

Uthelm Bechtel

# OS X Lion

Der PowerFinder für Ihren Mac!



# Anatomie

*Technisches zur Hardware*

► 33  
Chipsatz

Zum Betrieb eines Computers werden verschiedene Komponenten gebraucht. Die wichtigsten Komponenten dabei sind der Prozessor und der Chipsatz.

## Der Prozessor

Der Prozessor ist die zentrale Recheneinheit eines Computers. Er führt Befehle aus, nimmt Berechnungen vor und verteilt Aufgaben an die verschiedenen Subsysteme.

### Prozessortechnik oder: Wie funktioniert ein Prozessor?

Ein Prozessor besteht aus verschiedenen funktionalen Einheiten, die erst das Abarbeiten von Programmen ermöglichen. Man kann hier zwischen dem Frontend und dem Backend unterscheiden. Das Frontend ist dafür zuständig, dem Programmablauf zu folgen und das Backend mit den Befehlen und Daten aus dem Arbeitsspeicher zu versorgen. Das Backend übernimmt die eigentliche Berechnung der Daten anhand der Befehle.

### Programmablauf

Alle Prozessoren arbeiten nach dem gleichen Ablauf – Fetch, Decode/Dispatch, Execute, Writeback: Ein Prozessorbefehl wird von einer bestimmten Speicheradresse angenommen, abgefertigt und verarbeitet (d.h. auf bestimmte Daten angewendet). Dann wird das Ergebnis zurückgeschrieben. Gleichzeitig wird der Programmzähler um Eins erhöht, und der nächste Befehl im Programmablauf in der nächst höheren Speicheradresse wird angenommen. Diese

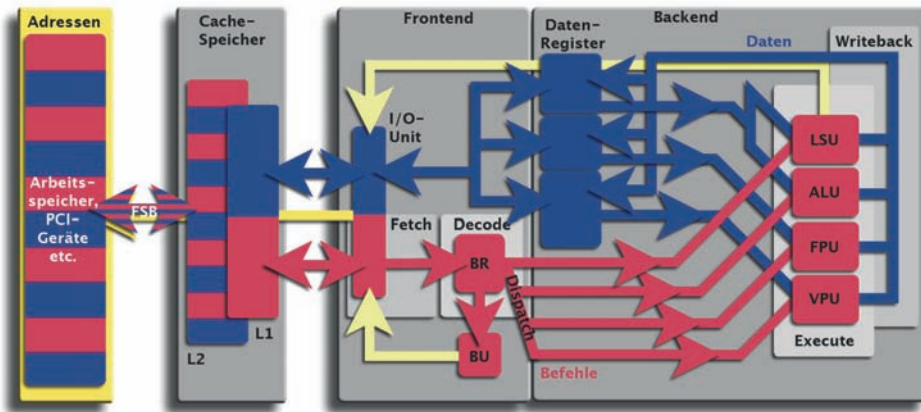
Sequenz durchläuft der Prozessor immer und immer wieder, von dem Zeitpunkt, an dem der Computer eingeschaltet wird, bis zum Ausschalten.

### Register

Register sind die Speichereinheiten innerhalb des Prozessors, der einzige Speicher, auf den der Prozessor direkten Zugriff hat. Befehle können nur auf Daten angewendet werden, die sich in Registern befinden. Jeder Befehl und die zu bearbeitenden Daten müssen zuerst in ein Register geladen werden. Um Daten aus dem Arbeitsspeicher in die Register zu laden, werden Lade-Befehle verwendet bzw. Speicher-Befehle, um Daten aus Registern in den Arbeitsspeicher zu schreiben (Load/Store-Befehle). Die unterschiedlichen Recheneinheiten für Festkomma-, Fließkomma- und Vektor-Berechnungen besitzen jeweils einen eigenen Satz an Registern. Speicheradressen werden in den Festkomma-Registern gespeichert. Zusätzlich gibt es Befehlsregister sowie Status- und Steuer-Register (z.B. für den Programmzähler).

### Programmverzweigungen

Normalerweise wird ein Befehl eines Programmes nach dem anderen abgearbeitet. In bestimmten Situationen kommt es jedoch zu Sprüngen oder Schleifen. Diese werden von der **Branch-Unit** bearbeitet. Ein Branch-Befehl setzt den Programmzähler unter bestimmten Umständen auf einen anderen Wert, sodass die folgenden Befehle von einer anderen Speicheradresse geladen werden.



Schematische Darstellung eines superskalaren RISC-Prozessors. In vier Pipeline-Stufen und mit vier parallelen Recheneinheiten werden die Daten (blau) mithilfe der Befehle (rot) verarbeitet. Die LSU berechnet die Adressen (gelb) für den Speicherzugriff.

BR = Befehls-Register    LSU = Load/Store Unit    FPU = Floating Point Unit  
 BU = Branch Unit      ALU = Arithmetic Logical Unit    VPU = Vector Processing Unit

### Aufbau von Prozessor-Befehlen

Ein Befehl besteht aus einer Reihe von Informationen. Die erste Information ist die Kennung, der »Opcode«. Dieser bestimmt, welche Berechnung ausgeführt werden soll (z.B. Addieren, Laden von Adresse etc.). Dahinter finden sich Registeradressen für die Quelle und das Ziel der Berechnung. Einige Befehle beinhalten auch eine so genannte »Immediate«, eine Zahl, die beim Übersetzen des Programms schon bekannt war.

