

## Aufbau eines Computersystems - Grundlagen

---

Wir werden uns im Informatikunterricht (auch) mit Aufbau und Funktion von Computersystemen befassen. Dazu ist es sinnvoll, zunächst die relevanten Geräte und Bauteile kennen zu lernen.

① Begriffe rund um Computersysteme

- Benenne die abgebildeten Geräte und gib kurz ihre jeweilige Funktion an.

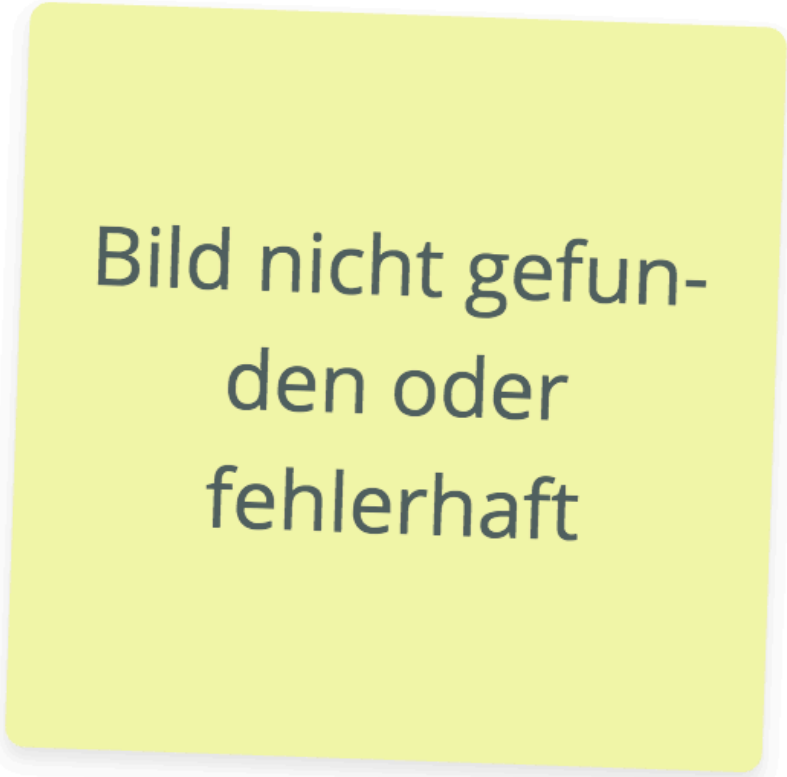


Bild nicht gefunden  
oder  
fehlerhaft

② Eine Ordnung entsteht

- Ordne die gefundenen Begriffe in Gruppen an.
- Notiere auch, nach welchen Kriterien Du die Gruppen gebildet hast.

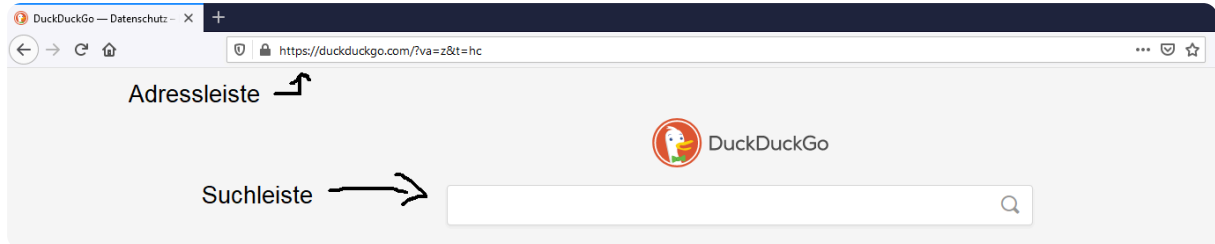
③ Es gibt noch viel mehr!

- Ergänze weitere Geräte und Bauteile (auch aus der Hausaufgabe) - schreibe sie in einer anderen Farbe zu den gebildeten Gruppen dazu.

## Funktionsprinzip eines Computersystems

Doch wie arbeitet nun ein Computersystem? Eine Übersicht findet sich hinter der folgenden URL: <https://t1p.de/ASS-Neumann1>

Achte darauf, die URL in die Adressleiste, und nicht etwa in die Suchleiste einzugeben!



### ④ Das EVA-Prinzip und die von-Neumann-Architektur

- Lies den hinter der URL bereitgestellten Text aufmerksam und übernehme die rot markierten Definitionen ins Heft.
- Prüfe Dein Verständnis mit der folgenden Learningapp: <https://t1p.de/ASS-EVA-sort>
- Achte darauf, nach der Besprechung an der Tafel die Abbildung zum EVA-Prinzip ins Heft zu übernehmen.



#### **Achtung!**

Mit "Controller" ist in der Grafik nicht etwa Spiele-Hardware gemeint, sondern eine Steuereinheit im Rechnerinneren.

