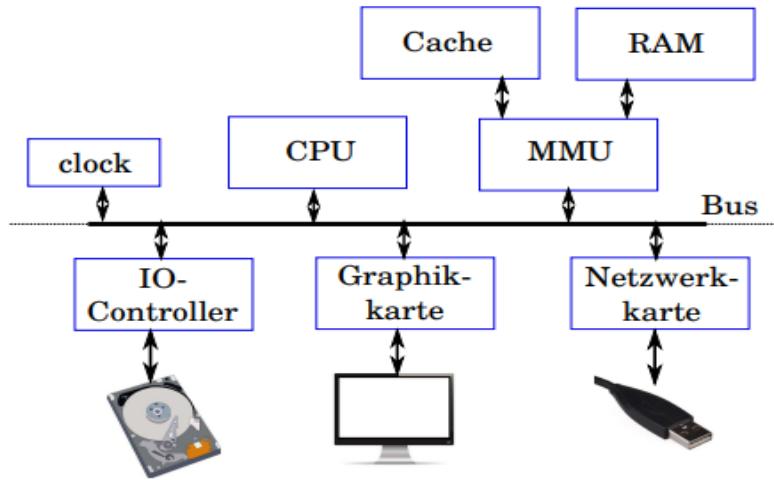


# Grundlagen der technischen Informatik (Hardware)

## 1. Der Prozessor (CPU = Central Processing Unit)

Die heutige Architektur eines Computers geht auf den ersten frei programmierbaren Rechner zurück, den kurz nach dem 2. Weltkrieg von dem Mathematiker *John von Neumann* entwickelten MANIAC Rechner. Sein Entwurf war so überzeugend, dass man heute noch von der *von Neumann Architektur* spricht. Das folgende Bild gibt einen ersten groben einen Überblick über diese Architektur, allerdings in modernisierter Form.

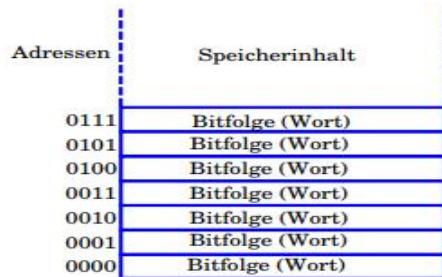


Ein sog. Bus dient als zentraler Kommunikationskanal zwischen den einzelnen Komponenten des Rechners. Über ihn laufen die Bitsequenzen. Meist gibt es aber nicht nur einen, sondern mehrere Busse, die verschiedenen Zwecken dienen. Insbesondere unterteilt man ihn in den Datenbus, über den die Daten laufen, und den Adressbus, über den die Adressen der Daten verschickt werden.

### Speicheradressierung

Hauptspeicher und Cache sind aufgeteilt in Folgen von *Wörtern*. Jedes Wort besteht aus einer festen Anzahl von Bits bzw. Bytes. Typische Wortgrößen sind 32 Bits (4 Bytes) oder 64 Bits (8 Bytes). I.A. ist ein Wort so groß, dass alle Bits in dem Wort in einem Takt verarbeitet werden können.

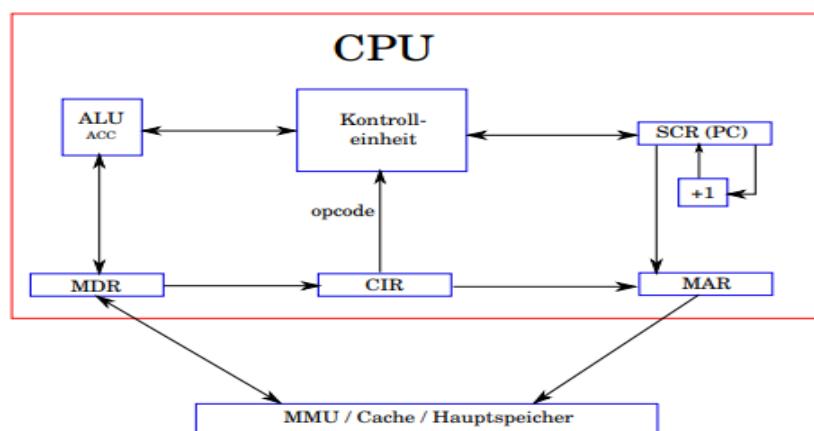
Die Folge von Wörtern ist durchnummeriert. Jedes Wort kann über seine Nummer (Adresse) angeprochen werden. Man kann sich das vorstellen wie die einzelnen Stockwerke eines Hochhauses. Jedes Stockwerk hat eine Nummer.



Da die Adressen, sie sind ja nur Nummern, in Binärdarstellung ebenfalls nur Bitfolgen sind, kann man sie auch selbst in den Speicher schreiben. In Adresse 5 könnte also als Speicherinhalt die Adresse 7 stehen. Diese Möglichkeit wird sehr häufig in Programmen ausgenutzt.

Maschinensprache: Eine weitere wichtige Erkenntnis, die John von Neumann in die Architektur hat einfließen lassen, bestand darin, dass man Programm und Daten nicht trennen muss. Auch Programme kann man als Bitfolgen repräsentieren, und diese genauso im Hauptspeicher halten wie die Daten. Dazu braucht man eine binäre Sprache, mit der man Anweisungen an die CPU ausdrücken kann, die sog. Maschinensprache. Ein sog. Maschinenbefehl besteht dann immer aus der binären Nummer des jeweiligen Befehls, dem OP-Code, gefolgt von binär dargestellten Argumenten. Z.B. könnte der Addierbefehl die Nummer 5 haben, und die Argumente enthalten dann die Adressen der Zahlen, welche zu addieren sind. Alles das lässt sich als Bitfolge repräsentieren, so dass es keinen strukturellen Unterschied mehr gibt zwischen Programm und Daten.

Die Bestandteile der CPU selbst in folgendem Bild, wiederum vereinfachend und schematisch, dargestellt:



#### **ALU: (Arithmetic Logic Unit)**

Dies ist die eigentliche Recheneinheit. Sie hat meist eigene Speicher, sog. *Register* und führt arithmetisch / logische Operationen mit den Werten in diesen Registern durch. Davon gibt es mindestens eines, den sog. *Akkumulator*.

**Kontrolleinheit:** Sie analysiert die Maschinbefehle und steuert damit die Abläufe innerhalb der CPU. Wenn es sich z.B. um einen arithmetischen Befehl handelt, gibt sie der ALU die Anweisung, diesen auszuführen.

#### **SCR (PC): (Sequence Control Register, Program Counter (PC) oder Programmzähler)**

ist ein Register, welches die Adresse des nächsten auszuführenden Befehls enthält. Bei normaler Programmausführung, ohne Sprungbefehle, wird dieser Zähler nach jedem Programmschritt um 1 weiter gezählt<sup>1</sup>.

#### **CIR (Current Instruction Register)**

enthält den Maschinencode des gerade auszuführenden Befehls.

#### **MAR: (Memory Address Register)**

In dieses Register wird die Adresse einer Speicherzelle geschrieben, von der die Daten geholt werden sollen. Das kann entweder die Adresse des nächsten Maschinbefehls sein, oder auch die Adresse des nächsten Datenwortes, welches verarbeitet werden muss. Der Inhalt des MAR wird über den Bus an die Memory Management Unit geschickt, welche dann die Daten aus dem Cache oder Hauptspeicher holt.

#### **MDR (Memory Data Register oder auch Memory Buffer Register)**

Sobald ein Datenwort über den Bus aus dem Speicher kommt, wird es im MDR zwischengespeichert. Von dort kommt es entweder in das CIR, falls es sich um einen Maschinbefehl handelt, oder in die ALU, falls es sich um ein Argument für eine Berechnungen handelt.

### 3.1 Der Fetch-Execute Zyklus

Die Verarbeitung eines Maschinenbefehls geschieht in folgenden Schritten:

1. Zunächst ist die Adresse des Befehls im Programmzähler (SCR/PC).
2. Von dort wird sie in das Memory Address Register (MAR) kopiert. Das hat zur Folge, dass die Adresse über den Bus an die MMU geschickt wird.
3. Der Inhalt der Adresse, d.h. der Maschinenbefehl selbst, kommt über den Bus in das Memory Data Register (MDR)
4. Da der CPU in diesem Moment bekannt ist, dass es sich um einen Maschinenbefehl handeln muss, wird er in das Current Instruction Register (CIR) kopiert.
5. Jetzt kann die Kontrolleinheit den Befehl analysieren, und abhängig davon die passenden Aktionen einleiten:
  - Falls es sich um einen arithmetisch/logischen Befehl handelt, dessen Argumente, bzw. deren Adressen ebenfalls im Maschinenbefehl enthalten sind, wird die ALU aktiviert, und über die Sequenz MAR → MMU → Speicher → MDR → Akkumulator werden die Daten bereit gestellt.  
Sobald die ALU mit der Berechnung fertig ist, wird das Ergebnis in das MDR geschrieben, und die Adresse des Zielwertes, in das die Daten kopiert werden sollen, in das MAR. Beide gelangen über den Bus und die MMU in den Speicher.
  - Falls es sich um einen Sprungbefehl handelt, wird die Adresse des Sprungziels unmittelbar in das Sequence Control Register (SCR) geschrieben.
6. Falls der letzte Befehl kein Sprungbefehl war, wird das Sequence Control Register automatisch weiter gezählt, um den nächsten Maschinenbefehl laden zu können.

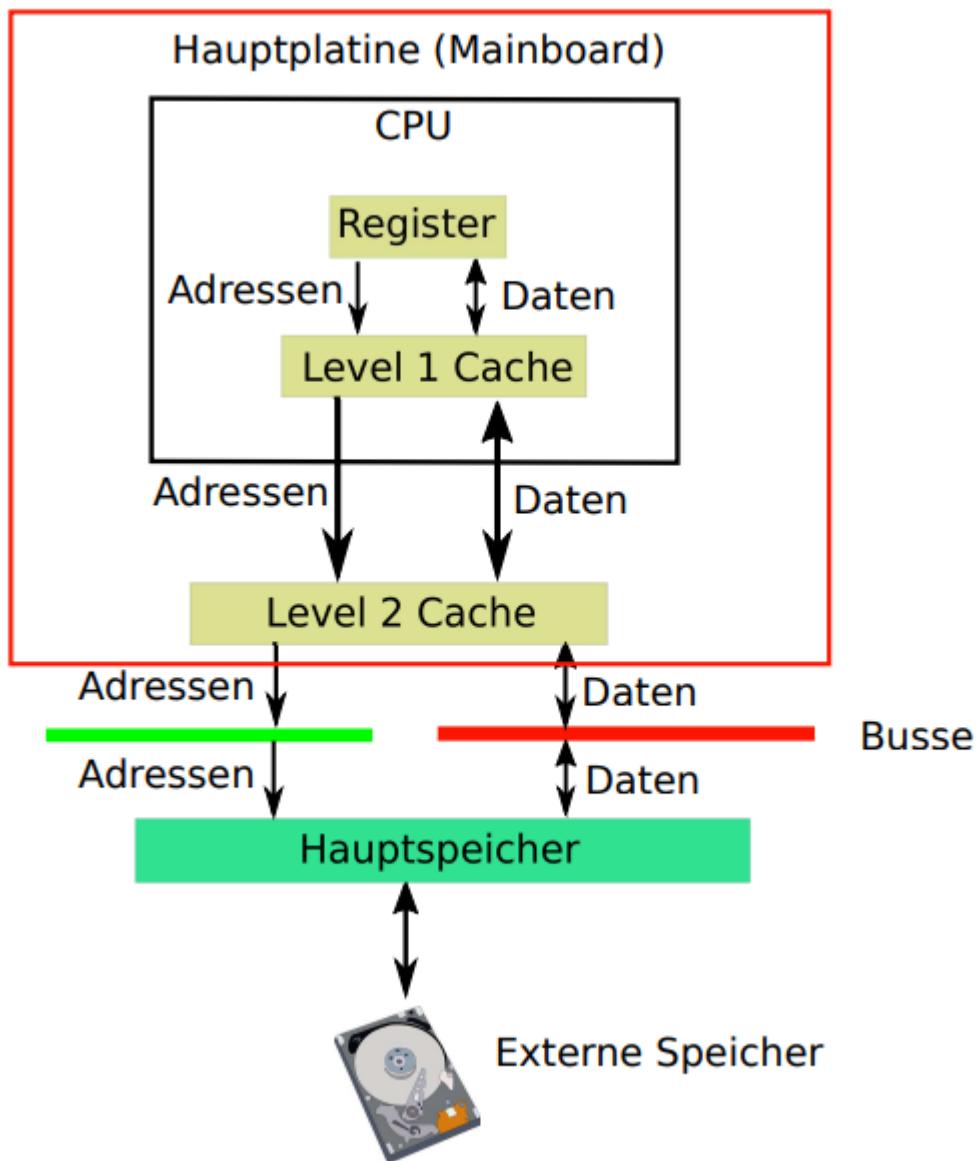
In konkreten Prozessoren sind noch andere Abfolgen als den oben geschilderten möglich. Insbesondere haben die meisten ALUs nicht nur ein Register, den Akkumulator, sondern bis zu 32 interne Register. Daher ist es oft nicht nötig, die Berechnungsergebnisse zurück in den Speicher zu schreiben. Sie können in einem der Register zwischengespeichert werden, um sofort für anschließende Berechnungen bereit zu stehen. Das beschleunigt die Abarbeitung ungemein.

### 4.1 Cache Speicher

In den ersten Computern war die CPU langsam und der Speicherzugriff schnell. Das hat sich irgendwann umgekehrt. Seit ca. Mitte der 90er Jahre sind die CPUs schnell, aber die Busse und Speicher langsam. D.h. die CPU könnte mehr verarbeiten als sie Daten liefert bekommt.

Man hat auf verschiedene Arten versucht, dem Problem entgegenzuwirken:

- Schnellere und mehr Busse, z.B. ein direkter Bus zwischen CPU und Hauptspeicher.
- Breitere Busse, 64 Bits parallel statt 32 Bits parallel. Das begrenzt aber die Geschwindigkeit, da man alle 64 Leitungen synchron halten muss.
- Schnelle Zwischenspeicher (Caches) in der Nähe der CPU.



Bei jedem Speicherzugriff muss also zuerst im Cache nachgeschaut werden, ob die Daten schon dort sind. Wenn nicht, werden sie sowohl in die Register als auch in den Cache geladen. Falls dabei festgestellt wird, dass der Cache schon voll ist, muss man Platz machen, d.h. andere Daten rauswerfen. Die Entscheidung, für welche Daten es sich lohnt, sie längerfristig im Cache zu lassen, wird nach *Lokalitätsprinzipien* getroffen:

**Zeitliche Lokalität:** Ein einmal gebrauchtes Wort aus dem Arbeitsspeicher wird wahrscheinlich öfter wieder gebraucht (in Schleifen, z.B.)