### Techniken in der Mikroarchitektur

- **Pipelines:** Bei den allerersten Prozessoren wurde ein neuer Befehl erst angenommen, wenn der vorherige Befehl fertig bearbeitet war. Damit verbringt aber ein Teil des Prozessors sehr viel der Zeit mit Leerlauf. Daher ist man auf die Idee der Befehlspipeline gekommen. Der nächste Befehl wird schon angenommen, wenn der erste sich noch in Bearbeitung befindet. Damit sind immer alle Funktionseinheiten ausgelastet. Die kleinste Pipeline besteht aus vier Stufen (die klassische RISC-Pipeline – für die vier Phasen der Bearbeitung eines Befehls: Fetch, Decode/

Dispatch, Execute, Writeback). Dieser Ablauf kann wiederum in kleinere Einheiten unterteilt und die Pipeline so deutlich verlängert werden (z.B. bis zu 31 Stufen beim Intel Pentium 4, 16 Pipeline-Stufen bei den Intel-Core-Prozessoren). Längere Pipelines ermöglichen eine höhere Taktfrequenz, da die kleineren Bearbeitungsschritte weniger Zeit benötigen.

Allerdings funktioniert das nur gut, solange es keine Abhängigkeiten von den Ergebnissen vorheriger Befehle oder Verzweigungen gibt. Dann müssen schon in Bearbeitung befindliche Befehle wieder gelöscht werden, es entstehen Blasen in der Pipeline, so genannte »Bubbles«. Je länger die Pipeline, desto öfter treten diese auf und desto größer sind sie.

- **Sprungvorhersage:** Bei Sprung-Befehlen ist es aber möglich, bestimmte Befehle schon auf Verdacht auszuführen, um Verzögerungen durch Bubbles zu vermeiden. Beispielsweise ist die Wahrscheinlichkeit, dass sich eine Schleife wiederholt, ungleich höher, als dass die Schleife verlassen wird (z.B. wird beim Umwandeln eines Bildes mit

1000 Pixeln die Schleife 999 mal wiederholt, aber nur einmal verlassen). Moderne Prozessoren haben eine eigene Recheneinheit, die darauf spezialisiert ist zu berechnen, welcher Zweig am wahrscheinlichsten als nächstes genommen wird (den Branch-Predictor).

- **Superscalare Prozessoren:** Die ersten Prozessoren besaßen nur eine Recheneinheit im Backend, mit der jede Art von Daten berechnet wurden. Später, als die Zahl der Transistoren zunahm, wurden in den Prozessor zuerst zwei gleiche Recheneinheiten eingebaut, die die Daten parallel berechnen konnten. Noch später wurden die Recheneinheiten in Unter-Einheiten für unterschiedliche Arten von Berechnungen geteilt (z.B. eine Einheit für die einfachen Berechnungen Addition, Subtraktion und Multiplikation, plus eine für die komplizierte Division). Beim Intel Core z.B. arbeiten 13 unterschiedliche Recheneinheiten parallel. Moderne Prozessoren arbeiten »superscalar«. Damit kann der Prozessor bei einem Takt nicht nur einen Befehl, sondern mehrere ausführen. Ein superscalarer Prozessor braucht jedoch eine zusätzliche Steuereinheit, die dafür sorgt, dass die Ergebnisse in der richtigen Reihenfolge ausgegeben werden.
- **Weitere Techniken:** Im Laufe der weiteren Entwicklung wurden weitere Techniken erfunden, die die Leistungsfähigkeit des Prozessors steigern sollten, wie das Umsortieren der Befehle (Out of Order Execution, OoOE) vor dem Ausführen oder das Ausführen von Befehlen auf Verdacht (Speculative Execution), unsichtbare Register, in denen Daten zwischengespeichert werden, wenn ein Register belegt ist (Rename-Register) o.Ä. Allerdings kommen einige neuere Prozessor-Designs (wie der POWER6 von IBM oder der Intel Atom) von manchen dieser Techniken

wieder ab, da man – wie in den 80er-Jahren bei CISC – gerade feststellt, dass sie u.U. mehr Kosten als Nutzen bringen.

## Speicher

Wie oben beschrieben, hat der Prozessor nur auf die Register einen direkten Zugriff. Zugriff auf den Arbeitsspeicher ist nur über den Speicherbus anhand von Adressen möglich. Auch Ein- und Ausgabegeräte (die Festplatte, die Tastatur, die Grafikkarte o.Ä.) werden über Adressen behandelt. In Programmen finden sich jedoch keine absoluten Speicheradressen (diese ändern sich, z.B. je nachdem, wie groß der eingebaute Arbeitsspeicher ist), Programme arbeiten mit einem virtuellen Adressraum. Heutige Prozessoren besitzen eine eigene Funktionseinheit für das Berechnen von Speicheradressen, die **Load/Store-Unit**, sowie eine weitere Einheit für das Umrechnen der in den Programmen gespeicherten Adressen in absolute Adressen, die **Memory Management Unit**.

