

Arbeitsheft

zu dem Fach

Rechnersystemtechnik

(RST)

Ein Skript zum Unterricht

Verfasst von S. Epping

Letzte Änderung: 09.08.2021

Inhaltsverzeichnis

1	Ein einfaches IT-System	1
1.1	Das EVA-Prinzip	1
1.1.1	Aufgabe „EVA-Prinzip“	1
1.2	Das Bussystem	2
1.2.1	Aufgabe „Lückentext - Was ist ein Bussystem?“	4
1.2.2	Aufgabe „Daten einlesen“	5
2	Rechnerorganisation	6
3	Die Speicherhierarchie	8
3.1	Aufgabe „Speicherpyramide“	8
3.2	Aufgabe „Kenngrößen der Speicher“	8
4	Von Prozessen und Threads	9
4.1	Single-Core	9
4.2	Multi-Core	10
4.3	Aufgabe „Multitasking, Multithreading, Multiprocessing“	12
5	Von Bit und Byte	13
5.1	Abbildung von logischen Zuständen in einem PC-System	13
5.2	Anzahl der Bit-Zustände	15
5.2.1	Aufgabe „Wie viele Bit werden benötigt?“	16
5.2.2	Aufgabe „Schach und Reis“	17
6	Speicherangaben in Bit und Byte	18
6.1.1	Aufgabe „Kundendaten“	20
6.1.2	Aufgabe „Kamerabilder“	20
6.1.3	Aufgabe „Videodatei“	20
7	Die Übertragung von Bits	21
7.1	Aufbau von digitalen Signalen	21
7.2	Taktleitung nutzen	22
7.3	Berechnung von Transferraten	24
7.3.1	Serielle Datenübertragung	24
7.3.2	Parallele Datenübertragung	24
7.3.3	Hohe Taktraten	25

7.3.4	Single-Data-Rate und Double-Data-Rate	26
8	Die Dauer einer Datenübertragung.....	27
8.1.1	Aufgabe „Dauer Kopiervorgang“	27
8.1.2	Die Datenrate einer Internetleitung.....	28
8.1.3	Aufgabe „Dauer eines Downloads“	28
9	Aufbau des Mainboards/Motherboards	29
9.1	Die rückseitigen Anschlüsse	30
9.2	Anschlüsse im Detail.....	31
9.3	Anschlüsse und Komponenten auf dem Mainboard.....	32
9.4	Chipsätze	34
10	Serielle Datenübertragung	36
10.1	Nachteile der parallelen Datenübertragung	37
10.1.1	Bandbreite wird geteilt.....	37
10.1.2	Gleiche Länge für alle Leitungen	38
10.2	Aufbau einer Lane	39
10.3	Differentielle Spannungspegel	40
10.3.1	Unsymmetrische Signalübertragung	40
10.4	Differentielle Übertragungstechnik.....	41
10.4.1	Aufgabe „Differentielle Spannungspegel (LVDS)“	43
10.5	Kodierungsverfahren	44
10.5.1	Die 8b/10b-Kodierung	44
10.5.2	Berechnung der Netto-Datenrate	45
10.5.3	Aufgabe „Kodierungsverfahren“	46
11	Lane Sharing	47
11.1.1	Aufgabe „Lane-Sharing“	48

1 Ein einfaches IT-System

Im ersten Kapitel sollen die Hauptkomponenten eines einfachen IT-Systems sowie das grundsätzliche Zusammenspiel dieser Komponenten untereinander erläutert werden.

1.1 Das EVA-Prinzip

Computer arbeiten nach dem EVA-Prinzip. EVA steht für Eingabe, Verarbeitung und Ausgabe.

- **Eingabe:** Es erfolgt zunächst eine **Eingabe** der Daten.
- **Verarbeitung:** Die Daten werden mit Hilfe von Recheneinheiten verarbeitet.
Zwischenergebnisse können in einer Speichereinheit abgelegt werden.
- **Ausgabe:** Anschließend erfolgt die **Ausgabe** der neuen Daten.

1.1.1 Aufgabe „EVA-Prinzip“

Welche Ein- und Ausgabegeräte kennen Sie? Füllen Sie die leeren Felder aus.

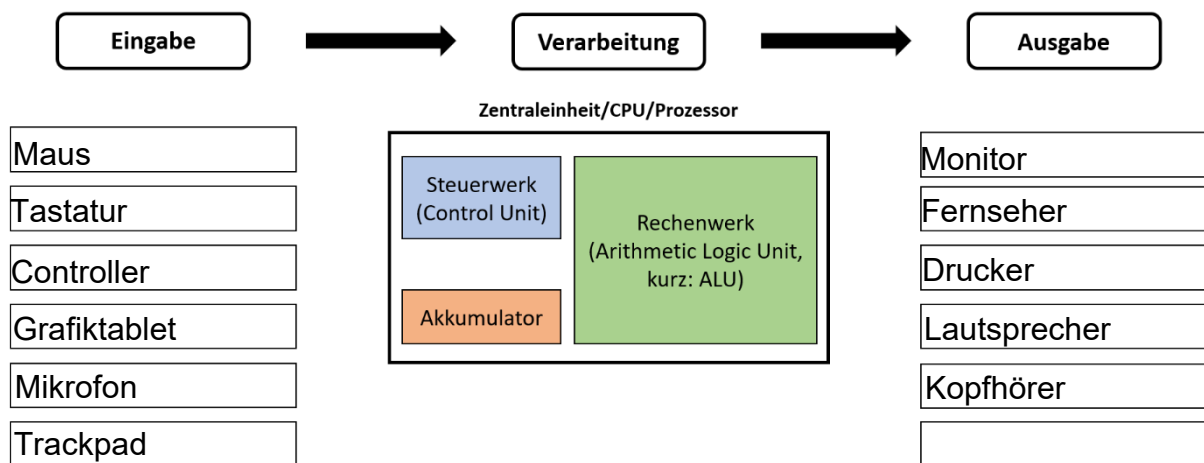


Abbildung 1-1: Das EVA-Prinzip

Erklärung der Komponenten:

Die Verarbeitungseinheit ist der **Prozessor**. Dieser besteht aus einem **Rechenwerk**, einem **Steuerwerk** sowie einem **Akkumulator**.

Das **Steuerwerk** regelt die Abfolge der Schritte für die Verarbeitung der Daten. Es ist neben dem Rechenwerk die Hauptkomponente des Prozessors.

Das **Rechenwerk** (Arithmetisch-logische Einheit, **ALU**) führt die Elementaroperationen eines Rechners durch. Es kann sowohl arithmetische (+, -, *, /) als auch logische (z.B. AND, OR, NOT) Operationen ausführen

Der **Akkumulator** dient als kleiner Speicher, um Zwischenergebnisse aufzunehmen.

1.2 Das Bussystem

Die Komponenten der Eingabe, Verarbeitung und Ausgabe sind über ein Bussystem miteinander verbunden. Unterschieden werden drei Gruppen (drei Busse):

- Datenbus
- Steuerbus
- Adressbus

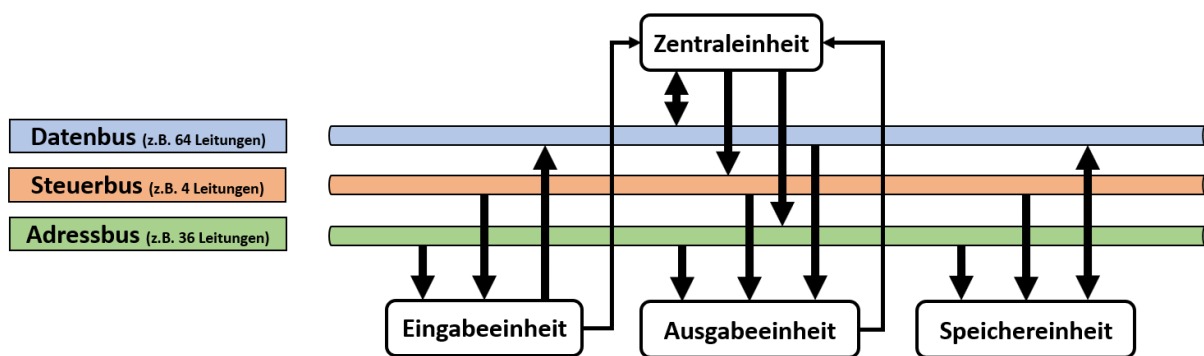


Abbildung 1-2: Das Bussystem

Erklärung des Bussystems:

Datenbus

- Daten gelangen von den Eingabegeräten hin zur CPU.
- Daten werden von der CPU raus an die Ausgabeeinheit geschickt.
- Zwischen der CPU und der Speichereinheit ist der Bus bidirektional. Das heißt, die Daten werden in beide Richtungen auf dem Bus verschickt (nicht gleichzeitig).
- Die Verbindung zwischen der CPU und dem Hauptspeicher (RAM) ist ein 64-Bit breiter Datenbus, d.h. auf 64 Leitungen werden 64 Bit (also 8 Byte) gleichzeitig übertragen.

Steuerbus

- Über den Steuerbus regelt die CPU die Kommunikation mit den weiteren Teilnehmern (unidirektional).

- Folgende Signale werden vom Steuerbus verarbeitet:
 - Lesen aus dem Speicher oder Schreiben in den Speicher
 - Eingabe von peripheren Geräten
 - Ausgabe an periphere Geräte
 - evtl. Interruptsignale (siehe nächste Seite)

Bevor die CPU ein Steuersignal an einen anderen Teilnehmer verschicken kann, muss erst klar sein, welcher Teilnehmer etwas senden oder empfangen soll. Da die komplette Peripherie mit dem Bussystem verbunden ist, empfängt jedes Gerät das Steuersignal. Folglich würde jede Komponente seine Daten schicken oder möchte Daten empfangen.

Zu diesem Zweck besitzen alle Teilnehmer in einem System eine Adresse.

Adressbus

- Es kann ein Gerät der Ein- oder Ausgabeeinheit oder eine Stelle im Speicher adressiert werden - das regelt die CPU (unidirektional).
- Wird eine Adresse im Speicher angesprochen, können entweder dort neue Daten abgelegt oder bereits vorhandene Daten ausgelesen werden – das regelt die CPU über den Steuerbus.

Interrupt Request (IRQ)

Jedes PC-System besteht aus einer Vielzahl von Ein- und Ausgabegeräten, welche Daten senden oder empfangen möchten. Damit der Prozessor mitbekommt, dass eine neue Ein- oder Ausgabe abgearbeitet werden soll und sofort reagiert, senden die Teilnehmer ein Unterbrechungssignal zur CPU.

Beispielsweise sendet die Tastatur eine Unterbrechungsanforderung (engl. Interrupt Request, IRQ), wenn eine Taste gedrückt wurde. Dazu wird ein Signal auf den Steuerbus oder direkt an einen speziellen Prozessorpin (IRQ-Eingang) gelegt. Der Prozessor merkt sich die Stelle, an der er gerade war, unterbricht seine aktuelle Tätigkeit und widmet sich zunächst der Unterbrechungsanforderung. Anschließend wird das wartende Programm dort fortgeführt, wo es unterbrochen wurde.

Weitere Beispiele, bei denen Geräte eine Unterbrechungsanforderung senden:

- Netzwerkkarte: wenn Daten empfangen wurden und im Puffer bereitliegen.
- Festplatte: wenn die vorher angeforderten Daten gelesen wurden und abholbereit sind (das Lesen von der Festplatte dauert relativ lange).
- Grafikkarte: wenn das aktuelle Bild fertig gezeichnet wurde.

- Soundkarte: wenn wieder Sound-Daten zum Abspielen benötigt werden, bevor der Puffer leer wird.

Die Ein- und Ausgabegeräte können auch ein Signal schicken, wenn ein Fehler vorliegt.

1.2.1 Aufgabe „Lückentext - Was ist ein Bussystem?“

Unter einem Bussystem, oder kurz gesagt Bus, ist bei einem PC ein Bündel

elektrischer Leitungen gemeint, an dem alle Komponenten parallel angeschlossen sind.

Die Anzahl der Leitungen ist die Busbreite. Es gibt 3 verschiedene Bussysteme:

Adressbus (Address Bus)

Da alle Komponenten parallel an dem Datenbus angeschlossen sind, aber gewährleistet sein muss, dass immer nur eine Einheit Daten empfängt oder sendet ist ein Adress-System notwendig. Nur wenn eine Geräteeinheit eine Adresse aus dem Adressbus empfangen hat, darf sie Daten über den Datenbus senden oder empfangen. Der Adressbus ist unidirektional, Adressdaten werden nur vom Prozessor zu den Baugruppen gesendet.

Datenbus (Data Bus)

Über den Datenbus werden Daten vom Prozessor zu verschiedenen Geräteeinheiten und auch wieder zurück zum Prozessor gesendet. Hier muss natürlich unterschieden werden, ob es sich um Ein- oder Ausgabeeinheiten handelt. Der Datenbus für Prozessor und Speicher ist bidirektional, Daten können in beide Richtungen bewegt werden.

Steuerbus (Control Bus)

Über den Steuerbus wird mitgeteilt, ob Daten empfangen oder gesendet werden sollen. Der Steuerbus ist unidirektional, Daten werden nur vom Prozessor zur Baugruppe gesendet.

Adressbus, Datenbus und Steuerbus bilden zusammen den Systembus.

Begriffe:

parallel, vom Prozessor, beide Richtungen, Systembus, elektrischer Leitungen, empfangen, den Datenbus, Busbreite, eine Einheit, unidirektional

1.2.2 Aufgabe „Daten einlesen“

Ein Eingabegerät möchte Daten senden. Welche Arbeitsschritte werden ausgeführt? Bringen Sie die untenstehenden Aktionen in die richtige Reihenfolge, indem Sie die Felder nummerieren.

3 Der Prozessor legt die Adresse des Eingabegerätes auf den Adressbus.

2 Der Prozessor registriert ein Interruptsignal, prüft dessen Herkunft und beendet seinen aktuellen Prozess.

10 Der Prozessor löscht die Adresse auf dem Adressbus und den Lesebefehl auf dem Steuerbus.

6 Die Eingabeeinheit registriert den Lesebefehl (über den Steuerbus).

7 Die Eingabeeinheit sendet ihre Daten.

9 Check ob die Datenübertragung abgeschlossen ist:
Wenn "nein" dann weiter mit 8
Wenn "ja" dann weiter mit 10

1 Das Eingabegerät sendet eine Unterbrechungsanforderung an die CPU.

4 Die Eingabeeinheit registriert ihre Adresse (über den Adressbus).

8 Der Prozessor liest die Daten (vom Datenbus) und verarbeitet sie.

5 Der Prozessor gibt einen Lesebefehl auf den Steuerbus.

11 Der Prozessor kehrt zurück zum unterbrochenen Programm.

2 Rechnerorganisation

Wenn Sie am PC sitzen, ist eine Menge los im Rechner. Sie surfen im Internet, hören Musik, schreiben einen Text usw. Alle Programme, Dateien und Prozesse, die gerade im Rechner laufen bzw. genutzt werden, befinden sich im Arbeitsspeicher.

Der **Arbeitsspeicher** ist vergleichbar mit Ihrem Arbeitstisch. Wenn Sie z.B. die Hausaufgaben für das Fach Englisch erledigen möchten, bereiten Sie sich auf diesen Arbeitsprozess vor. Sie legen das Fachbuch, das Heft, die Federmappe, evtl. ein Vokabelheft oder ein Wörterbuch auf den Tisch. All das sind Dinge, die Sie zur Bewältigung des Arbeitsschrittes „Englischhausaufgaben machen“ benötigen und an die Sie schnell rankommen möchten. Wenn die Aufgabe abgearbeitet wurde, können die Materialien wieder weggeräumt werden. Der Tisch ist jetzt frei für die nächste Aufgabe. So läuft es auch mit dem RAM.

