Das Rechnermodell von John von Neumann

Historisches

Die ersten mechanischen Rechenmaschinen wurden im 17. Jahhundert entworfen. Zu den Pionieren dieser Entwichlung zählen Wilhelm Schickard, Blaise Pascal und Gottfried Wilhelm Leibniz. Einen wichtigen Schritt in Richtung der modernen Computer gab es durch Charles Babbage, der Mitte des 19. Jahrhunderts das Konzept einer programmierbaren Rechenmaschine entwarf. Die eigentliche Entwicklung von Computern begann aber erst im 20. Jahrhundert. Es ist umstritten, welche der in den 30er und 40er Jahren gebauten Maschinen als erster Computer bezeichnet werden sollte, aber man kann mit Sicherheit sagen, dass die Z3 von Konrad Zuse über viele Eigenschaften eines Computers verfügte (elektrisch, nach binärlogischen Prinzipien aufgebaut und programmierbar) und im Jahr 1941 allen damaligen Konkurrenten konzeptionell überlegen war. Einer dieser Kokurrenten war der von Howard Aiken entwickelte Rechner Mark 1. Das war der erste funktionierende Großrechner, der aber noch dekadische Zahrräder zur Speicherung verwendete. Der erste rein elektronische digitale Großrechner war ein für die US-Armee entwickeltes Gerät zur Berechnung ballistischer Tabellen, das auf den Namen ENIAC (Electronic Numerical Integrator and Computer) getauft wurde. Hauptbestandteil dieses Rechners waren über siebzehntausend Elektronenröhren. Mit 174 kW Leistungsaufnahme und 27 Tonnen Gewicht war der ENIAC wahres Schwergewicht. Sein wesentlicher Nachteil bestand darin, dass er keinen Befehlsspeicher hatte und zur Programmierung neu verkabelt werden musste. Diese Nachteile konnten durch eine von John von Neumann erabeitete Konzeption mit dem Titel EDVAC (Electronic Discrete Variable Automatic Computer) behoben werden. Das Konzept wurde gemeinsam mit Presper Eckert und John Mauchly aus der ENIAC-Gruppe realisiert.

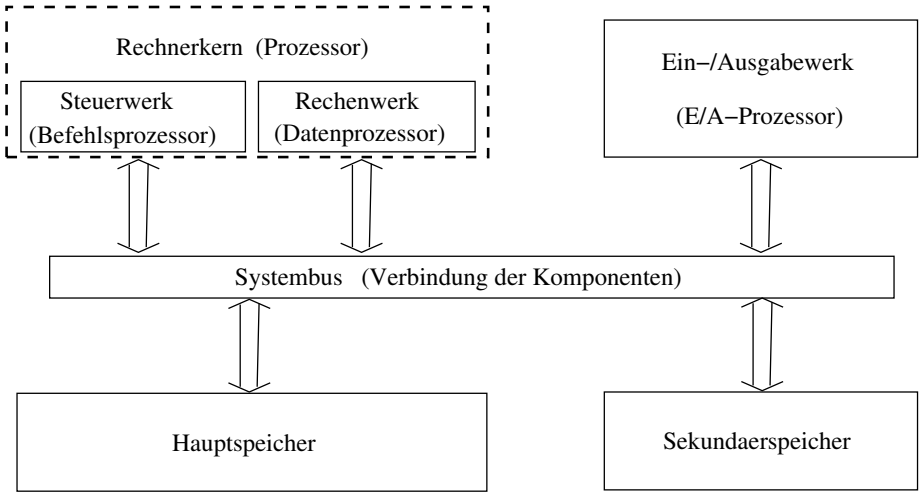
Die Von-Neumann-Rechnerarchitektur

Die nach von Neumann benannten Prinzipien der Rechnerorganisation kann man wie folgt zusammenfassen:

1. Ein Rechner besteht aus vier Werken:

   - Ein Hauptspeicher für Programme und Daten;
   - Ein Steuerwerk zur Interpretation der Programme;
   - Ein Rechenwerk zur Ausführung arithmetischer Operationen
   - Ein Ein- und Ausgabewerk eventuell verbunden mit einem Sekundärspeicher

   Die folgende Abbildung zeigt den schematischen Aufbau eines Computers und seine Hauptkomponenten. Steuerwerk und Rechenwerk bilden zusammen den Kern des Rechners, den sogenannten Prozessor. Die Verbindung zwischen den Komponenten wird durch einen Systembus realisiert.



2. Die Struktur des Rechners ist unabhängig vom zu bearbeitenden Problem, d.h. die Arbeit wird vom Programm (der Software) gesteuert.

