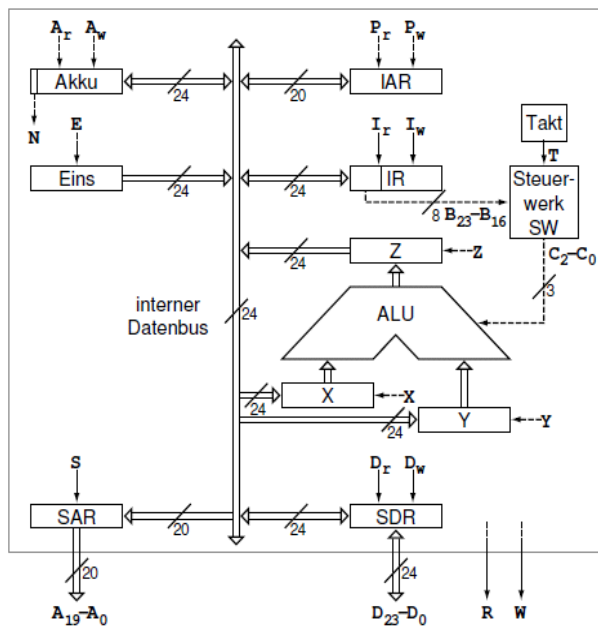


MIMA

Architektur der MIMA



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

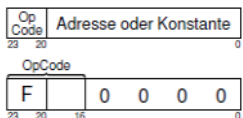
– für den Speicher

- R : Leseanforderung
- W : Schreibenanforderung

Meldesignale zum SW

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

Befehlsformate



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	$\text{Akku} \text{ AND } \langle a \rangle \rightarrow \text{Akku}$
5	OR a	$\text{Akku} \text{ OR } \langle a \rangle \rightarrow \text{Akku}$
6	XOR a	$\text{Akku} \text{ XOR } \langle a \rangle \rightarrow \text{Akku}$
7	EQL a	if ($\text{Akku} == \langle a \rangle$) { -1 \rightarrow Akku; } else { 0 \rightarrow Akku; }
8	JMP a	Sprung zur Adresse a
9	JMN a	wenn $\text{Akku} < 0$, dann Sprung nach a
A	LDIV a	$\langle \langle a \rangle \rangle \rightarrow \text{Akku}$
B	STIV a	$\text{Akku} \rightarrow \langle \langle a \rangle \rangle$
C	JMS a	jump subroutine (s.u.)
D	JIND a	jump indirect (s.u.)
E		frei
F0	HALT	stoppt die MIMA
F1	NOT	Einskomplement von Akku \rightarrow Akku
F2	RAR	rotiere Akku eins nach rechts \rightarrow Akku (s.u.)
F3 - FF		frei

Anmerkungen:

- Was beim **RAR** Befehl rechts rausgeschoben wird, wird links wieder eingeführt.
- Der Befehl **JMS ziel** bewirkt, daß die Adresse des nachfolgenden Befehls, also die Rücksprungadresse, an der Speicheradresse **ziel** abgelegt und dann zum Befehl mit der Adresse **ziel+1** verzweigt wird.
- Der Befehl **JIND ziel** bewirkt, dass zu derjenigen Adresse verzweigt wird, die in der Speicherzelle **ziel** abgespeichert ist.

Register

Akku: Akkumulator

X: 1. ALU Operand Y: 2. ALU Operand Z: ALU Ergebnis

Eins:Konstante 1

IAR: Instruktionsadreßregister

Speichert die Adresse des aktuell auszuführenden Befehls zur Beginn der Lese-Phase (fetch phase). Nach der Lese-Phase enthält es die Adresse des im nächsten Zyklus auszuführenden Maschinenbefehls.

IR: Instruktionsregister

Speichert das aktuell auszuführende Befehlswort.

SAR: Speicheradreßregister

Enthält die Speicheradresse auf die bei der nächsten Lese- (R = 1)/Schreibenanfrage (W = 1) zugegriffen wird.

SDR: Speicherdatenregister

Enthält bei einer Schreiboperation (W = 1) das zu schreibenden Datenwort. Nach einer Leseoperation (R = 1) enthält es das gelesene Datenwort.

FETCH-PHASE:

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

LDC:

7. Takt: IR \rightarrow Akku

LDV:

7. Takt: IR \rightarrow SAR; R = 1
8. Takt: R = 1
9. Takt: R = 1
10. Takt: SDR \rightarrow Akku

STV:

7. Takt: Akku \rightarrow SDR
8. Takt: IR \rightarrow SAR; W = 1
9. Takt: W = 1
10. Takt: W = 1

AND:

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 AND
12. Takt: Z \rightarrow Akku

EQL:

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 Vergleich
12. Takt: Z \rightarrow Akku

JMP:

7. Takt: IR \rightarrow IAR

JMN:

7. Takt: B = 1
8. Takt: IR \rightarrow IAR

HALT:

7. Takt: Endlose Schleife:
- Folgebefehl = aktueller Befehl

LDIV:

7. Takt: IR ! SAR; R = 1
8. Takt: R = 1
9. Takt: R = 1
10. Takt: SDR ! SAR; R = 1
11. Takt: R = 1
12. Takt: R = 1
13. Takt: SDR ! Akku

STIV:

7. Takt: IR ! SAR; R = 1
8. Takt: R = 1
9. Takt: R = 1
10. Takt: SDR ! SAR
11. Takt: Akku ! SDR; W = 1
12. Takt: W = 1
13. Takt: W = 1