Der RAM ist ein flüchtiger Speicher. Flüchtig bedeutet, dass der Speicher seinen Inhalt verliert, sobald die Spannungsversorgung unterbrochen wird. Schaltet man den Rechner also aus, werden die Daten im RAM gelöscht.

Wo liegen die Programme und Dateien vorher?

Programme und Dateien liegen zuvor auf einem nicht-flüchtigen Speicher, wie z.B. einer **HDD** oder **SSD**. Hiervon werden **Kopien** der Programme und Dateien in den RAM geladen.

Der Zugriff auf den RAM erfolgt wesentlich schneller als bei einer Festplatte. Dadurch kommt die CPU schneller an die Daten und das steigert die Performance.

Dateien gelangen auch wieder von dem RAM zur Festplatte. Das geschieht, wenn Sie z.B. einen Text schreiben und dann auf „Speichern“ oder „Speichern unter“ klicken. Dann wird eine Kopie der Datei vom RAM erstellt und auf die Festplatte geschrieben.

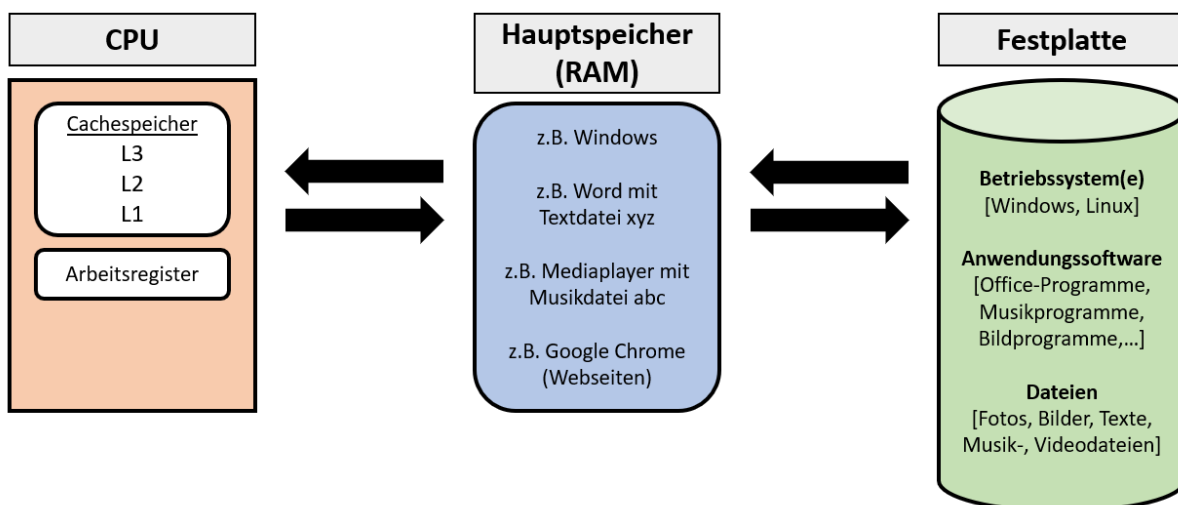


Abbildung 2-1: Zusammenspiel der Hauptkomponenten

Noch schneller als RAM

Dateien, die während eines aktuell laufenden Prozesses sehr häufig gebraucht werden, können noch näher zur CPU gebracht werden. Vom Hauptspeicher gelangen die Daten in den Cache-Speicher. Die Zugriffszeiten sind hier nochmals um ein Vielfaches geringer. Der Cache-Speicher ist nochmals in drei Level unterteilt – L1, L2 und L3.

- Hierbei ist L1 der kleinste, aber schnellste Speicher.
- L2 ist langsamer als L1, besitzt jedoch mehr Speicherplatz.
- L3 ist langsamer als L2, aber immer noch schneller als der RAM. Dafür hat L3 im Vergleich zu L2 eine größere Speicherkapazität.

Noch schneller als Cache

Die schnellsten Speicher befinden sich direkt mit in der CPU. Die CPU verfügt über mehrere Arbeitsregister, die als (sehr) kleine Zwischenspeicher dienen und Dateien sowie Befehle bereithalten, die unmittelbar von der Zentraleinheit benötigt werden.

3 Die Speicherhierarchie

Die Speicherhierarchie verfolgt ein Grundkonzept:

Häufig benötigte Speicherinhalte in kleinen Speichern, seltener benötigte Inhalte in großen Speichern ablegen! Die Daten, die die CPU in dem jeweils aktuellen Prozess benötigt, werden in die Nähe der CPU transportiert. Kleine Speicher liegen somit näher an der CPU und haben eine geringere Zugriffszeit. Das ist die Zeit, die nach der Adressierung einer Speicherzelle vergeht, bis die Daten tatsächlich anliegen.

3.1 Aufgabe „Speicherpyramide“

Füllen Sie die Speicher-Pyramide aus. Tragen Sie die folgenden Speicher an die richtige Stelle ein.

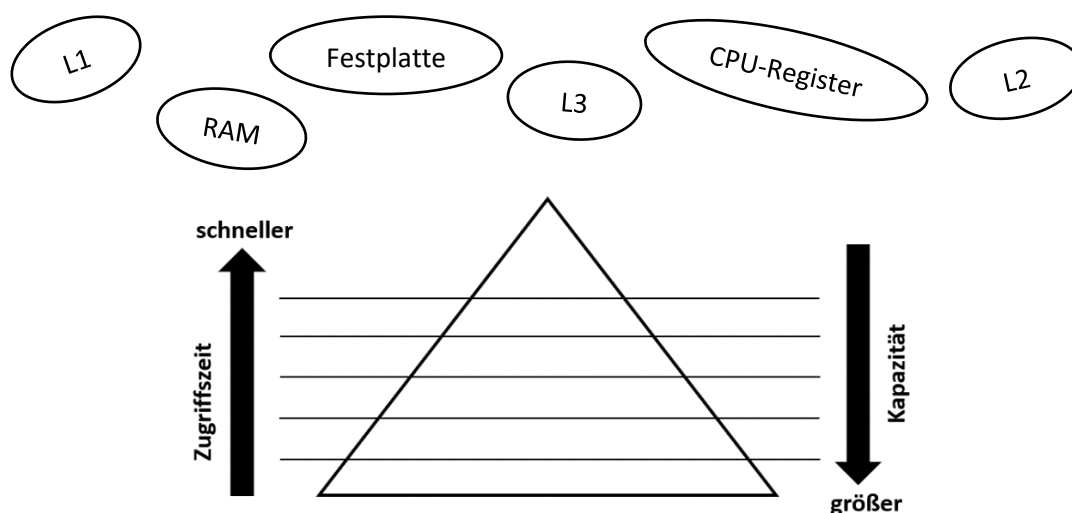


Abbildung 3-1: Speicherpyramide

3.2 Aufgabe „Kenngrößen der Speicher“

Ordnen Sie die Angaben der Speichergröße und der Zugriffszeit den passenden Speichern zu.

Speichergrößen: ~4–32 GB, ~8–64kB, ~1–4TB, ~128–500GB, ~500Byte, ~1–16MB, ~256kB

Zugriffszeiten: ~50ns, ~ 5ns, 0,3ms, ~ 250ps, ~ 1ns, ab 3,5ms, ~25ns

Speicher	Typische Speichergröße	Ungefähre Zugriffszeit	Eigenschaften
CPU-Register			flüchtig
L1-Cache			flüchtig
L2-Cache			flüchtig
L3-Cache			flüchtig
Hauptspeicher			flüchtig
SSD			nicht flüchtig
HDD			nicht flüchtig

4 Von Prozessen und Threads

Ein Prozess (manchmal auch Task genannt) ist die Ausführung eines Programms (Word, Webbrowser, Musik abspielen) auf einem Prozessor. Ein Prozessor(kern) kann zu einem Zeitpunkt immer nur einen Prozess abarbeiten. Ein Programm wird also genau genommen nur dann ausgeführt, wenn es sich auf dem Prozessor befindet.

4.1 Single-Core

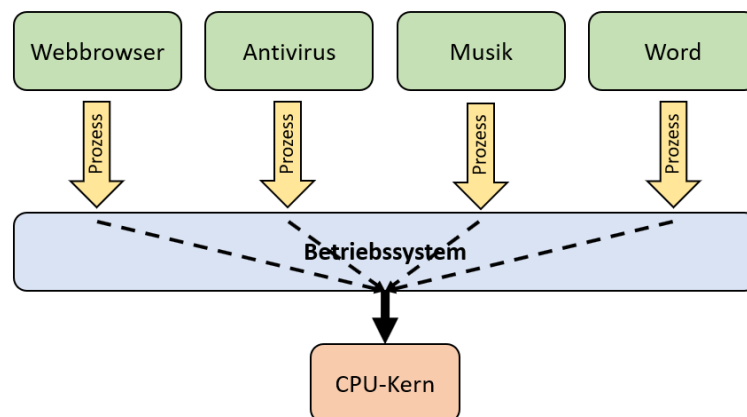


Abbildung 4-1: Single-Core

Bei einem Computer mit einem Single-Core-Prozessor wird immer nur ein Task auf einmal bearbeitet. Dennoch kommt es uns Anwendern so vor, als würden viele Programme parallel ablaufen.

Multitasking

Das Betriebssystem, welches die Ausführung der Prozesse kontrolliert, wendet einen Trick an. Er besteht darin, einzelne Prozesse jeweils nur für kurze Zeit (ungefähr 1...50 ms) zu bearbeiten und danach auf einen anderen Prozess umzuschalten. Man spricht auch von einer verschachtelten Bearbeitung (Zeit-Multiplexverfahren). Diese Abwechslung von Prozessen findet sehr schnell statt, so dass der Eindruck entsteht, als würde der Computer alle Anwendungen zur selben Zeit ausführen. In Wirklichkeit werden die Prozesse nur quasi-parallel ausgeführt. Dieses Konzept nennt sich in der Computertechnologie **Multitasking**.

Multithreading

Neben dem Begriff „**Multitasking**“ gibt es noch das „**Multithreading**“. Eine Anwendung bzw. ein Prozess kann nochmals in kleinere Unterprogramme, sogenannte Threads (englisch: „Faden“, „Strang“), untergliedert werden. Ein Thread ist also ein Teil eines Prozesses. Threads mit einem gemeinsamen Kontext gehören einem Prozess an. Sowohl Multithreading als auch Multitasking können auf nur einem Prozessorkern ausgeführt werden. Die Abarbeitung ist jedoch nur quasi-parallel, aber eben nicht echt-parallel.

4.2 Multi-Core

Eine echt-parallele Ausführung von Programmen kann nur mit Hilfe von mehreren Kernen erfolgen. Ein Multicore-System besteht aus einem Prozessor mit zwei oder mehreren Prozessorkernen, die unabhängig voneinander Prozesse verarbeiten können. Die Performance des Systems wird dadurch verbessert.

Multiprocessing

Können Prozesse, dank mehrerer CPU-Kerne, echt-parallel ausgeführt werden, spricht man nicht mehr vom Multitasking, sondern von „**Multiprocessing**“. In diesem Zusammenhang ist der Begriff des „**Multiprocessing**“ also aus „Multi“ (mehrere) und „Prozesse“ entstanden. Voraussetzung für Multiprocessing sind mindestens zwei unabhängige CPU-Kerne. In modernen Multicore-Systemen treten beide Formen, Multitasking und Multiprocessing, für die Abarbeitung von Prozessen auf. Je nachdem, wie die Kerne von den Programmen in Anspruch genommen werden (s. Abb. 4-2).

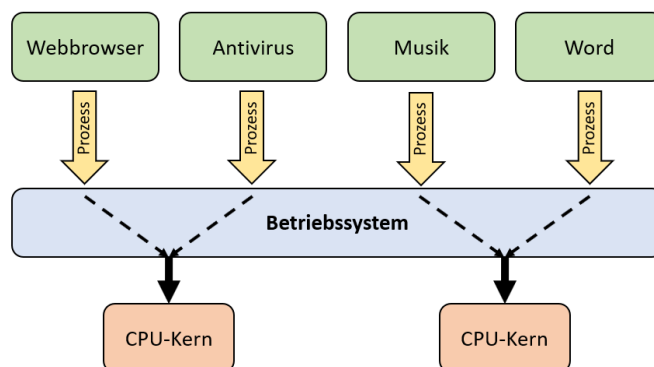


Abbildung 4-2: Multiprocessing kombiniert mit Multitasking

Wie schon erwähnt, wird ein Prozess selbst noch einmal in Unterprogramme zergliedert (Threads). Diese Threads werden vom Betriebssystem ebenfalls auf die CPU-Kerne verteilt, so dass echt-paralleles Multithreading erfolgt.

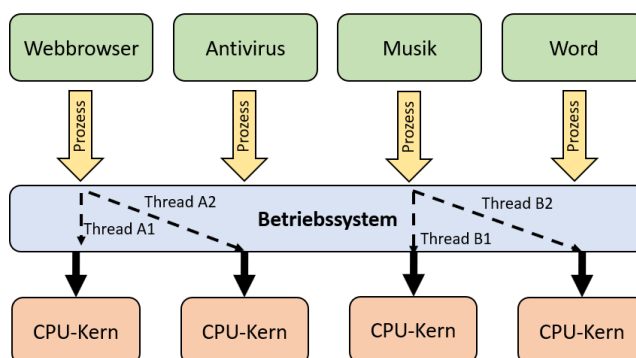


Abbildung 4-3: Multiprocessing kombiniert mit Multithreading

Abbildung 4-4 zeigt modellhaft den Aufbau eines Prozessors mit 4 Kernen. Wie man sieht, besitzt jeder CPU-Kern einen eigenen L1-Cache, welcher direkt mit im Kern verbaut ist. Weiterhin verfügt jeder Kern über einen Cache-Speicher vom Level 2. Gemeinsam greifen die Kerne auf den größeren L3-Cache zu.

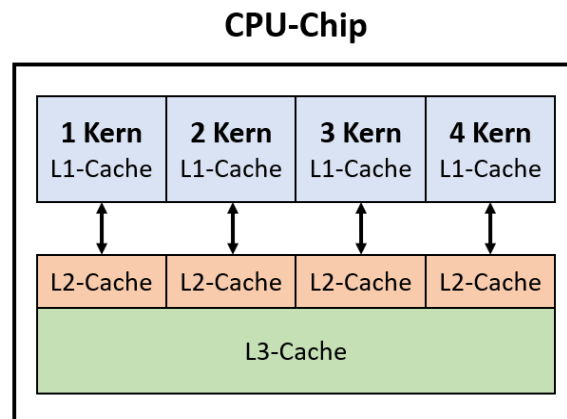


Abbildung 4-4: CPU mit 4 Kernen (Quad-Core)

Mehrprozessorsysteme (auch: Multiprozessorsysteme)

Eine konsequente echt-parallele Ausführung kann nur bei Rechnern mit mehreren Prozessoren garantiert werden. Besitzt ein Computer z.B. zwei unabhängige CPUs, können zwei Anwendungen ausgeführt werden, von denen jede einen eigenen Prozessor hat. Mehrere Prozesse können somit garantiert echt-gleichzeitig verarbeitet werden. Solche Systeme werden meist als Server oder Hochleistungs-Arbeitsplatzrechner genutzt.