## Fokus: Die Verarbeitung

Vor allem die Verarbeitung (unterstützt durch den Arbeitsspeicher RAM) ist für uns von Interesse. Leider bringt es uns nicht weiter, die entsprechenden Bauteile aufzuschneiden oder einfach zu vergrößern - daher werden wir demnächst auf ein Simulationsmodell zurückgreifen, um die Funktionsweise nachzuvollziehen.

Bild nicht gefunden oder fehlerhaft

Prozessor - Außenansicht

Bild nicht gefunden oder fehlerhaft

Prozessor - Innenansicht

## Abkürzungen und Fachbegriffe rund um Computersysteme

CPU, RAM, BUS, ... - bei der Beschäftigung mit Computersystemen begegnen wir unweigerlich vielen Fachbegriffen und Abkürzungen. Vervollständige die angegebenen Tabellen in Deinem Heft, um einen Überblick zu erhalten. Verwende die angegebenen Abkürzungen bzw. Begriffe!

Verwende dabei als Quelle das digitale Schulbuch (<https://t1p.de/ASS-EVA-V>) und eigene Recherchen!

### ⑤ Tabelle mit Abkürzungen

Abkürzung	engl. Bedeutung	deutsche Bedeutung
BIOS	Basic Input Output System	Firmware zur Konfiguration der Hardware eines PCs
...	...	..



#### Liste mit Abkürzungen:

CD, CPU, DVD, EVA, GUI, RAM, ROM, USB, GPU

### ⑥ Glossar

Bauteil	Funktion
Arbeitsspeicher (RAM)	Speicher, der die gerade auszuführenden Programmteile und die benötigten Daten enthält
...	...



#### Liste mit Begriffen:

Bus(-System), Festplatte, Grafikkarte, Hauptplatine/Motherboard, Peripherie, Prozessor (CPU)

### ⑦ Unterscheidungsaufgabe

- Erkläre den Unterschied zwischen RAM und ROM.
- Recherchiere den Unterschied zwischen den Speichern HDD, SSD und SD-Karte.

## Schneller-Höher-Weiter: Leistungsdaten von Computersystemen

Computersysteme werden oft mit Leistungsangaben beworben oder beschrieben. Dabei gibt es für die Kategorien Speicher und Rechengeschwindigkeit jeweils eine Grundeinheit, die dann mit Präfixen für Größenordnungen erweitert werden können:

	Speicher	Taktfrequenz
Basiseinheit	1 Byte (=8 Bit)	Hertz
Bedeutung	Sammeleinheit für 8 Informationen vom Typ „0 oder 1“	Rechenvorgänge pro Sekunde



### Hinweis

Bei den Präfixen existieren zwei konkurrierende Systeme: Das eine basiert auf dem Dezimalsystem (Ziffern 0 bis 9), das andere auf dem Binärsystem (Ziffern 0 und 1).

Frage: Warum könnte es diese beiden Systeme geben?

Präfix	Bedeutung	Präfix	Bedeutung

### ⑧ Rechenaufgaben

- Wie viele Fotos à 1,7 MB passen auf eine Speicherkarte mit 2 GB Speicherplatz?
- Wie könnte ein Gigabyte in die entsprechende binäre Einheit umgerechnet werden?

## Modell zur Verarbeitung: Der Murmelrechner

Der Murmelrechner ist ein Simulationsmodell, bei dem ihr die Rolle ausgewählter Bauteile eines Rechnersystems übernehmt und damit die Verarbeitung von Rechenaufgaben durch ein Computersystem analysieren könnt.

Der Name kommt von den Murmeln (es geht auch mit Holzperlen, Bohnen, ...), die zur Darstellung von Zahlenwerten verwendet werden.

Damit bei der Durchführung auch alles funktioniert, müsst ihr die Bauteile und ihre Funktion verstehen. Lest bitte alle Erklärungen genau durch und stellt, wenn nötig, Fragen!

### Der Speicher

Der Speicher wird durch Plastikbecher symbolisiert, die von Null an aufsteigend nummeriert sind. Die Anzahl der Murmeln in den Behältern gibt den darin gespeicherten Zahlenwert an. Die Nummern der Gefäße sind wichtig, damit immer klar ist, welche Speicherzelle gemeint ist. Das Bild zeigt einen Speicher aus sechs Behältern:

Die dargestellte Situation entspricht der folgenden Speicherbelegung:

Adresse	0	1	2	3	4	5
Inhalt	6	8	10	12	4	0

### Grundoperationen des Murmelrechners

Um die Möglichkeiten eines Computersystems auf unterster Ebene zu simulieren, müssen wir die möglichen Operationen auf jene begrenzen, die dem Computersystem zur Verfügung stehen.

