

Rechneraufbau und Rechnerstrukturen

1 Grundmodell der von Neumann Architektur

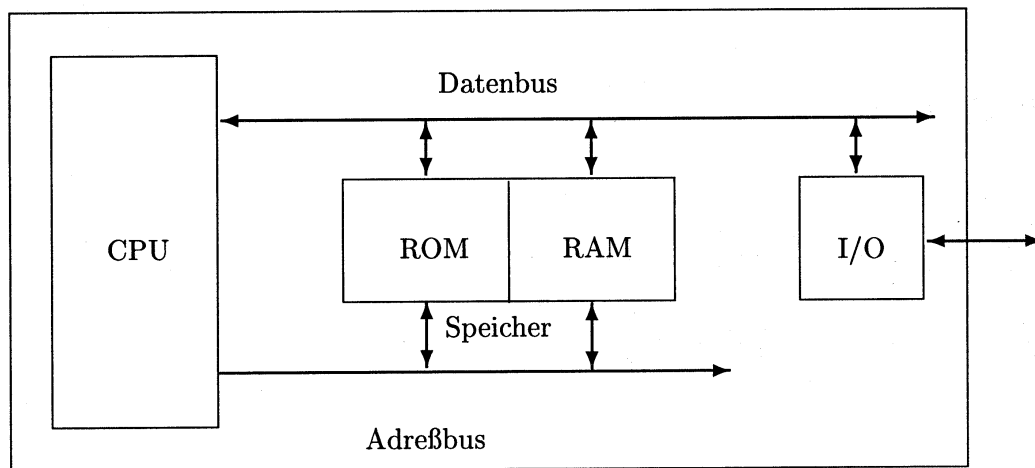


Abbildung 1: Struktur eines von Neumann-Rechners

2 Aufbau des Prozessors

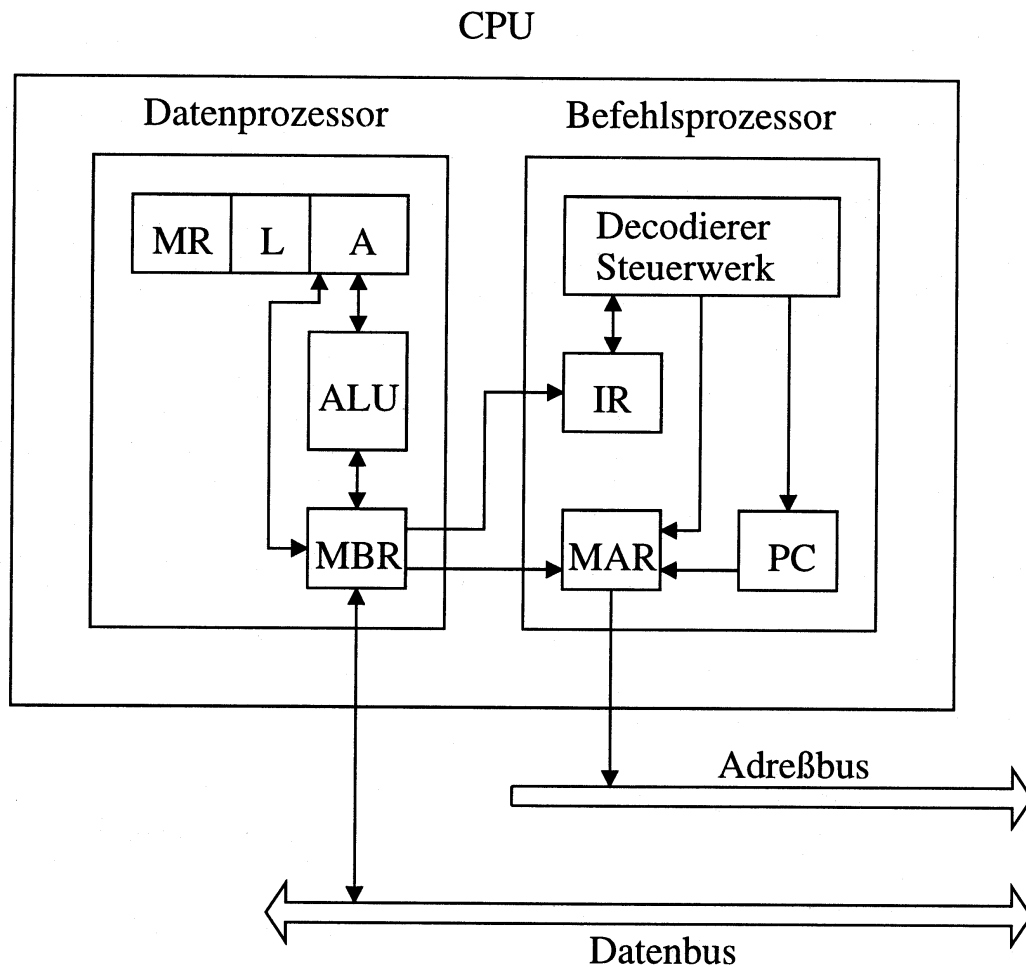


Abbildung 2: Struktur einer CPU

3 Register

MAR	Memory Adress Register	Speicheradress-Register für die Adresse des nächsten anzusprechenden Speicherplatz
MBR	Memory Buffer Register	Puffer-Register für die Kommunikation mit dem Speicher
A	Akkumulator	General Purpose Register für alle anfallenden Aufgaben
MR	Multiplier Register	Multiplikator (Aufnahme von Multiplikationsergebnissen)
L	Link Register	Einstelliges Register für Überträge und Ähnliches
IR	Instruction Register	Befehlsregister, enthält den aktuellen Befehl
PC	Program Counter	Befehlszähler, enthält die Adresse des nächsten Befehls

- Der Akkumulator A enthält die Register MR und L, d.h. MR und L sind Erweiterungen des Akkus. Aus diesem Grund zerstört ein Schreiben auf MR den Inhalt von L und A (wichtig für Multiplikation, s.u.).
- Die Register A (und damit auch MR und L) und MBR sind Teil des Datenprozessors, ebenso das Rechenwerk ALU (Arithmetic Logical Unit).

- Die Register MAR, IR und PC sind Teil des Befehlsprozessors, ebenso wie der Dekodierer zum Entschlüsseln der Befehle und das Steuerwerk zur Steuerung der Ausführung.

4 Mikro-Instruktionen

MAR	\leftarrow PC	Inhalt von PC (Adresse) wird nach MAR gebracht
MBR	\leftarrow <MAR>	Inhalt der Speicheradresse, die im MAR steht, wird nach MBR gebracht