4.3 Aufgabe „Multitasking, Multithreading, Multiprocessing“

Richtig oder falsch? Kreuzen Sie zu jeder Aussage die korrekte Antwort an.

- Sowohl Multithreading als auch Multitasking funktionieren nur bei Prozessoren mit mindestens zwei Kernen.
☐ Richtig ☐ Falsch
- Echt-paralleles Multithreading erfordert mindestens zwei Kerne.
☐ Richtig ☐ Falsch
- Threads, die voneinander abgeschottet sind, gehören verschiedenen Prozessen an.
☐ Richtig ☐ Falsch
- Ein Mehrprozessorsystem ist immer multiprocessing-fähig.
☐ Richtig ☐ Falsch
- Multiprocessing kann auch bei Single-Core-Systemen funktionieren.
☐ Richtig ☐ Falsch
- Eine Gruppe von Threads, die von allen anderen Threads, jedoch nicht voneinander, abgeschirmt sind, ist ein sogenannter Prozess, oft auch Task genannt.
☐ Richtig ☐ Falsch
- Wenn mehrere Aufgaben versetzt im Zeitscheibenverfahren von nur einem Rechenkern abgearbeitet werden, handelt es sich um Multitasking.
☐ Richtig ☐ Falsch
- Für Multiprocessing sind mehrere (mind. zwei), weitgehend unabhängige Prozessorkerne notwendig.
☐ Richtig ☐ Falsch
- In modernen Computersystemen werden heutzutage beide Verfahren gleichzeitig eingesetzt: Es sind mehrere Prozessorkerne vorhanden, die echt-gleichzeitig Threads ausführen können (voneinander abgeschottete oder im selben Kontext laufend) – zusätzlich werden jedem Prozessorkern für sich nochmals im Zeitscheibenverfahren Prozesse zugeteilt.
☐ Richtig ☐ Falsch

5 Von Bit und Byte

Der Computer speichert bzw. verarbeitet Informationen mit Hilfe von Spannungssignalen. Hierbei werden nur zwei Zustände unterschieden – entweder liegt eine Spannung an oder nicht. In der Informatik wird dieser Sachverhalt mit den Ziffern 1 und 0 zum Ausdruck gebracht:

- Logisch 1 bedeutet, dass eine Spannung anliegt
- Logisch 0 bedeutet, dass keine Spannung anliegt

Die kleinste Informationseinheit ist das **Bit**. Ein einzelnes Bit kann gesetzt sein (Spannung an), es steht somit auf logisch 1 oder es ist nicht gesetzt (Spannung aus), es steht dann auf logisch 0. **Acht Bit** ergeben eine Gruppierung, die als **Byte** bezeichnet wird.

Ein mögliches Byte könnte folgendermaßen aussehen:

1	0	0	1	1	0	1	1
---	---	---	---	---	---	---	---

Wie werden die zwei Zustände logisch 1 und logisch 0 in einem PC realisiert?

Hier einige Beispiele, wie Bits abgebildet werden:

5.1 Abbildung von logischen Zuständen in einem PC-System

Hauptspeicher (RAM)

Im Hauptspeicher (RAM) befinden sich winzige Kondensatoren. Ein Kondensator ist ein elektrisches Bauteil, das in der Lage ist, eine Ladung aufzunehmen und zu speichern. Vergleichbar mit einer Mini-Batterie. Für das Setzen und Auslesen der Bit ist ein Input-Output-Controller (I/O-Controller) zuständig. Soll eine logische 1 gesetzt werden, legt der I/O-Controller für einen kurzen Augenblick eine Spannung an den Kondensator an. Der Kondensator nimmt dann Ladungsträger auf und bewahrt diese.

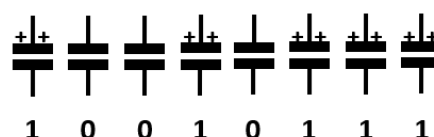


Abbildung 5-1: Bitmuster beim RAM

Beim Auslesen fließt dann die gesetzte Ladung wieder zurück zum I/O-Controller. Falls auf einem Kondensator keine Ladung vorhanden ist, handelt es sich um eine logische 0. So erkennt der I/O-Controller das Bitmuster.

Festplatte (Hard Disc Drive, HDD)

Eine Festplatte besteht aus rotierenden Scheiben. Auf den Scheiben befinden sich schmale Laufbahnen, in denen magnetisierbare Minipartikel liegen. Sehr dicht über der Platte schwebt ein Schreib-/Lesekopf, der aus einem Eisenkern mit einer Spule besteht. Beim Schreiben von Daten erzeugt der Schreib-/Lesekopf Magnetfelder, welche die Mini-Magnete ausrichten. Entweder nach links oder nach rechts. An den Stellen, an denen ein Wechsel der Ausrichtung stattfindet, liegt eine logische 1 an. Ansonsten logische 0en.

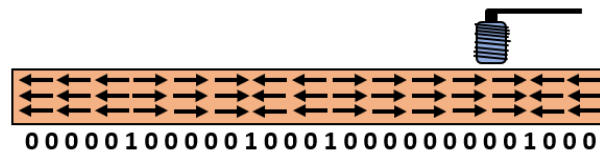


Abbildung 5-2: Bitmuster bei einer HDD

Aufgrund der geordneten Ausrichtung der Minimagnete erzeugen diese nun selbst kleine Magnetfelder. Beim Auslesen fährt der Schreib-/Lesekopf wieder über die Laufbahn und erkennt genau die Stellen, an denen ein Polaritätswechsel stattfindet.

Optische Datenträger (CD, DVD, Blu-ray)

Bei einem optischen Datenträger (CD, DVD, Blu-ray) wird ein Muster erzeugt, welches aus Höhen und Tiefen besteht. Man spricht auch von Pits (engl. Grube) und Lands (engl. Boden). An den Stellen, an denen ein Wechsel zwischen Pit und Land bzw. Land und Pit stattfindet, liegt eine logische 1 an. Zwischen den Kanten befinden sich logische 0en.

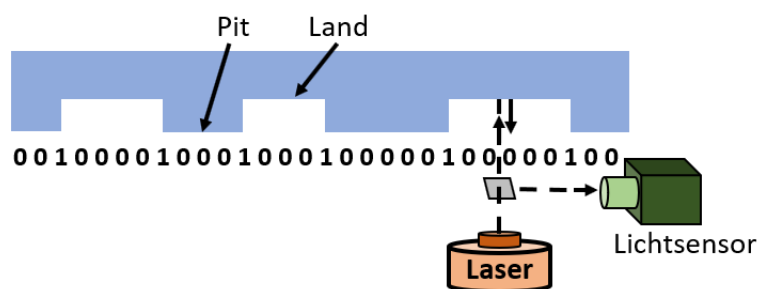


Abbildung 5-3: Bitmuster bei einer CD/DVD/Blu-ray

Das Auslesen des Musters bzw. der Daten erfolgt durch einen Laser mit Hilfe von Licht. Darum auch der Begriff „Optische Datenträger“. Der Lichtstrahl trifft auf die Oberfläche des Trägers und wird reflektiert. Das reflektierte Licht trifft anschließend auf einen halbdurchlässigen Spiegel, der den Lichtstrahl zu einem Lichtsensor umlenkt. Die Kanten im Muster sorgen dafür, dass der Lichtstrahl für

einen kurzen Augenblick unterbrochen wird. Genau diese Momente detektiert der Lichtsensor, so dass eine logische 1 erkannt wird.

Das waren drei Beispiele, die eben verdeutlichen sollen, wie digitale Daten abgebildet werden.

Neben den reinen Nutzdaten gehören noch viele andere Daten in Form von Zusatzinformationen.

Denn diese Bit bzw. Byte können alles Mögliche darstellen:

- einen numerischen Zahlenwert
- einen Buchstaben
- die Farbe von einem Pixel etc.

5.2 Anzahl der Bit-Zustände

In diesem Kapitel soll geklärt werden, wie viele Zustände mit einer bestimmten Anzahl von Bit dargestellt werden können. Zur Veranschaulichung stellen wir uns für ein Bit eine Glühlampe vor.

Diese kann zwei Zustände haben:

Zustand 1
Lampe ist aus



Zustand 2
Lampe ist an



Mit Hilfe einer Glühlampe könnte man Ja/Nein-Fragen bzw. Wahr/Falsch-Fragen beantworten.

Sender und Empfänger müssen sich lediglich auf eine Zuordnung einigen. Sie definieren einen gemeinsamen Code. Lampe aus könnte die Antwort „Nein“ (Falsch) und Lampe ein die Antwort „Ja“ (Wahr) bedeuten.

Beantworten Sie folgende Fragen, indem Sie die Lampe gelb ausmalen oder nicht.

1) Die Hauptstadt von Spanien ist Barcelona.



2) Die Wurzel aus 81 ist 9.



3) Der Durchmesser der Erde beträgt 12.742 Km



4) In der Klausur schreibe ich mindestens die Note 2.



Für einfache Ja-Nein-Fragen genügt eine Lampe. Möchte man jedoch die Lage deutscher Städte mit den Himmelsrichtungen Norden, Osten, Süden, Westen beschreiben, kommt man mit einer Glühlampe nicht mehr aus, wenn die Antwort auf „einem Blick“ entnommen werden soll.

Um die Frage „Wo liegt Hamburg innerhalb von Deutschland“ zu beantworten, werden zwei Lampen benötigt. Es ergeben sich vier Kombinationsmöglichkeiten:



Auch hier muss für Sender und Empfänger die Bedeutung der Signale bekannt sein. Andere Zuordnungen wären denkbar.

Wie viele Zustände können Sie mit drei Lampen darstellen? Wie viele mit vier Lampen? Füllen Sie die Tabelle aus. Wie lässt sich die Anzahl rechnerisch ermitteln?

Anzahl Lampen	1	2	3	4	5	6	7	8	9
Anzahl Zustände									
Rechnung									

5.2.1 Aufgabe „Wie viele Bit werden benötigt?“

1) Wie viele Zahlen könnte man mit seinen beiden Händen darstellen (10 Finger)?

2) Geben Sie für die folgenden Modelle die gebrauchte Anzahl an Glühlampen an, um die Vielfalt der Zustände abdecken zu können.

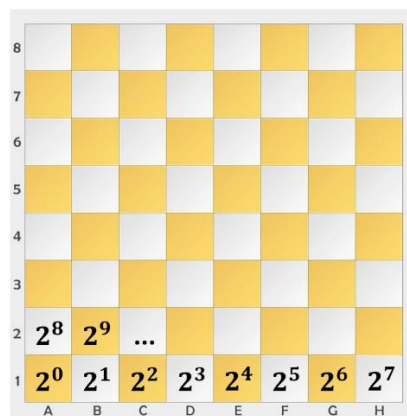
- a) Sie sollen das deutsche Alphabet (nur Großbuchstaben, ohne Umlaute) abbilden.
- b) Aus einem Kartenspiel (32 Karten) wird eine Karte gezogen.
- c) Die Position auf einem Schachbrett soll bestimmt werden.
- d) Eine einzelne Schachfigur wird auf irgendein Feld des Schachbrettes gesetzt. Die Farbe spielt hierbei keine Rolle. Es kommt nur auf den Figurentypen an (Bauer, Läufer, ...)
- e) Eine einzelne Schachfigur wird auf irgendein Feld des Schachbrettes gesetzt. Dabei muss jede einzelne Spielfigur aus dem Spiel unterscheidbar sein.

- f) Ein Buch umfasst 346 Seiten. Jede Seite besitzt maximal 28 Zeilen und in jeder Zeile stehen maximal 22 Wörter. Aus dem Buch soll ein ganz bestimmtes Wort bestimmt werden.
- g) Ein Kino besitzt 8 Kinosäle. Jeder Saal umfasst maximal 18 Sitzreihen. Eine Sitzreihe besteht aus maximal 24 Sitzplätzen. In jedem Saal finden maximal 5 Filmvorführungen statt. Für eine Vorstellung soll nun ein ganz bestimmter Sitz reserviert werden.

5.2.2 Aufgabe „Schach und Reis“

Stellen Sie sich folgendes Szenario vor: Gegeben ist ein Schachbrett. Auf das erste Feld des Schachbrettes (A1) wird 1 Reiskorn gelegt. Auf das zweite Feld (B1) werden 2 Reiskörner gelegt. Auf das dritte Feld kommen 4 Reiskörner, dann 8, 16, 32 u.s.w., bis zum letzten Feld H8.

Die Reihe entspricht der Zahlenfolge des Binärsystems ($2^0, 2^1, 2^2 \dots 2^n$)



8								
7								
6								
5								
4								
3								
2	2^8	2^9	...					
1	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7
	A	B	C	D	E	F	G	H

Betrachten Sie die gesamte Menge der Reiskörner, die dann (rein hypothetisch) auf dem Schachbrett liegt.

Angenommen, jeder Deutsche isst einmal in der Woche eine Portion Reis (100g). Wie viele Wochen kann sich die deutsche Bevölkerung von dieser Reismenge ernähren, wenn 46 Reiskörner 1g wiegen.

Um diese Frage beantworten zu können, müssen Sie vorab noch andere Fragen klären. Worüber müssen Sie sich informieren? Welche Infos fehlen Ihnen noch?

Halten Sie alle Ihre Ergebnisse schriftlich fest.

6 Speicherangaben in Bit und Byte

Die kleinste Informationseinheit in der digitalen Welt ist das Bit. **Acht Bit** ergeben ein **Byte**.

Aufgabe: Wie viel Byte entsprechen wie viel Bit – und umgekehrt. Füllen Sie die Tabelle aus!

Angabe in Bit	Angabe in Byte
72	
	14
	340
424	

Nehmen Werte eine große Dimension an, verwendet man andere Größeneinheiten. So vermeidet man die Angabe von langen Zahlen. Keiner würde z.B. sagen: „Ich wiege 70000 Gramm“. Stattdessen gibt es die nächste Größeneinheit Kilogramm. Genauso würde man nicht sagen, dass die Entfernung zwischen Hilden und Berlin 550.000.000 Millimeter beträgt.

Die Größenordnungen Kilo (k), Mega (M), Giga (G), Tera (T) etc. findet sich auch bei Speicherangaben wieder. Der Faktor beträgt auch hier stets **1000**, also **1000 Kilo = 1 Mega**, **1000 Mega = 1 Giga**

Aufgabe: Schreiben Sie die Werte in die gesuchten Größeneinheiten um.

8000Bit → 8 kBit → 1 kByte

48.000.000Bit → 48000 kBit → 48 Mbit → 6 MByte

512.000.000.000Bit → 512000000 kBit → 512000 Mbit → 512 GBit → 64 GByte

7,2MByte → 7200 kByte → 7200000 Byte → 57600000 Bit

3,2GByte → 3200 MByte → 3200000 kByte → 25600000000 Bit

Kurzschreibform:

Für Byte wird in der Regel der Großbuchstabe „B“ verwendet.

Für Bit wird in der Regel der Kleinbuchstabe „b“ benutzt.

Ich denke
binär!



Jetzt ist es jedoch so, dass ein Computer mit dem Binärsystem arbeitet.
Folglich ist 1 Kilo für den PC nicht 1000, sondern $2^{10} = 1024$.

Somit sind auch $1024k = 1M$ und $1G = 1024M$.

Damit man unterscheiden kann, ob 1 Kilo nun 1000 oder 1024 bedeutet, gibt es andere Bezeichnungen, wenn mit dem Binärsystem gerechnet werden soll.