## Cache

Regulärer Arbeitsspeicher ist um ein Vielfaches langsamer als der Prozessor. Es gibt zwar schnelleren Speicher, doch der ist erheblich teurer (je schneller, desto teurer). Damit der Prozessor so wenig wie möglich auf den langsamen Arbeitsspeicher warten muss, wird schneller Speicher in mehreren Stufen als Cache für den langsamen Arbeitsspeicher verwendet (wobei der Arbeitsspeicher wiederum ein Cache für die Festplatte ist). Im Cache werden Kopien der aktuell benötigten Daten und Programmbefehle zwischengespeichert. Es gibt folgende Cache-Stufen: Am schnellsten, aber auch am kleinsten ist der Level-1-Cache (L1), er ist in den Prozessor integriert. Im L1 werden Befehle und Daten in getrennten Bereichen gespeichert. Auch der größere, etwas langsa-

mere L2 befindet sich bei heutigen Prozessoren zusammen mit dem Prozessor auf einem Chip. Bei manchen Prozessoren steht zusätzlich noch ein Level-3-Cache zur Verfügung. Die Verwaltung des L1- und L2- (und evtl. L3-) Cache übernimmt der Prozessor, sie ist von Prozessor zu Prozessor unterschiedlich. Mithilfe der Cache-Hierarchie wird der Speicherzugriff stark beschleunigt. Wie stark, wird nicht nur durch die Größe des Cache, sondern auch durch seine Organisation bestimmt.

### Recheneinheiten

Ein Prozessor besteht nicht nur aus einer Recheneinheit, sondern aus verschiedenen, die für unterschiedliche Rechen- und Steueraufgaben zuständig sind. Frühere Prozessoren besaßen als Recheneinheit nur die Einheit für Ganzzahl-Berechnungen (**ALU**, Arithmetic Logical Unit). Seit längerem enthalten Prozessoren mit der **FPU** zusätzlich eine Einheit für Fließkomma-Berechnungen in doppelter Genauigkeit (d.h. für 64-bit-Fließkomma-Werte). Alle Einheiten können auch mehrfach vorhanden sein.

### Vektor-Einheiten

Heutige Prozessoren enthalten zusätzlich zu den bekannten Recheneinheiten für Ganzzahl- und Fließkomma-Berechnungen eine weitere Recheneinheit, mit der multiple Daten bzw. Vektoren gleichzeitig mit einem Befehl bearbeitet werden können (SIMD – Single Instruction, Multiple Data).

### 64 bit

In den letzten Jahren wurden die meisten Prozessorarchitekturen auf 64 bit erweitert. Die Bezeichnung 64-bit-Prozessor bezieht sich auf die Breite der Register für die ALU. Mit den 64 bit breiten Registern kann der Prozessor eine ungleich größere Menge an (virtuellem und

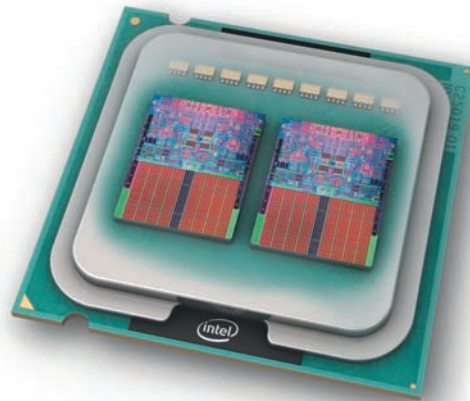


Abbildung: Intel

Blick ins Innere des Intel Core 2 Quad Prozessors. Dieser enthält zwei Prozessoren mit jeweils zwei Prozessorkernen. Erst der Core i7 ist ein richtiger Vierkern-Prozessor. Die großen Felder unten sind der L2-Cache-Speicher.

physikalischem) Speicher adressieren. Außerdem wird die Berechnung von ganzen Zahlen, die über 32 bit lang sind, beschleunigt.

### Multicore

Bei einem Multicore-Prozessor sind zwei oder mehr Prozessorkerne auf einen gemeinsamen Chip zusammengefasst. Wie bei einem Computer mit mehreren einzelnen Prozessoren kann dadurch die Rechenleistung gesteigert werden. Sie verdoppelt sich jedoch nicht, da zusätzlicher Verwaltungsaufwand entsteht. Mehrere Prozessorkerne auf einen Chip zu bringen, hat gegenüber einzelnen Prozessoren den Vorteil, dass die Prozessoren direkt, ohne den Umweg über die Northbridge, kommunizieren und auf einen gemeinsamen Cache-Speicher zugreifen können.

Der erste Dualcore-Prozessor wurde schon im Jahre 2001 eingeführt. Es war der POWER4 von IBM, aus dem später der PowerPC 970 (G5) – paradoxerweise zuerst als Singlecore-Prozessor – entwickelt wurde.

## Befehlsätze

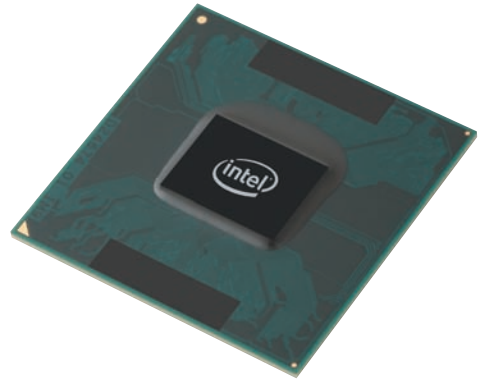
Jede Prozessorfamilie arbeitet mit einem eigenen Satz von Prozessorbefehlen (Instruction Set Architecture, ISA). Die **ISA** wurde erstmals in den 60er-Jahren im IBM System/360 eingeführt, um die Programmierung von der eigentlichen Hardware unabhängig zu machen. Die ISA liegt als Abstraktionsebene über der Mikroarchitektur des Prozessors. Vorher musste ein Programm immer für die Mikroarchitektur angepasst werden, selbst bei einer kleinen Änderung des Prozessors mussten die Programme neu geschrieben werden. Mit Einführung der ISA konnte vorhandene Software auch auf neuer Hardware mit der gleichen ISA verwendet werden. Die Befehle der ISA wurden dabei zuerst mithilfe der Microcode Engine verarbeitet. Diese ist eine Art Prozessor im Prozessor, der aus jedem Befehl eine Reihe von Befehlen erzeugt und diese dann dem Backend zuführt.