Weder wir noch ein Computersystem kann leicht (für uns: "mit einem Blick" erkennen, welches von zwei Gefäßen mehr Murmeln enthält, sofern nicht gerade eines davon leer ist. Das "Leersein" eines Gefäßes hingegen ist leicht festzustellen.

Ebenso ist es relativ leicht, eine Murmel hinzuzufügen oder zu entfernen.

#### ⑨ Überlegung

- Überlege, wie man nur mit den beschriebenen Grundoperationen die Zahlen 3 und 2 addieren kann.



#### Der Befehlssatz

Es ergibt sich ein "Werkzeugkasten" von Befehlen, die das System ausführen kann:

Eine Murmel hinzufügen

Eine Murmel entfernen

Testen, ob das Gefäß leer ist

## Lösung: Schrittweise Addieren

Du hast es bestimmt schnell herausgefunden: Um die Zahlen zu addieren, kann wie folgt vorgegangen werden - das Ergebnis ist im Gefäß "0" ablesbar:

*entferne eine Murmel aus Gefäß 1  
lege eine Murmel in Gefäß 0  
entferne eine Murmel aus Gefäß 1  
lege eine Murmel in Gefäß 0*

Bild nicht gefunden oder

Speicherbelegung vorher

Bild nicht gefunden oder

Speicherbelegung nachher

### **Achtung - Schwierigkeit**

So einfach ist es natürlich nur, wenn man mit festen Eingangswerten arbeitet. Sobald zwei beliebige Zahlen addiert werden müssen, wird es komplizierter: Dann muss auch - wie im folgenden Beispiel - der Nulltest verwendet werden.

*prüfe, ob in Gefäß 1 noch etwas vorhanden ist  
wenn ja, dann  
entferne eine Murmel aus Gefäß 1  
lege eine Murmel in Gefäß 0  
beginne wieder am Anfang  
wenn nicht, dann  
beende den Vorgang*

### **alternative Notation**

```
0 tst 1
1 jmp 3
2 jmp 6
3 dec 1
4 inc 0
5 jmp 0
6 hlt
```

### ⑩ Analyse

- Vergleiche die beiden Notationen und ermittle die Bedeutung der Befehle:
- tst
- jmp
- dec
- inc
- hlt

## Einschub: Fachsprache

Für das Hinzufügen und das Wegnehmen von Murmeln gibt es Fachbegriffe, die wir zukünftig verwenden werden:

- Inkrementieren: Erhöhen eines Wertes um 1 / Hinzufügen einer Murmel
- Dekrementieren: Senken eines Wertes um 1 / Entfernen einer Murmel

Eine Abfolge von Murmelrechnerbefehlen, die der Reihe nach ausgeführt werden können, um eine Aufgabe zu lösen, wird als Murmelrechnerprogramm bezeichnet. Pro Zeile gibt es immer nur einen Befehl.

## Befehle und Befehlskategorien

Die Befehle lassen sich in verschiedene Kategorien einsortieren:

### Rechenbefehle

Die Befehle `dec` und `inc` greifen nicht in den Programmablauf ein, sondern beeinflussen nur den Inhalt von Speicherstellen. Daher werden sie als Rechenbefehle bezeichnet.

Rechenbefehle erhalten als Parameter die Nummer der zu inkrementierenden bzw. zu dekrementierenden Speicherstelle.

### Kontrollbefehle

Die Befehle `jmp`, `tst` und `hlt` greifen in den Ablauf des Programms ein, verändern also, ob und welche Befehle als nächstes ausgeführt werden.

Daher werden sie als Kontrollbefehle bezeichnet.

Der Kontrollbefehl `jmp` erhält als Parameter die Programmcodezeile, in der das Programm weiter ausgeführt werden soll. Der Befehl `tst` erhält als Parameter die Nummer der zu testenden Speicherstelle und `hlt` benötigt keine Parameter.

### Genauer zum `tst`-Befehl

Der `tst`-Befehl ermöglicht es überhaupt erst, Programme für beliebige Probleme zu schreiben. Ohne Nulltest könnte man nur Programme für konkrete Aufgaben (wie "3+2") schreiben, da es keine Möglichkeit gibt, das Programm erst zu beenden, wenn eine der Eingabespeicherstellen leer ist.

Der `tst`-Befehl funktioniert so:

1. Test, ob die angegebene Speicherstelle noch Murmeln enthält
2. Wenn ja, dann wird der Befehl in der nächsten Zeile ausgeführt (danach geht die Ausführung je nach Inhalt dieser Zeile normal weiter).
3. Wenn nein, dann wird stattdessen in der übernächsten Zeile weiter ausgeführt.