MBR	\leftarrow A	Inhalt des Akkumulators in das MBR laden
<MAR>	\leftarrow MBR	Inhalt von MBR auf die Adresse speichern, auf die MAR zeigt
IR	\leftarrow MBR	Instruction Register erhält Befehl aus MBR
PC	\leftarrow PC + n	PC um n erhöhen
PC	\leftarrow MBR	PC auf Adresse setzen, die im MBR steht
MAR	\leftarrow MAR+n	Adresse in MAR auf Inhalt + n setzen
MBR	\leftarrow MAR	Inhalt des MAR (Adresse) in das MBR bringen
PC	\leftarrow MAR	PC auf Adresse im MAR setzen
A	\leftarrow A + MBR	Berechnung im Akkumulator, Addition
A	\leftarrow -A	unäres Minus
A	\leftarrow \neg A	logische Negation
MR	\leftarrow A * MBR	Multiplikation
A	\leftarrow MBR	Konstante in Akku laden
A	\leftarrow A < 0	logischer Vergleich mit Konstanten, Ergebnis im Akku
MAR	\leftarrow MBR	Umladen von Adressen

Beim Maschinencode werden die tatsächlichen Register der CPU angegeben.

5 Pseudo-Assembler

LOAD <i>x</i>	Lade den Inhalt von Adresse <i>x</i> in den Akkumulator
STORE <i>x</i>	Speichere den Inhalt des Akkumulators in der Speicherzelle mit der Adresse <i>x</i>
ADD <i>x</i>	Addiere den Wert an Adresse <i>x</i> zum Inhalt des Akkumulators
SUB <i>x</i>	Subtrahiere den Wert von Adresse <i>x</i> vom Inhalt des Akkumulators
JMPNEG <i>x</i>	Springe zur Marke <i>x</i> , wenn der Inhalt des Akkumulators < 0
JMPEQ <i>x</i>	Springe zur Marke <i>x</i> , wenn der Inhalt des Akkumulators = 0
JMP <i>x</i>	Springe zur Marke <i>x</i>
NOT	logische Negation - bitweises Komplement des Wertes im Akkumulator
NEG	unäres Minus angewendet auf den Wert im Akkumulator.
HALT	Beendet das Programm