- 29 RISC Moderne ISAs sind jedoch so designt, dass der Prozessorkern direkt versorgt werden kann.

Mit der Wahl des Befehlsatzes werden in einem gewissen Maße auch die Eigenschaften des Prozessors bestimmt. Einige Hersteller bieten Prozessoren mit verschiedenen Befehlsätzen für unterschiedliche Einsatzzwecke an.

- ◀ 15 Geschichte, ARM Der Befehlsatz des **ARM**-Prozessors (Advanced RISC Machine) zeichnet sich durch besondere Effizienz aus. Prozessoren mit diesem Befehlsatz sind besonders sparsam. Wie das iPhone ist auch praktisch jedes andere Mobiltelefon mit einem ARM bestückt. Auch das iPad verwendet einen ARM-Prozessor. Außerdem werden ARM-Prozessoren im Embedded-Bereich eingesetzt, also in Computern, die in andere Geräte integriert sind. In diesem Bereich sind aber auch z.B. MIPS, 68k (der Befehlsatz des

- ◀ 13 Geschichte, PowerPC Prozessors der ersten Macs) und **PowerPC** (der später, bis zum Wechsel zu Intel, in den Macs verbaut war) vertreten. Letzterer besitzt einen besonders flexiblen Befehlsatz, der Sparsam-



Intel Core-2-Duo-Prozessor

Abbildung: Intel

keit, Leistungsfähigkeit und Zuverlässigkeit vereint. PowerPC-Prozessoren finden in den unterschiedlichsten Gebieten Einsatz. Die Prozessoren der aktuellen Spielekonsolen von Sony, Microsoft und Nintendo sind PowerPCs. PowerPCs steuern die Infrastruktur der Mobilfunknetze und auch im Bereich Automobil und Luft- und Raumfahrt ist der PowerPC stark vertreten. Die Palette geht bis zu den großen Server-Systemen. Hier ist auch der SPARC-Prozessor vertreten.

Der **x86** dagegen ist praktisch ein reiner Personalcomputer-Prozessor. Seine herausragende Eigenschaft ist die Kompatibilität zu Windows, dem meist verbreiteten PC-Betriebssystem. Damit konnte der x86 sich am Markt halten und zum Teil sogar die modernen Prozessoren im RISC-Design verdrängen. Intel versucht, mit dem x86 in die verschiedenen anderen Bereiche vorzustoßen. In Bereichen, in denen besondere Sparsamkeit, besondere Leistungsfähigkeit oder besondere Zuverlässigkeit gefordert ist (oder alles gleichzeitig), hat der x86 jedoch kaum eine Chance. Der x86 zeigt, dass mit der Lösung eines Problems ein neues geschaffen werden kann: Durch die definierte ISA wurde der Markterfolg eines Prozessors von der vorhandenen Software abhängig.



## Eigenschaften und Probleme der x86-ISA

Die Befehle des x86-Befehlssatzes haben eine extrem **unterschiedliche Länge** (zwischen 1 und 17 Bytes; im Durchschnitt 3,5, in tatsächlich verwendetem Programm-Code kann der Wert höher oder niedriger sein). Aufgrund der unterschiedlichen Länge der Befehle kann der Prozessor beim Annehmen der Befehle nicht automatisch erkennen, wann der nächste Befehl beginnt. Im Vergleich zu modernen Prozessoren, bei denen die Befehle alle gleich lang sind, ist das Dekodieren der Befehle beim x86 also ungleich aufwendiger. Heutige x86-Prozessoren nehmen die Befehle paketweise an. x86-Befehle können nicht einfach wie ein gleichmäßiger Datenstrom angenommen und aufgeteilt werden. Bevor diese weiterverarbeitet werden können, muss der Prozessor immer zuerst anhand des Inhalts feststellen, ob es sich dabei um einen oder mehrere Befehle handelt.

Häufig gebrauchte Befehle sind kurz, um Speicher zu sparen. Da die Zahl der kurzen Befehle rein mathematisch beschränkt ist, gibt es – um möglichst viele Befehle zu realisieren – zusätzlich seltener gebrauchte, längere Befehle. Zum Teil handelt es sich um zwei Versionen des gleichen Befehls: Ein kurzer Befehl, der automatisch auf ein bestimmtes Register angewendet wird, und ein langer, der auf jedes beliebige Register angewendet werden kann. Der x86-Befehlssatz enthält auch Befehle, die Speicherinhalte direkt berechnen (der Prozessor muss die Inhalte natürlich trotzdem in die Register laden, um sie berechnen zu können). Heutzutage ist die Einsparung an Speicherplatz, die sich theoretisch durch den Befehlssatz ergibt, irrelevant. Bei einem Mac-OS-X-Programm beträgt sie, wenn überhaupt, nur wenige Prozent.

