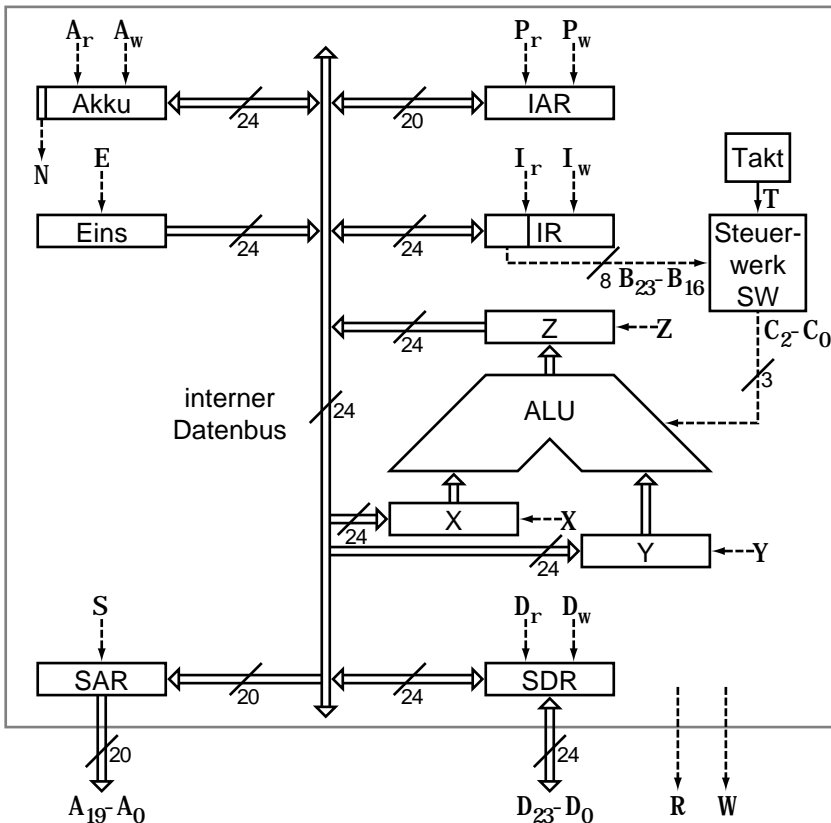


## Architektur der MIMA



$C_2 C_1 C_0$	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	$c \rightarrow$ Akku
1	LDV a	$\langle a \rangle \rightarrow$ Akku
2	STV a	Акку $\rightarrow \langle a \rangle$
3	ADD a	Акку + $\langle a \rangle \rightarrow$ Akku
4	AND a	Акку AND $\langle a \rangle \rightarrow$ Akku
5	OR a	Акку OR $\langle a \rangle \rightarrow$ Akku
6	XOR a	Акку 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

## 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: Schreib Anforderung

## Meldesignale zum SW

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

## Befehlsformate