Kilo heißt dann Kibi, Mega heißt dann Mebi und Tera heißt dann Tebi. Hinter den Buchstaben wird dann einfach ein i drangehängt (ki, Mi, Gi, Ti).

Hier noch einmal die Größeneinheiten im Vergleich:

Der Hersteller rechnet mit dem Faktor 1000:

- $1k = 1000 = 10^3$ (Kilo)
- $1M = 1000k = 1000 * 1000 = 10^3 * 10^3 = 10^6$ oder 1000^2 (Mega) $\rightarrow 1.000.000$
- $1G = 1000M = 1000 * 1000 * 1000 = 10^3 * 10^3 * 10^3 = 10^9$ oder 1000^3 (Giga) $\rightarrow 1.000.000.000$

PC-Angaben beruhen auf dem Faktor 1024:

- $1ki = 1024 = 2^{10}$ (Kibi) $\rightarrow 1024$
- $1Mi = 1024ki = 2^{10} * 2^{10} = 2^{20}$ oder $1024 * 1024 = 1024^2$ (Mebi) $\rightarrow 1.048.576$
- $1Gi = 1024Mi = 2^{10} * 2^{10} * 2^{10} = 2^{30}$ oder $1024 * 1024 * 1024 = 1024^3$ (Gibi) $\rightarrow 1.073.741.824$

Aufgabe: Füllen Sie die Tabelle aus. Schreiben Sie in die mittlere Spalte die Speichergröße mit Hilfe der Potenzangabe hinein, z.B. ist Kilo = 10^3 . In die Spalte rechts kommt die ausgeschriebene Anzahl der Byte hinein.

Speichergröße	Schreibweise als Potenz	Anzahl der Bytes ausgeschrieben
1,8 kByte	$1,8 * 10^3 \text{Byte}$	1800Byte
5 kiByte	$5 * 2^{10} \text{Byte}$	5120Byte
26,2 MByte		
4,6 GByte		
12 MiByte		
6 GiByte		
0,5 GiByte		

6.1.1 Aufgabe „Kundendaten“

Jeden Tag melden sich Personen bei einem Onlineportal für Lehrvideos an. Bei der Anmeldung müssen die Personen verschiedene Angaben machen. Die Daten werden von der Firma nach den Richtlinien der DSGVO abgespeichert. Ermitteln Sie den Speicherplatzbedarf für die Kundendaten für 1 Monat (30 Tage). Gehen Sie bei der Rechnung vom folgenden Sachverhalt aus:

- Die Angaben zu einer Person belegen im Schnitt 128 Byte
- Täglich melden sich ca. 300 neue Personen an

Geben Sie den Wert in Wert in MiB an. Runden Sie auf zwei Stellen nach dem Komma.

6.1.2 Aufgabe „Kamerabilder“

Zwei Überwachungskameras nehmen jede Sekunde ein Bild von einem Raum auf. Die Größe eines Bildes beträgt 34kiB. Die Bilder werden auf einer Festplatte gespeichert. Berechnen Sie den Speicherplatzverbrauch für die Bilder für einen Zeitraum von 8 Stunden. Geben Sie den Wert in GiB an. Runden Sie auf zwei Stellen nach dem Komma.

6.1.3 Aufgabe „Videodatei“

Ein Video wird mit einer hochwertigen Videokamera aufgenommen. Die Bildgröße beträgt 3840 x 2160 Pixel. Jeder Pixel hat einen Farbwert von 24 Bit. Man spricht auch von der Farbtiefe. Die Bildwiederholungsrate liegt bei 50fps. Berechnen Sie die Größe der Videodatei für eine Aufnahmezeit von 1 Minute. Geben Sie den Wert in Wert in GiB an. Runden Sie auf zwei Stellen nach dem Komma. Berücksichtigen Sie, dass die Videodatei mit Hilfe eines Kompressionsverfahrens von 1:10 abgespeichert wird.

7 Die Übertragung von Bits

7.1 Aufbau von digitalen Signalen

Bits werden in Form von Spannungspegeln über Leitungen (meistens aus Kupfer) von einer Komponente zur nächsten Komponente weitergegeben.

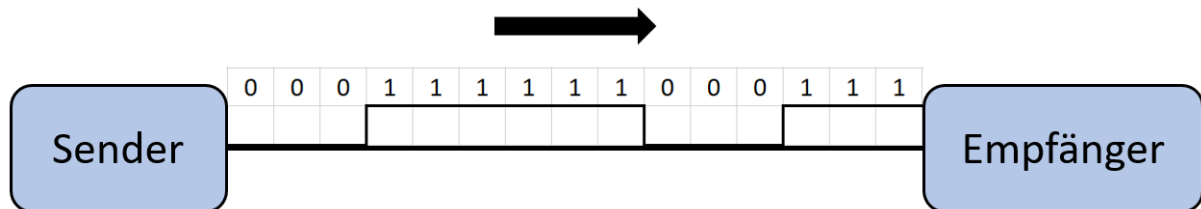
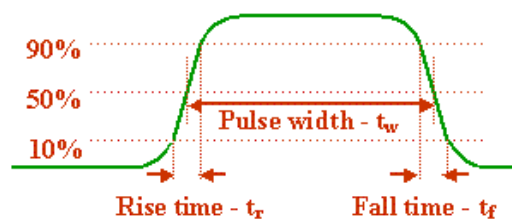


Abbildung 7-1: Übertragung von Bits a)

Misst der Empfänger eine Spannung, wird dieser Wert als eine logische 1 interpretiert. Wird keine Spannung gemessen, so handelt es sich um den logischen Zustand 0.



In Wirklichkeit haben die digitalen Datensignale nicht so eine ideale Rechteckform. Auch ein digitales Signal muss sich erst aufbauen. Die Spannung steigt zunächst an (Rise time), dann muss der Spannungswert für einen gewissen Zeitwert gehalten werden (Pulse width). Anschließend braucht es etwas Zeit, bis die Spannung komplett abgefallen ist (Fall time).



Bei der Übertragung ist es wichtig, dass der Empfänger das Datensignal zum richtigen Zeitpunkt erfasst. Dieser liegt in der Mitte der Pulsbreite (Pulse Width). Wird das Signal zu weit links (siehe Rise time) oder zu weit rechts (siehe Fall time) abgetastet, misst der Empfänger womöglich einen Spannungswert, der nicht zum eigentlichen logischen Wert führt.

7.2 Taktleitung nutzen

Woher weiß der Empfänger, wann die Mitte eines Signals erreicht ist?

Betrachtet man noch einmal das Signal aus der Abbildung 11 der vorherigen Seite, könnte der Empfänger das Signal auch wie folgt auslesen:

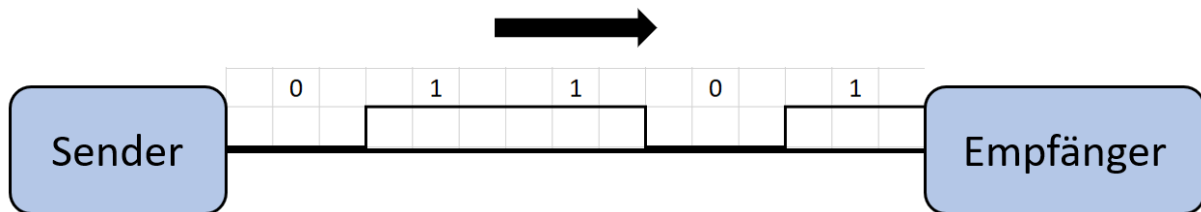


Abbildung 7-2: Übertragung von Bits b)

Woher weiß der Empfänger, wie viele logische Zustände auf dem Datensignal liegen? Sender und Empfänger müssen synchron miteinander arbeiten. Das gelingt, wenn sich die Teilnehmer nach einem gemeinsamen Takt richten. Es wird somit eine zusätzliche Leitung benötigt, auf der in perioden Abständen ein Taktsignal gesendet wird.

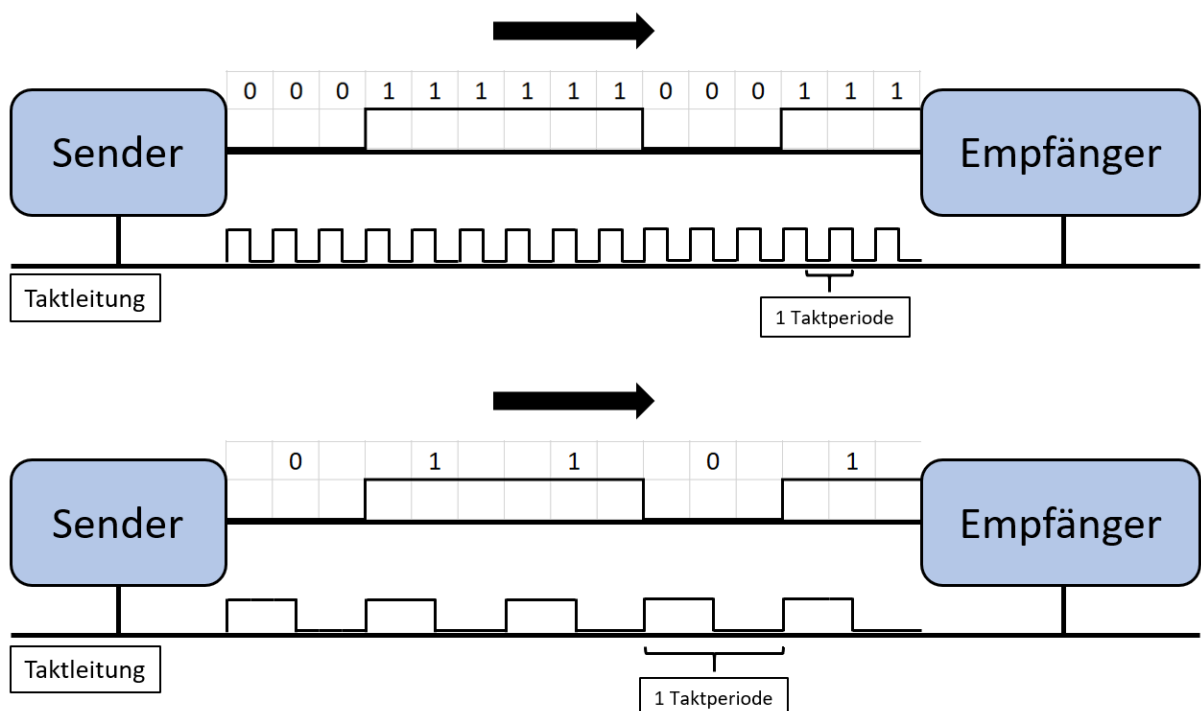


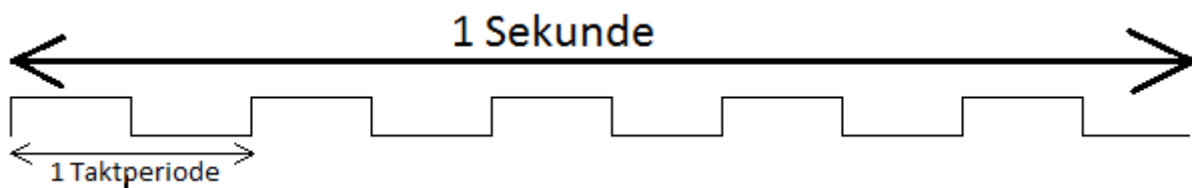
Abbildung 7-3: Übertragung mit Taktleitung

Sender und Empfänger sind so konzipiert, dass die Bits bei der aufsteigenden Flanke des Taktsignals gesendet bzw. abgetastet werden. Die Frequenz des Taktsignales steht somit in direktem

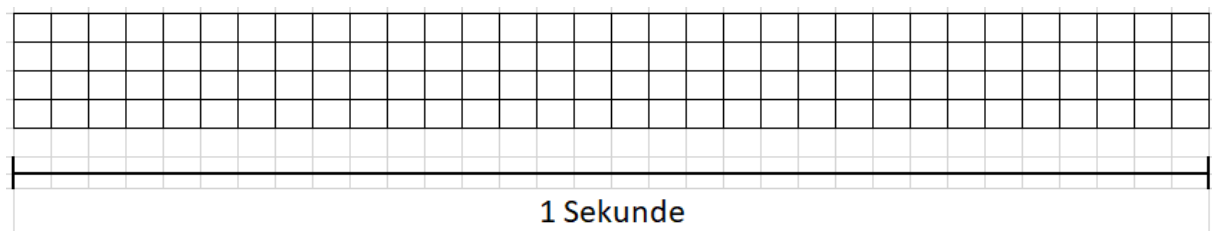
zusammenhang mit der Anzahl der Bits, die pro Sekunde übertragen werden. Pro Taktperiode wird ein Bit gesendet. Abbildung 7-3 zeigt zwei mögliche Lösungen mit verschiedenen Taktraten.

Angenommen, der Datenstrom umfasst den Zeitraum von 1 Sekunde. Der obere Takt hätte dann eine Frequenz von 15 Hz [$\text{Hz} = \text{Hertz} = \frac{1}{\text{Sekunde}} = \frac{1}{s}$]. Es gibt 15 Schwingungen bzw. Taktperioden und somit liegt die Transferrate bei 15 Bit pro Sekunde (15Bit/s).

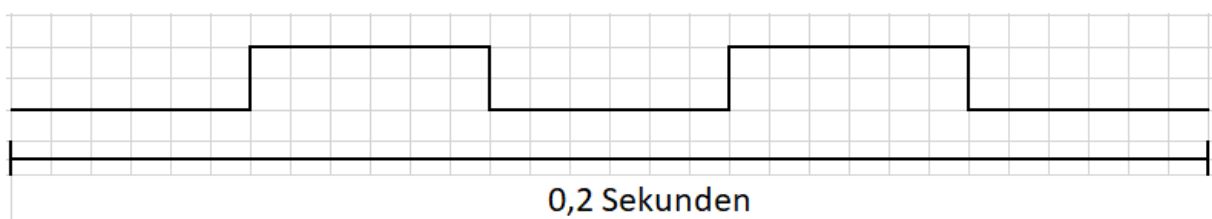
Beim zweiten Beispiel lassen sich 5 Taktperioden zählen. Die Transferrate beträgt somit 5 Bit/s. Hier noch einmal die Darstellung einer 5Hz-Frequenz.



Aufgabe: Zeichnen Sie in das Feld ein Taktsignal von 8Hz ein.



Aufgabe: Ermitteln Sie aus der Abbildung die Frequenz des Taktsignals.



7.3 Berechnung von Transferraten

Bits werden über Leitungen von einer Komponente zur nächsten Komponente weitergegeben. Die Übertragung kann auf zwei unterschiedlichen Arten erfolgen:

- seriell
- parallel

7.3.1 Serielle Datenübertragung

Bei einer seriellen Datenübertragung existiert in der Regel nur eine einzelne Datenleitung. Über diese Datenleitung werden die Bits einzeln hintereinander versendet. Die Anzahl der Bit, die pro Sekunde übertragen werden, bezeichnet man bei seriellen Verbindungen als Transferrate. Die Einheit ist T/s (Transfers pro Sekunde). Das ist gleichbedeutend mit bit/s.