3. Programme und Daten stehen im selben Speicher und können durch die Maschine verändert werden.

4. Der Hauptspeicher ist in Zellen gleicher Größe geteilt, die fortlaufend nummeriert sind (Adressierung).

5. Ein Programm ist eine Folge von Befehlen, die in der Regel in der gegebenen Reihenfolge ausgeführt werden (Prinzip der Sequentialität).

6. Das Prinzip der Seqentialität kann durch bedingte oder unbedingte Sprungbefehle außer Kraft gesetzt werden. Bedingte Sprünge sind von gespeicherten Werten abhängig.

7. Befehle und Daten werden binär dargestellt.

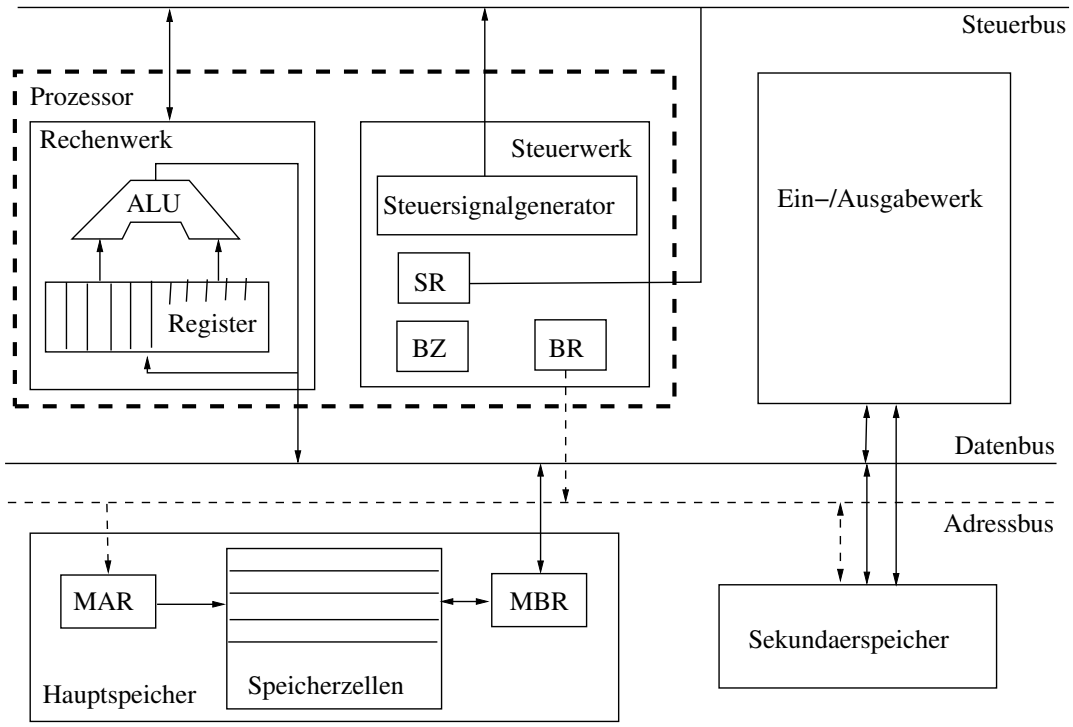
Diese Prinzipien haben sich weitgehend bewährt und bilden, abgesehen von einigen Abweichungen und Erweiterungen, bis heute das Grundkonzept moderner Rechner. Die folgenden Punkte enthalten die wichtigsten Modifikationen des Konzepts:

- Zur Leistungsteigerung durch Parallelverarbeitung können einige Komponenten (insbesondere im Rechenwerk) mehrfach auftreten (Konsequenz: Sequentialitätsprinzip wird durch das Konzept der nebenläufigen Prozesse erweitert).

- Die Trennung zwischen Haupt- und Sekundärspeicher wird zu einer vielstufigen Speicherhierarchie erweitert.

- Auf unterster Ebene werden getrennte Speicher für Daten und Programme verwendet. Dadurch wird ein gleichzeitiger Zugriff über getrennte Busse möglich.

In der folgenden Abbildung werden einige Komponeten eines von-Neumann-Rechners etwas detaillierter dargestellt. Diese Details ermöglichen es, die Abarbeitung eines Programms besser zu verstehen.



Beginnen wir mit dem Steuerwerk. Es enthält einen Steuersignalgenerator, der als zentraler Taktgeber die Arbeit des gesamten Rechners steuert. Die Kommunikation zwischen den einzelnen Komponenten wird auf drei getrennte Bussysteme aufgeteilt: Steuerbus, Datenbus und Adressbus. Die Taktsignale werden durch den Steuerbus an die anderen Komponenten übertragen (die Verbindungen zu Haupt- und Sekundärspeicher wurden in der Darstellung eingespart, um das Bild nicht zu überfrachten).