Die **Zahl der Register** ist im Vergleich zu modernen Prozessoren sehr klein. Pro Funktionseinheit stehen im 32-bit-Modus 8

Register zur Verfügung. Die Menge an Registern reicht nicht aus, um die Grundparameter zu speichern, die für den Programmablauf benötigt werden. Diese müssen beim x86 im Speicher abgefragt werden, was viel Zeit kostet. Im 64-bit-Modus stehen 16 Register zur Verfügung. Das soll laut AMD in 90% der Fälle ausreichend sein.

Im Gegensatz zu modernen Prozessor-Designs kennt der x86 nur Befehle mit **zwei Operanden**, der Inhalt eines Registers (oder einer Speicheradresse) wird mit dem Inhalt eines anderen verändert (nach dem Schema  $A + B \rightarrow A$ ). Bei modernen Prozessoren dagegen haben die meisten Befehle drei Operanden ( $A + B \rightarrow D$ ) oder gar vier ( $A + B + C \rightarrow D$ ).

Der x86 bearbeitet Daten »**Little Endian**«. D.h. wenn das Wort »UNIX« ein 64-bit-Wert wäre, würde es in den Registern im 32-bit-Modus als »NU« und »XI« bzw. im 64-bit-Modus als »XINU« gespeichert. Little-Endian brachte früher den Vorteil, dass beim Umwandeln von Werten weniger Befehle gebraucht werden (die Speicheradresse ändert sich nicht). Moderne Prozessoren können meist im Little- oder Big-Endian-Modus betrieben werden. Sie werden aber vorzugsweise Big-Endian betrieben, da Netzwerk-Protokolle in der Regel Big-Endian-Zahlen verwenden. Auch das Mac-OS-X-Dateisystem HFS+ ist Big-Endian. Beim x86 müssen diese Werte entsprechend vom Kernel hin- und hergedreht werden.

Die **SSE** genannte Vektor-Einheit übernimmt beim x86 auch Fließkomma-Berechnungen, da die ursprüngliche x87-FPU problematisch zu programmieren ist. Dafür werden die Register der immer noch vorhandenen x87-FPU für bestimmte Vektorberechnungen gebraucht, die mit Intels erstem Versuch einer Multimedia-Erweiterung namens MMX eingeführt wurden. Da SSE den Beschränkungen des x86-Befehlssatzes unterliegt, ist die Vektor-Einheit





Umwandlungseinheiten in ein oder zwei interne Befehle umgewandelt, die komplexeren Befehle werden über ein Mikroprogramm in eine Sequenz von – teils Dutzenden – internen Befehlen umgerechnet.

### x86-Overhead

Sowohl für die Verarbeitung der x86-Befehle nach der ursprünglichen CISC-Methode, als auch für die Umwandlung der x86-Befehle werden viele Ressourcen verschwendet, für x86-Translation tendenziell sogar mehr, als für die Verarbeitung über Mikroprogrammierung. Viele Millionen Transistoren sind damit beschäftigt, den Prozessor mit Windows kompatibel zu machen und verbrauchen Strom, ohne zur Rechenleistung beizutragen. Da die verfügbaren Ressourcen sich mit jeder Prozessor-Generation vervielfachen, dachte man, man könnte diesen Overhead bald vernachlässigen. Heute aber wird mehr auf Stromsparen geachtet und Leistungssteigerungen bei Prozessoren werden nicht mehr durch Vergrößern eines einzelnen Prozessors und höhere Taktung erreicht, sondern durch Mehrprozessor-Technik. Mit der Vervielfachung kompletter Prozessorkerne steigt die Menge der von der x86-Übersetzungshardware verbrauchten Ressourcen nun doch wieder an.

Über die tatsächliche Menge der Transistoren, die für den x86-Overhead verbraucht werden, schweigen sich die Hersteller von x86-Prozessoren weitestgehend aus. Ein Pentium Pro, aus dem die aktuellen Core-Prozessoren entwickelt wurden, soll ca. 40% seiner Transistoren gebraucht haben, um x86-Befehle zu übersetzen. In heutigen x86-Prozessoren dürfte der x86-Overhead im Bereich von 10 Millionen Transistoren pro Prozessorkern liegen. (Diese Zahl entspricht etwa zwei bis drei kompletten ARM-Prozessorkernen, von denen einer im iPhone zum Einsatz kommt!)