**Räumliche Lokalität:** Zu einem gebrauchten Wort aus dem Arbeitsspeicher werden auch häufiger die Nachbarworte gebraucht. Insbesondere bei sequentieller Programmabarbeitung oder bei Arrayzugriffen ist das oft so. Daher werden u.U. mit einem angeforderten Wort auch vorsorglich mehrere Wörter aus seiner Nachbarschaft in den Cache Speicher geladen.

## 4.2 Pipelining, Superskalarität

Die bisher vorgestellte Arbeitsweise im Fetch-Execute Zyklus hat den großen Nachteil, dass fast alle Komponenten der CPU die meiste Zeit arbeitslos sind. Als Beispiel betrachten wir den Maschinenbefehl

add t3, t1, t2

mit der Bedeutung: Addiere die Daten an Adresse t1 zu den Daten an Adresse t2 und speichere das Ergebnis in Adresse t3.

Um so einen Befehl auszuführen, müssen folgende Schritte gemacht werden:

1. Lade Daten aus Adresse t1      Die ALU tut nichts
2. Lade Daten aus Adresse t2      Die ALU tut nichts
3. addiere                          Die Kontrolleinheit tut nichts
4. Speichere das Ergebnis in t3    Die ALU tut wieder nichts.

Man kennt das Phänomen auch aus dem Alltag, z.B. beim Wäsche waschen:

- |                                 |   |
|---------------------------------|---|
| 1. Waschen in der Waschmaschine | der Trockener tut nichts, der Bügler tut nichts       |
| 2. Trocknen im Trockner         | die Waschmaschine tut nichts, der Bügler tut nichts   |
| 3. bügeln                       | die Waschmaschine tut nichts, der Trockner tut nichts |
| 4. Wäsche einräumen             | Waschmaschine, Trockner und Bügler tun nichts.        |

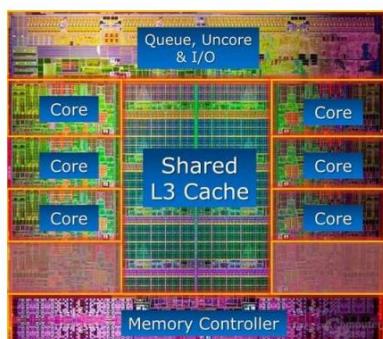
Bei nur einer Ladung Wäsche macht das nichts aus. Niemand würde aber mehrere Ladungen Wäsche nacheinander auf diese Weise bearbeiten. Stattdessen arbeitet man *versetzt*: Sobald die Waschmaschine mit der ersten Ladung Wäsche fertig ist, wird sie in den Trockner getan, und sofort kommt die zweite Ladung Wäsche in die Waschmaschine.

Dieses Prinzip des *Pipelining* hat man auch in den Prozessoren verwirklicht. Dazu werden die Arbeitsschritte in der CPU in möglichst kleine Teilschritte aufgeteilt, so dass eine ganze Befehlsfolge versetzt durch diese Teilschritte laufen kann, und damit alle Teile der Hardware permanent beschäftigt werden können.

Zur Illustration betrachten wir folgende Aufspaltung der Instruktionsausführung in mehrere Stufen

## 4.3 Hyper-Threading und Mehrkernrechner

Die Taktfrequenzen von Computern haben lange Zeit rasant zugenommen, von ca. 1 Herz beim Maniac Anfang der 50er Jahre, bis zu 3 Gigaherz beim Intel Core i3 in 2007. Seither stagniert die Entwicklung der Taktfrequenzen. Das liegt im wesentlichen daran, dass für höhere Taktfrequenzen auch mehr Energie benötigt wird. Der Energieverbrauch spielt aber eine immer größere Rolle.

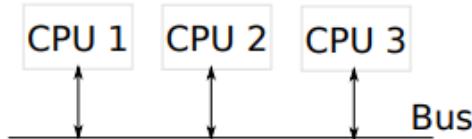


## 4.6 Mehrrechnersysteme

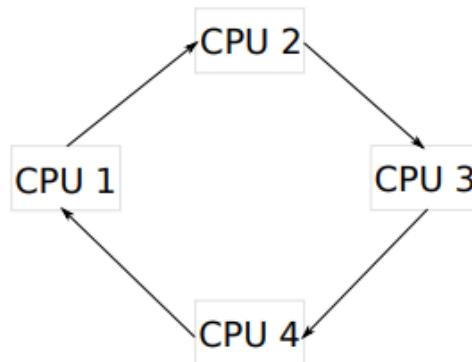
Sobald wirklich viel Rechenleistung gebraucht wird, kommt man mit einem Prozessorchip nicht mehr aus. Dann muss man viele davon zu einem Prozessornetzwerk zusammenschalten. Seit 2016 ist der Rekordhalter, d.h. der schnellste Supercomputer der Welt der chinesische *Sunway TaihuLight*. Er besteht aus 40960 64-Bit Prozessoren, von denen jeder 256 Rechenkerne hat. Damit hat man insgesamt 10,649,600 Rechenkerne. Damit schafft er 93 Petaflops. Das sind  $93 \cdot 10^{15}$  Floating Point Operationen pro Sekunde. Allerdings verbraucht er dafür 15 Megawatt Leistung.

Die vielen Prozessoren müssen alle miteinander verbunden werden, damit sie auch miteinander kommunizieren können. Dafür hat man verschiedene *Topologien* entwickelt:

**Bus:** Wenige Computer, die auch wenig miteinander kommunizieren, verbindet man über das Ethernet als Bus miteinander. Das Problem dabei ist, dass immer nur zwei Computer gleichzeitig Daten austauschen können. Während sie das tun, müssen alle anderen warten.



**Token Ring:** Wenn man den Bus schließt, bekommt man einen Ring. In diesem laufen sog. Token herum, die die Daten transportieren. Das kann man sich vorstellen wie ein Zug mit mehreren Waggons, in die man Daten hinein laden kann. Der Zug läuft immer im Ring herum. Wenn eine CPU etwas schicken will, wartet sie bis ein leerer Waggon vorbei kommt, und lädt ihre Daten hinein. Die Ziel-CPU liest alle vorbeikommenden Waggons mit für sie bestimmten Daten aus. Sobald eine CPU ausfällt, ist allerdings der ganze Token Ring blockiert.



**Kompletter Zusammenschluss:** Hier verbindet man jeden Prozessor mit jedem anderen. Damit erreicht man natürlich die bestmögliche Kommunikation dazwischen. Da man dafür quadratisch viele Leitungen braucht ist das leider nur praktikabel wenn man wenige CPUs hat.

## Der CPU-Befehlszyklus

- **Holen:** Die CPU ruft die nächste Anweisung aus dem Arbeitsspeicher ab. Die Adresse des nächsten Befehls wird im Programmzähler (PC) gespeichert.
- **Dekodieren:** Die Steuereinheit (CU) (Control Unit) interpretiert den abgerufenen Befehl, um zu verstehen, welche Operation ausgeführt werden soll.
- **Ausführen:** Die CPU führt die Anweisung aus. Wenn eine arithmetische oder logische Operation erforderlich ist, ist das Rechenwerk (ALU) hierfür zuständig.
- **Speichern:** Das Ergebnis der Operation wird zurück in den Speicher oder in ein internes Register geschrieben, falls dies in der Anweisung vorgesehen ist.
- **Wiederholen:** Der Prozess wiederholt sich für jede nachfolgende Anweisung, um sicherzustellen, dass alle Teile eines Programms ausgeführt werden.

## Weitere wichtige Funktionen

- **Steuerung und Koordination:** Die CPU koordiniert die Arbeit zwischen verschiedenen Komponenten des Computers, um sicherzustellen, dass alles reibungslos funktioniert.
- **Taktung:** Eine Computeruhr synchronisiert den Zyklus und bestimmt die Geschwindigkeit, mit der Anweisungen verarbeitet werden.
- **Multitasking:** Obwohl der Zyklus schrittweise abläuft, können moderne CPUs durch schnelles Wechseln zwischen verschiedenen Prozessen den Anschein von Parallelität erwecken und Multitasking durchführen.

Der Takt einer CPU ist die Geschwindigkeit, mit der sie Befehle ausführt, gemessen in Hertz (Hz). Man kann ihn sich als den „Herzschlag“ des Prozessors vorstellen, wobei ein höherer Takt bedeutet, dass der Prozessor pro Sekunde mehr Zyklen durchlaufen kann und somit schneller arbeitet. Diese Geschwindigkeit wird oft in Gigahertz (GHz) angegeben, was Milliarden von Takten pro Sekunde bedeutet.

- **Taktfrequenz:** Die Taktfrequenz (auch Taktrate genannt) gibt an, wie viele Zyklen ein Prozessor pro Sekunde ausführt. Ein Taktzyklus ist die grundlegende Zeiteinheit, in der die CPU eine bestimmte Operation durchführt, wie z.B. das Abrufen oder Ausführen eines Befehls.
- **Messung:** Eine Taktfrequenz von 1 GHz entspricht 1 Milliarde Takten pro Sekunde. Ein Prozessor mit 2,2 GHz führt demnach 2,2 Milliarden Takte pro Sekunde aus.
- **Bedeutung:** Eine höhere Taktfrequenz führt in der Regel zu einer höheren Leistung, aber die Gesamtleistung hängt auch von anderen Faktoren wie der Anzahl der Kerne und der Architektur der CPU ab.
- **Dynamische Anpassung:** Moderne Prozessoren passen ihre Taktfrequenz dynamisch an. Sie können die Frequenz erhöhen, um mehr Leistung zu erzielen, und sie senken, um Energie zu sparen und die Wärmeentwicklung zu reduzieren.
- **Anzeigen:** Die aktuelle und maximale Taktfrequenz der CPU können Sie im Task-Manager von Windows unter „Leistung“ überprüfen.

## CPU-Kerne

- **Was sie sind:** Physische Hardware-Einheiten innerhalb der CPU, die für die eigentliche Berechnung zuständig sind. ↗
- **Funktionsweise:** Jeder Kern kann eine Befehlsabfolge (einen Prozess) bearbeiten. ↗
- **Leistung:** Mehr Kerne bedeuten im Allgemeinen mehr parallele Verarbeitungsmöglichkeiten, was die Leistung steigert, wenn die Software für mehrere Kerne optimiert ist. ↗

## Threads

- **Was sie sind:** Virtuelle, softwarebasierte Befehlsfolgen oder Aufgaben, die an die CPU gesendet werden. ↗
- **Funktionsweise:** Ein Thread ist der kleinste Teil eines ausführbaren Codes. Mehrere Threads können innerhalb eines Prozesses ausgeführt werden. ↗
- **Leistung:** Die Fähigkeit, mehr Threads zu verarbeiten, ermöglicht es dem Prozessor, mehrere Aufgaben gleichzeitig zu bearbeiten und verbessert so das Multitasking. ↗
- **Zusammenarbeit mit Kernen:** Ein einzelner Kern kann zwar nur eine Sache auf einmal tun, aber er kann sehr schnell zwischen mehreren Threads wechseln, um den Eindruck von paralleler Ausführung zu erwecken. ↗

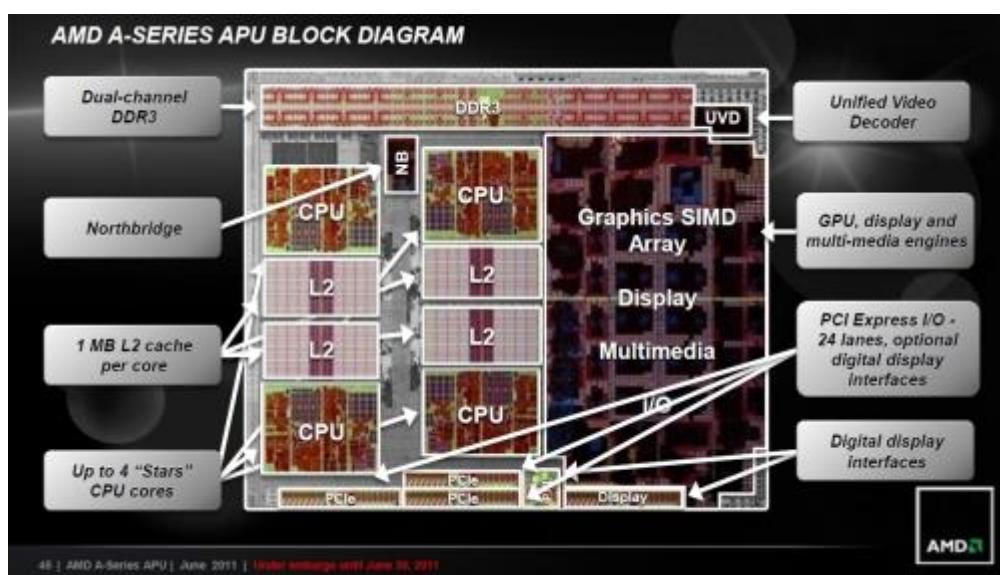
## Beispiel: Hyper-Threading

- **Technologie:** Ermöglicht es einem einzelnen physischen Kern, zwei oder mehr Threads gleichzeitig zu verwalten. ↗
- **Vorteil:** Erhöht die Effizienz, indem der Kern an mehr Aufgaben arbeiten kann, insbesondere wenn ein Thread auf eine Datenquelle warten muss. ↗
- **Ergebnis:** Eine CPU mit 4 Kernen und aktiviertem Hyper-Threading kann als 8-Thread-CPU auftreten und wird vom Betriebssystem als 8 logische Kerne behandelt. ↗

Eine APU (Accelerated Processing Unit) ist ein Prozessor, der **sowohl eine CPU (Central Processing Unit) als auch eine GPU (Graphics Processing Unit) auf einem einzigen Chip vereint**. Diese Kombination ermöglicht eine effizientere Datenübertragung, da CPU und GPU physisch näher beisammen liegen, was die Leistung verbessern kann, insbesondere bei Aufgaben, die sowohl Rechen- als auch Grafikverarbeitung erfordern.