Alle Komponenten besitzen spezielle Register. Sie dienen als Speicher für Daten, Befehle, Adressen oder sonstige Werte. Im Steuerwerk findet man das Statusregister (SR), den Befehlszähler (BZ) und das Befehlsregister (BR oder IR). Das Befehlsregister enthält den aktuellen Befehl und der Befehlszähler die Adresse des nächsten auszuführenden Befehls. Nach Ausführung des aktuellen Befehls wird der nächste über den Adressbus aus dem Hauptspeicher angefordert und über den Datenbus in das Befehlsregister transportiert. Das Statusregister enthält spezielle Informationen über die letzte Operation des Rechenwerks (Vorzeichenbit, Überlaufbit, usw.) und spielt eine wichtige Rolle bei bedingten Sprüngen.

Der aktuelle Befehl wird im Steuerwerk interpretiert. Wenn er eine logische oder arithmetische Operation beinhaltet, werden die notwendigen Operanden über ihre Adressen identifiziert (die Adressen sind Teil des Befehls) und gegebenfalls aus dem Hauptsspeicher in spezielle Register des Rechenwerks geladen. Dann führt das Rechenwerk (genauer gesagt die arithmetisch-logische Einheit ALU) die Operation aus, speichert das Ergebnis in einem Register und schickt es gegebenfalls (wenn der Befehl das

erfordert) in eine adressierte Zelle des Hauptspeichers.

Die Adresse eines aus dem Hauptspeicher angeforderten Speicherinhalts wird im Speicheradressregister (MAR - memory adres register) gehalten. Der Inhalt der im MAR adressierten Speicherzelle wird zuerst in ein Speicherdatenregister (MBR - memory buffer register) ausgelesen und von dort über den Datenbus an das Zielregister im Steuer- oder Rechenwerk transportiert.

Befehlsverarbeitung im Maschinenbefehlszyklus

Um die Vorgänge in Steuerwerk besser zu verstehen, muss man sich die Struktur eines Maschinenbefehls genauer ansehen. Ein Befehl besteht aus einem Operationsteil, einem Adressteil und einem Indikatorteil. Im Operationteil wird codiert, welche Operation auszuführen ist und der Adressteil beinhaltet die Speicheradressen der Operanden, bzw. die Speicheradresse der Zelle auf die der Inhalt eines bestimmten Registers geschrieben werden soll. Der Indikatorteil kann zusätzliche Informationen enthalten. Zur Konstruktion eines einfachen Befehlsatzes (wie er z.B. in Rechnern der ersten Generation verwendet wurde) reicht es aus, dass der Adressteil nur eine Adresse enthält. Werden mehrere Operanden benötigt, muss man sie nacheinander in Register des Rechenwerks laden. In modernen Rechnern stehen sogenannte Komplexbefehle zur Verfügung, in denen man mehrere Operanden und zusätzlich eine Adresse für das Operationsergebnis angeben kann, aber da es uns hier um das prinzipielle Verständnis geht, begnügen wir uns mit einem möglichst einfachen Befehlssatz. Wir unterscheiden dabei zwischen den folgenden Befehlsarten:

- Transport- und Ladebefehle
- Arithmetische Befehle
- Logische Befehle
- Adressoperationen
- Sprungbefehle
- Ein- und Ausgabebefehle
- Steuerungsbefehle (privilegierte Befehle zur Änderung des Programmstatus)

Die Bearbeitung eines Befehls gliedert sich in die folgenden Phasen, die zusammen einen Maschinenbefehlszyklus bilden:

1. Befehlsholphase: Im Befehlszähler steht die Adresse des Befehls, der in dieser Phase aus dem Hauptsspeicher in das Befehlregister geladen wird.
2. Decodierphsae; Der Operationsteil des Befehls wird decodiert und entsprechende Steuersignale generiert.

3. Operandenholphase: Wenn der Befehl einen Operanden aus den Hauptspeicher verarbeitet, wird dieser aus der im Adressteil codierten Speicherzelle in das Rechenwerk geladen.

4. Ausführungsphase: Die ALU verknüpft die Operanden mit der angefoderten Operation

5. Rückschreibephase: Wenn der Befehl die Speicherung des Operationsergebnisses (oder eines Registerinhalts) in den Hauptspeicher erfordert, wird dieser Wert in die im Adressteil codierte Speicherzelle transportiert

6. Adressierungsphase: Der Inhalt des Befehlszählers wird aktualisiert (in der Regel um 1 angehoben, bei Sprung- und Steuerbefehlen kann auch ein anderer Wert festgelegt werden).