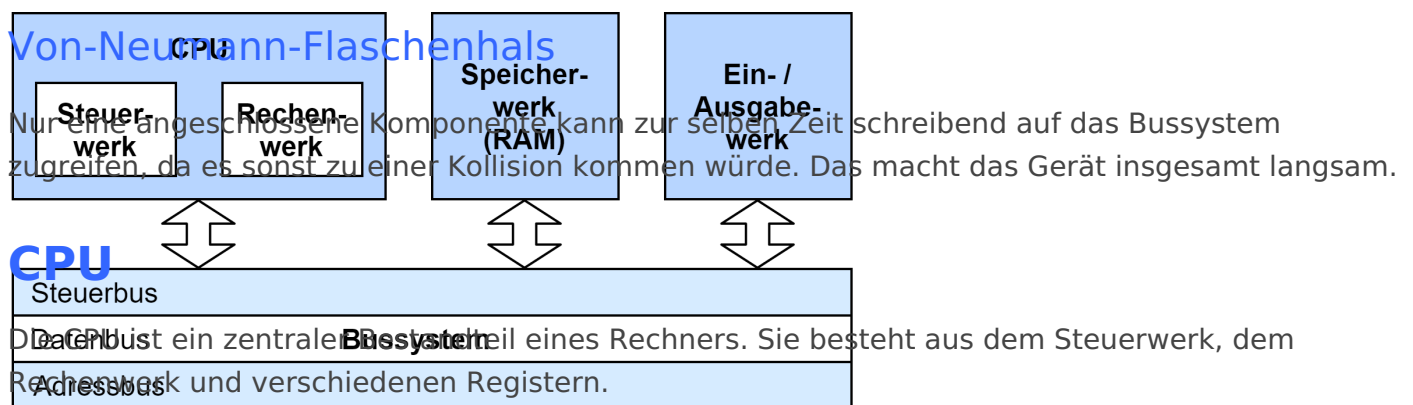


Computerarchitektur

Die Computerarchitektur ist ein Teilgebiet der Informatik, welche sich mit dem internen sowie externen Aufbau eines Computersystems beschäftigt.

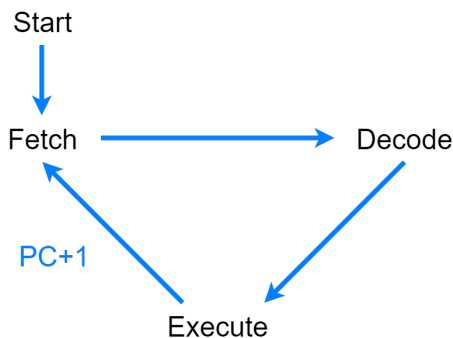
Von-Neumann-Architektur

Die Von-Neumann-Architektur bildet ein Referenzsystem für Computer und besteht aus einer CPU mit Steuerwerk und Rechenwerk, einem Ein- / Ausgabewerk, einem Speicherwerk und einem Bus-System. Das Bus-System verbindet die Komponenten miteinander.



Steuerwerk

Das Steuerwerk dient der sequentiellen Abarbeitung von Programmen.



Fetch

- Adresse aus Befehlszähler auslesen → RAM
- Befehl aus Steuerwerk → RAM
- RAM liefert Daten zurück
- Daten → Befehlsregister

Decode

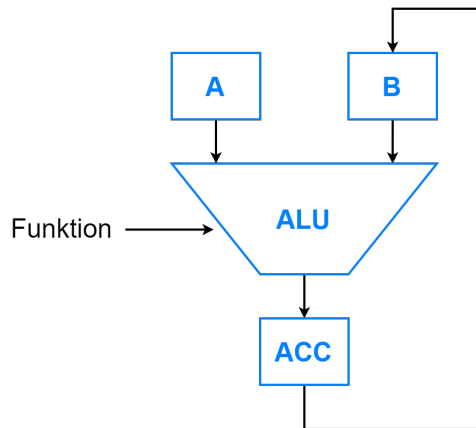
→ Befehl decodieren, Weichen stellen

Execute

→ Befehl ausführen

Rechenwerk

Das Rechenwerk führt vom Steuerwerk in Auftrag gegebene Berechnungen durch.