## Funktionsweise

- **Vereinigung von CPU und GPU:** Der Hauptunterschied zu einer herkömmlichen CPU ist die integrierte Grafikeinheit. Das bedeutet, dass die APU die Aufgaben der zentralen Verarbeitung (CPU) und der Grafikverarbeitung (GPU) auf demselben Chip ausführt.
- **Gemeinsamer Speicherzugriff:** Sowohl die CPU-Kerne als auch der integrierte Grafikprozessor greifen auf denselben Hauptspeicher (RAM) zu, was die Kommunikation beschleunigt und den Bedarf an separaten Grafikspeichern reduziert.
- **Beschleunigte Leistung:** Durch die Nähe der beiden Einheiten werden Daten schneller übertragen und verarbeitet. Dies ist vorteilhaft für Anwendungen, die eine hohe grafische Leistung benötigen, wie z. B. Spiele, Videobearbeitung oder andere rechenintensive Aufgaben.
- **Effizienz und Platzersparnis:** Die Integration von CPU und GPU auf einem Chip reduziert die Anzahl der benötigten Komponenten, was zu geringeren Kosten, weniger Platzbedarf und einem geringeren Energieverbrauch führen kann.



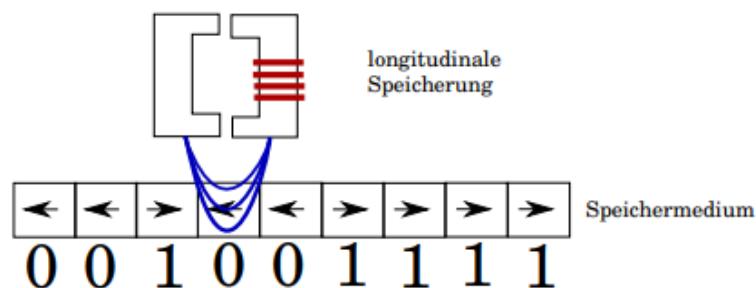
## 2. Speichertechniken

Magnetspeicher:

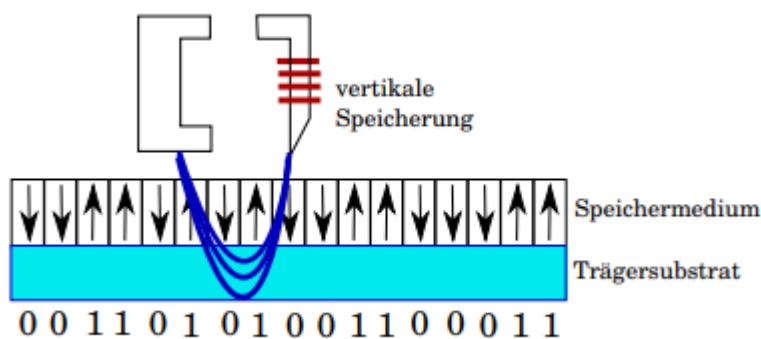
Eines der ältesten der auch heute noch in Gebrauch befindlichen Speicherverfahren ist die *magnetische Speicherung*. Hier nutzt man aus, dass ferromagnetischen Stoffe, z.B. Eisen, magnetisiert werden können, d.h. die magnetischen Momente der Atome (sog. Elementarmagnete) zeigen alle in eine Richtung, und der Stoff wirkt als Magnet. Allerdings müssen nicht wirklich *alle* Elementarmagnete eines magnetischen Stoffes in eine Richtung zeigen. Es können sich kleine Bereiche bilden, sog. *Weiss-Bezirke*, wo alle Elementarmagnete in eine Richtung zeigen. Die Elementarmagnete eines benachbarten Weiss-Bezirks können dann aber in eine andere Richtung zeigen. Die einzelnen Bezirke sind durch sog. *Bloch Wände* voneinander getrennt. Die Magnetisierungsrichtung eines Weiss-Bezirks lässt sich durch einen Elektromagneten abtasten (lesen) und auch verändern (schreiben).

Dies nutzt man jetzt folgendermaßen aus: Ein ferrormagnetischer Stoff wird entweder als Magnetband oder als Magnetplatte geformt. Diesem Stoff werden magnetisch Folgen von Weiss-Bezirken aufgeprägt. Bei Magnetbändern liegen die nebeneinander auf dem Band. Bei Magnetplatten liegen die in Kreisen angeordnet (sog. *Tracks*). Die Weiss-Bezirke können jetzt in zwei verschiedene Richtungen magnetisiert werden. Eine Richtung gilt dann als 0 und die andere Richtung gilt als 1. Mit einem Elektromagneten, der die Reihenfolge der Weiss-Bezirke abfährt, kann man jetzt die Magnetisierungsrichtung lesen und verändern.

In den ersten Magnetspeichern war die Magnetisierungsrichtung horizontal entlang des Bandes oder der Platte wie in diesem Bild.



Seit ca. 2010 gibt es Magnetspeicher, wo die Magnetisierungsrichtung vertikal liegt (perpendicular recording). Damit konnte man die Speicherdichte verzehnfachen.



## Bänder und Platten

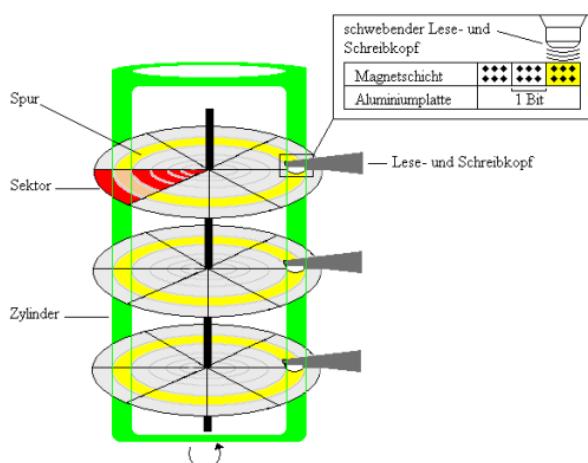
Bei *Magnetbändern* wird die ferromagnetische Schicht auf einem flexiblen Band aufgetragen. Dieses Band wird unter einem fest installierten Schreib-Lesekopf hin und her gespult. Man kennt sie von früher als Tonbänder, Musikkassetten, Videobänder und in der IT als Datenbänder. Heute spielen sie kaum noch eine Rolle, außer als Backupspeicher für große Datenmengen. Die Kapazität solcher Bandkassetten ist in der Größenordnung bis zu einem Terabyte. Der große Nachteil von Bandkassetten ist, dass sie nur sequentiell gelesen werden können. Um ganz bestimmte Daten zu lesen, muss man das Band von vorne bis zu der Stelle, wo diese Daten sind, vorspulen.

Bei *Festplatten* wird die ferromagnetische Schicht auf runde rotierende Platten aufgetragen. Die Weiss-Bezirke werden darauf als konzentrische Ringe (Spuren, Tracks) angelegt. Der Schreib-Lesekopf sitzt auf einem beweglichen Arm, der die einzelnen Spuren anfahren kann, und dann die unter ihm vorbei rotierenden Bits lesen und schreiben kann.



Die Platten sind i.A. oben und unten beschichtet, und zu mehreren aufeinander gestapelt. Für jede Plattenseite gibt es dann einen eigenen Arm mit dem Schreib- Lesekopf. Damit können dann mehrere Bits parallel gelesen und geschrieben werden. Die in den verschiedenen Platten übereinander liegenden Spuren fasst man zu einem *Zylinder* zusammen. Da alle Schreib-Leseköpfe fest miteinander verbunden sind, bewegen sie sich nicht nur zu einer bestimmten Spur einer Plattenoberfläche, sondern simultan zu allen übereinander liegenden Spuren, dem Zylinder.

## Festplattengeometrie



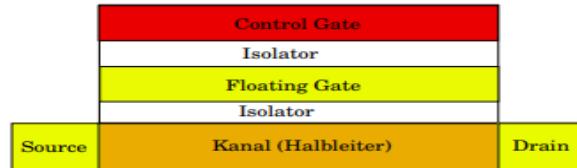
## Flash-Speicher

### 3 Flash-Speicher

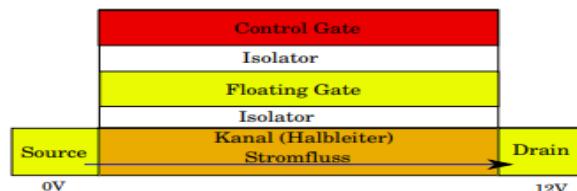
Die *Flash-Speichertechnik* ist heute u.a. in den Speicherkarten, USB-Sticks und Solid State Drives (SSD) zu finden. Durch ihre vielen Vorteile verdrängt sie allmählich immer mehr die anderen externen Speichertechniken.

#### Funktionsprinzip

Wir erklären das Funktionsprinzip für die *Floating-Gate* Transistortechnik. Eine Alternative ist die Charge-Trapping Technik, die im Prinzip ähnlich, aber im Detail anders funktioniert. Die wichtigste Eigenschaft eines Floating-Gate Transistors, ist, dass er auch im ausgeschalteten Zustand seine Information behält. Der Aufbau sieht so aus:

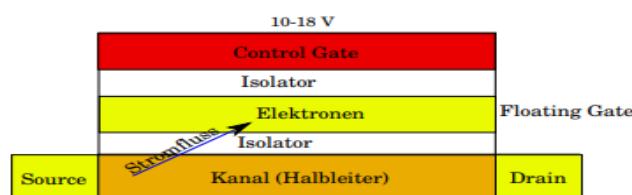


Das *Floating Gate* ist zunächst vollkommen isoliert von dem Rest des Transistors. Ungeladen hat es keinen Einfluss auf den Rest des Transistors. Jetzt kann man den Speicherinhalt folgendermaßen testen. Zwischen Source und Drain wird eine Spannung angelegt. Durch den Kanal fließt ein Strom. Der Stromfluss zeigt das Bit 1 an.



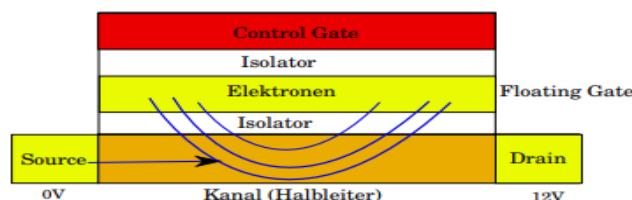
5

Nun wollen wir das Bit 0 einspeichern. Diese Aktion soll den Stromfluss zwischen Source und Drain unterbrechen. Dazu müssen Elektronen in das Floating Gate hineingezogen werden. Dafür wird eine hohe positive Spannung an das Control Gate angelegt. Diese bewirkt, dass Elektronen in das Floating Gate gezogen werden.



Obwohl zwischen Floating Gate und dem Kanal ein Isolator liegt, kann Strom fließen. Das ist möglich wegen des quantenmechanischen Tunneleffekts. Mit Hilfe des Tunneleffekts können Elektronen auch Hindernisse überwinden (oder durchtunnellen), die sie mit normaler Elektrodynamik nicht schaffen könnten. Alle unsere USB-Sticks und sonstigen Flash-Speicher funktionieren daher nur mit quantenmechanischen Phänomenen.

Die Elektronen im Floating Gate erzeugen nun ein elektrisches Feld, welches durch den Isolator in den Halbleiter wirkt, und damit den Kanal zwischen Source und Drain sperrt.

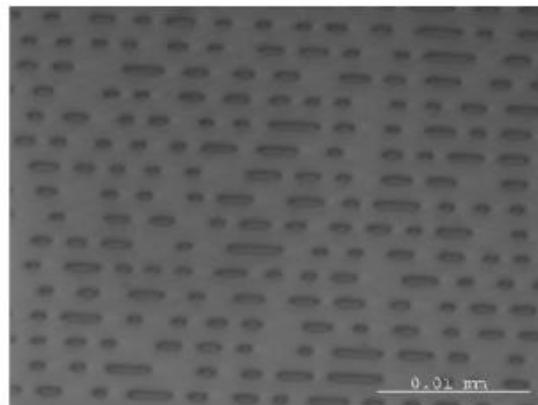


## CDs, DVDs, Blue-Rays

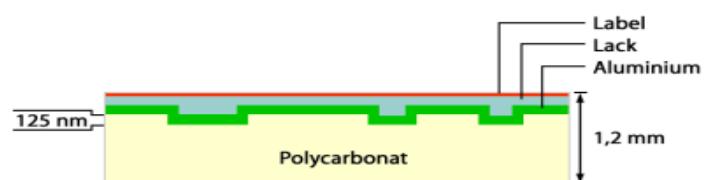
Jeder kennt sie. Aber wie sie aufgebaut sind, ist wahrscheinlich weniger bekannt.

### CDs

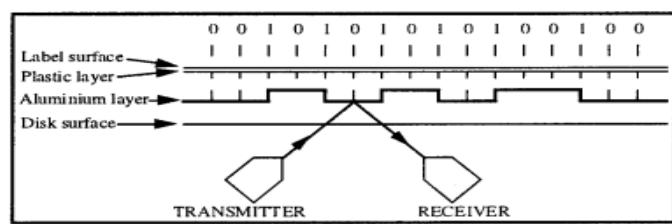
Auf den altbekannten CD-Scheiben sind die Bits durch eingepresste Löcher auf einer spiralförmigen Bahn kodiert.



Im Querschnitt sieht das so aus:



Die CDs werden mechanisch von Masterplatten gepresst, und können dann nicht mehr verändert werden. Gelesen werden sie, indem ein Laserstrahl die Spur entlangfährt. Dabei wird die Reflexion gemessen und in Bitfolgen umgesetzt.

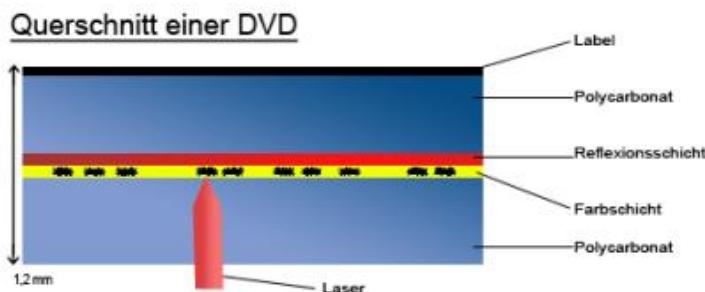


Es ist allerdings nicht so, dass z.B. ein Loch die 1 repräsentiert und kein Loch die 0. Stattdessen repräsentieren Flanken, d.h. der Abstieg in ein Loch, und der Aufstieg aus dem Loch, die 1 und gerade Strecken repräsentieren die 0.

Die Kapazität einer solchen CD ist ca. 700 Megabyte.

## DVDs

Bei DVDs werden Bits durch unterschiedliches Reflexionsvermögen einer Farbschicht dargestellt. Bei beschreibbaren DVDs lässt sich das Reflexionsvermögen durch einen Laser verändern.



Die Kapazität einer solchen DVD ist ca. 4,7 Gigabyte bzw. 9,4 Gigabyte doppelseitig.