### Vorteile durch x86?

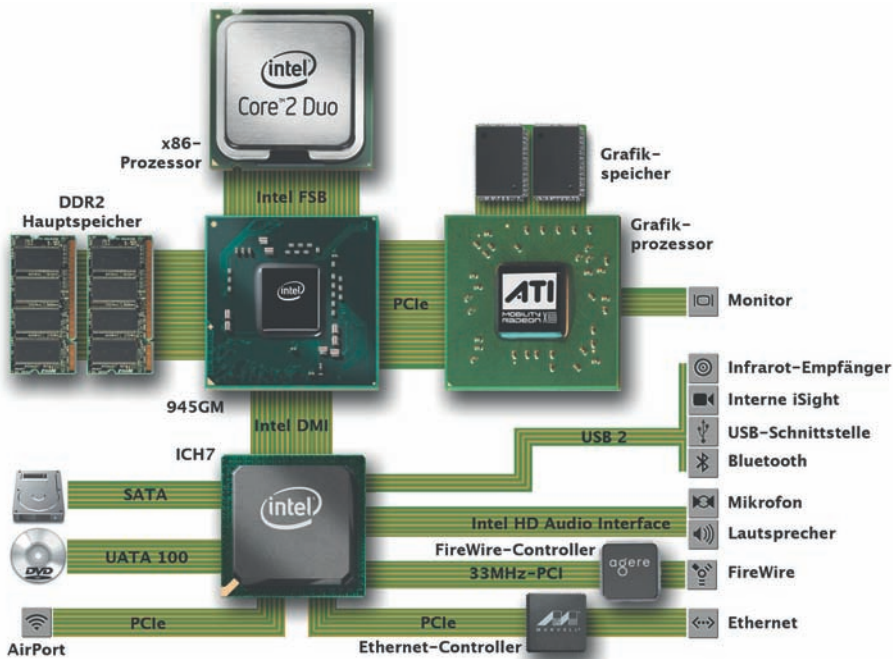
Intels Marketingabteilung behauptet häufig, dass die Verwendung von x86-Prozessoren Vorteile bringen würde, z.B. dass man so Zugriff auf die große Menge an Programmen hätte, die für x86 geschrieben sind. Diese Behauptungen sind allerdings völlig falsch. Für einen Mac, der unter Mac OS X betrieben wird (oder auch bei Verwendung von Linux oder einem anderen Betriebssystem) bringt es keinen Vorteil, den x86-Befehlssatz zu verwenden. Programme sind in der heutigen Zeit nicht mehr in Assembler für einen bestimmten Prozessor programmiert, sondern in Hochsprachen wie »C« oder »C++« bzw. »Objective C« für Cocoa. Diese Sprachen sind prozessorunabhängig. Beim Erzeugen eines Programms sorgt der Compiler dafür, die Hochsprache in die Prozessorsprache umzusetzen.

Außerdem besteht der größte Teil eines Programms (bei Photoshop z.B. sind es 80%) heutzutage aus Code für die Kommunikation mit dem Betriebssystem. Beim Portieren eines Programms von Windows nach Mac OS X muss dieser Teil aufwendig von Hand geändert werden, da Mac OS X andere Schnittstellen verwendet, als Windows. Nur der mathematische Teil (bei Photoshop, einem Programm, das verhältnismäßig viele Berechnungen durchführt, ungefähr 20%), kann eventuell auf einen Prozessor-Befehlssatz optimiert sein. Meist allerdings reicht es, den Code neu zu kompilieren. Die Verwendung von x86-Prozessoren bringt allenfalls den Vorteil (falls man das als solchen sieht), dass Windows – und damit Windows-Programme – auf dem Computer in voller Geschwindigkeit betrieben werden kann.

◀ 31  
x86-  
Translation

◀ 12  
Geschichte,  
Pentium Pro

▶ 92  
Praxis,  
Virtualisierung



Das Design des Mainboards im Intel-iMac entspricht dem des MacBook Pro bzw. ohne den Grafikprozessor und -speicher dem des Mac mini und MacBooks – und dem Design der meisten PC-Notebooks mit Intel-Prozessor.

## Der Chipsatz

Auf dem Mainboard des Macs befinden sich neben dem Prozessor weitere Komponenten, die zum Betrieb eines vollwertigen Computers gebraucht werden. Bei diesen Komponenten spricht man vom »Chipsatz«.

Ein Chipsatz besteht traditionell aus zwei Hauptkomponenten. Die Northbridge kontrolliert den Speicher und verbindet den Prozessor mit dem lokalen Bus. Außerdem ist die Grafikhardware über die Northbridge angebunden. Die Southbridge ist für die Verwaltung der I/O-Schnittstellen wie Festplatten-, USB-, FireWire- und die Netzwerk-Schnittstellen zuständig. North- und Southbridge sind direkt miteinander verbunden.

Der Trend geht allerdings zu einer immer weiteren Integration der Komponenten. Bei moderner Hardware, wie sie z.B. im iPad einge-

setzt wird, sind der Prozessor und alle Komponenten des Chipsatzes auf einen Chip vereint, den so genannten SoC (System on a Chip). Im Bereich der x86-PC-Hardware, speziell der von Intel, hängt die Entwicklung aber noch deutlich hinterher. Bei vielen Herstellern wurde schon länger der Speicher-Controller direkt in den Prozessor integriert. Diesem Trend folgte Intel etwas verspätet. Mittlerweile sind Speicher-Controller und Grafikhardware in den Prozessor integriert. Für die Schnittstellen wird aber noch immer ein zusätzlicher Chip benötigt.

Die Macs mit Intel-Prozessor verwenden die gleichen Chipsätze wie jeder andere PC. Der zuerst verwendete 945-Chipsatz von Intel war eigentlich für 32-bit-Prozessoren gedacht und kann daher – selbst, wenn mehr eingebaut ist – nur bis 3 GB Speicher verwalten. Erst

36ff ►  
Schnittstellen



- ◀ 27  
Prozessor,  
64 bit
- der nachfolgende 965-Chipsatz kann mehr Speicher verwalten. In die Intel-Southbridges ICH7 und ICH8 sind lediglich Festplatten- und USB-Schnittstellen integriert. Die Controller für die Netzwerk-Schnittstellen sind über PCIe angebunden, FireWire bei manchem Modell sogar nur über den langsamen PCI-Bus. In einigen Modellen enthält der Chipsatz einen in die Northbridge integrierten Grafikprozessor (GMA). Dieser Grafikprozessor besitzt im Gegensatz zu normalen Grafikprozessoren keine eigene Speicherschnittstelle, er greift auf einen reservierten Teil des Hauptspeichers zu. Außerdem fehlen Recheneinheiten für bestimmte Grafikberechnungen, die moderne Grafikhardware ausführen kann. Der Intel GMA suggeriert dem System jedoch, diese Funktionen zu beherrschen und gibt einen Teil der Berechnungen dann aber wieder an den Hauptprozessor zurück.
- ◀ 10  
Geschichte,  
x86, BIOS
- ◀ 10  
Geschichte,  
8086