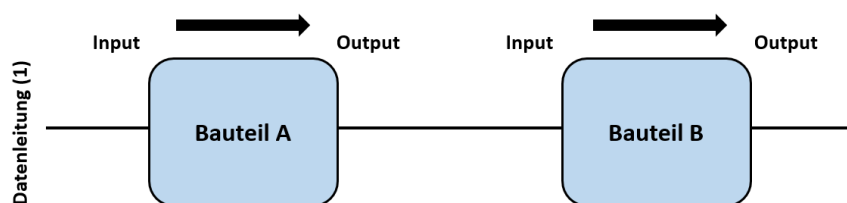


Abbildung 7-4: Serielle Datenübertragung

Aufgabe: Füllen Sie die rechte Spalte aus. Wie hoch ist die Transferrate? Gehen Sie davon aus, dass stets bei der aufsteigenden Flanke des Taktsignals ein Bit übertragen wird.

Anzahl der Leitungen	Taktrate [Hz]	Transferrate [bit/s]
1	500	500 bit/s
1	1400	1400 bit/s
1	4600	4600 bit/s

7.3.2 Parallele Datenübertragung

Für eine parallele Datenübertragung werden mehrere Datenleitungen benötigt. Die Bits können dann parallel über die Leitungen verschickt. Vergleichbar mit einer mehrspurigen Straße, auf der mehrere Autos parallel nebeneinander herfahren. Die Anzahl der Bit, die pro Sekunde übertragen werden, bezeichnet man bei parallelen Verbindungen in der Regel als Datenübertragungsrate (kur: DÜR). Die Einheit ist bit/s (kurz: b/s, aber eher selten verwendet) bzw. Byte/s (kurz: B/s).

Hier ein Beispiel von einer Schnittstelle mit 16 Datenleitungen.

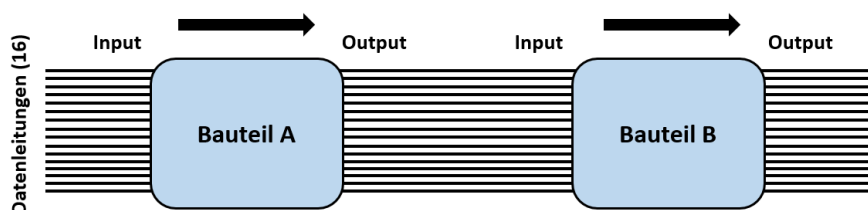


Abbildung 7-5: Parallele Datenübertragung

Aufgabe: Berechnen Sie die Datenübertragungsrate (DÜR) einmal in bit/s und in Byte/s

Anzahl der Leitungen	Taktrate [Hz]	Transferrate [bit/s]	Transferrate [Byte/s]
16	30	480 bit/s	60 byte/s
32	260	8320 bit/s	1040 byte/s
64	3200	204800 bit/s	2560 byte/s

7.3.3 Hohe Taktraten

Pro Sekunde werden von der CPU und anderen Komponenten Millionen von Anweisungen verarbeitet. Dementsprechend haben diese Takte eine Größe, die in den Giga-Bereich geht.

Aufgabe: Schreiben Sie die folgenden Werte in die gewünschte Einheit um: **Kilo (k)** – **Mega (M)** – und **Giga (G)**. Hinweis: Der Umrechnungsfaktor von einer benachbarten Größe zur anderen beträgt 1000!

1kHz = 1000 Hz 1MHz = 1000000 Hz 1 GHz = 1000000000 Hz
 1200Hz = 1,2 kHz 24.500 Hz = 24,5 kHz 1.125.000Hz = 1125 kHz
 1.125.000Hz = 1,125 MHz 1540kHz = 1,54 MHz 12.600 MHz = 12,6 GHz

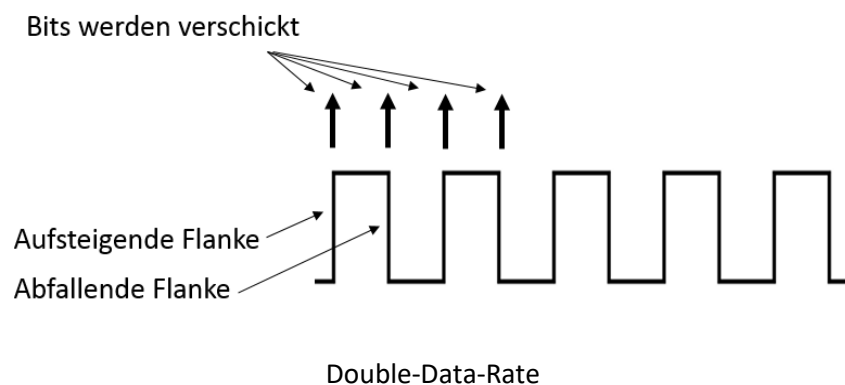
Aufgabe) Berechnen Sie die Datenübertragungsrate (DÜR) einmal in bit/s und in Byte/s. Wandeln Sie das Ergebnis direkt in eine angemessene Größe (Kilo, Mega, Giga) um.

Anzahl der Leitungen	Taktrate	Transferrate [bit/s]	Transferrate [Byte/s]
1	3,2GHz	3,2 gbit/s	400 mbyte/s
16	260MHz	4,16 gbit/s	520 mbyte/s
32	400MHz	12,8 gbit/s	1,6 gbyte/s
64	200kHz	12,8 mbit/s	1,6 mbyte/s
1	4GHz	4 gbit/s	500 mbyte/s

Aufgabe: Die langjährige Standard-Schnittstelle für Laufwerke (Festplatten, namens Parallel ATA (kurz: P-ATA oder PATA) besitzt eine 16-Bit-Datenübertragung. Die Datenübertragungsrate beträgt 133MB/s. Berechnen Sie die Taktrate.

7.3.4 Single-Data-Rate und Double-Data-Rate

Wie bereits schon erwähnt, werden die Bits bei der aufsteigenden Taktflanke übertragen. Dieses Prinzip nennt sich Single-Data-Rate (SDR). Viele Bussysteme beherrschen jedoch eine andere Technik, die sich Double-Data-Rate (DDR) nennt. Hier werden die Bits bei der aufsteigenden und bei der abfallenden Flanke verschickt. Pro Taktzyklus werden also zwei Bit übertragen.



Aufgabe: Berechnen Sie die Datenübertragungsrate (DÜR) einmal in bit/s und in Byte/s. Wandeln Sie in eine angemessene Größe (Kilo, Mega, Giga) um. Achten Sie auf den Typ der Data-Rate!

Anzahl der Leitungen	Taktrate	Data-Rate	Transferrate [bit/s]	Transferrate [Byte/s]
4	1,25GHz	DDR		
16	200MHz	SDR		
2	2,5GHz	DDR		
64	266MHz	SDR		
1	4GHz	DDR		

8 Die Dauer einer Datenübertragung

Soll eine Datei von einem Medium auf ein anderes Medium übertragen werden, kann die theoretische Dauer der Datenübertragung berechnet werden. Grundsätzlich gilt:

$$\text{Dauer} = \frac{\text{Dateigröße}}{\text{DÜR}}$$

Die Dateigröße kann in unterschiedlichster Form angegeben sein: kb, kibit, kB, kiB, Mbit, etc. Hierbei muss beachtet werden, ob der Wert zur Basis 10 bzw. 1000 oder zur Basis 2 bzw. 1024 gehört.

Die DÜR ist entweder in bit/s oder B/s in den entsprechenden Größen Kilo, Mega, Giga, etc. angegeben. Beachten Sie, dass eine DÜR stets auf dem Faktor 1000 beruht.

8.1.1 Aufgabe „Dauer Kopiervorgang“

- 1) Eine Datei der Größe von 800MB soll auf ein anderes Speichermedium kopiert werden. Die Datenübertragungsrate beträgt 40MB/s. Wie lange dauert der Kopiervorgang?
- 2) Eine Datei der Größe 800MB soll auf ein anderes Speichermedium kopiert werden. Die Transferrate beträgt 40Mbit/s. Wie lange dauert der Kopiervorgang?
- 3) Eine Datei der Größe von 800MiB soll auf ein anderes Speichermedium kopiert werden. Die Datenübertragungsrate beträgt 40MB/s. Wie lange dauert der Kopiervorgang?
- 4) Eine Datei der Größe 800MiB soll auf ein anderes Speichermedium kopiert werden. Die Transferrate beträgt 40Mbit/s. Wie lange dauert der Kopiervorgang?
- 5) Eine Datei der Größe 2,4GB soll auf ein anderes Speichermedium kopiert werden. Die Datenübertragungsrate beträgt 90MB/s. Wie lange dauert der Kopiervorgang?
- 6) Eine Datei der Größe 2,4 GiB soll auf ein anderes Speichermedium kopiert werden. Die Transferrate beträgt 300Mbit/s. Wie lange dauert der Kopiervorgang?

8.1.2 Die Datenrate einer Internetleitung

Wenn es um die Schnelligkeit einer Internetverbindung geht, so benutzt man häufig Ausdrücke wie: „Ich habe eine 16er-Leitung, mein Bruder hat eine 50er-Leitung, und mein Onkel hat eine 100er-Leitung.“

Was bedeuten diese Ausdrücke wie „16er-Leitung“ oder „50er-Leitung“?

Gemeint ist die Anzahl der Bits in Mega, die in einer Sekunde übertragen werden können. Das bedeutet:

Leitung	Transferrate
16er	$16 \frac{Mbit}{s} = 16 * 10^6 \frac{bit}{s} = 16\,000\,000 \frac{bit}{s}$
50er	$50 \frac{Mbit}{s} = 50 * 10^6 \frac{bit}{s} = 50\,000\,000 \frac{bit}{s}$
100	$100 \frac{Mbit}{s} = 100 * 10^6 \frac{bit}{s} = 100\,000\,000 \frac{bit}{s}$

Hierbei werden die Bits seriell und nicht parallel verschickt. Die Bits „laufen“ also hintereinander durchs Kabel. Bei einer seriellen Übertragung spricht man häufig von einer **Transferrate**, die in **bit/s** angegeben wird.

8.1.3 Aufgabe „Dauer eines Downloads“

- 1) Gegeben ist eine DSL 50er-Leitung. Wie lange dauert der Download einer Datei mit einer Größe von 24 MiB?
- 2) Gegeben ist eine DSL 200er-Leitung. Wie lange dauert der Download einer Datei mit einer Größe von 1,6 GiB?
- 3) Gegeben ist eine DSL 400er-Leitung. Wie lange dauert der Download einer Datei mit einer Größe von 0,4 GiB?

9 Aufbau des Mainboards/Motherboards

Das Mainboard (auch Motherboard genannt) ist die Hauptplatine eines Computers, auf dem die grundlegende Elektronik befestigt ist.

Auf einem Mainboard werden alle anderen Hardware-Komponenten aufgesteckt oder per Kabel angeschlossen:

- Dazu gehören etwa CPU, RAM, Grafikkarte, Festplatte(n) und Lüfter.
- Auf modernen Mainboards sind die meisten Hardware-Komponenten wie Soundkarte, (einfache) Grafikkarte und Netzwerkkarte bereits integriert.
- Daher nennt man sie dann auch Onboard-Grafikkarte, Onboard-Soundkarte etc.

Mainboards haben unterschiedliche Größen für kleine oder große Gehäuse. In PCs nutzt man standardmäßig die Größe ATX. Für kleinere Computer genügt ein entsprechendes kleineres Mainboard. Die Größe wird auch als Formfaktor bezeichnet. Ein Formfaktor in einem PC legt nicht nur allein die Maße des Mainboards fest. Der Formfaktor spezifiziert auch über welche Größe und Befestigungsmöglichkeiten die Hardware-Komponenten verfügen müssen. Neben Mainboards haben auch Festplatten einen Formfaktor. Hier kennt man vor allem 2,5- und 3,5-Zoll. Neben der reinen Größenangabe spielen aber auch die Bohrungen für die Befestigungen des Mainboards mit dem Gehäuse eine Rolle. Die Löcher sind hierbei nicht willkürlich gewählt, sondern ermöglichen, dass auch kleinere Mainboards in größere Gehäuse befestigt werden können.

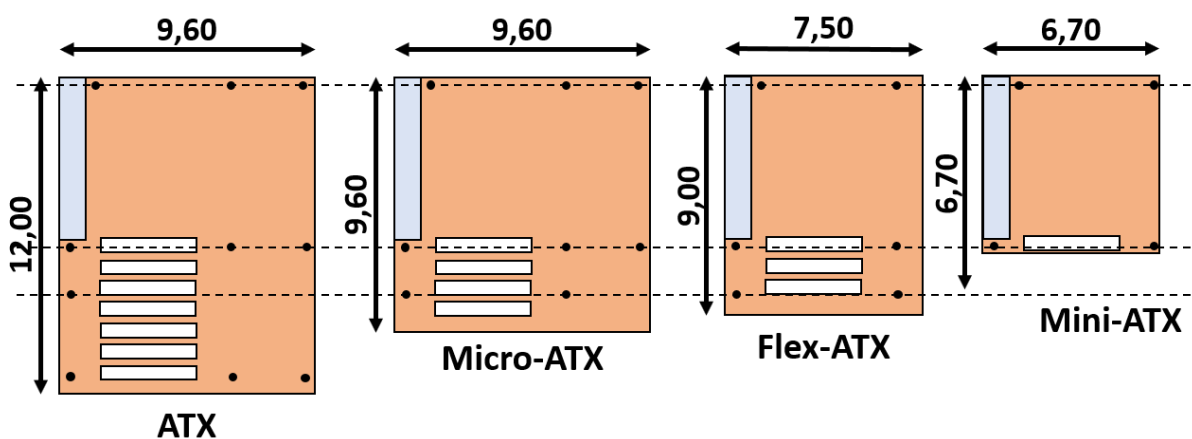
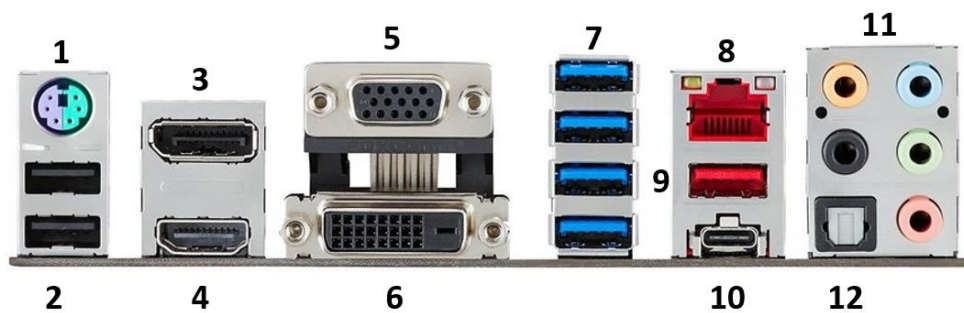


Abbildung 9-1: Mainboard Formfaktoren

9.1 Die rückseitigen Anschlüsse

Aufgabe: Kennen Sie diese Anschlüsse? Füllen Sie die Tabelle aus.



Nummer	Bezeichnung	Verwendung
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		

9.2 Anschlüsse im Detail

Die Anschlüsse HDMI, DisplayPort und Thunderbolt übertragen Bild- und Audiosignale. Aber worin unterscheiden sich diese drei Anschlüsse? Erstellen Sie für jede Schnittstelle kleine Steckbriefe mit wichtigen Eigenschaften.

HDMI



Steckbrief:

DisplayPort



Steckbrief:

Thunderbolt



Steckbrief:

9.3 Anschlüsse und Komponenten auf dem Mainboard

Aufgabe: Kennen Sie diese Anschlüsse bzw. Komponenten? Füllen Sie die Tabelle auf der nächsten Seite aus.