## Blue-Ray Discs

Blue-Ray Discs funktionieren im Prinzip nach dem gleichen Prinzip wie DVDs, nur sind alle Parameter viel präziser. Der Laser hat eine kleinere Wellenlänge (violett, daher der Name Blue-Ray), der Spurabstand ist kleiner, die Linsen sind präziser, der Abstand zwischen Laser und Speicherschicht ist kleiner, die Datenrate ist höher usw.

Darüber hinaus ist für Blue-Rays eine interaktive Anwendungsschicht spezifiziert, womit man dem Benutzer mehr Freiheiten in der Steuerung des Abspielvorgangs geben kann.

Eine Blue-Ray Disc fasst 25 Gigabyte bzw. 50 Gigabyte doppelseitig.

## Random Access Memories (RAMs)

### 5 Random Access Memories (RAMs)

Die Speichertechniken mit den schnellsten Schaltzeiten, heute im Nanosekundenbereich, werden für den Arbeitsspeicher eines Rechners und für die internen Register der Prozessoren benutzt. Random Access Memories können Informationen nahezu gleich schnell lesen und schreiben, allerdings nur solange die Stromversorgung gewährleistet ist. Wird der Rechner ausgeschaltet, dann verlieren sie ihre Information.

Man unterscheidet dynamische und statische RAMs: In dynamischen RAMs (DRAM) wird die Information in Kondensatoren gespeichert. Für jedes Bit reicht ein Kondensator und ein Transistor. Allerdings verlieren die Kondensatoren relativ schnell ihre Ladung, und müssen daher regelmäßig wieder aufgeladen werden (refreshing). Während des Aufladens hat man jedoch keinen Zugriff auf die Information. Daher sind DRAMs oft langsamer als statische RAMs. Sie erlauben jedoch eine höhere Packungsdichte.

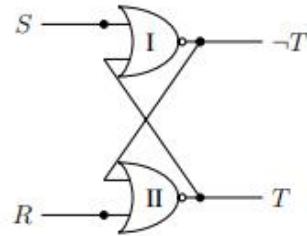
In statischen RAMs (SRAM) wird die Information in rückgekoppelten Schaltkreisen sog. Flipflops gespeichert.

#### 5.1 Flipflops

(Zum Studium dieses Abschnitts sollte man mit Schaltnetzen vertraut sein.)

Ein Flipflop, oft auch bistabile Kippstufe oder bistabiles Kippglied genannt, ist eine Schaltung, mit der es möglich ist, ein Bit an Information in eine Rückkopplungsschleife zu speichern. Die einfachste Variante ist ein *RS-Flipflop* (auch Latch genannt). Es hat einen S-Eingang (S wie Set), und einen R-Eingang (R wie Reset), sowie Ausgänge  $T$  und  $\neg T$  (die Negation von  $T$ ).

Mit NOR-Gatter Technik sieht die Schaltung so aus:

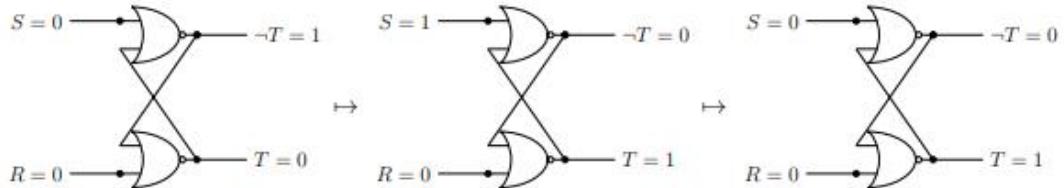


Man sieht direkt die Rückkopplung: die Eingabe von I hängt von der Ausgabe von II ab, und umgekehrt. Man kann daher nicht direkt eine Wertetabelle für die Schaltung angeben, sondern man muss sie abhängig machen vom Startwert von  $T$  und  $\neg T$ . Dann sieht sie so aus:

Startwert $T = 0, \neg T = 1$				Startwert $T = 1, \neg T = 0$			
Eingabe		Ausgabe		Eingabe		Ausgabe	
R	S	$T$	$\neg T$	R	S	$T$	$\neg T$
0	0	0	1	0	0	1	0
0	1	1	0	0	1	1	0
1	0	0	1	1	0	0	1
1	1	ungültig		1	1	ungültig	

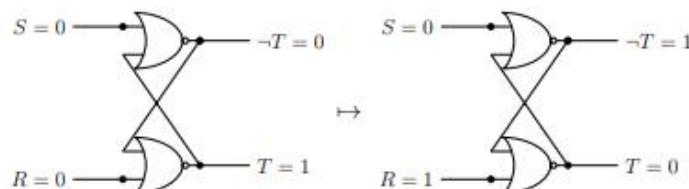
Die Kombination  $S = 1, R = 1$  bewirkt einen nicht-logischen Zustand mit  $T = 0$  und  $\neg T = 0$ , was man tunlichst vermeiden sollte.

Um zu verstehen, wie die Schaltung als Speicher funktioniert, starten wir mit  $T = 0$  und  $\neg T = 1$  und speichern dann eine 1 ein, indem kurzfristig das Signal  $S = 1$  kommt, welches aber dann wieder auf  $S = 0$  gesetzt wird.



Aus der Wertetabelle kann man ersehen, dass die Umschaltung von  $S = 1$  auf  $S = 0$  keine Änderung an der Ausgabe bewirkt. D.h.  $T = 1$  ist eingespeichert, und bleibt so.

Jetzt wollen wir  $T = 1$  wieder löschen, d.h. das gespeicherte Bit auf 0 setzen. Dabei ist wichtig, dass in diesem Moment  $S = 0$  anliegt. Wir legen jetzt  $R = 1$  an (reset).



Auch wenn man jetzt  $R = 0$  anlegt, ändert sich an der Ausgabe nichts.  $R = 0$  sollte aber trotzdem angelegt werden, bevor mit  $S = 1$  wiederum ein Bit gespeichert wird.

Die Abfolge ist also folgendermaßen:

1. Beim Aktivieren des Speichers (Einschalten des Rechners) sollte  $T = 0$  und  $\neg T = 1$ , sowie  $S = 0$  und  $R = 0$  angelegt werden. Damit wird der Speicher mit Inhalt = 0 initialisiert.
2. Um eine 1 einzuspeichern, wird kurzfristig das Signal  $S = 1$  angelegt (set), aber dann wieder auf  $S = 0$  zurückgesetzt.
3. Um die 1 zu löschen wird kurzfristig das Signal  $R = 1$  angelegt (reset), und dann auch wieder auf  $R = 0$  zurückgesetzt.

Oft ist es gewünscht, dass ein Flipflop nur in ganz bestimmten Situationen überhaupt reagiert, z.B. damit mehrere Flipflops synchron arbeiten. Um das zu erreichen, wird ein RS-Flipflop mit Zusatzschaltungen bestückt, mit denen man die Reaktionen zeitlich kontrollieren kann. Man hat dann z.B. taktgesteuerte Flipflops.

Es wurden auch eine Reihe von weiteren Flipflop-Typen entwickelt, z.B. D-Flipflops, mit denen man Signale verzögern kann, oder JK-Flipflops, die keinen dieser unerlaubten Zustände haben.

Heutige Speicherbausteine haben Millionen oder gar Milliarden solcher Flipflops zusammen geschaltet. Durch geschickte Steuerungsschaltkreise kann man dabei ganz gezielt einzelne Gruppen von Flipflops ansprechen, und damit Informationen an ganz bestimmten Adressen einspeichern und auslesen.

### 3. Die Grafikkarte (GPU – Graphics Processing Unit)

Die GPU ist die Hauptkomponente jeder Grafikkarte und ist maßgeblich für alle grafischen Berechnungen zuständig. Die GPU ist damit das Hauptrechenwerk für die Grafikberechnung und ist organisatorisch das Äquivalent zu der CPU (Central Processing Unit). Sie kann auf dem Motherboard (Integrated Graphics Processor) installiert sein, aber auch als externe Lösung, in Form einer Grafikkarte. Im Wesentlichen bestehen ihre Aufgaben aus 2D und 3D Grafikberechnungen, so wie auch komplexe mathematische Berechnungen in Form von GPGPU (General Purpose on Graphics Processor Unit). Prinzipiell beschränkt sich die GPU auf 2 Hauptaufgaben: Geometrie Erzeugung / Berechnung, welche in Form von Polygonen (bzw. eine Ansammlung von Dreiecken bei der GeForce Graphics Pipeline) und die Erzeugung von Pixeln [KiHw10, S. 22f.]. Diese Erkenntnis hat dazu geführt, dass man zwei wichtige Komponenten in die GPU integriert hat. Den Programmable vertex processor, welcher für die Geometrie zuständig ist und der Programmable fragment processor, welcher die Pixelerzeugung übernimmt.[KiHw10, S. 26ff.]

Durch die Einführung einer höheren Softwareabstraktionsschicht und die damit verbundenen APIs (DirectX und OpenGL) kann man diese beiden Prozessoren direkt ansprechen und mit ihnen Berechnungen durchführen. Unter anderem bieten moderne GPUs die Möglichkeit des Antialiasings, welche die Kanten der Polygone glättet und Anisotropes Filtern, welches für Rasterung der Texturen zuständig ist.

Moderne GPUs bestehen aus mehreren SMs (Streaming Multiprocessors), welche im Fall von Nvidia CUDA-Kerne genannt werden. Mehrere SMs werden zu einem Block zusammengefasst, bei der Geforce 8800er Reihe bilden zwei SMs einen Block. Jeder SM hat eine bestimmte Anzahl von SPs (Streaming Processors), welche sich die Steuerungslogik wie auch den Programmspeicher teilen [KiHw10, S. 8]. Aktuelle Grafikkarten besitzen bis zu 512 Stream Prozessoren.

## 2.2 Bussystem

Das Bussystem verbindet logisch die GPU mit den anderen Busteilnehmern. Der heute übliche Standard ist PCIe (Peripheral Component Interconnect express), welcher den AGP (Accelerated Graphics Port) ersetzt hat. Bei aktuellen Grafikkarten sind mit PCIe 2.0 und mit 16 Lanes theoretisch Datenraten bis zu 8000 MB/s möglich [Whit04, S.93ff].

## 2.3 Grafikspeicher

Der Grafikspeicher ist für die temporäre Speicherung der grafikrelevanten Daten zuständig. Es gibt ihn in 3 verschiedenen Ausführungen, als Reservierung des Hauptspeichers, als eigenen Speicherbaustein auf der Grafikkarte oder als Kombination aus den beidem. Der Grafikspeicher unterscheidet sich von dem Arbeitsspeicher im wesentlichen in der erhöhte Taktrate. Dadurch wird die Datentransferrate erhöht. Moderne Grafikkarten haben Datentransferraten von bis zu 177 Gbyte/s [iX,2011] . Diese erhöhte Geschwindigkeit ermöglicht Grafikkarten eine deutlich schnellere Berechnung von komplexen Daten, da diese schneller nachgeladen werden können. Durch diesen Vorteil und durch die Einführung von GPGPU ist im Vergleich zu einem Programm, welches auf der CPU alleine ausgeführt wird ein 10x Speed-Up möglich [KiHw10, S. 10f.]. Weitere Optimierungen am Code, welche die Datenzugriffe des Arbeitsspeichers wie auch die des Grafikspeichers mindern, und eine Verminderung des sequentiellen Teils des Codes, ermöglicht einen bis zu 100x Speed-Up [KiHw10, S. 12]. Der heute übliche Standard für Grafikkarten ist GDDR5 (Graphics Double Data Rate), welcher eine Weiterentwicklung des GDDR Grafikspeicher ist. Allgemein gehört der GDDR Speicher zu dem DDR-SDRAM-Standard, in dem die Taktfrequenz erhöht wurde. GDDR5 Speicher hat Taktfrequenzen von bis zu 4 GHz[Davi, 2011]. Die Größe des Speichers liegt im Durchschnitt der aktuellen Grafikkarten bei 2 GB. Dieser wird größtenteils zur Bildspeicherung benutzt aber auch für Berechnung wie zum Beispiel beim Rendern.

2.4 RAMDAC Der RAMDAC-Chip (Random Access Memory Digital/Analog Converter) liest in regelmäßigen Abständen digitale Daten aus dem Framebuffer im Grafikspeicher aus und wandelt diese in analoge Bildsignale um, welche von Monitoren wiedergegeben werden können. Durch die Einführung von digitalen Ausgängen wie zum Beispiel HDMI und DVI verlor der RAMDAC an Bedeutung. Die Bilder müssen daher nicht mehr Digital/Analog umgewandelt werden. Bei dieser Art von Ausgängen kommt ein TMDS-Modul (Transition Minimized Differential Signaling) zu tragen, welches die digitalen Daten aus dem Grafikspeicher zu anderen Geräten transportiert.

## 2.5 Sonstiges

Erwähnenswert sind die Ausgänge. Heute übliche Standards sind VGA (Video Graphics Array), als heute noch einziger analoge Ausgang, DVI (Digital Visual Interface), HDMI (High Definition Multi-media Interface) und DisplayPort. Ebenso nennenswert sind die Kühlervarianten, welche die Grafikkarte vor Überhitzung schützt. Grafikkartenkühlsysteme können aus elektrisch angetriebenen Lüftern (Axial- Radiallüfter), aus passiven Kühlkörper oder als Wasserkühlung bestehen.

[Vött,2011, S.8]. Keine direkte Hardware, aber dennoch sehr elementar für das Verständnis von Grafikkarten, ist die sogenannte Grafik-Pipeline. Die Grafik-Pipeline ist eine Abarbeitungskette, welche für die Erzeugung von Grafik nötig ist. Sie besteht aus mehreren Unterschritten. Im Prinzip ist es notwendig, dass die CPU der GPU Geometrie- und Pixel-Daten zur Verarbeitung übergibt und die Grafikkarte, nach Ablauf dieser Kette, Grafik erzeugt. Im nächsten Kapitel wird u.a. darauf eingegangen, an welchen Stellen der Abarbeitungskette eingegriffen werden kann um die Grafikerzeugung zu optimieren bzw. zu vereinfachen [KiHw10, S. 23f.].

## 4 Parallelität

Im folgenden Kapitel, wird erläutert warum Parallelität einen enormen Faktor zur Leistungssteigerung beiträgt, wie man sie anwendet und versteht und wie diese Übertragbar ist auf Grafikkarten. Ebenfalls wird ein Vergleich zwischen CPUs und GPUs genannt, welcher verdeutlicht zu welcher Rechenleistung Grafikkarten im derzeitigen Stadium fähig sind.