Zwischenzeitlich verwendete Apple einen Chipsatz von **NVIDIA**, welcher sich durch eine größere Integration auszeichnet. In diesem Chip sind nicht nur – wie schon viele Jahre vorher bei dem Chipsatz, den Apple selber für den G4 entworfen hatte (siehe Grafik Seite 13) – die North- und die Southbridge sowie die Netzwerk-Schnittstelle zusammengefasst, sondern auch der Grafikprozessor integriert. Im Gegensatz zur integrierten Grafik in den Intel-Chipsätzen wird im NVIDIA-Chipsatz jedoch ein vollwertiger Grafikprozessor eingesetzt. Lediglich in Bezug auf die Speicheranbindung gibt es Nachteile gegenüber dedizierten Grafikprozessoren, da hier statt auf speziellen Grafikspeicher, auf den (etwas langsameren) Hauptspeicher zugegriffen wird.

Durch die Produktpolitik von Intel, mit der NVIDIA vom Markt der Chipsätze für Intel-Prozessoren verdrängt werden sollte, ist Apple bei den neuesten Modellen gezwungen, wieder auf einen Intel-Chipsatz mit entspre-

chend schlechterer integrierter Grafikhardware zu setzen.

Im Mac Pro werden Server-Prozessoren (Xeon) und -Chipsätze verbaut. Bei Intel ist nur mit diesen der Betrieb mehrerer Prozessoren möglich.

## Die Firmware

Damit ein Computer starten kann, braucht das Mainboard ein eigenes kleines Betriebssystem, das den Arbeitsspeicher initialisiert und das auf der Festplatte befindliche System starten kann.

### EFI

Beim x86-PC ist dafür normalerweise das BIOS zuständig, das mittlerweile schon 25 Jahre alt ist und im Laufe der Zeit um viele Funktionen erweitert wurde. Es setzt aber immer noch den »Real Mode« des x86-Prozessors voraus. Im Real Mode verhält der Prozessor sich wie ein 8086, einem Prozessor aus den 70er Jahren, mit entsprechend geringen Möglichkeiten bezüglich des adressierbaren Speichers und der Programmiersprache. Selbst für den neuesten x86-PC mit 64-bit-Prozessor muss ein x86-BIOS mit 1 MB Arbeitsspeicher auskommen und kann nur in Assembler – einer Sprache, die sich mehr an der Maschine, als am Menschen orientiert – programmiert werden.

In den Macs mit Intel-Prozessor übernimmt EFI (Extensible Firmware Interface) diese Aufgabe. EFI, welches von Intel ursprünglich als Firmware für den 64-bit-Server-Prozessor Itanium entwickelt wurde, ist beim x86 nur eine Abstraktionsschicht und ein Interface-Aufsatz für das BIOS. Das BIOS wird im Gegenzug von den vielen Funktionen, die im Laufe der Zeit hinzugekommen waren, befreit und wird wieder zu einer reinen Firmware. EFI ermöglicht für den Systemstart das Auffinden der Startdatei

auf der Festplatte und es können Firmware-Programme (hier »EFI-Module« genannt) ausgeführt werden.

Apple verwendet die fertige EFI-Implementation von Intel namens »Tianocore«, welche auch Intel für den größten Teil der Intel-Mainboards verwendet. Daher wird die EFI-Software für einen Mac unter Windows entwickelt.

Apple hat das Intel-EFI nur um eigene wenige Module erweitert – z.B. ein geändertes Modul mit grafischer Benutzeroberfläche für die Startvolume-Auswahl oder ein Modul für den FireWire-Target-Modus. Damit bietet EFI im Intel-Mac die gleichen Möglichkeiten, wie sie seit Jahren in Macs selbstverständlich waren, die auf der moderneren PowerPC-Architektur aufbauten.

Der Zugriff über eine Kommandozeilen-Oberfläche (EFI-Shell) ist bei Intel-Macs nicht ohne Weiteres möglich, da Apple das entsprechende Modul nicht integriert hat.

Eines der Module, die beim Start ausgeführt werden können, liefert das BIOS-Interface. Dieses ist zum Start von Windows notwendig. (Bei den meisten PCs anderer Hersteller mit EFI ist dieses Compatibility Support Module (CSM) das einzige EFI-Programm, das ausgeführt wird. Es liefert dort auch die im PC-Bereich bekannte BIOS-Bedienoberfläche.)

### Mac OS X ohne EFI?

Für den Betrieb von Mac OS X ist EFI nicht notwendig. EFI schreibt lediglich beim Systemstart bestimmte Werte in den Arbeitsspeicher. Eine angepasste Bootloader-Software, wie die im Internet frei verfügbare, nur wenige kB große Software »Chameleon« kann diese Aufgabe auf normalen PCs mit BIOS übernehmen. Ist diese Software installiert, läuft Mac OS X ohne Änderungen auf jedem beliebigen PC mit Intel-Prozessor und -Chipsatz. Für den Betrieb

auf PCs mit Chipsätzen oder Prozessoren von anderen Herstellern muss zusätzlich ein Kernel mit kleineren Änderungen verwendet werden. Auch der »Voodoo«-Kernel ist im Internet frei verfügbar.