In der ALU werden Funktionen durchgeführt. Die Werte

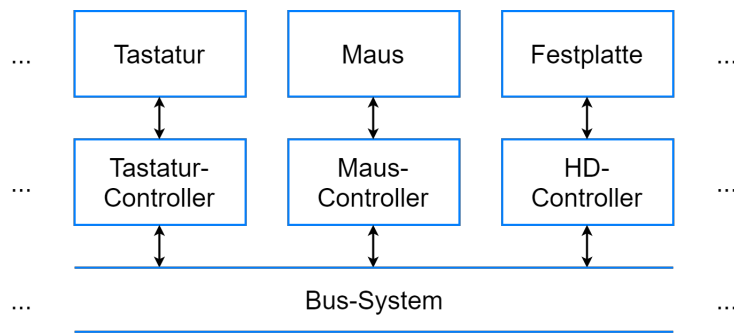
werden schließlich im Akkumulator zwischengespeichert, wo sie ausgelesen werden können.

Speicherwerk

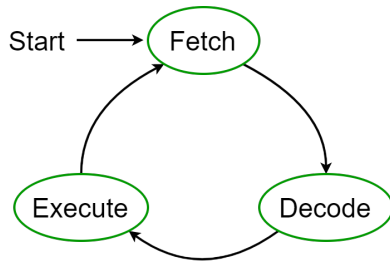
Das Speicherwerk ist in eine endliche Anzahl gleichgroßer Speicherzellen unterteilt. Jede Zelle verfügt über eine eindeutige Adresse. Es kann den Wert einer adressierten Speicherstelle auslesen oder einen Wert in eine Speicherzelle schreiben.

Ein- / Ausgabewerk

Das Ein- / Ausgabewerk dient der Ein- und Ausgabe von Informationen über verschiedene Peripheriegeräte (Maus, Tastatur, Monitor, ...). Für jedes Gerät gibt es einen Controller.



Von-Neumann-Zyklus



Fetch:

- Sende Adresse des Befehls an das Speicherwerk
- Empfange aktuellen Befehl vom Speicherwerk
- Hole aktuellen Befehl

Decode:

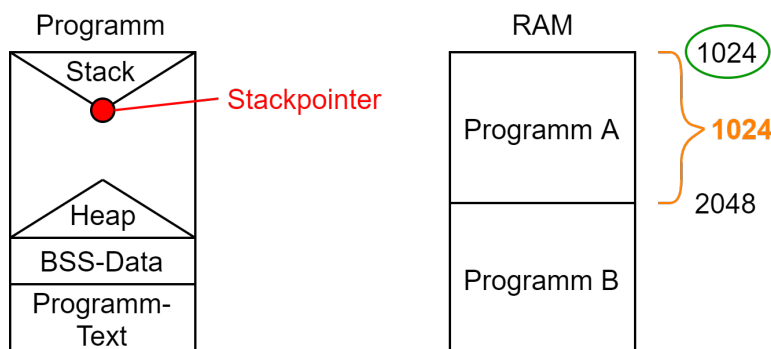
- Analysiere Befehl und treffe Vorbereitungen
- Erhöhe Programcounter (um 1)

Execute:

- Führe den Befehl aus

Weitere Komponenten der Von-Neumann-Architektur

Register



Im Stackregister wird der

Stackpointer

, der oberste Punkt des Stacks, gespeichert.

Das **Basisregister** legt die Adresse fest, zu der das Programm beginnt.

Im **Limitregister** wird die Größe des Speicherbereiches gespeichert.

Um mehrere Programme im RAM zu halten, werden das Basis- und Limitregister benötigt.

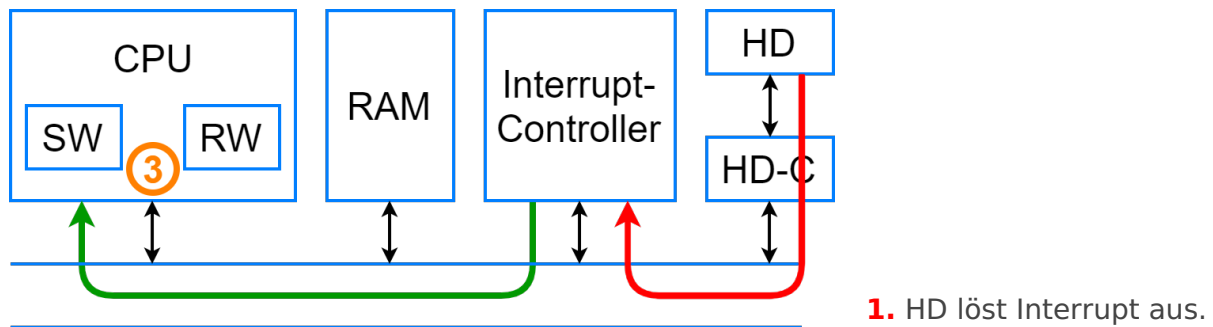
Interrupt-Controller

Der Interrupt-Controller nimmt Interruptsignale entgegen und informiert die CPU über das Vorliegen eines Interrupts.