### 4.1 Zeitalter der Parallelität

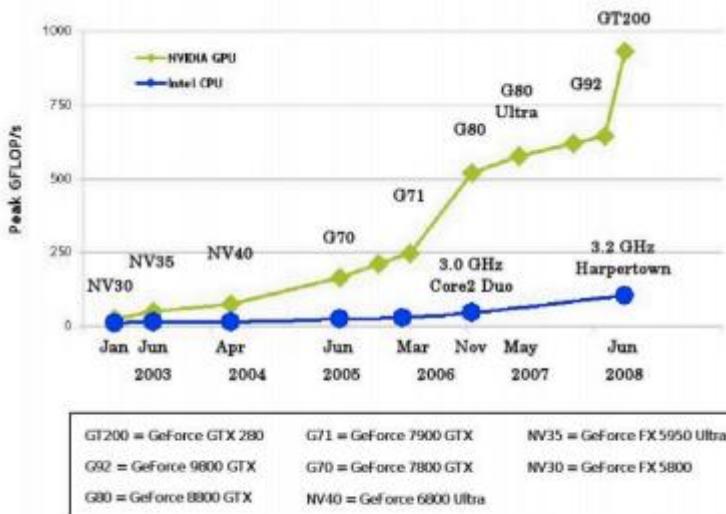
Einführend zu der historischen Entwicklung von GPGPU sei ein Zitat genannt welches sich auf die Entwicklung von CUDA bezieht: „*There was a time in the not-so distant past when parallel computing was looked upon as an „exotic“ pursuit and typically got compartmentalized as a specialty within the field of computer science This perception has changed in recent years. The computing world has shifted to the point where far from being an esoteric pursuit, nearly every aspiring programmer needs training in parallel programming to be fully effective in computer science*“.

[SaKa11,S.1] Im Laufe der Entwicklung von CPUs im Desktop-Bereich stieß man an Grenzen, da die Taktrate nicht beliebig erhöht werden konnte. Infolge dieser Problematik wurde die Entwicklung von Mehrkern-Prozessoren vorangetrieben, welche bereits bekannt waren aus dem Server-Bereich. Man erhoffte sich theoretisch eine erhöhte Leistungssteigerung, welche aber nicht ohne Anpassung am Code möglich war. In den Entwicklungsphasen der CPU, wurde zeitgleich die Entwicklung des ausführbaren Codes von Betriebssystemen vorangetrieben. Mit der Entwicklung dieser angepassten Betriebssysteme konnte die CPU parallel mehrere Prozesse ausführen. Aber nicht nur die Hersteller solcher Betriebssysteme profitierten von dieser Leistungssteigerung, sondern auch die Entwickler von Anwendersoftware, da diese größtenteils Threads benutzen, welche ebenfalls auf Kerne der CPU verlagert werden können und damit Echt-Parallel ausgeführt werden können. Die Entwicklung und Akzeptanz solcher parallelen Software, führte die großen Hardwarehersteller dazu vermehrt Mehrkern-Prozessoren zu verbauen um eine Verbesserung der Leistung

herbeizuführen. Wie bereits oben erwähnt führte gerade das Nvidia dazu, ihre Grafik-Pipeline so umzubauen, dass sie die neuen Einheitsprozessoren mit Hunderten von diesen Prozessorkernen ausnutzen. In der folgenden Tabelle ist ein Vergleich zwischen einer modernen CPU und einer aktuellen Grafikkarte mit solch einer GPU dargestellt.

Prozessor/ Grafikkarten-typ	Intel Core i7-980X Gulftown	Nvidia Geforce GTX 580
Taktrate	3,33 GHz	772 MHz
Prozessorkerne	6	512
GFLOPS	107,58	1581

Quelle: In Anlehnung an SiSoftware Sandra Gflops Test und Pc Games Hardware Ausg. 05/11



Quelle: Cuda Präsentation Uni Kassel.

## 4.2 Werkzeuge der Parallelität

Um den hohen Grad der Parallelität von Grafikkarten ausnutzen zu können, bedarf es einiger Vorkenntnisse und Empfehlungen. In erster Linie ist ein Umdenken erforderlich. Parallele Programme müssen vom Aufbau her komplett neu organisiert werden. In der Literatur wird von einem Pfirsich-Modell gesprochen [KiHw10,S.12] . Das Pfirsich-Modell beschreibt wie Programme aufgebaut werden müssen, damit die Rechenleistung von parallelen Prozessoren bestmöglich ausgenutzt wird. Bestandteile des Pfirsich-Modells sind der Kern und das Fleisch. Das Fleisch ist der parallele Teil eines Programms. Im Gegensatz zu dem Kern ist dieser weitaus größer. Daher sollte ein Programm möglichst viele Berechnungen parallel ausführen, weshalb findet die Ausführung auf einer Grafikkarte oder ähnlichen Multiprozessorsystemen statt. Der Kern repräsentiert den sequentiellen Teil eines Programms, welcher von der CPU ausgeführt wird. Der sequentielle Teil sollte größtenteils sequentiell bleiben und man sollte sich bemühen keine Optimierung zu unternehmen, um diesen in irgendeiner Form zu einem parallelen Teil umzuformen. Das Pfirsich-Modell beschreibt wie eine Art Software zu entwickeln ist und diese auf verschiedene Komponenten eines Systems zu verlagern. Durch dieses Modell ist es möglich, die Stärken von CPU und GPU softwareseitig zu

kombinieren, denn die CPU ist dafür ausgelegt den sequentiellen Teil zu managen und auszuführen, während die GPU schnelle und massiv parallele Berechnungen durchführt. Ebenfalls empfehlenswert ist es den Grad der Parallelität in Programmen möglichst hoch zu halten. Der Grad der Parallelität bestimmt wieviel Berechnungen parallel ausgeführt werden können. Ist der Grad gering so ist die maximale Leistungsausbeutung geringer. Weitaus wichtiger als Programmierrichtlinien ist die zu programmierende Hardware. Wie bereits oben erwähnt benötigt man hierfür eine Multiprozessorsystem, welches sich durch Schnittstellen ansprechen lässt. Im besten Fall ist eine API vorhanden um sich einen Arbeitsaufwand von hardwarespezifischen Zugriffe zu vermindern.

## 5 CUDA

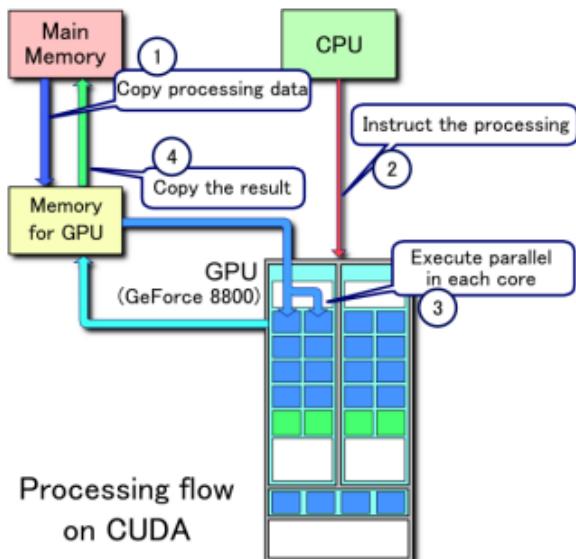
Als großes Beispiel für die Nutzung von GPGPU wird im folgenden Kapitel CUDA vorgestellt. CUDA gilt momentan als die größte Innovation für das Rechnen auf der Grafikkarte und ist zumindest in diesem Bereich weit verbreitet. Im Folgenden wird eine prinzipielle Übersicht, was CUDA ist und wie es funktioniert gegeben.

### 5.1 Einführung in CUDA

CUDA ist in erster Linie eine auf C basierende API, welche Funktion hinzufügt, aber auch Funktionen einschränkt. Mit CUDA ist es möglich, beliebige Algorithmen auf der Grafikkarte auszuführen und dies massiv parallel. Als Voraussetzung für die Nutzung von CUDA benötigt man eine Grafikkarte mit der oben bereits erwähnten CUDA-Hardware-Architektur, welche aus einem Verbund von hunderten Einheitsprozessoren besteht. Eine CUDA-Hardware-Architektur bieten die Karten von Nvidia, welche nach der Geforce 8800er Reihe (8800er Reihe mit einbezogen) gefertigt wurden. Vor allem in der Forschung wurden bereits große Erfolge erzielt. So schreibt zum Beispiel der Hersteller selbst auf seiner Website: „*Im Bereich wissenschaftlicher Forschung wurde CUDA mit Begeisterung aufgenommen. CUDA beschleunigt jetzt zum Beispiel AMBER, ein Molekulardynamik-Simulationsprogramm, das weltweit von mehr als 60.000 Forschern an Universitäten und in Pharmaunternehmen zur Beschleunigung der Medikamentenentwicklung eingesetzt wird.*“ [Nvidia,2011] und bezieht sich dabei auf eins von vielen Forschungsgebieten. Weitere CUDA Fallbeispiele werden in Kapitel 6 erläutert. Prinzipiell ist CUDA nur in C verfügbar, mit Wrapper kann man aber auch Programmiersprachen wie Python, Java oder eine .NET Sprache benutzen. Seit der neusten Architektur ist auch eine Benutzung von C++ vorgesehen [Nvidia2011].

### 5.2 Funktionsweise von CUDA

Abstrakt gesehen funktioniert die Berechnung auf der Grafikkarte mit Hilfe von CUDA in 4 Schritten. Im Folgenden wird eine Grafik dargestellt, welche dies sehr gut verdeutlicht.

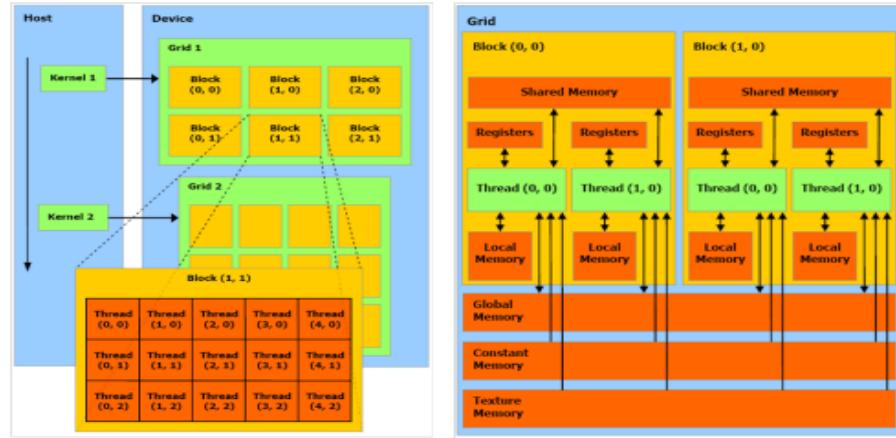


Quelle: Wikipedia Cuda englisch

Im folgenden werden die Schritte erläutert. Zuerst kopiert die CPU die zu berechnenden Daten von dem Hauptspeicher zu dem Grafikkartenspeicher, was nötig ist damit die Grafikkarte nicht permanente Speicherzugriffe auf den Hauptspeicher ausübt und somit der Hauptspeicher zu dem Flaschenhals des Gesamtsystems wird. Weiterhin ist zu Punkt 1 zu erwähnen, dass durch das kopieren von den zu berechnenden Daten in den Grafikkartenspeicher, die Zugriffsgeschwindigkeit des Grafikspeichers erst ausgenutzt wird und die CUDA-Hardware-Architektur Speicherzugriffe optimieren bzw. ermöglichen kann. Im 2. Schritt erteilt die CPU den Befehl den zu berechnenden Algorithmus auf der Grafikkarte auszuführen, womit die Grafikkarte die komplette Kontrolle über den auszuführenden Code bekommt. In Schritt 3 organisiert die Grafikkarte die parallele Abarbeitung. Dies geschieht in dem die Daten sinnvoll auf alle SMs verteilt werden, welche dann die interne Kontrolle über ihre SPs haben. Um dies genau zu erklären bedarf es eines Vorgriffs auf das nächste Kapitel. Kurz zusammengefasst teilt man zusammenhängende Daten wie es z.B. bei Arrays der Fall ist, auf verschiedene SMs auf, um somit möglichst viel der parallelen Architektur auszunutzen. In Schritt 4 werden die fertig abgearbeiteten Daten wieder der CPU übergeben, welche dann mit der sequentiellen Abarbeitung fortfährt. [KiHw10,S.48ff.]

### 5.3 Interne parallele Abarbeitung

Um die massive Parallelität der Einheitsprozessoren ausnutzen zu können Bedarf es einer Umstrukturierung des Codes und einem Umdenken bei der Organisierung des Codes. CUDA gibt drei Möglichkeiten Daten zu organisieren um sie möglichst effektiv ausführen zu lassen. Alle drei Möglichkeiten sind hierarchisch angeordnet. Anschließend ist ein Bild für die Organisation dieser Hierarchie dargestellt.



Quelle: In Anlehnung an Programming Massively Parallel Processors

### 5.4 CUDA Code

Nun soll verdeutlicht werden, wie CUDA angewendet wird. Gezeigt wird eine simple Vektor Addition. Zunächst ist ein Beispiel dargestellt, welches in C ohne CUDA programmiert wurde. Die Kommentare helfen zum Verständnis.

```

1 #define N 10 // Größe des Vektors
2
3 void add(int *a, int *b, int *c) {
4     int tid = 0; // Element des Vektors, initialisiert bei 0;
5     while (tid < N) {
6         c[tid] = a[tid] + b[tid];
7         tid += 1;
8     }
9 }
10
11 int main(void) {
12     int a[N], b[N], c[N]; //Initialisierung der Vektoren.
13
14     /*Die beiden Vektoren, welche zusammen addiert werden sollen,
15      * werden mit Werte initialisiert.
16      */
17     for (int i = 0; i < N; i++) {
18         a[i] = -i;
19         b[i] = i * i;
20     }
21     /* Funktionsaufruf der Methode add.
22      * Alle Vektoren werden als Referenz übergeben.
23      */
24     add(a, b, c);
25     // Ausgabe
26     for (int i = 0; i < N; i++) {
27         printf("%d + %d = %d \n", a[i], b[i], c[i]);
28     }
29 }
30
31

```

Wie man sieht ist dies eine simple Vektor-Addition, welche vollkommen von der CPU ausgeführt wird. Das nächste Beispiel zeigt wie dieser Code in CUDA-C aussieht.

## 4. PC-Netzteile

Ein PC-Netzteil, auch Computer-Netzteil, ist ein Schaltnetzteil für den Einsatz als Stromversorgung in Mikrocomputern. Die in den meisten Fällen eingebauten Lüfter dienen nicht nur der Eigenkühlung, sondern ganz oder teilweise auch der Kühlung der sonstigen im Computergehäuse eingebauten Komponenten.

