \rightarrow \langle a \rangle$ ;  $a + 1 \rightarrow IAR$
- Schreibt die Folgeadresse der aktuellen Adresse an die Speicherstelle a und springt zur Adresse, die auf a folgt
- Wie sieht das Mikroprogramm für die Ausführungsphase aus?

# Übungsaufgabe 1.2: JIND-Befehl

- Jump Indirect: Neuer Befehl mit dem Opcode D
- JIND a:  $\langle a \rangle \rightarrow \text{IAR}$
- Springt zu der Adresse, die in den 20 niederwertigen Bits der Speicherstelle steht, die durch a referenziert wird
- Wie sieht das Mikroprogramm für die Ausführungsphase aus?

# Übungsaufgabe 1.2: JIND-Befehl

- Jump Indirect: Neuer Befehl mit dem Opcode D
- JIND a:  $\langle a \rangle \rightarrow \text{IAR}$
- Springt zu der Adresse, die in den 20 niederwertigen Bits der Speicherstelle steht, die durch a referenziert wird
- Wie sieht das Mikroprogramm für die Ausführungsphase aus?

- Aufgabe: Mit einem MIMA-Programm zu einem beliebigen Datum in 2004 den Wochentag (0 = Sonntag, 1 = Montag, ...) berechnen
- Vorgegebene Speicherstellen:
  - 0x00020: Tag (Eingabe)
  - 0x00021: Monat (Eingabe)
  - 0x00030: Wochentag (Ausgabe)
  - 0x00100: Startpunkt des Programms

- Algorithmus (in C):

```
int ersterTag[] = {-1, 4, 0, 1, 4, 6, 2, 4, 0, 3, 5, 1, 3};  
int wochentag = TAG + ersterTag[MONAT] - 1;  
wochentag %= 7;
```

AN x64 PROCESSOR IS SCREAMING ALONG AT BILLIONS OF CYCLES PER SECOND TO RUN THE XNU KERNEL, WHICH IS FRANTICALLY WORKING THROUGH ALL THE POSIX-SPECIFIED ABSTRACTION TO CREATE THE DARWIN SYSTEM UNDERLYING OS X, WHICH IN TURN IS STRAINING ITSELF TO RUN FIREFOX AND ITS GECKO RENDERER, WHICH CREATES A FLASH OBJECT WHICH RENDERS DOZENS OF VIDEO FRAMES EVERY SECOND

BECAUSE I WANTED TO SEE A CAT  
JUMP INTO A BOX AND FALL OVER.



I AM A GOD.

Quelle: <http://xkcd.com/676/>