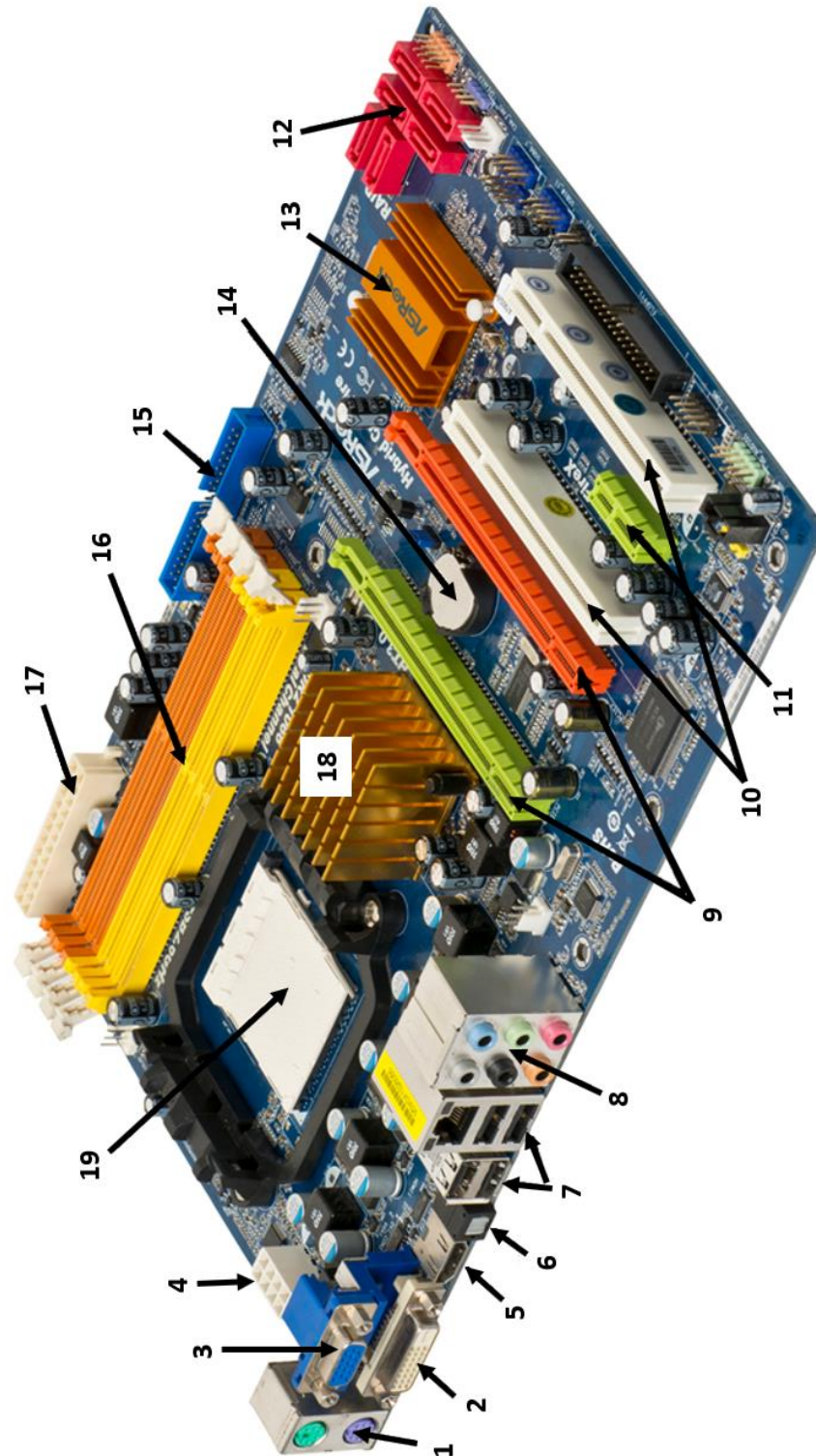


Abbildung 9-2: Übersicht eines Mainboards; Quelle: www.pixabay.com

Nummer	Bezeichnung	Verwendung
1	PS/2	Maus/Tastatur
2	DVI	Bildschirm
3	VGA	Bildschirm
4	CPU Strom	Strom für CPU
5	HDMI	Bildschirm
6	S/PDIF	Digitale Soundübertragung
7	USB	Universelle Datenübertragung
8	3.5mm Chinch	Analoge Soundübertragung
9	PCIE x16	PCIE Karten die viele Daten übertragen müssen, z.B. Grafikkarten
10	PCI	PCI Karten
11	PCIE x1	PCIE Karten die nicht so viele Daten übertragen
12	SATA	Speichermedien
13	Southbridge	
14	CMOS Batterie	Speichert Die BIOS Einstellungen
15	IDE Laufwerk	
16	DIMM-Slot	RAM
17	Stromversorgung	Versorgt das Mainboard mit Strom
18	Kupfer Kühlkörper	Kühlt das Mainboard
19	CPU-Slot	CPU

9.4 Chipsätze

Das zentrale Element auf einem Motherboard ist der Chipsatz (engl. Chipset). Zwar ist der Hauptprozessor das schlagende Element in einem Computer, aber der Chipsatz sorgt erst dafür, dass die verschiedenen Komponenten miteinander kommunizieren und arbeiten können. Der Chipsatz ist das Bindeglied zwischen den einzelnen Komponenten eines Computers. Egal was in einem Computer passiert, der Chipsatz ist immer daran beteiligt.

Der Chipsatz ist meist nur ein Halbleiterbaustein, der viele Controller in sich vereint. Die Controller sorgen für die Anbindung von Hauptprozessor (CPU), Laufwerken, Erweiterungskarten und anderen Geräte über Schnittstellen.

Der Chipsatz sorgt dafür, dass die Controller untereinander verknüpft und mit dem Hauptprozessor verbunden sind. Dabei werden unterschiedliche Spannungspegel, Taktfrequenzen und Protokolle berücksichtigt bzw. untereinander umgewandelt.

Im Laufe der Zeit entstanden verschiedene Chipsatz-Architekturen. Die Veränderungen an der Architektur hatten vor allem die Erhöhung der Systemgeschwindigkeit zur Folge. Zwangsläufig kam es dazu, dass immer mehr Funktionen und Schnittstellen in den Prozessor wanderten.

Abbildung 18 zeigt zum Beispiel ein Mainboard, welches nur noch über einen Chipsatz verfügt.

Vergleicht man das Mainboard, mit dem von Abbildung 17, scheint es so, als wäre die Northbridge verschwunden.

Allerdings stellt sich nun die Frage, welche Komponente die Aufgaben der Northbridge übernimmt.

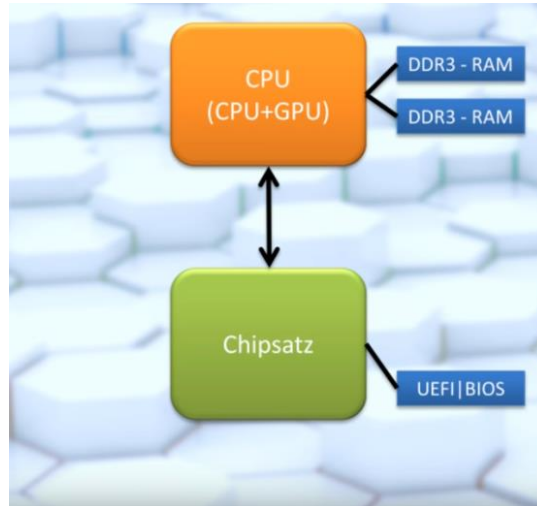


Abbildung 9-3: Mainboard mit nur einem Chipsatz; Quelle: www.wikipedia.de

Aufgabe: Schauen Sie sich folgendes Youtube-Video an und bearbeiten Sie das Arbeitsblatt:

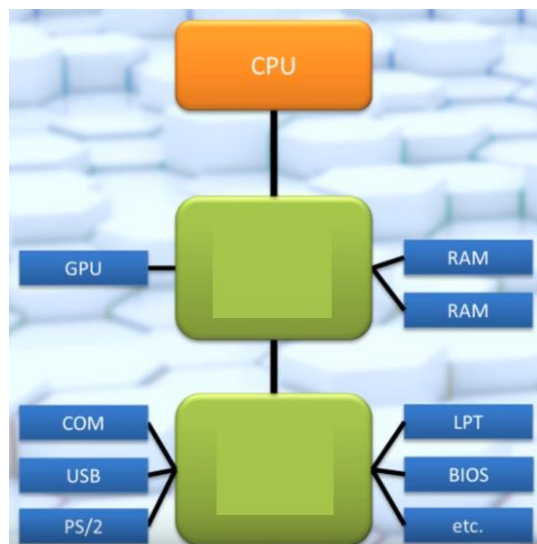
<https://www.youtube.com/watch?v=yPmx89bp62M>

1. Ergänzen Sie die Abbildung um die fehlenden Komponenten.



2. Neben der GPU befindet sich auch ein IMC mit in der CPU. Was ist das?

3. Ergänzen Sie die Abbildung um die fehlenden Komponenten.



4. Wie wird die Schnittstelle zwischen der CPU und dem (klassischen) Chipsatz genannt?

10 Serielle Datenübertragung

Alle modernen Schnittstellen beruhen auf der seriellen Datenübertragung. Es gibt nur noch wenige Bussysteme, die die Signale parallel übertragen. Beispiele wären

- die Schnittstelle PATA bzw. IDE (Vorgänger von SATA)
- PCI (Vorgänger von PCIe)
- der Speicherbus (Verbindung zwischen CPU und RAM)

Während der Speicherbus ein fester Bestandteil aller Mainboards ist, findet man PATA und PCI nur noch selten. Die parallele Datenübertragung wurde durch die Serielle verdrängt. Das wirft jedoch Fragen auf. Können bei einer parallelen Verbindung nicht mehr Bits übertragen werden? Schließlich gibt es hier mehr Leitungen, während die serielle Verbindung nur aus einer Leitung besteht.

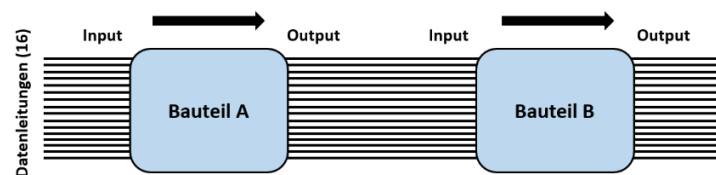


Abbildung 10-1: Parallele Schnittstelle

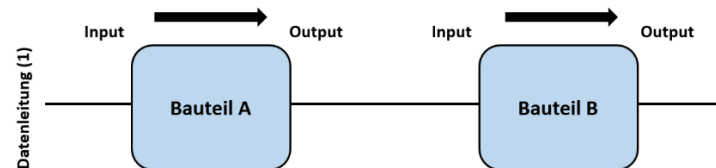


Abbildung 10-2: Serielle Schnittstelle

Tatsächlich ist es so, dass durch das serielle Datenübertragungsverfahren eine wesentlich höhere Datenübertragungsrate erzielt werden kann. Serielle ist also schneller als parallel.

Wie das sein kann, ist Thema der nächsten Abschnitte.

10.1 Nachteile der parallelen Datenübertragung

Das parallele Bussystem PCI war lange Zeit die Standard-Schnittstelle für Erweiterungskarten wie Grafik-, Sound-, Netzwerkkarten und so weiter. Mittlerweile etablierte sich der Nachfolger PCIe als Standard. Anhand von zwei Eigenschaften soll der Wechsel hin zur seriellen Schnittstelle verdeutlicht werden.

10.1.1 Bandbreite wird geteilt

Die PCI-Slots auf einem Mainboard sind alle über ein gemeinsames Bussystem miteinander verbunden. Das rechte Bild veranschaulicht dies schematisch.

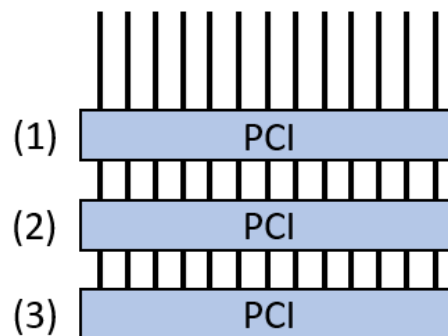


Abbildung 10-3: Bussystem bei PCI

Da sich alle PCI-Teilnehmer einen Bus teilen, kann zu einem Zeitpunkt immer nur eine Karte Daten senden oder empfangen.

Somit müssen sich die PCI-Teilnehmer die Bandbreite teilen. Die Bandbreite bezeichnet allgemein die

Übertragungsgeschwindigkeit - sprich die DÜR.

Eine gängige Datenrate für PCI lag bei 266MB/s. Wären in unserem Beispiel alle drei PCI-Slots belegt, so läge die gesamte Datenrate bei immer noch 266MB/s. Denn die Datenrate vervielfacht sich nicht.

Über PATA können bis zu zwei Laufwerke bzw. Festplatten angeschlossen werden. Auch hier teilen sich beide Teilnehmer dieselben Leitungen, so dass zu einem Zeitpunkt nur ein Gerät aktiv sein kann.



Abbildung 10-4: PATA-Kabel

10.1.2 Gleiche Länge für alle Leitungen

Ein weiterer Nachteil bei einem parallelen Bussystem ist die Vermeidung von zu großen Längenunterschieden bei den Leitungen.

Jede einzelne Leitung bei PCI oder bei PATA müssen exakt dieselbe Länge aufweisen. Die Ausbreitung der Spannungspegel auf den Leitungen ist überall gleich schnell. Muss ein Bit einen längeren Weg zurücklegen als seine Nachbarbits, benötigt dieses mehr Zeit und trifft verspätet beim Empfänger ein.

Wie Sie in Kapitel 8 erfahren haben, muss der Empfänger die Spannungspegel zu genauen Zeitpunkten abtasten. Dafür sorgt das Taktsignal. Allerdings kann bei der Fertigung von Mainboards nicht immer eine einheitliche Länge von Leitungen garantiert werden, wie auch die Abbildung 23 zeigt.

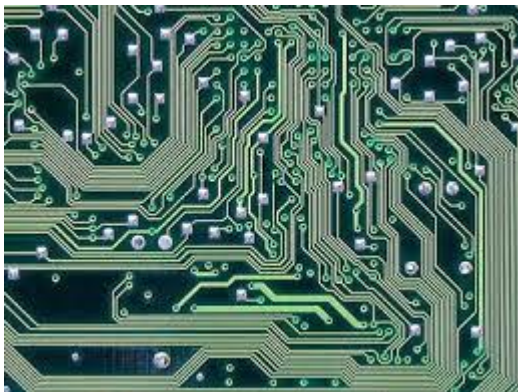


Abbildung 10-5: Unterschiedliche Leitungslängen

Die Leiterbahnen müssen oft Kurven nehmen, da andere Bauteile im Weg sind. Dadurch ergeben sich unterschiedlich lange Wege für die Bits. Eine Erhöhung der Taktrate ist demnach eingeschränkt. Schließlich orientiert sich die maximale Taktrate an dem langsamsten Bit und endet bei ca. 533MHz.

Zusammenfassend führten also zwei Kriterien dazu, dass die Technik der seriellen Übertragung weiterentwickelt wurde.

Bei einer seriellen Verbindung gibt es kein gemeinsames Bussystem – jede Leitung arbeitet für sich. Des Weiteren gibt es kein langsamstes Bit. Da es nur eine Leitung gibt, gibt es keine Längenunterschiede zu anderen Leitungen. Dies eröffnet die Möglichkeit die Taktrate um ein Vielfaches von 533MHz anzuheben (PCIe 5.0: 16GHz, Thunderbolt: 20,625GHz).