### Nennleistung

Die Nennleistung eines Netzteils bezeichnet die von den Herstellern garantierte maximale Stromabgabe des Netzteils. Dieser Wert liegt bei gängigen Modellen zwischen 300 und 600 W. Die Nennleistung steht – bis auf einen kleinen Einfluss auf den Wirkungsgrad (siehe dort) – in keinem Zusammenhang mit dem tatsächlichen Stromverbrauch des Rechners.

Neben der Gesamt-Nennleistung wird in den technischen Daten die Belastbarkeit der einzelnen Spannungsschienen in dem Netzteil angegeben. Hierbei liegt der Wert für 3,3 und 5 V unabhängig von der Gesamtleistung bei etwa 100 W, der Rest verteilt sich auf die 12-V-Schiene(n). Bei hochwertigen Netzteilen wird als Nennleistung des Gesamtnetzteils nur eine aufgerundete 12-V-Leistung angegeben; bei besonders einfachen Modellen wird dagegen zum Teil die Summe aller Einzelleistungen abgerundet.

Die für ein Rechnersystem benötigte Nennleistung ist vom eingesetzten Mainboard und der darauf vorhandenen CPU sowie den verbauten Komponenten abhängig. Eine empfohlene Nennleistung ist meist aus dem Handbuch der Hauptplatine ersichtlich. Zu knapp dimensionierte Netzteile können Leistungsspitzen nicht hinreichend ausgleichen, was zu schwer eingrenzbaren Stabilitätsproblemen führen kann.

PC-Netzteile regeln die 5-V-Ausgangsspannung. Diese muss sehr stabil sein, um die Funktion des PCs zu erhalten. Die 12 V laufen über denselben Transformator mit, daher schwanken die 12 V Nennspannung zwischen 10,5 V und 13,5 V je nach Lastverhältnis der Ausgänge 5 V und 12 V. Die 12 V werden für Laufwerksmotoren verwendet, bei denen stets die Drehzahl geregelt wird und zur Speisung von Abwärtswandlern auf Mainboards und Grafikkarten. Da diese Wandler Ausgangsspannungen unter 2,5 V erzeugen und von der jeweiligen CPU bzw. GPU abhängig eingestellt werden, können sie nicht im Netzteil untergebracht werden. Eine feinere Regelung der 12 V ist daher nicht notwendig. Die nach den sekundären Gleichrichtern geschalteten Speicherdrösseln für 5 V und 12 V sind über einen gemeinsamen Ringkern gewickelt.

Netzteile für Spieler-PCs mit einer Ausgangsleistung über 600 W haben weitere separate 12-V-Netzteile für die Grafikprozessoren und Laufwerke im selben Gehäuse eingebaut. Technisch ist dies wie Siamesische Zwillinge verschaltet, um die Bauteilkosten zu begrenzen, sowie bei identischen Kabelquerschnitten Kurzschlussfestigkeit zu gewähren. Diese Ausgänge sind mit 12V1, 12V2, 12V3 usw. gekennzeichnet.

Die 3,3 V wurden Anfangs über einen Abwärtswandler aus den 5 V generiert. Heute besitzt der Übertrager Wicklungen oder Abgriffe für die 3,3 V. Häufig sind gemeinsame Angaben über Gesamtausgangsleistung der 3,3 V und 5 V spezifiziert. Da bei niedrigen Spannungen und höheren Strömen ein erheblicher Spannungsabfall an den Leitungswiderständen von Kabeln und Steckkontakte auftritt, werden die 3,3 V am mainboardseitigen Stecker abgetastet und auf den Regler zurückgeführt. In der Praxis werden die 3,3 V mit zwei Leitungen im selben Schuh des ATX-Steckers gespeist und zurückgemessen. Verlängerungen und Adapter zwischen Netzteil und Board können sich unter hoher Last störend auswirken, da der verlängerte Kabelabschnitt nicht Teil dieses Regelkreises ist. Als Regler werden u.a. die pin- und funktionsgleichen IR3M02, TL494CN, KB7500B verwendet.

Das PG-Signal (Power Good) wird über ein Zeitglied oder Komparator auf die 5 V geschaltet. Sobald es 5 V führt, ist sichergestellt, dass das Netzteil nach dem Einschalten seine Ausgangsspannungen aufgebaut hat. Für das Mainboard bedeutet dies, dass es mit einem internen RESET-Signal den Betrieb starten kann. Dies wird vom Taktgenerator verarbeitet.



Blick ins PC-Netzteil



PC-Netzteil ohne Netzfilter, ohne PFC



PC-Netzteil mit vier (drei weiteren) Optokopplern statt Übertrager

## Eingangsspannungsumschaltung

Netzteile mit Spannungsumschalter 115/230 V sind auf eine mittlere Eingangsspannung von 160 V optimiert und erreichen dabei ihren höchsten Wirkungsgrad. Sie besitzen zwei in Serie geschaltete Kondensatoren mit 200 V und mindestens 220  $\mu$ F (bei 250 bis 300 W Nennleistung). Um die Spannung gleichmäßig auf die Kondensatoren zu verteilen, sind Varistoren (spannungsabhängige Widerstände) parallel zum jeweiligen Kondensator geschaltet. Der Spannungsumschalter brückt in der Stellung 115 V eine Phase der Netzspannung zwischen die beiden Kondensatoren. Auf diese Weise wird aus dem Brückengleichrichter ein Zweiwegegleichrichter. Hierdurch entsteht der klassische Spannungsverdoppler in Delon-Brückenschaltung.

Einige Netzteile können ab 80 bis 240 V Eingangsspannung verarbeiten. Diese Netzteile erreichen dies mit einem überdimensioniertem primärseitigen 450-V-Kondensator. Die Spezifikationen der Passiv-PFC sind damit weitgehend erfüllt. Sie werden ggf. über eine am Gleichrichter in Reihe geschaltete Induktivität gewährleistet. Leider kommen teilweise 400-V-Kondensatoren zum Einsatz. Diese können eine Fehlerquelle darstellen, da bereits bei 230 V die Spitzenspannung zuzüglich Netzspannungstoleranz über 400 V liegt.

Im Eingang sind sowohl eine Schmelzsicherung, als auch ein NTC (Heißleiter zur Begrenzung des Einschaltstromes) Serie geschaltet.

## PFC

In der EU verkauftene elektrische Verbraucher mit einer Nennleistung über 75 W – also sämtliche PC-Netzteile – müssen über einen Leistungsfaktorkorrekturfilter (*englisch Power Factor Compensation*, daher *PFC* abgekürzt) verfügen. Von diesen gibt es zwei Bauformen. Die passive Version ist vergleichsweise billig, schwer und nur begrenzt wirksam. Die zweite, teurere Lösung ist ein aktiver Filter, der zudem eine bessere Wirkung aufweist. Die 80-PLUS-Zertifizierung schreibt einen Leistungsfaktor über 0,9 vor, der mit einer passiven PFC nur schwer erreichbar ist, sodass de facto sämtliche 80-PLUS-Netzteile einen aktiven Filter besitzen. In der Praxis bewegen sich aktive PFCs zwischen 0,95 und 1.

Ein wichtiges Merkmal eines PC-Netzteils ist sein Wirkungsgrad, der von der technischen Qualität der Konstruktion und der elektrischen Belastung abhängt. Allgemein gilt ein Wert von 80 % als untere Grenze für ein Netzteil mit „gutem“ Wirkungsgrad. Ursächlich hierfür ist vermutlich die 80-PLUS-Kampagne, für die ein Mittelwert von Bedeutung ist, der an den Leistungspunkten 20 %, 50 % und 100 % gemessen wird. Die derzeit besten Netzteile liegen im Bereich von 88 % Wirkungsgrad bei 20 % und 100 % Belastung und über 90 % Wirkungsgrad bei 50 % Last. In unteren Preisklassen sind noch Modelle mit einem Wirkungsgrad von weniger als 50 % erhältlich. Wie die Wirkungsgrade für andere Lastwerte aussehen, lässt sich aus dem angegebenen Wirkungsgrad nicht ersehen, es wird jedoch allgemein davon ausgegangen, dass es hierbei nur kleine Abweichungen gibt und einzelne Netzteil-Tests bestätigen dies auch. Bei einer Belastung von unter 20 % sinkt der Wirkungsgrad allerdings stark.

Letzteres führt zu Problemen, da sich der Netzteil-Markt in den letzten Jahren immer mehr von der sonstigen Entwicklung im PC-Markt abgekoppelt hat. Netzteile, die über eine „80 PLUS“-Kennzeichnung verfügen, sind erst ab einer Nennleistung von 300 bis 350 Watt im Einzelhandel zu bekommen; Modelle mit bis zu 90 % Wirkungsgrad oft erst ab 500 Watt. Technisch wäre anderes möglich. So ist beispielsweise bei Dell ein 235-Watt-Modell im Angebot, das an den Prüfpositionen 20 %, 50 % und 100 % im Mittel einen Wirkungsgrad von knapp über 90 % erzielt.<sup>[1]</sup>

Im Gegensatz dazu hat ein moderner PC ohne dedizierte Grafikkarte, wie oft im Office-Bereich eingesetzt, im Leerlauf meist nur eine Leistungsaufnahme von 40 bis 50 Watt, die unter Last selten über 100 Watt steigt.<sup>[2]</sup> Hierfür wären also Netzteile mit einer Nennleistung von maximal 200 bis 250 Watt sinnvoll, um die 20 % Last nicht zu unterschreiten. Eine Leistungsaufnahme von 350 Watt wird hingegen normalerweise erst bei der Verwendung von Hochleistungsgrafikprozessoren (insbesondere sogenannter *Dual-GPU-Karten* wie Nvidia GeForce GTX295 oder ATI Radeon HD 4870 X2) überschritten.<sup>[3]</sup> Bei noch umfangreicheren Konfigurationen ist dann eine Leistungsaufnahme von mehr als 1000 Watt möglich.<sup>[4]</sup>

Eine Optimierung des Wirkungsgrades hängt von den verwendeten Siebkondensatoren ab, je schneller sie altern desto häufiger muss vermeidbar geschaltet werden, was einen Verlust mit sich bringt. Wesentliche Einsparungen an Energie brächte der Einsatz Synchrongleichrichtern auf der Sekundärseite mit sich. Auf Dioden fallen je nach Technologie und Halbleitermaterial 0,7 bis 0,3 V ab, während Transistoren auf weit unter 0,2 V möglich sind, was sich auf die Ausgänge der 3,3 V und 5 V besonders (mit bis zu 10 W Einsparung je nach Anwendung) auswirkt, die nichtmehr über den Kühlkörper abgegeben werden müssen. Die Erhöhung des Schaltfrequenzbereiches verkleinert und optimiert die Übertrager, verteuert zudem die Schalttransistoren. Ein weiterer Verlust kann bauartbedingt in der PFC auftreten.

## Stecksysteme

Seit ca. 2006 werden bei einigen Netzteilen Stecksysteme für die internen Anschlüsse angeboten, oft „Kabelmanagement“ genannt. Dabei sitzen an der Innenseite des Netzteils mehrere Buchsenleisten. Die Kabel zu Grafikkarten und Laufwerken, in Einzelfällen auch die zur Versorgung der Hauptplatine, sind daran steckbar. Diese Systeme führen einerseits zu einer größeren Flexibilität, da nicht benötigte Kabel entfernt werden können und nicht den Kühlstrom behindern können oder Kabel in unterschiedlichen Längen angeboten werden können. Allerdings führen diese Steckverbindungen durch den Übergangswiderstand am Steckverbinder zu einer leicht verschlechterten Effizienz und Überlastungen oder Wackelkontakte können zu Ausfällen führen.

## Passive gekühlte Netzteile

Als „passiv gekühlt“ wird ein Netzteil bezeichnet, das über keinen Lüfter zur Wärmeabfuhr verfügt, sondern hierfür relativ große Rippenkühler verbaut sind. Selten wird ein Peltier-Element eingesetzt. Letzteres erlaubt die Abfuhr größerer Wärmemengen, erhöht wegen des sehr schlechten Wirkungsgrads aber den Stromverbrauch und produziert selbst Wärme, die das Gehäuseinnere des PCs weiter aufheizt und die anderweitig abgeführt werden muss. Passiv gekühlte Netzteile sind selbst zwar im Gegensatz zu aktiv gekühlten Netzteilen geräuschlos, die gegebenenfalls erforderlichen zusätzlichen Gehäuselüfter oder auch Wasserkühlungen nicht.

Eine Abwandlung sind semi-passive Netzteile, die zwar über einen Lüfter verfügen, der aber nur bei Bedarf eingeschaltet wird. Dies ist bei einigen modernen Netzteilen der Fall, die einerseits einen hohen Wirkungsgrad (~90 %) besitzen, der die Wärmemenge reduziert, und andererseits einen hohen Leistungsbereich abdecken, der zusätzliche Kühlung erfordert kann.

Die AT-Netzteile (*Advanced Technology*) unterscheiden sich von heutigen Netzteilen insbesondere dadurch, dass sie einen herausgeführten Schalter besitzen. Diese Schalter befanden sich ursprünglich direkt im Netzteilgehäuse und waren auf der Rückseite oder durch eine Gehäuseaussparung seitlich erreichbar. Später wurden sie mittels einer Netzteilungsverlängerung in der Front des PC-Gehäuses untergebracht, wobei es keinen übergreifenden Standard für die Art, die mechanische Bauform und die Montage des verwendeten Schalters gibt.

Sie schalten das Netzteil und damit den PC auf der Netzspannungsseite physikalisch aus. Es gibt also keine Stand-By-Modi.

Ferner ist der Stromanschluss für ein AT-Mainboard mit zwei mechanisch identischen, aber unterschiedlich belegten Steckern realisiert. Im Normalfall werden sie nebeneinander gesteckt, mit den schwarz markierten Adern zueinander. Die beiden Steckerhälften waren nicht mechanisch kodiert und konnten daher auch vertauscht aufgesteckt werden, was zu einem Kurzschluss führt.

