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.

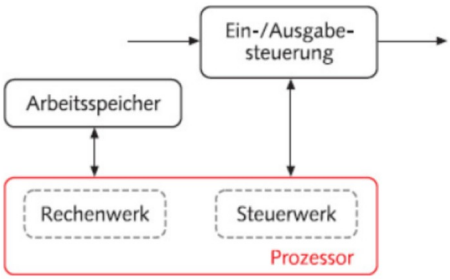
Historische Entwicklung – von-Neumann-Rechner

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

Auch in der Präsentation: ***Maschinenbefehlszyklus***

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 Registermaschine beschrieben.

Die ***Registermaschine*** besteht aus folgenden Teilen:



- Programme, Daten, 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 regelt die Befehlsabfolge.
- Das ***Rechenwerk*** 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.

Ein Programm besteht aus einer Folge von Befehlen, die im Allgemeinen nacheinander

ausgeführt werden und in aufeinanderfolgenden Speicherzellen abgelegt sind. Von der sequenziellen Abfolge der Befehlsausführung kann durch Sprungbefehle abgewichen werden. Die Speicherelemente kennen zwei Schaltzustände: ein oder aus. Daten, Befehle und Adressen sind also ***binär codiert***.

Grundlagen Rechnerarchitektur – Zahlen

→ siehe Präsentation Darstellungsweise binärer Zahlen

- Speicherung/Kodierung von positiven ganzen Zahlen
- Speicherung/Kodierung von ganzen Zahlen (auch negative Zahlen) – Vorzeichen-Bit, 1er-Komplement, 2er-Komplement
- Hexadezimalzahlen
- Speicherung/Kodierung von Dezimalzahlen (Kommazahlen) nach IEEE 754 single

Maschinennahe Programmierung – Assembler

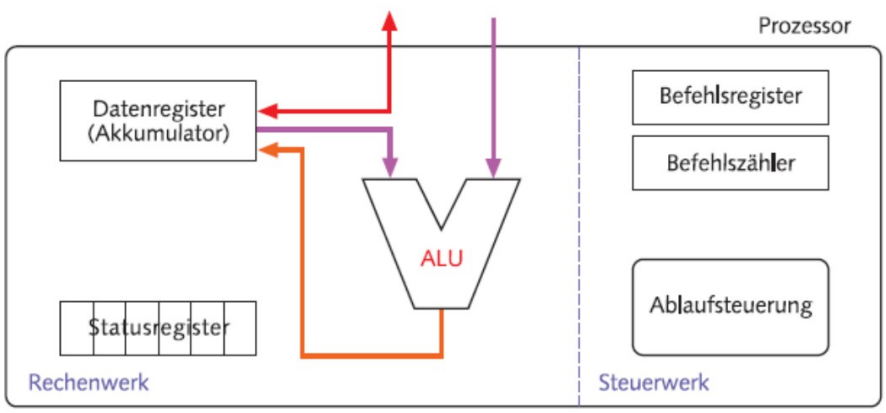
Eine Maschinensprache ist dazu entwickelt worden, um eine konkret vorliegende Hardware direkt ansprechen zu können.

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

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

Prozessor/Hardware genauer betrachtet

Wie beim obigen Kapitel aufgeführt, besteht der Prozessor bei der logischen Betrachtung nach der von-Neumann-Architektur aus dem Rechenwerk und Steuerwerk. Dies ist notwendig, da man jetzt ***die Hardware über Maschinenbefehle direkt ansprechen möchte***.



- 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.

Assembler programmieren mit der Minimaschine

→ siehe Präsentation Assembler