

Funktionsweise eines Rechners

Aufbau/Bestandteile eines Rechners

Aufgabe 1:

Nimm dir ein Blatt in DIN-A4-Größe im Querformat her und entwerfe eine Mindmap mit den Bestandteilen die in einem Rechner vorhanden sind. Peripheriegeräte (Tastatur, Maus, Bildschirm, ...) sollen nicht in die Mindmap aufgenommen werden.

Historische Entwicklung – von-Neumann-Rechner

→ Siehe Präsentation zu **Historische Entwicklung – von-Neumann-Rechner**

Rechnermodell nach den von-Neumann-Prinzipien

1. Rechner besteht aus 4 Werken: Rechenwerk, Steuerwerk, Speicherwerk (Arbeitsspeicher), Ein-/Ausgabewerk
2. Programmgesteuert: ein Programm ist eine Folge von Verarbeitungsvorschriften; sie müssen von außen eingegeben und im Speicher abgelegt werden.
3. Programme und Daten (Eingabedaten, Zwischenergebnisse und Endergebnisse) liegen im selben Speicher.
4. Hauptspeicher in Zellen gleicher Größe aufgeteilt, die fortlaufend nummeriert sind und deren Inhalt über die Nummer (Adresse) abgerufen/verändert werden kann.
5. Aufeinanderfolgende Befehle eines Programms werden in aufeinanderfolgenden Speicherzellen abgelegt und sequentiell ausführt.
6. Von der sequentiellen Ausführung der Befehle kann durch Sprungbefehle abgewichen werden, die die Programmfortsetzung an einer anderen Stelle im Speicher fortsetzt.
7. Speicherelemente kennen zwei Schaltzustände: „ein“ und „aus“. Aus diesem Grund werden Daten, Befehle und Adressen binär codiert



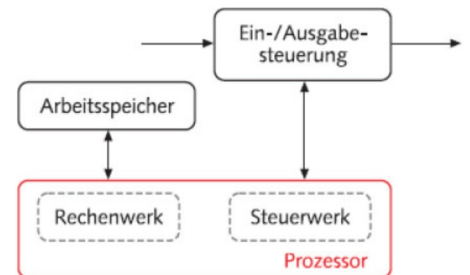
John von Neumann
1903 - 1957

Registmaschine nach Von-Neumann-Architektur

Die Bestandteile eines Rechners lassen sich im wesentlichen gliedern in Prozessor, Arbeitsspeicher und Peripheriegeräte; letztere sind über die Ein-/Ausgabesteuerung angebunden. Die Hauptkomponenten ohne Peripherie werden durch das von-Neumann-Modell einer Registmaschine beschrieben.

Die **Registmaschine** besteht aus folgenden Teilen:

- Programme, Daten, Eingabedaten, Zwischen- und Endergebnisse werden im selben Speicher, dem **Arbeitsspeicher**, abgelegt und können vom Rechenwerk verändert werden. Der Arbeitsspeicher ist in Zellen gleicher Größe aufgeteilt, die durchnummeriert sind. Dies ermöglicht den wahlfreien Zugriff (-> RAM).
- Das **Steuerwerk** interpretiert Programmbefehle; koordiniert das Rechenwerk, die Ein-/Ausgabesteuerung und den Arbeitsspeicher. Es ist Teil der CPU, regelt die Befehlsabfolge und den gesamten Datenfluss innerhalb des Prozessors.
- Das **Rechenwerk** ist Teil der CPU führt einfache arithmetische Rechenoperationen und logische Verknüpfungen durch.
- Die **Ein-/Ausgabesteuerung** kommuniziert mit den Peripheriegeräten und steuert die Ein- und Ausgabe von Daten.



Der wesentliche Fortschritt dieses Prinzips war:

Unterschiedliche Programme lassen sich auf der gleichen Maschine ohne Veränderung der Hardware realisieren.