An Steckverbindungen bieten diese Netzteile:

- zweiteiliger Stecker zur Stromversorgung der Hauptplatine ( $\pm 12\text{ V}$ ,  $\pm 5\text{ V}$  und GND)
- 4-Pin Molex (für interne Peripherie wie Festplatten u. Ä.) (5 und 12 V)
- 4-Pin Floppy Disk-Stromversorgung (5 und 12 V)



Kaltgerätestecker Netzspannungs Aus- und Eingang.  
Bei ATX ist der Ausgang nur in Ausnahmen vorhanden und auf Dauerstrom oder über ein Relais geschaltet

### Netzteilstecker

Farbe	Pin	Signal
orange	P8.1	Speisung gut
rot	P8.2	+5 V
gelb	P8.3	+12 V
blau	P8.4	-12 V
schwarz	P8.5	Masse
schwarz	P8.6	Masse
schwarz	P9.1	Masse
schwarz	P9.2	Masse
weiß	P9.3	-5 V
rot	P9.4	+5 V
rot	P9.5	+5 V
rot	P9.6	+5 V

ATX steht für *Advanced Technology Extended*. Bei ihrer Einführung 1995 verfügten diese PC-Netzteile über folgende Steckverbinder:

- 20-Pin (Steckerbelegung) zur Stromversorgung der Hauptplatine
- 4-Pin Molex (für interne Peripherie)
- Floppy Disk-Stromversorgung

Außerdem in neueren Ausführungen

- 24 statt 20-Pin (Steckerbelegung) zur Stromversorgung der Hauptplatine
- 4-Pin Hauptplatine (ATX12V bzw. „Intel P4-fähig“)
- 8-Pin Hauptplatine (EPS12V-Standard, statt dem 4-Pin Stecker, Intel-Multicoreprozessoren nach Core2, i7 etc.)
- 6 oder 8 Pin PCI-E (Anzahl variiert, meist für leistungsstarke PCI-E Steckkarten wie Grafikkarten)
- S-ATA (zum Anschluss interner Peripherie bzw nur S-ATA Geräte)
- optional Tachosignalstecker zum Anschluss auf die Hauptplatine zum Auslesen der Netzeillüfter-Drehzahl

Die übliche Bauform eines PC-Netzteils ist ein quaderförmiges Blechgehäuse mit einer Euro-Kaltgeräte-Buchse für die Netzspannung, optional einem Netzschalter (dann in seltenen Fällen mit einer Euro-Kaltgeräte-Steckerbuchse zur mitgeschalteten Monitor-Stromversorgung) sowie einem oder mehreren Lüftern. Ursprünglich dienten die Lüfter nicht nur der Kühlung des Netzteils selbst, sondern der Kühlung des Rechners insgesamt, indem der oder die Lüfter des Netzteiles Luft nach außen fördern, somit im Rechnergehäuse Luft durch diverse Schlitze und Öffnungen angesaugt wird und am Netzeil das Gehäuse wieder verlässt.

Bei leistungsstarken Computern reicht die Kühlwirkung des Netzeillüfters allein meist nicht mehr aus und muss durch andere Maßnahmen, beispielsweise zusätzliche Gehäuselüfter, ergänzt werden.

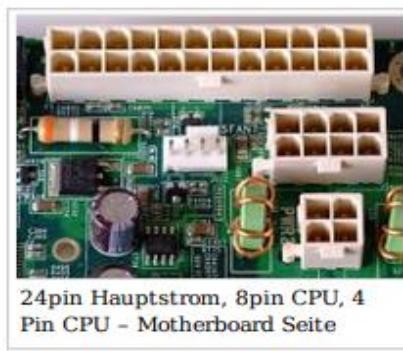
PC-Netzteile lassen sich – sofern spezifiziert – an Stromnetzen mit unterschiedlicher Spannung betreiben. Die Umschaltung kann manuell oder automatisch erfolgen. Bei manueller Umschaltung, die meist über einen nur mit einem Werkzeug zu betätigenden Schalter erfolgt, besteht bei falscher Einstellung die Gefahr einer sofortigen Zerstörung des Netzteils. Moderne Netzteile weisen häufig eine automatische Anpassung auf, die den Betrieb zwischen 100 V (teilweise ab 80 V) und 240 V Netzspannung erlauben.

PC-Netzteile müssen mindestens folgende Ausgangsspannungen zur Verfügung stellen: +12 V, +5 V, +3,3 V, -12 V, +5 VSB (Standby-Spannung), wobei nach dem neuesten ATX-Standard mindestens zwei +12-V-Schienen vorhanden sein müssen. Der Nutzen mehrerer 12-V-Leitungen ist aber umstritten, da diese ursprünglich dafür vorgesehen waren, eine stabile Stromversorgung bei zunehmender Last zu gewährleisten. Eine 12-V-Leitung darf laut ATX-Spezifikation nicht mehr als 20 Ampere haben, bevor eine zusätzliche Leitung notwendig wird. Es hat sich aber gezeigt, dass die Netzeilhersteller keine Probleme haben, ihre Netzteile so zu entwickeln, dass sie eine höhere Leistung weit über 20 Ampere hinaus leisten können. Bei Verlängerungs-, Y- und Adapterkabeln kann die Kurzschlussfestigkeit beeinträchtigt sein.

Die Spannungen werden u. A. für Folgendes gebraucht:



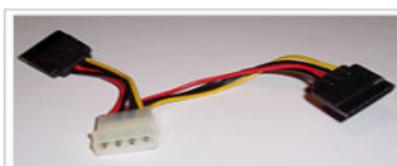
24pin Hauptstrom, 8pin CPU, 4 Pin CPU – Netzteil Seite



24pin Hauptstrom, 8pin CPU, 4 Pin CPU – Motherboard Seite



Anschlüsse eines ATX-2-Netzteils



SATA-Molex 4-Pol-Adapter, der keine 3,3 V an das SATA-Laufwerk gibt

Häufig werden angeschlossene USB-Geräte auch über die +5VSB mit Strom versorgt, sodass man sie z. B. zum Aufladen anderer Geräte benutzen kann.

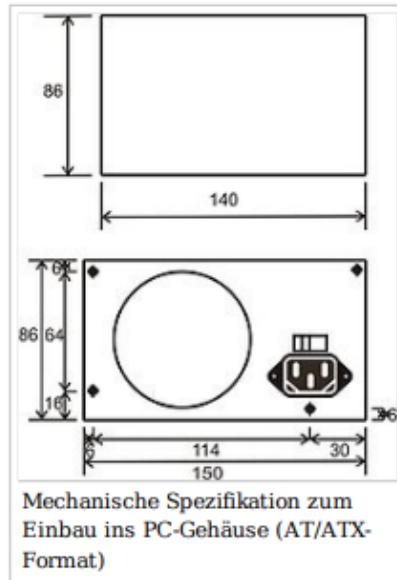
Für den Einbau in kleine Gehäuse sind die Maße wichtig.

	<b>Tiefe</b>	<b>Breite</b>	<b>Höhe</b>
ATX	140	150	84
ATX large	160	150	84

Zu beachten ist in obiger Tabelle jedoch, dass die Werte für die Tiefe nur Richtwerte darstellen, deren Einhaltung im ATX-Standard nicht verbindlich vorgeschrieben ist. Leistungsfähige Netzteile erreichen auch 19 cm Einbautiefe und mehr.

#### ATX-Netzteilstecker

Beim 20-poligen Stecker fehlen die letzten 4 Pins (11, 12, 23 und 24).



<b>Farbe</b>	<b>Signal</b>	<b>Pin</b>	<b>Pin<sup>1</sup></b>	<b>Signal</b>	<b>Farbe</b>
orange	+3,3 V	1	13 (11)	+3,3 V + Rückmeldung <sup>2</sup>	orange braun
orange	+3,3 V	2	14 (12)	-12 V	blau
schwarz	Masse	3	15 (13)	Masse	schwarz
rot rot	+5 V	4	16 (14)	Speisung ein	grün
schwarz	Masse	5	17 (15)	Masse	schwarz
rot	+5 V	6	18 (16)	Masse	schwarz
schwarz	Masse	7	19 (17)	Masse	schwarz
grau	Speisung gut	8	20 (18)	-5 V <sup>3</sup>	weiß
violet	+5 V Standby	9	21 (19)	+5 V	rot
gelb	+12 V	10	22 (20)	+5 V	rot
	+12 V	11	23	+5 V	
	+3,3 V	12	24	Masse	

<sup>1</sup> In Klammern die Anschlußzählung für den 20-Pin-Stecker



## 5. Mainboard/Motherboard

# Identifying Components of Motherboards

The spine of the computer is the *motherboard*, otherwise known as the system board and mainboard. This is the printed circuit board (PCB)—a conductive series of pathways laminated to a nonconductive substrate—that lines the bottom of the computer and is often of a uniform color, such as olive, brown, or blue. It is the most important component in the computer because it connects all the other components together. Figure 1.1 shows a typical PC system board, as seen from above. All other components are attached to this circuit board. On the system board, you will find the central processing unit (CPU), underlying circuitry, expansion slots, video components, random access memory (RAM) slots, and a variety of other chips. We will be discussing each of these components throughout this book.



## System Board Form Factors

System boards are classified by their form factor (design), such as ATX, micro ATX, and ITX. Exercise care and vigilance when acquiring a motherboard and case separately. Some cases are less accommodating than others and might not be physically compatible with the motherboard you choose.

## System Board Components

Now that you understand the basic types of motherboards and their form factors, it's time to look at the components found on the motherboard and their locations relative to each other. Many of the following components can be found on a typical motherboard:

- Chipsets
- Expansion slots and buses
- Memory slots and external cache
- CPUs and their sockets
- Power connectors
- Onboard disk drive connectors
- Keyboard connectors
- Integrated peripheral ports and headers
- BIOS/firmware
- CMOS battery
- Jumpers and DIP switches
- Front-panel connectors

In the following sections, you will learn about some of the most common components of a motherboard, what they do, and where they are located on the motherboard. We'll show what each component looks like so you can identify it on most any motherboard you run across. In the case of some components, this chapter provides only a brief introduction, with more detail to come in later chapters.

Before we can talk about specific components, however, you need to understand the concepts underlying serial and parallel connectivity, the two main categories of bus architecture.

## Bus Architecture

There has been a quiet revolution taking place in the computer industry for quite some time now. Unlike in the early days of personal computing, when parallel communication pathways (made up of multiple synchronized wires or traces) dominated single-file serial connections, this revolution has brought a shift toward serial communications. Once engineers created transmitters and receivers capable of sustaining data rates many times those of parallel connections, they found no need to tie these pathways together in a parallel circuit. The downside of parallel communications is the loss of circuit length and throughput—how far the signal can travel and the amount of data moved per unit of time, respectively—due to the careful synchronization of the separate lines, the speed of which must be controlled to limit skewing the arrival of the individual signals at the receiving end.

The only limitation of serial circuits is in the capability of the transceivers, which tends to grow over time at a refreshing rate due to technical advancements. Examples of specifications that have heralded the migration toward the dominance of serial communications

are Serial ATA (SATA), Universal Serial Bus (USB), IEEE 1394/FireWire, and Peripheral Component Interconnect Express (PCIe).

Parallel computer-system components work on the basis of a bus. A *bus*, in this sense, is a common collection of signal pathways over which related devices communicate within the computer system. Expansion buses of various architectures, such as PCI and AGP, incorporate slots at certain points in the bus to allow insertion of external devices, or adapters, into the bus, usually with no regard to which adapters are inserted into which slots; insertion is generally arbitrary. Other types of buses exist within the system to allow communication between the CPU and components with which data must be exchanged. Except for CPU slots and sockets and memory slots, there are no insertion points in such closed buses because no adapters exist for such an environment.

The term *bus* is also used in any parallel or bit-serial wiring implementation where multiple devices can be attached at the same time in parallel or in series (daisy-chained). Examples include Small Computer System Interface (SCSI), USB, and Ethernet.

## Chipsets

A *chipset* is a collection of chips or circuits that perform interface and peripheral functions for the processor. This collection of chips is usually the circuitry that provides interfaces for memory, expansion cards, and onboard peripherals and generally dictates how a motherboard will communicate with the installed peripherals.

Chipsets are usually given a name and model number by the original manufacturer. Typically, the manufacturer and model also tell you that your particular chipset has a certain set of features (for example, type of RAM supported, type and brand of onboard video, and so on).

Chipsets can be made up of one or several integrated circuit chips. Intel-based motherboards, for example, typically use two chips. To know for sure, you must check the manufacturer's documentation, especially because today's chipset chips are frequently obscured by cooling mechanisms, sometimes hindering visual brand and model identification.

The functions of chipsets can be divided into two major functional groups, called Northbridge and Southbridge. Let's take a brief look at these groups and the functions they perform.

## Northbridge

The *Northbridge* subset of a motherboard's chipset is the set of circuitry or chips that performs one very important function: management of high-speed peripheral communications. The Northbridge is responsible primarily for communications with integrated video using AGP and PCIe, for instance, and processor-to-memory communications. Therefore, it can be said that much of the true performance of a PC relies on the specifications of the Northbridge component and its communications capability with the peripherals it controls.



When we use the term *Northbridge*, we are referring to a functional subset of a motherboard's chipset. There isn't actually a Northbridge brand of chipset.

The communications between the CPU and memory occur over what is known as the *frontside bus (FSB)*, which is just a set of signal pathways connecting the CPU and main memory, for instance. The clock signal that drives the FSB is used to drive communications by certain other devices, such as AGP and PCIe slots, making them local-bus technologies. The *backside bus (BSB)*, if present, is a set of signal pathways between the CPU and Level 2 or 3 (external) cache memory. The BSB uses the same clock signal that drives the FSB. If no backside bus exists, cache is placed on the frontside bus with the CPU and main memory.

The Northbridge is directly connected to the Southbridge (discussed next). It controls the Southbridge and helps to manage the communications between the Southbridge and the rest of the computer.

## Southbridge

The *Southbridge* subset of the chipset is responsible for providing support to the onboard slower peripherals (PS/2, parallel ports, serial ports, Serial and Parallel ATA, and so on), managing their communications with the rest of the computer and the resources given to them. These components do not need to keep up with the external clock of the CPU and do not represent a bottleneck in the overall performance of the system. Any component that would impose such a restriction on the system should eventually be developed for FSB attachment.

In other words, if you're considering any component other than the CPU, memory and cache, AGP slots, or PCIe slots, the Southbridge is in charge. Most motherboards today have integrated PS/2, USB, LAN, analog and digital audio, and FireWire ports for the Southbridge to manage, for example, all of which are discussed in more detail later in this chapter. The Southbridge is also responsible for managing communications with the slower expansion buses, such as PCI, and legacy buses.

Figure 1.2 is a photo of the chipset of a motherboard, with the heat sink of the Northbridge, at the top left, connected to the heat-spreading cover of the Southbridge, at the bottom right.

Figure 1.3 shows a schematic of a typical motherboard chipset (both Northbridge and Southbridge) and the components they interface with. Notice which components interface with which parts of the chipset.

## Expansion Slots

The most visible parts of any motherboard are the *expansion slots*. These are small plastic slots, usually from 1 to 6 inches long and approximately  $\frac{1}{2}$  inch wide. As their name suggests, these slots are used to install various devices in the computer to expand its capabilities. Some expansion devices that might be installed in these slots include video, network, sound, and disk interface cards.

