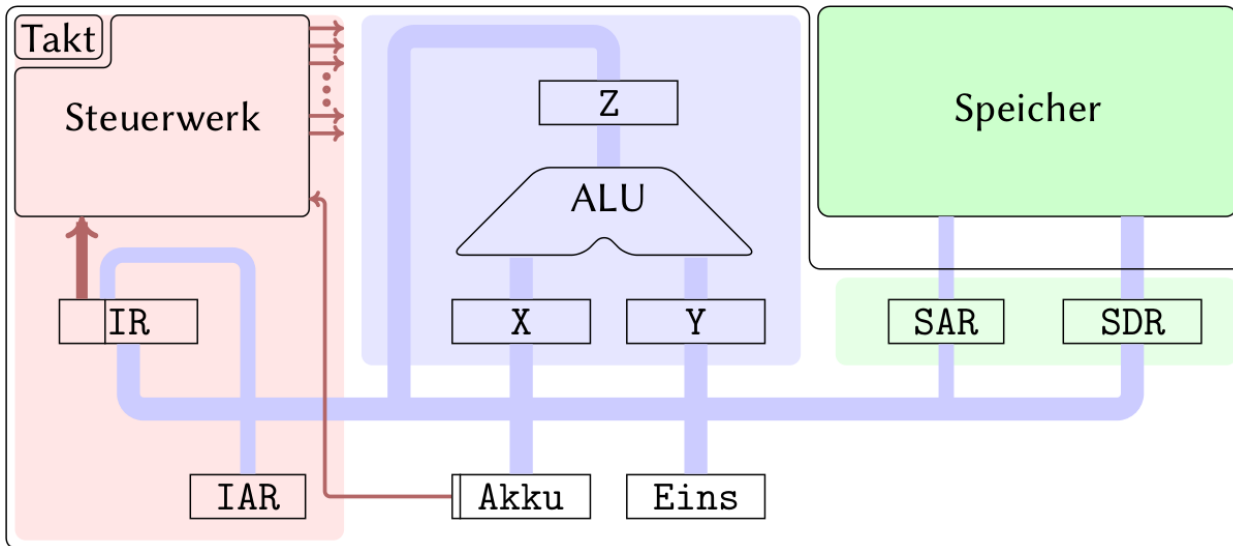


Die MIMA

Aufbau:



Register

IR	Adresse des aktuellen Befehls
IAR	Der aktuelle Befehl
SAR	Adresse für Speicherzugriff
SDR	Wert von/für Speicherzugriff
X, Y	Input für die ALU
Z	Output der ALU
Akku	Zwischenspeicher für Daten
Eins	1-Wert (readonly)

Sonstige Bauteile

Takt	Steuert die Ausführungsrounden
Steuerwerk	Öffnet/Schließt Register und verwaltet Operationen von ALU und Speicher
ALU	Führt Rechenoperationen aus
Speicher	Speicher für Befehle und Daten

Befehle (Auswahl):

Rechenoperationen:

(ERGEBNISSE EINER RECHENOPERATION LANDEN STETS IM AKKU)

- ADD *addr*** : Addiert den Wert an Adresse *addr* zum Akku
- AND *addr*** : Bildet das binäre UND zwischen dem Wert an Adresse *addr* und dem Akku
- OR *addr*** : Bildet das binäre ODER zwischen dem Wert an Adresse *addr* und dem Akku
- NOT** : Invertiert den Wert im Akku ($0 \rightarrow -1, -1 \rightarrow 0$)
- EQL *addr*** : Speichert -1 im Akku, wenn der Wert an Adresse *addr* gleich dem Akku ist. Ansonsten 0.

Zugriffsoperationen:

- LDC *val*** : Lädt den Wert *val* in den Akku
- LDV *addr*** : Lädt den Wert an Adresse *addr* in den Akku
- STV *addr*** : Speichert den Wert im Akku an der Adresse *addr*
- LDIV *addr*** : Lädt den Wert an der Adresse, die an Adresse *addr* steht in den Akku
- STIV *addr*** : Speichert den Wert im Akku an die Adresse, die an Adresse *addr* steht.

Sprünge:

- JMP *val*** : Springt zum Befehl *val* (= Setzt IR auf *val*)
- JMN *val*** : Springt zum Befehl *val*, falls der Akku < 0 ist