Die CPU arbeitet den Interrupt mit der Interruptbehandlungsroutine ab.

Die Interruptbehandlungsroutine ist eine Reihe von Anweisungen, die einem bestimmten Interrupt zugeordnet sind und auf der CPU abgearbeitet werden können.

Was passiert bei einem Interrupt?



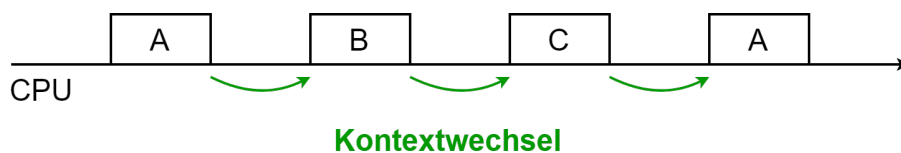
Interrupt wird an Interrupt-Controller gesendet und in Queue gespeichert.

2. Der Interrupt-Controller meldet den Interrupt dem Steuerwerk.

3. Nach der letzten Execute-Phase des aktuellen Prozesses werden aktuelle Registerinhalte gesichert. Die Interruptbehandlungsroutine wird ausgeführt.

Kontextwechsel

Damit mehrere Prozesse quasi gleichzeitig auf der CPU laufen können, müssen regelmäßig Kontextwechsel durchgeführt werden.

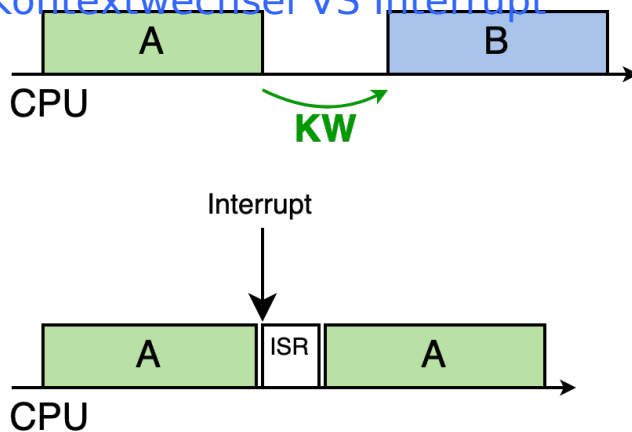


Während eines Kontextwechsels werden Registerinhalte des vorherigen Prozesses gesichert und Registerinhalte des neuen Prozesses geladen.

Zwischengespeichert werden die Registerinhalte auf dem Stack des jeweiligen gerade ausgeführten Programmes.

Der Stackpointer wird in einer bestimmten Verwaltungseinheit vom Betriebssystem hinterlegt.

Kontextwechsel VS Interrupt



Bei einem Kontextwechsel werden alle

Registerinhalte eines Prozesses zwischengespeichert. Bei hunderten Registern ist das entsprechend zeitaufwendig.

Ein Interrupt setzt einen Prozess nur kurzzeitig aus und lässt die wesentlichen Registerinhalte im Prozessor unangetastet. Es werden nur eine Hand voll Registerinhalte kurzzeitig ausgetauscht und zwischengespeichert. Die Interruptserviceroutine läuft durch und anschließend wird der vorher laufende Prozess wieder fortgesetzt.

DMA-Controller

Der Direct-Memory-Access-Controller dient dazu, die CPU bei dem Datentransfer (z.B. HD zu RAM) zu entlasten.

Prozess ohne DMA

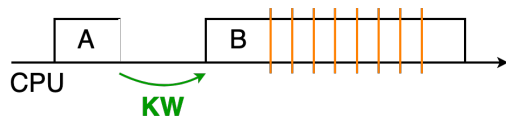
LHD → 1. Teil laden (~~XXX~~ → Kontextwechsel)

STORE → Datenwort in RAM ablegen

LHD

...

Die CPU ist ca. 1 Mio mal schneller als die HD, daher wird ein **Kontextwechsel** zu einem anderen Prozess gemacht. Sobald ein Programmteil geladen wurde, wird ein **Interrupt** gesendet. Das führt zu vielen Interrupts in Prozess B.



Der DMA-Controller löst das Problem. Er agiert als

Mini-CPU und kontrolliert den Datentransfer. Sobald das Programm vollständig in den RAM geladen wurde, sendet er einen Interrupt.

Solange kann die CPU an einem anderen Prozess weiterarbeiten.

Er benötigt dabei Quellsystem, Zielsystem, die jeweilige Startadresse und die zu übertragende Menge (Datenwörter).