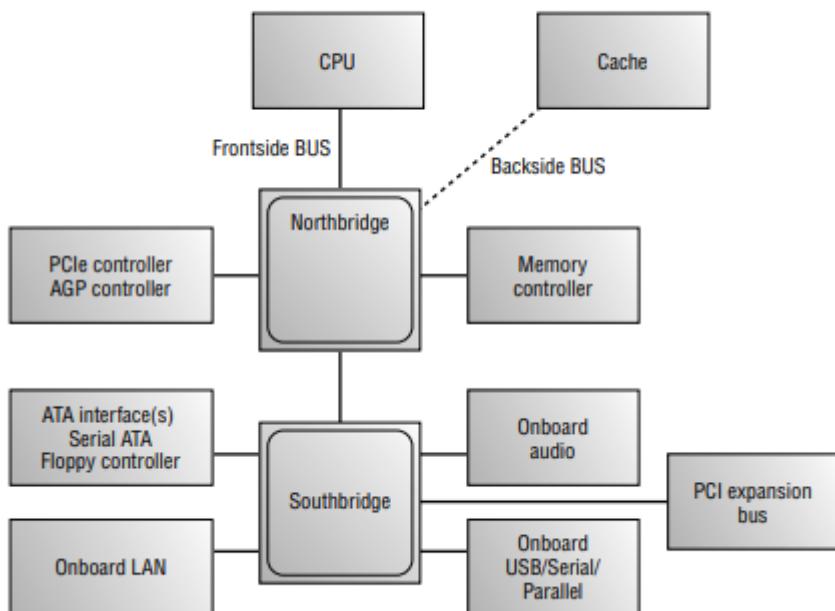
If you look at the motherboard in your computer, you will more than likely see one of the main types of expansion slots used in computers today:

- PCI
- AGP

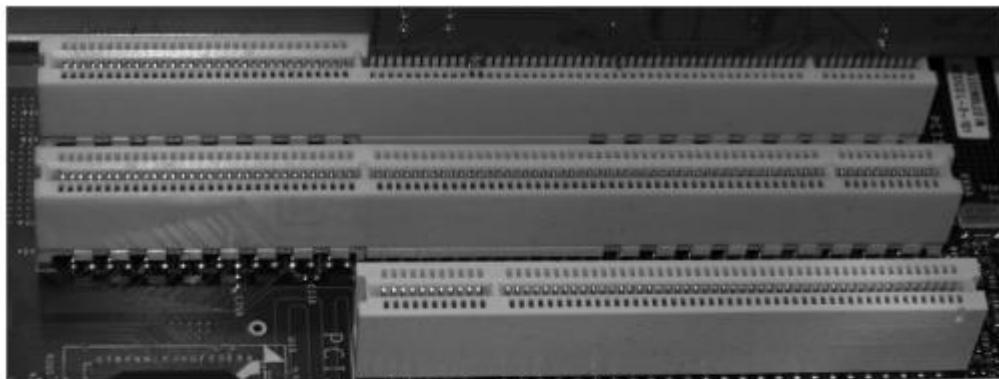
## Identifying Components of Motherboards

- PCIe
- PCI-X
- CNR

**FIGURE 1.3** A schematic of a typical motherboard chipset



**FIGURE 1.4** PCI expansion slots



#### Arriving at the Exact Answer

To get the math exactly right when dealing with frequencies and data rates ending in 33 and 66, you have to realize that every 33 has an associated one-third ( $\frac{1}{3}$ ), and every 66 has an associated two-thirds ( $\frac{2}{3}$ ). The extra quantities are left off of the final result but must be added back on to get the math exactly right. The good news is that omitting these small values from the equation still gets you close, and a bit of experience with the numbers leads to being able to make the connection on the fly.

#### PCI-X Expansion Slots

Visually indistinguishable from 64-bit PCI, because it uses the same slots, PCI-Extended (PCI-X) takes the 66MHz maximum frequency of PCI to new heights, to the most common, 133MHz, and the current maximum, 533MHz. With an 8-byte (64-bit) bus, this translates to maximum throughput of 4266MBps, roughly 4.3GBps. Additionally, PCI-X supports a 266MHz bus as well as the only frequency it shares with PCI, 66MHz, making PCI-X slots compatible with PCI adapters.

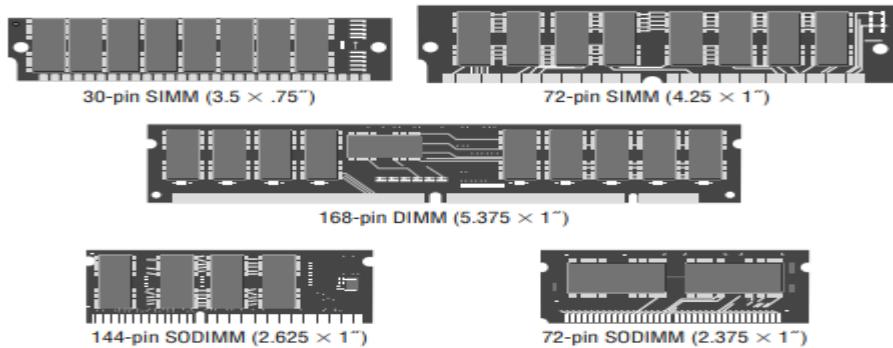
PCI-X is targeted at server platforms with its speed and support for hot-plugging but is still no match for the speeds available with PCIe, which all but obviates PCI-X today. PCI-X also suffers from the same shared-bus topology as PCI, resulting in all adapters falling back to the frequency of the slowest inserted adapter.

#### Memory Slots and Cache

Memory or random access memory (RAM) slots are the next most notable slots on a motherboard. These slots are for the modules that hold memory chips that make up primary memory that is used to store currently used data and instructions for the CPU. Many and varied types of memory are available for PCs today. In this chapter, you will become familiar with the appearance and specifications of the slots on the motherboard so you can identify them.

For the most part, PCs today use memory chips arranged on a small circuit board. A dual inline memory module (DIMM) is one type of circuit board. Today's DIMMs differ in the number of conductors, or pins, that each particular physical form factor uses. Some common examples include 168-, 184-, and 240-pin configurations. In addition, laptop memory comes in smaller form factors known as small outline DIMMs (SODIMMs) and MicroDIMMs. The single inline memory module (SIMM) is an older memory form factor that began the trend of placing memory chips on modules. More detail on memory packaging and the technologies that use them can be found later in this chapter in the section "Identifying Purposes and Characteristics of Memory." Figure 1.8 shows the form factors for some once popular memory modules. Notice how they basically look the same but the module sizes and keying notches are different.

**FIGURE 1.8** Different memory module form factors



## Central Processing Unit and Processor Socket

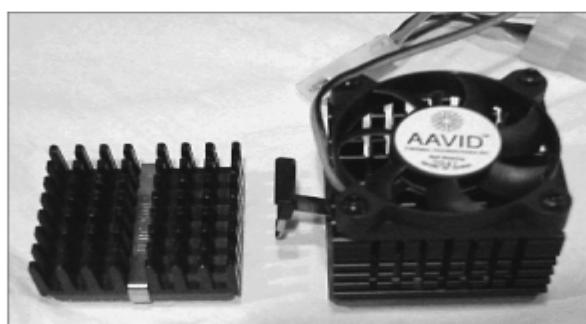
The “brain” of any computer is the central processing unit (*CPU*). There’s no computer without a CPU. There are many different types of processors for computers—so many, in fact, that you will learn about them later in this chapter in the section “Identifying Purposes and Characteristics of Processors.”

Typically, in today’s computers, the processor is the easiest component to identify on the motherboard. It is usually the component that has either a fan or a heat sink (usually both) attached to it (as shown in Figure 1.9). These devices are used to draw away and disperse the heat a processor generates. This is done because heat is the enemy of micro-electronics. Theoretically, a Pentium (or higher) processor generates enough heat that without the heat sink it would permanently damage itself and the motherboard in a matter of hours or even minutes.

CPU sockets are almost as varied as the processors they hold. Sockets are basically flat and have several columns and rows of holes or pins arranged in a square, as shown in Figure 1.10. The top socket is known as Socket A or Socket 462, made for earlier AMD processors such as the Athlon, and has holes to receive the pins on the CPU. This is known as a pin grid array (PGA) arrangement for a CPU socket. The holes and pins are in a row/

column orientation, an array of pins. The bottom socket is known as Socket T or Socket LGA 775, and there are spring-loaded pins in the socket and a grid of lands on the CPU. The land grid array (LGA) is a newer technology that places the delicate pins (yet more sturdy than those on chips) on the cheaper motherboard instead of on the more expensive CPU, opposite to the way the aging PGA does. The device with the pins has to be replaced if the pins become too damaged to function. The PGA and LGA are mentioned again later in this chapter in the section “Identifying Purposes and Characteristics of Processors.”

**FIGURE 1.9** Two heat sinks, one with a fan



## Firmware

*Firmware* is the name given to any software that is encoded in hardware, usually a read-only memory (ROM) chip, and can be run without extra instructions from the operating system. Most computers and large printers use firmware in some sense. The best example of firmware is a computer's Basic Input/Output System (BIOS) routine, which is burned in to a chip. Also, some expansion cards, such as SCSI cards and graphics adapters, use their own firmware utilities for setting up peripherals.

## BIOS and POST

Aside from the processor, the most important chip on the motherboard is the Basic Input/Output System (*BIOS*) chip, also referred to as the ROM BIOS chip. This special memory chip contains the BIOS system software that boots the system and allows the operating system to interact with certain hardware in the computer in lieu of requiring a more complex device driver to do so. The BIOS chip is easily identified: If you have a brand-name computer, this chip might have on it the name of the manufacturer and usually the word

*BIOS*. For clones, the chip usually has a sticker or printing on it from one of the major BIOS manufacturers (AMI, Phoenix/Award, Winbond, and so on). On later motherboards, the BIOS might be difficult to identify or even be integrated into the Southbridge, but the functionality remains, regardless of how it's implemented.

### BIOS

Figure 1.12 gives you an idea of what a modern BIOS might look like. Despite the 1998 copyright on the label, which refers only to the oldest code present on the chip, this particular chip can be found on motherboards produced as late as 2009. Notice also the Reset CMOS jumper at lower left and its configuration silkscreen at upper left. You might use this jumper to clear the CMOS memory, discussed next, when an unknown password, for example, is keeping you out of the BIOS configuration utility. The jumper in the photo is in the clear position, not the normal operating position. System boot-up is typically not possible in this state.

Some BIOS firmware can monitor the status of a contact on the motherboard for intrusion detection. If the feature in the BIOS is enabled and the sensor on the chassis is connected to the contact on the motherboard, the removal of the cover will be detected and logged by the BIOS. This can occur even if the system is off, thanks to the CMOS battery. At the next boot-up, the BIOS will notify you of the intrusion. No notification occurs over subsequent boots unless additional intrusion is detected.

### POST

A major function of the BIOS is to perform a process known as a power-on self-test (POST). POST is a series of system checks performed by the system BIOS and other high-end components, such as the SCSI BIOS and the video BIOS. Among other things, the POST routine verifies the integrity of the BIOS itself. It also verifies and confirms the size of primary memory. During POST, the BIOS also analyzes and catalogs other forms of hardware, such as buses and boot devices, as well as manages the passing of control to the specialized BIOS routines mentioned earlier. The BIOS is responsible for offering the user a key sequence to enter the configuration routine as POST is beginning. Finally, once POST has completed successfully, the BIOS selects the boot device highest in the configured boot order and executes the master boot record (MBR) or similar construct on that device so that the MBR can call its associated operating system's boot loader and continue booting up.

The POST process can end with a beep code or displayed code that indicates the issue discovered. Each BIOS publisher has its own series of codes that can be generated. Figure 1.13 shows a simplified POST display during the initial boot sequence of a computer.

## Flashing the System BIOS

If ever you find that a hardware upgrade to your system is not recognized, even after the latest and correct drivers have been installed, perhaps a BIOS upgrade, also known as *flashing the BIOS*, is in order. Only certain hardware benefits from a BIOS upgrade, such as drives and a change of CPU or RAM types. Very often, this hardware is recognized immediately by the BIOS and has no associated driver. So, if your system doesn't recognize the new device, and there's no driver to install, the BIOS is a logical target.

Be clear about the fact that we are not talking about entering the BIOS setup utility and making changes to settings and subsequently saving your changes before exiting and rebooting. What we are referring to here is a replacement of the burned-in code within the BIOS itself. You might even notice after the upgrade that the BIOS setup utility looks different or has different pages and entries than before.

On older systems and certain newer ones, a loss of power during the upgrade results in catastrophe. The system becomes inoperable until you replace the BIOS chip, if possible, or the motherboard itself. Most new systems, however, have a failsafe or two. This could be a portion of the BIOS that does not get flashed and has just enough code to boot the system and access the upgrade image. It could be a passive section to which the upgrade is installed and switched to only if the upgrade is successful. Sometimes this is controlled onscreen, and other times, there may be a mechanism, such as a jumper, involved in the recovery of the BIOS after a power event occurs. The safest bet is to make sure your laptop has plenty of battery power and is connected to AC power, or your desktop is connected to an uninterruptible power supply (UPS).

In all cases, regardless of BIOS maker, you should not consult BIOS companies—AMI, Award, Phoenix, etc. Instead, go back to the motherboard or system manufacturer; check its website, for example. These vendors have personalized their BIOS code after licensing it from the BIOS publisher. The vendor will give you access to the latest code as well as the appropriate flashing utility for its implementation.

## CMOS and CMOS Battery

Your PC has to keep certain settings when it's turned off and its power cord is unplugged:

- Date
- Time
- Hard drive/optical drive configuration
- Memory
- CPU settings, such as overclocking

## Front-Panel Connectors

From the time of the very first personal computer, there has been a minimum expectation as to the buttons and LEDs that should appear on the front of the case. Users expect a *power button* to use to turn the computer on (although these were on the side or back of very early PCs). The soft power feature available through the front power button, which is no more than a relay, allows access to multiple effects through the contact on the motherboard, based on how long the button is pressed. These effects can be changed through the BIOS or operating system. They expect a *power light*, often a green LED, to assure them that the button did its job. As time progressed, users were introduced to new things on the front panel of their computers. Each of these components depends on connectivity to the motherboard for its functionality. As a result, most motherboards have these standardized connections in common. The following list includes the majority of these landmarks:

- Power button
- Power light
- Reset button
- Drive activity lights
- Audio jacks
- USB ports