Maschinennahe Programmierung – Assembler

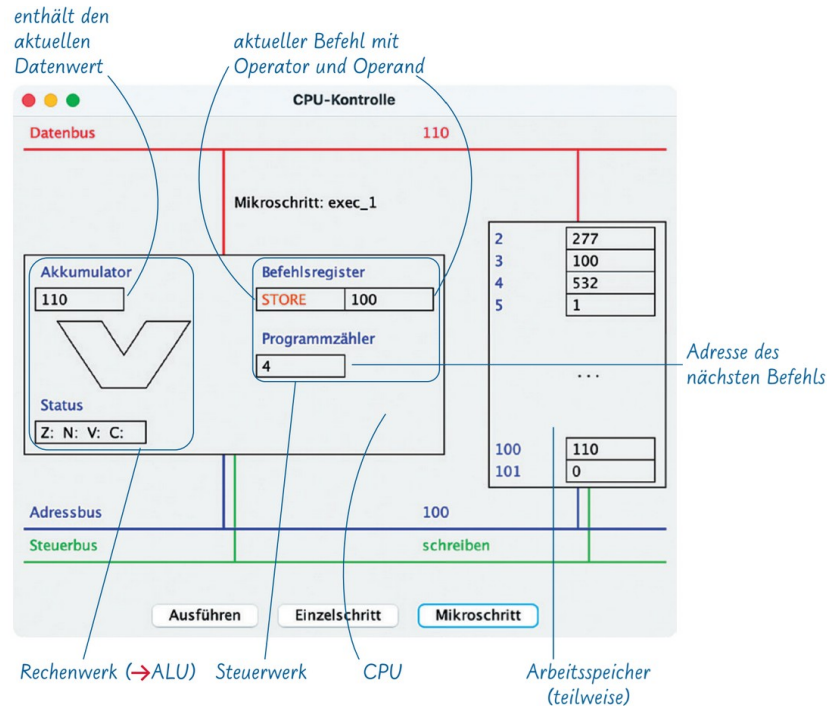
Eine Maschinensprache ist dazu entwickelt worden, um eine konkret vorliegende Hardware direkt ansprechen zu können.

Das nebenstehende Bild zeigt, dass z.B. das Betriebssystem die Hardware über Maschinenbefehle anspricht.

Anwendungsprogramme		
Systemprogramme		
Compiler	Interpreter	Editoren
Betriebssystem		
Maschinensprache		
Mikroprogrammierung		
Physikalische Geräte		

Höhere Programmiersprachen wie z.B. JAVA setzen dagegen nicht direkt auf der Maschinenebene auf sondern eine Abstraktionsebene weiter oben auf.

Die Minimaschine als von-Neumann-Rechner



- Der **Akkumulator** ist ein Register des Rechenwerks; er besteht aus einer Anzahl Speicherzellen (in der Minimaschine: 1 mit 16 Bit), in die vor Ausführung einer Operation die Operanden geladen werden. Nach der Ausführung eines Befehls landet der Wert wieder im Akkumulator zur Weiterverwendung oder zum Zurückschreiben in den Arbeitsspeicher.
- Das **Statusregister** gibt Aufschluss über bestimmte Eigenschaften des letzten Rechenergebnisses. In der Minimaschine sind dies **Zero-, Negativ- und Overflow-Flag** (jeweils binäre Werte). In der Simulation wird bei WAHR ein * angezeigt.
- In der **arithmetisch-logischen Einheit (ALU)**, dem zentralen Element des Rechenwerks, werden die Befehle – geregelt durch das Steuerwerk – ausgeführt. Bei der vorliegenden Simulation handelt es sich um eine Ein-Adressmaschine. Der Maschinenbefehl kann also höchstens eine Zelle im Arbeitsspeicher adressieren. Der gegebenenfalls – z. B. Bei der Summenbildung – erforderliche zweite Operand steht dann im Akkumulator.
- Der **Befehlszähler** (in der Simulation: Programmzähler) speichert die Adresse des nächsten auszuführenden Befehls. Der Inhalt wird nach der Ausführung eines Befehls um 1 erhöht. Bei **Sprungbefehlen** wird eine im Adressteil des Befehls stehende Adresse in den Befehlszähler kopiert.
- Das **Befehlsregister** enthält den Befehl, der gerade ausgeführt wird. Es besteht aus Operationskennung (Opcode, geladen aus einer Speicherzelle des Arbeitsspeichers) und aus dem Adressteil (folgende Speicherzelle). Falls kein Operand erforderlich ist, steht im Adressteil 0.

Programmieren mit der Minimaschine → siehe Präsentation **Assembler**