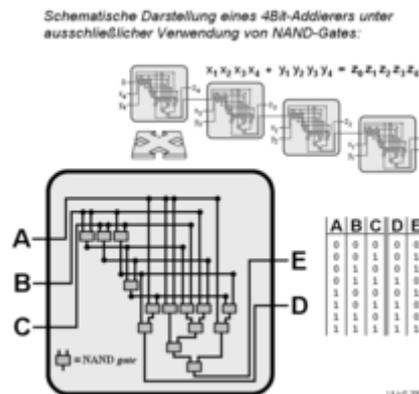




In dieser Stunde haben wir das binäre Addieren mit NAND-Gattern geschafft (wobei der Übertrag etwas kompliziert ist) und uns am Ende kurz die von-Neumann-Architektur eines Rechners angesehen. Mit den bisherigen prinzipiellen Vorüberlegungen dürfen wir von uns zu Recht behaupten, dass wir (ganz primitiv und grundsätzlich) wissen, wie ein PC funktioniert. Also nutzen wir ihn ab der kommenden Stunde endlich!

Addieren mit NAND-Gittern

Auf den ausgeteilten Arbeitsblättern wird gezeigt, wie man sich ein NAND-Gatter aus einfachen elektrischen Schaltungen konstruieren kann und dann wiederum mit einigen NAND-Gattern erst den Halb- und schließlich den Volladdierer baut. Um dann zwei Byte-Zahlen zu addieren, benötigt man dann mehrere der Volladdierer, denn die können nur bitweise addieren. Hier ein 4Bit-Addierer aus Wikipedia (unter NAND-Gatter zu finden):

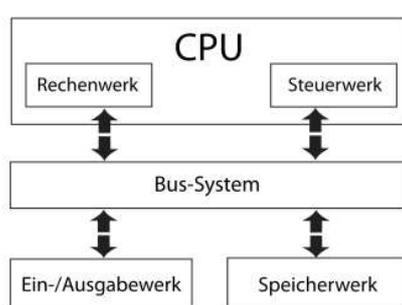


von-Neumann-Architektur

Wer sich das Programm „Johnny“ genauer ansehen möchte; hier geht's zum Download:

<http://www.heise.de/software/download/johnny/72728>

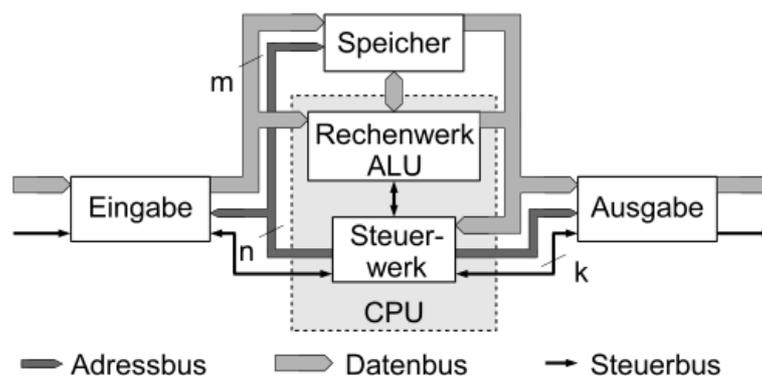
John von Neumann konzipierte einen Rechner, der einzelne Befehle nacheinander abarbeitet. Heutzutage gibt es andere Konzepte, aber damals waren seine Ideen revolutionär, da sein Rechner mit verschiedenen Programmen laufen kann. Die ersten Rechner hatten feste Programme (so ähnlich wie ein Colaautomat; der kann Trinken verkaufen, aber keine Spielekonsole ersetzen). Sein Konzept sieht so aus:



Der Traum der Menschen war damals eben ein universeller Rechner; einer, der praktisch alles kann, wenn man ihm das richtige Programm gibt. Heutige Ideen gehen Richtung KI (=künstliche Intelligenz) mit dem Wunsch einen Rechner zu bauen, der alles lernen kann, wenn er nur beobachten darf. Dazu empfehle ich (so doof es klingt) „Terminator“ mit Arnie :D

Der Aufbau ist so unterteilt:

- In der CPU ist das Rechenwerk (=Prozessor), der Rechenoperationen/logische Verknüpfungen durchführen kann (denkt an unsere NAND-Gatter, heute ist es eleganter, aber so geht's).
- Dann haben wir in der CPU noch das Steuerwerk, das auf Anweisungen eines Programms hört und diese ausführt (bspw. mit einem Lesekopf auf einem Speichermedium herumfährt oder dem Rechenwerk sagt, welche zwei Bytes im Speicher addiert werden sollen usw).
- Klar, ein Speicherwerk braucht es, weil sonst können keine Daten verarbeitet werden. Das Speichermedium haben wir uns nicht sehr genau angesehen, aber es wäre denkbar, dass jemand mit Birnchen (an/aus) Bits kodiert. Eleganter ist eine Magnetisierung (magnetisiert/nicht magnetisiert) von Metallbändern, was in den ersten Rechnern so war. Heute gibt es u.a. optische Methoden, ähnlich einer Schallplatte.
- Und Ein-/Ausgabewerke braucht man, um ein Programm zu schreiben bzw. das Ergebnis abzuholen (Tastatur, Drucker, Bildschirm,...).
- Dazwischen (siehe Bild oben) vermittelt das Bussystem (ganz anschaulich wie ein Bus). Ein Programm wird in einem temporären Speicher (RAM = Arbeitsspeicher) abgelegt und auf der Bus zu:



Der Bus ist im Prinzip einfach nur ein Knoten, an dem die ganzen Stromleitungen zusammenlaufen und sie sich gegenseitig „sehen“ können: So kann das Rechenwerk bspw. auf den Speicher zugreifen. Praktischerweise genau an der Stelle, die das Steuerwerk vorher via Bus im Speicher aufgerufen hat, was das Steuerwerk wiederum vom Programm erfahren hat...

Hier geht es noch zu einer Simulation eines von-Neumann-Rechners:

http://tams-www.informatik.uni-hamburg.de/applets/baukasten/DA/VNR_Simulation_1.html