## Akteure, Durchführung und Beobachtung

Innerhalb unseres Prozessmodells "Murmelrechner" gibt es (wie in der realweltlichen Vorlage, dem Prozessor) verschiedene Akteure, die jeweils eigene Aufgaben erledigen:

### Der Rechenkünstler / Die Rechenkünstlerin

Der/Die Rechenkünstler/in führt die Rechenbefehle dec und inc sowie den Testbefehl tst aus. Dazu erhält er/sie ein "Akkumulator" genanntes Gefäß, in dem er die Operationen ausführt, und ein Vorratsgefäß mit Murmeln, aus dem/in das er/sie die dabei benötigten Murmeln entnimmt bzw. ablegt. Außerdem kann er/sie an den Steuermann / die Steuerfrau das Ergebnis des Nulltests kommunizieren ("ja"/"nein").



### Der Laufbursche / Die Überbringerin

Diese Rolle hat die Aufgabe, dem/der Rechenkünstler/in die Murmeln aus den angegebenen Speicherstellen zur Verfügung zu stellen bzw. diese nach den Operationen wieder zurück zu bringen. Dazu verwendet er/sie ein eigenes Transportgefäß.



### Der Steuermann / Die Steuerfrau

Diese Rolle gibt den beiden anderen Rollen Anweisungen, was zu tun ist. Dazu erhält er/sie das Programm und eine Markierung, mit der verfolgt werden kann, welche Zeile gerade ausgeführt wird.



#### **Wichtig: Klare Trennung der Rollen**

Das Modell läuft nur richtig ab, wenn jede/r sich auf seine/ihre Aufgaben konzentriert und die andere ihre machen lässt. Im realen Prozessor gilt ebenso, dass jede Komponente eine klar definierte Aufgabe besitzt und speziell für diese gebaut und optimiert wurde.

Das bedeutet, die folgenden Rollen haben exklusiven Zugriff auf:

- RechenkünstlerIn: Vorrat
- Laufbursche/Überbringerin: Speichergefäße
- Steuermann/-frau: Programm



## Anwendung als Prozessmodell zur Berechnung

⑪ Führt die folgenden Aktivitäten in der Kleingruppe durch:

- Spielt das Programm von Seite 6 mit verschiedenen Startbefüllungen der Gefäße 0 und 1 durch. Erreicht ihr das erwartete Ergebnis?
- Versucht, ein Programm zur Subtraktion zu erstellen. Ihr dürft annehmen, dass die größere der Zahlen immer in Gefäß 0 liegt.
- Prüft den Ablauf eures eigenen Programms - wird das erwartete Ergebnis erreicht?
- Was macht das Erstellen von Programmen schwierig? Notiere hier:



### Dokumentation

Um die Übersicht zu behalten, schreibt tabellarisch auf, wie sich die Werte in den Speicherstellen nach und nach verändern. Dabei sollte nach jeder einzelnen Änderung eine neue Zeile geschrieben werden.

## Übertragung auf ein „echtes“ Computersystem

⑫ Um die Erkenntnisse aus dem Modell auf ein echtes Computersystem zu übertragen, beschreibe zunächst die Aufgaben der folgenden Komponenten kurz in eigenen Worten:

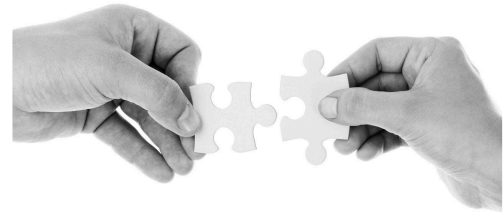
- 1) RechenkünstlerIn, 2) Steuermann/frau, 3) Laufbursche bzw. Überbringerin, 4) Speichergefäße

Zu welchen bereits bekannten Komponenten könnten diese Beschreibungen passen?

## Die Auflösung: Was entspricht was im realen System?

Natürlich handelt es sich beim Murmelrechner nur um ein Modell, mit dem die komplexe Funktion eines Computersystems leichter nachvollzogen werden kann. Dennoch besitzt jede Rolle im Modell ein realweltliches Gegenstück.

- ⑬ Verbinde die zusammen gehörenden Rollen/Komponenten, zunächst mit einem Bleistift:



Rechenkünstler/in

Steuermann/-frau

Laufbursche /  
Überbringerin

Speichergefäße

Steuerwerk  
(Teil der CPU)

BUS-System

Arbeitsspeicher

Rechenwerk  
(Teil der CPU)

- ⑭ Fülle den nachfolgenden Lückentext aus.

Der Prozessor (CPU) besteht im Wesentlichen aus zwei Teilen. Der eine ist das so genannte

(englisch: Control Unit, CU). Das  teilt große

Aufgaben in kleinste Schritte auf und koordiniert ihre Verarbeitung. Der andere Teil ist das

(englisch: Arithmetic Logic Unit, ALU). Hier werden die kleinsten

Aufgaben aus dem Steuerwerk verarbeitet.

Die notwendige Verbindung zum  wird dabei über das

als Datenverbindung hergestellt.