10.2 Aufbau einer Lane

Serielle Schnittstellen wie HDMI, DisplayPort, PCIe, SATA, Thunderbolt, M.2 bestehen aus sogenannten Lanes. Eine Lane besteht aus vier einzelnen Adern wobei zwei Adern ein Paar bilden. Das eine Leitungspaar ist für den Datenversand und das andere für den Datenempfang zuständig. Senden und Empfangen erfolgen gleichzeitig (Vollduplex).

Zur Steigerung der Datenrate besitzen Schnittstellen mehrere Lanes. Insgesamt lassen sich bis zu 32 Lanes bündeln (also 128 einzelne Adern). Die Verbindung wird zusammenfassend als Link bezeichnet. Also mehrere Lanes bilden einen Link. Häufig werden die Begriffe jedoch synonym verwendet.

In der Praxis sieht es so aus, dass einfache Erweiterungskarten nur eine Lane haben. Ausnahmen bilden Grafikkarten-Slots, die als PEG (PCI Express for Graphics) bezeichnet werden oder Audiokarten für höhere Ansprüche.

Jede Lane verschickt unabhängig von den anderen Lanes Daten gleichzeitig in beide Richtungen. Die Daten werden beim Empfänger gesammelt, sortiert und dann weitergegeben.

Pro Taktperiode werden zwei Bits übertragen → Double Data Rate

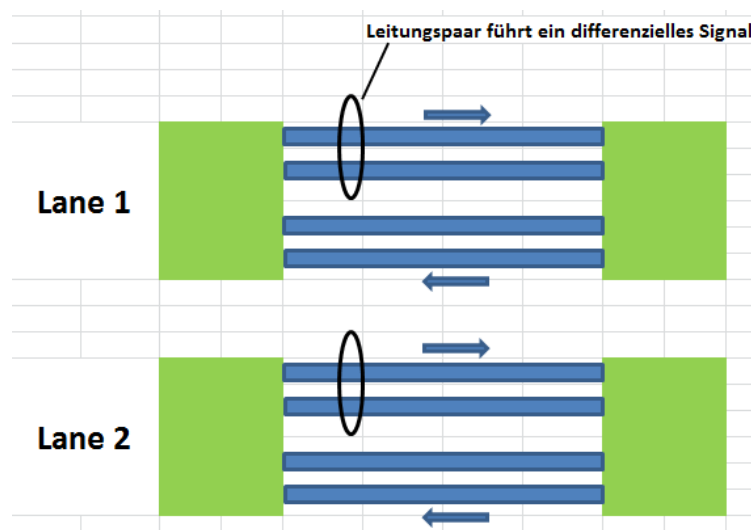


Abbildung 10-6: Aufbau mit 2 Lanes

Datenübertragung durch differenzielle Spannungspegel

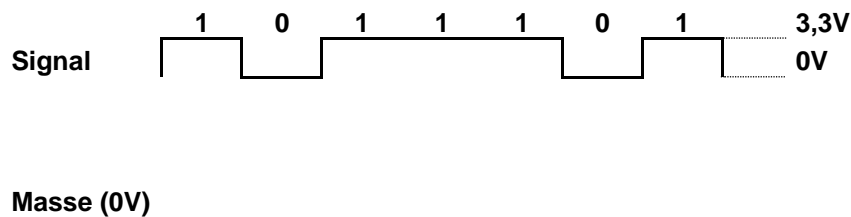
Die Übertragung erfolgt über sogenannte differenzielle Spannungspegel. Hierbei wird neben dem eigentlichen Datensignal zusätzlich auf einer zweiten Leitung das Datensignal invertiert verschickt. Dadurch werden auf Empfängerseite Störungen aus dem Signal herausgefiltert. Werden die Signale mit Hilfe von sehr niedrigen Spannungswerten übertragen, spricht man auch von LVDS (Low Voltage Differential Signaling).

10.3 Differentielle Spannungspegel

Die Übertragung von Daten mittels differentieller Spannungspegel kommt bei der seriellen Datenübertragung zum Einsatz. Was hat es mit den differentiellen Spannungspegeln auf sich? Um diese Technik und den Grund für die Verwendung verstehen zu können, betrachten wir zunächst die Übertragungstechnik, wie sie bei der parallelen Datenübertragung üblich war.

10.3.1 Unsymmetrische Signalübertragung

Bei der veralteten unsymmetrischen Übertragungstechnik nutzte man zwei Leitungen – eine Leitung für das Signal und eine Leitung für die Masse (Bezugsspannung, in der Regel 0V)



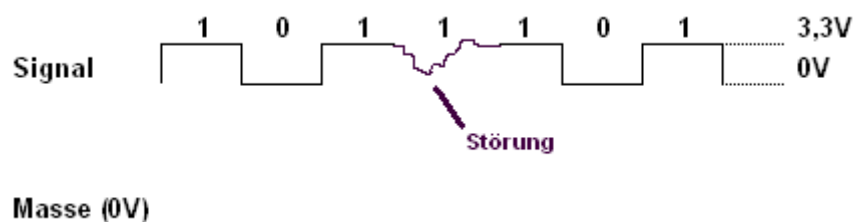
Die Signalleitung führt zwei Spannungswerte um die logischen Zustände darzustellen. Denkbar wäre:

- 3,3V entspricht einer logischen 1
- 0 V entspricht einer logischen 0

Der Spannungspegel auf der Signalleitung wird mit Hilfe der Bezugsspannung (Masseleitung 0V) ermittelt:

- Führt die Signalleitung den Spannungswert 0V, misst der Empfänger $0V - 0V = 0V$
- Führt die Signalleitung den Spannungswert 3,3V, misst der Empfänger $3,3V - 0V = 3,3V$

Je schneller man taktet und je höher der Spannungspegel ist, desto größer sind die elektromagnetischen Felder, die von den einzelnen Leitungen abgestrahlt werden. Die Leitungen stören sich also gegenseitig. Aufgrund dieser Störfelder kann das Signal in seiner eigentlichen Form, während der Übertragung, beeinflusst werden (siehe Bild). Es treten Störungen in den Spannungspegeln auf. Der Empfänger interpretiert das Datensignal falsch.



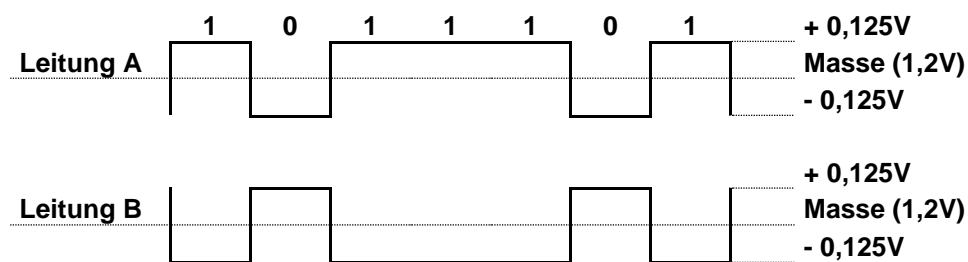
An der Stelle mit der Störung werden nun keine 3,3V mehr gemessen!

Wenn man bedenkt, dass jede einzelne Leitung ein elektromagnetisches Feld abstrahlt, müssen die angrenzenden Leitungen vor diesen Feldern geschützt werden. Abhilfe verschaffen zusätzliche Masseleitungen, die die Energie abfangen. Das wiederum verursacht mehr Kosten und macht die Übertragungsstrecken nur noch breiter.

Die Lösung lautet: Wechsel von der parallelen zur seriellen Datenübertragung unter Verwendung differentieller Spannungspegel.

10.4 Differentielle Übertragungstechnik

Die differentielle Signalübertragung benötigt 3 Adern. Die untere Abbildung zeigt ein Leitungspaar mit zwei einzelnen Signalleitungen plus einer Masseleitung. Auf Leitung A sehen wir das originale Datensignal. Welches Signal wird nun auf Leitung B geführt? Auf Leitung B sieht man das invertierte Signal von Leitung A. Das bedeutet, wenn beispielsweise das Signal auf Leitung A positiv ist, ist das Signal auf Leitung B negativ und umgekehrt.



Mittig ist die Masse angedeutet, die das Bezugspotenzial führt. Dieses ist nun nicht mehr 0V, sondern liegt bei ca. 1,2V.

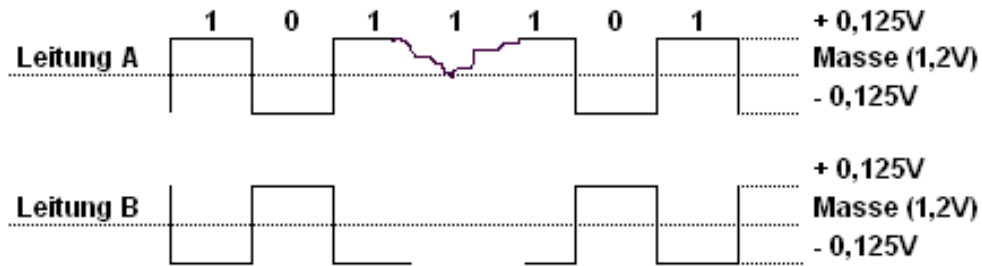
- Misst der Empfänger an Leitung A einen positiven Spannungshub, entspricht dies einer logischen 1.
- Wird an Leitung A ein negativer Spannungshub gemessen, entspricht dies einer logischen 0.

Jetzt kommt das Entscheidende:

Der Empfänger wertet die Differenz zwischen beiden Spannungshüben auf Leitung A und Leitung B aus. **Wie groß ist die Differenz laut Abbildung?**

Antwort: _____

Wir betrachten nun wieder den Fall einer Signalstörung. Wie Sie sehen, wurde das Signal auf Leitung A verändert. Aber wie wirkt sich die Störung auf der zweiten Leitung aus? Zeichnen Sie diese ein und überlegen Sie, welchen Spannungswert der Empfänger an dieser Stelle misst. Hinweis: Das invertierte Signal auf Leitung B wird im Sender erzeugt. Auf Ereignisse, die während der Übertragung auf dem Leiter passieren, hat der Sender keinen Einfluss mehr.



Lösung:

Wichtig: Zu beachten ist, dass pro Taktflanke zwei Datenbits verschickt werden → **Double Data Rate**.

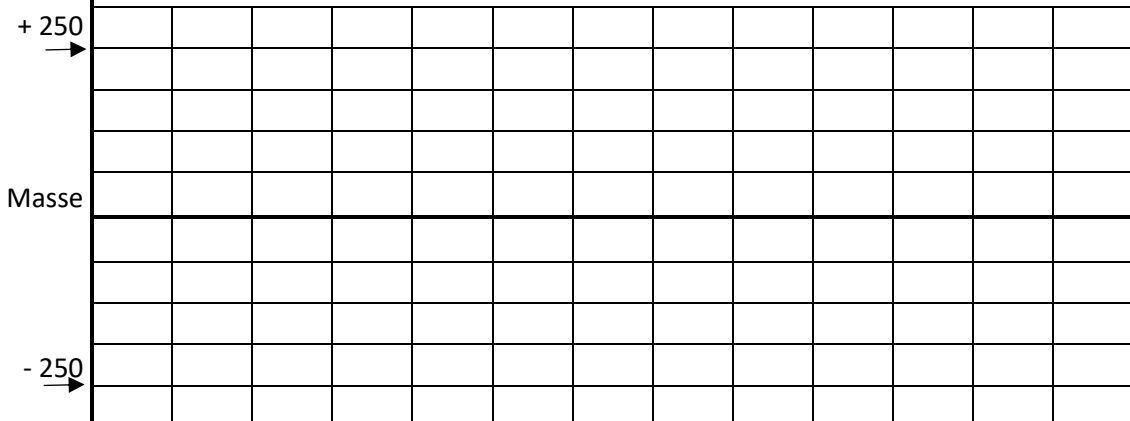
10.4.1 Aufgabe „Differentielle Spannungspegel (LVDS)“

Gegeben ist ein Leitungspaar, welches mit differentiellen Spannungspegeln ($\pm 125\text{mV}$) arbeitet. Sie sollen folgenden Binärcode als Signalfanken in die untere Skizze eintragen: 1001011101

Eine logische 1 erfolgt, wenn auf der Leitung 1 ein positiver Spannungspegel anliegt. Ein negativer Spannungshub auf Leitung 1 stellt eine logische 0 dar.

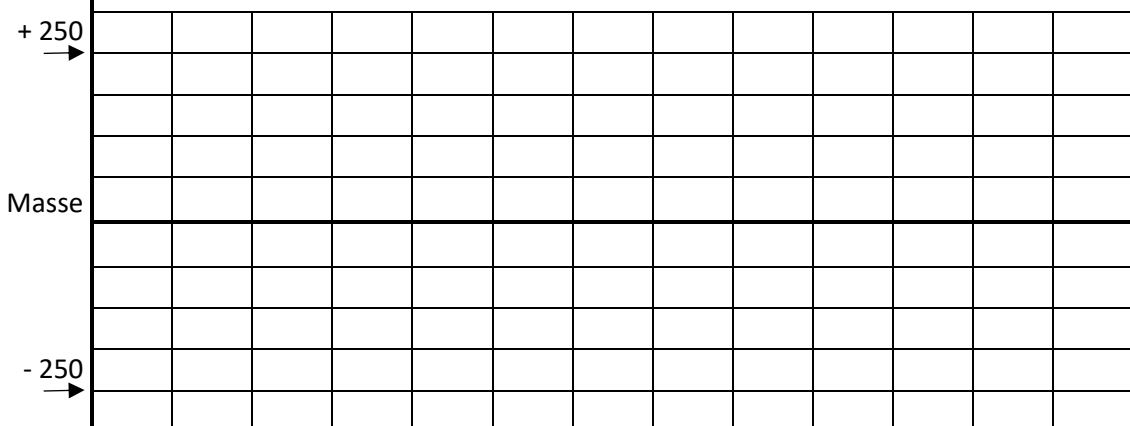
Leitung 1

Spannung
[mV]



Leitung 2 (invertiert)

Spannung
[mV]



Störung einbauen

Das Signal wird nun durch ein äußeres elektromagnetisches Feld gestört. Zeichnen Sie in die mittlere 1 der drei aufeinanderfolgenden 1en eine Signalstörung ein. Denkbar wäre eine Spannungsspitze, die sich auf das Signal überträgt. **Welche Wirkung hat die Störung auf das Signal beim Auslesen?**

10.5 Kodierungsverfahren

Serielle Datenverbindungen, senden die Datenbits nicht eins zu eins über die Datenleitungen, sondern bedienen sich einer Kodierung. PCIe benutze z.B. bei den älteren Versionen die 8b/10b-Kodierung. Bei den neueren Versionen wird das 128b/130b-Verfahren angewandt. Thunderbolt 3 nutzt hingegen die 64b/66b-Kodierung. Was es damit aus sich hat, soll nun beschrieben werden.

Die Daten werden über die vorhandenen Lanes unabhängig voneinander übertragen und hinterher wieder zusammengesetzt. Das bedeutet es wäre unvorteilhaft, wenn sich alle Lanes nach einem allgemeinen Systemtakt richten müssten. Jede Lane sollte selbstständig arbeiten können. Doch woher soll der Empfänger wissen, wie viele Bits innerhalb einer bestimmten Zeit übertragen werden, wenn es keine Taktleitung gibt? Möglich wäre es, den Takt aus dem Datenstrom zurück zu gewinnen. Hierbei müssen sich die Flanken im Bitstrom nur häufig genug abwechseln, damit der Empfänger den Takt rekonstruieren kann.

10.5.1 Die 8b/10b-Kodierung

Bei der 8b/10b-Kodierung ergeben 8 Bit ein Datenpaket oder auch Datenwort genannt. Nun gibt es allerdings Bitkombinationen, die wenige Flankenwechsel enthalten, z.B. 1000 0000.

Der Empfänger würde bei den vielen Nullen schnell aus dem Takt kommen. Darum wird jedes 8Bit-Datenwort einem 10-Bit-Datenwort zugeordnet. Ein 10-Bit-Wort bietet 1024 Bitkombinationen. Von diesen 1024 Möglichkeiten werden nur diejenigen genutzt, bei denen innerhalb des Wortes und beim Übergang in das nächste Wort nicht mehr als 5-mal der gleiche Zustand, also fünfmal 1 oder 0, erfolgen. Weiterhin darf die Differenz aus der Anzahl der Einsen und der Anzahl der Nullen nur maximal 2 betragen. Von diesen Möglichkeiten gibt es insgesamt mehr als 256, so dass jedes 8-Bit-Wort einem 10-Bit-Wort zugeordnet werden kann.

Die Tabelle zeigt Ihnen Beispiele für die Kodierung. Versuchen Sie keinen logischen Zusammenhang oder ein Muster zwischen den Wörtern zu erkennen. Die Zuordnung erfolgte einfach willkürlich und ist seither fest.

8-Bit-Wort		10-Bit-Wort
0000 0000	→	10 0111 0100
0000 0001	→	01 1011 0100
0000 0010	→	10 1101 0100
0000 0011	→	11 0001 1011
0000 0100	→	11 0101 0100
0000 0101	→	10 1001 1011
...

10.5.2 Berechnung der Netto-Datenrate

Aufgrund der Kodierung muss bei der Ermittlung der Datenrate zwischen einem Brutto- und einem Netto-Wert unterschieden werden. Durch die Kodierung ergibt sich ein Overhead, der nicht zu den eigentlichen Nutzdaten gezählt werden kann. Dies muss berücksichtigt werden.

Erklärung:

Eine Verbindung mit einem Takt von 1,25GHz ergibt zunächst dank **DDR** eine Transferrate von 2,5 GT/s. **Doch Transferrate ist nicht gleich Datenrate.** 2,5GT/s beziehen sich auf die kodierten 10 Bits, lediglich nur 80% von diesen 10Bits gehören zu den eigentlichen Nutzdaten.

Beispielrechnung: Berechnung der Netto-Datenrate für PCIe 1.0 (Taktrate = 1,25GHz):

1) Transferrate aus der Taktrate berechnen

- Transferrate = 1,25GHz x 2 Bit (DDR) = 2,5GT/s (bzw. 2,5Gbit/s)

2) Overhead wegen Kodierung abziehen

- DÜR[bit/s] = 2,5 GT/s x 0,8 = 2 Gbit/s
 - Der Faktor 0,8 ergibt sich aus der Kodierung 8b/10b

3) Umwandlung der Datenrate in Byte/s

- $DÜR[Byte/s] = \frac{2Gbit/s}{8bit/Byte} = 250 MB/s$

Angaben zu PCIe – Taktraten und Datenraten

	PCIe 1.0/1.1	PCIe 2.0/2.1	PCIe 3.0	PCIe 4.0	PCIe 5.0
Taktrate	1,25 GHz	2,5 GHz	4,0 GHz	8,0 GHz	16 GHz
Transfers/s (je Lane und Richtung)	2,5 GT/s	5,0 GT/s	8,0 GT/s	16,0 GT/s	32,0GT/s
Kodierung	8b10b	8b10b	128b130b	128b130b	128b/130b
Lanes (Breite)					
x1	250 MB/s	500 MB/s	985 MB/s	1,969 GB/s	3,938GB/s
x2	500 MB/s	1000 MB/s	1969 MB/s	3,938 GB/s	7,877GB/s
x4	1000 MB/s	2000 MB/s	3938 MB/s	7,877 GB/s	15,754GB/s
x8	2000 MB/s	4000 MB/s	7877 MB/s	15,754 GB/s	31,508GB/s
x16	4000 MB/s	8000 MB/s	15754 MB/s	31,508 GB/s	63,015GB/s

Können Sie die Angaben aus der Tabelle nachvollziehen?

10.5.3 Aufgabe „Kodierungsverfahren“

1) Welche der folgenden 10-Bit-Codes sind korrekt? Setzen Sie ein entsprechendes Kreuz

10-Bit-Wort	Korrekt	Nicht korrekt / Grund?
11010 11011		
10011 01010		
10000 00111		
11110 00001		
10010 10101		

2) Weisen Sie mit einer Rechnung nach, wie man bei PCIe 4.0 von der Taktrate auf die DÜR von 7,877GB/s für x4 kommt.

3) Ein Entwicklerteam testet zurzeit eine neue Kodierung für eine Verbindung. Hierbei werden 224 Bit auf 228 Bit abgebildet. Die Taktrate beträgt 4GHz. Die Signale werden über 8 Lanes mit der DDR-Technik übertragen. Berechnen Sie die Datenübertragungsrate.

11 Lane Sharing

Eine CPU verfügt in der Regel über eine Anbindung von 16 Lanes. Primär empfängt die CPU die Daten der Grafikkarte, welche auf den (primären) PCIe-x16-Anschluss gesteckt ist. Aber ein Mainboard besitzt im Allgemeinen nicht nur über eine einzige PCIe-Schnittstelle. Verschiedene Konstellationen sind auf dem Mainboard-Markt zu finden: 1x PCIe16 + 2x PCIe1, 1x PCIe16 + 1x PCIe4 + 1x PCIe1, 2x PCIe16 + 2x PCIe1.

Zwar kann eine PCI-Express-Lane nicht mehrere Geräte anbinden, aber man kann die Lanes über elektrische Splitter auf verschiedene Leiterbahnen legen. Typisch für Gaming-Mainboards sind beispielsweise zwei mechanische x16-Slots. 8 Lanes der CPU werden permanent zum ersten Slot geführt, 8 weitere Lanes aber zu einer Gruppe aus Splittern. Wird nur eine Grafikkarte am ersten Slot betrieben, leitet der Splitter die Signale ebenfalls an den ersten Slot, der dann als normaler elektrischer x16-Slot arbeitet. Wird eine zweite Grafikkarte genutzt, zum Beispiel für SLI (Scalable Link Interface, d.h. zwei Grafikkarten arbeiten im Verbund; wird bei AMD Crossfire genannt), leiten die Splitter hierfür die 8 Lanes um zur zweiten Grafikkarte. Der erste Slot läuft nun im x8-Modus, der zweite ebenfalls.

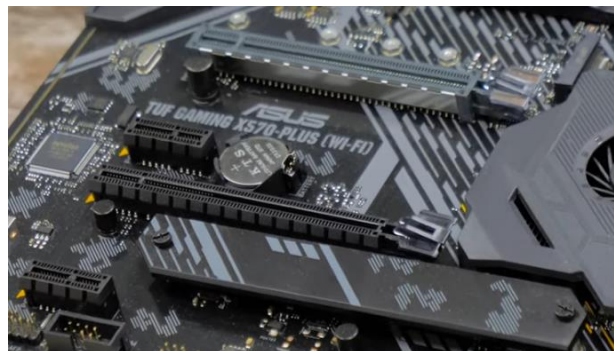


Abbildung 11-1: Mainboard mit 4 PCIe-Slots

Die obere Abbildung zeigt ein Mainboard mit 2 PCIe16- und 2 PCIe1-Slots. Werden beide Slots mit Grafikkarten bestückt, werden nur jeweils 8 der 16 Lanes pro Schnittstelle tatsächlich benutzt. Welche Kombinationen möglich sind, verrät ein Blick auf das Blockschaltbild des Chipsatzes (s. nächste Seite).

INTEL® Q170 AND Q150 CHIPSET BLOCK DIAGRAM

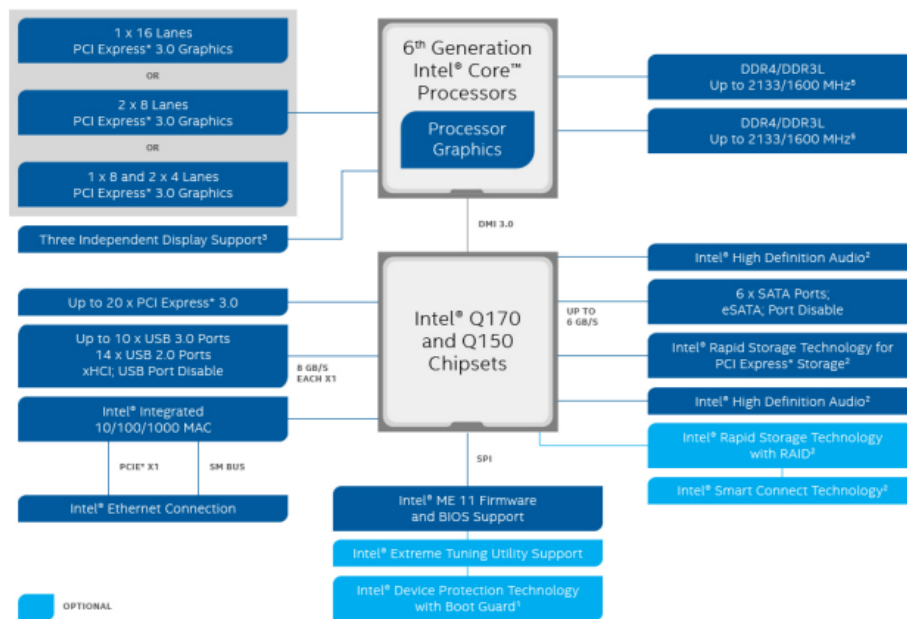


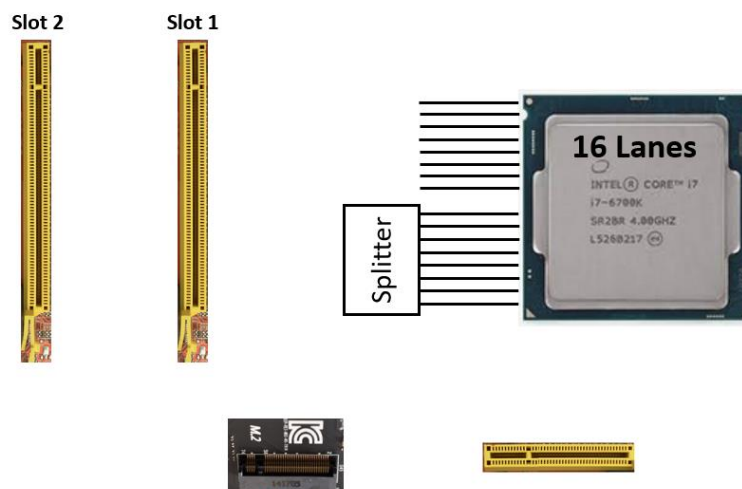
Abbildung 11-2: Blockschaltbild zum Chipsatz

11.1.1 Aufgabe „Lane-Sharing“

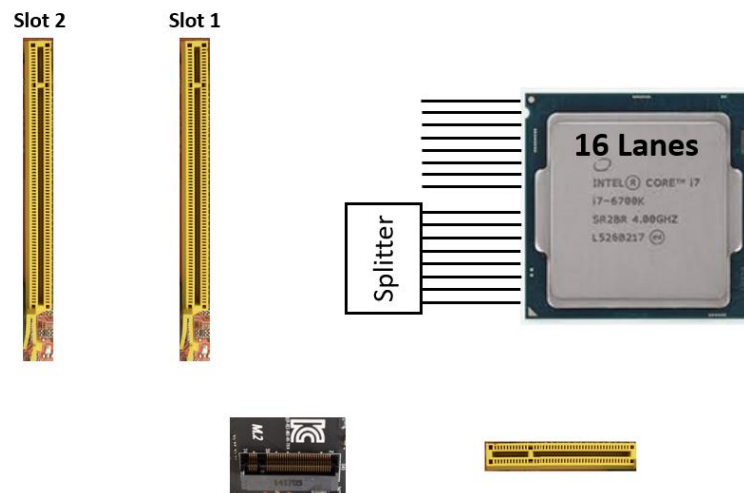
Zeichnen Sie für die verschiedenen Belegungsmöglichkeiten die Lanes ein (1 Lane = 1 Strich).

Grundsätzlich gehen von Slot1 8 Lanes direkt zur CPU. Die restlichen 8 Lanes werden wahlweise über einen Splitter zu den anderen Anschlüssen (PCIex16, PCIe4, M.2) geschaltet.

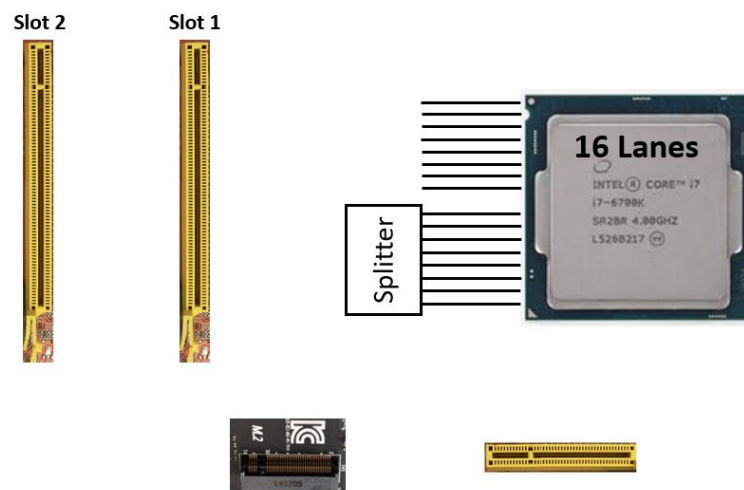
1. Belegung: 1x PCIe16 (Grafikkarte)



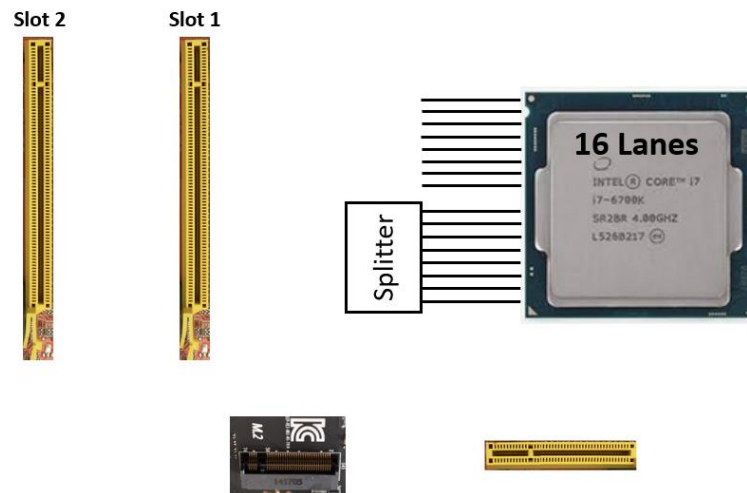
2. Belegung: 2x PCIe16 (Grafikkarten über Crossfire bzw. SLI verbunden)



3. Belegung: 1x PCIe16 (Grafikkarte), 1x M.2



4. Belegung: 2x PCIe16 (Grafikkarte), 1x M.2



5. Belegung: 1x PCIe16 (Grafikkarte), 1x M.2, 1x PCIe4