### NVRAM

Die Einstellungen für das EFI werden im NVRAM gespeichert, einem nicht-flüchtigen (engl. non-volatile) Flash-Speicherbaustein auf dem Mainboard. Sie können auch im Terminal von Mac OS X mit dem Befehl »nvrnm -p« betrachtet werden.

### SMC

Macs haben auf dem Mainboard einen speziellen Baustein, der unter anderem die Batterie und die Lüfter überwacht. Der SMC (System Management Controller) ist – entsprechend Intels Vorgabe – ein komplett vom Betriebssystem unabhängiges System aus einem kleinen RISC-Prozessor mit integriertem Arbeits- und Flash-Speicher, in dem ein eigenes kleines Betriebssystem gespeichert ist.

Im Speicher des SMC ist außerdem ein Entschlüsselungscode – ein Limerick – für das Ausführen bestimmter essenzieller Mac-OS-X-Programme (Loginwindow, Finder) gespeichert, deren x86-Binärkode in verschlüsselter Form vorliegt, um den Betrieb von Mac OS X auf Nicht-Apple-PCs zu verhindern. Auch gegen diese Maßnahme hat die Hackintosh-Szene entsprechende Software entwickelt (z.B. die Kernel-Erweiterung »dsmos.kext« oder »fakesmc.kext«).



Das Logo der uefi.org, welche die Weiterentwicklung der EFI-Spezifikation übernommen hat.

### 86 ▶

Praxis,  
FireWire-Target-Modus

### 298 ▶

Praxis,  
SMC-Reset

### 265ff ▶

Praxis, Shell

► 274

Praxis,  
Festplatte  
installieren

## Interne Schnittstellen

### Serial ATA

Die Macs mit Intel-Architektur verwenden Serial ATA als Festplatten-Schnittstelle. An einen Anschluss kann nur eine Festplatte angeschlossen werden, sie braucht nicht konfiguriert zu werden.

Serial-ATA-Festplatten können auch extern in speziellen eSATA-Gehäusen angeschlossen werden. Da Apple jedoch keine externen Anschlüsse einbaut, müsste dafür eine eSATA-Schnittstelle nachgerüstet werden. Das ist nur beim Mac Pro mittels einer PCIe-Karte möglich.

### ATA / IDE

Die DVD-Laufwerke sind bei einigen Modellen an eine IDE-Schnittstelle angeschlossen. IDE wird auch als ATA bezeichnet, seit sich Serial ATA durchsetzt, IDE häufig auch als parallel ATA oder PATA.

### PCIe

Der Mac Pro verwendet als interne Schnittstellen PCIe – auch PCI Express genannt. PCIe arbeitet mit Punkt-zu-Punkt-Verbindungen, so genannten Lanes. Karten bzw. Steckplätze können 1, 2, 4, 8 oder 16 Lanes besitzen. PCI-, PCI-X- und AGP-Karten sind nicht kompatibel zu PCIe.



PCI-  
Express-x1-  
Karte (1 Lane)  
mit zwei  
eSATA-  
Schnittstellen

## Peripherie-Schnittstellen

Jeder Mac besitzt eine Reihe von Schnittstellen, über die er mit der Außenwelt kommuniziert. Einige dieser Schnittstellen sind bei jedem Mac vorhanden, andere nur bei bestimmten Modellen. Hier werden alle Schnittstellentypen und ihre Stecker beschrieben.

### USB



Am Universal Serial Bus (USB) werden die Eingabegeräte – Tastatur, Maus – sowie andere »langsame« Geräte wie Modem, Drucker, Scanner usw. angeschlossen. Im laufenden Betrieb können bis zu 127 Geräte angeschlossen werden. Mit der an allen Intel-Macs verwendeten Version 2.0 wurde USB stark beschleunigt, sodass auch schnelle Geräte wie Festplatten etc. angeschlossen werden können.

Trotz der Bezeichnung ist USB eigentlich gar kein Bus. USB-Geräte können nicht direkt hintereinander angeschlossen werden. Sie müssen immer an einem Hub (dem internen im Computer oder einen externen) angeschlossen werden. Die Geräte können nicht direkt miteinander kommunizieren. Die Kommunikation läuft immer über den Computer, selbst wenn die Geräte an den gleichen Hub angeschlossen sind. Trotz der nominell hohen Geschwindigkeit bleibt auch USB 2.0 (welches von Intel entwickelt wurde, um FireWire zu verdrängen) trotzdem nur eine aufgebohrte Tastaturschnittstelle, ohne eine für eine Hochgeschwindigkeitsschnittstelle geeignete Bus-Topologie. Wegen der – um wenige Cent – billigeren Hardware werden häufig Geräte statt mit einer FireWire- mit einer USB-2-Schnittstelle ausgestattet. Wo immer möglich, sollten Sie FireWire den Vorzug geben.

## FireWire



In den meisten neueren Macs ist mit FireWire (IEEE 1394) eine Hochgeschwindigkeitsschnittstelle zu finden. FireWire ist ein serieller Bus, an den im laufenden Betrieb bis zu 63 Geräte angeschlossen werden können. FireWire kann beispielsweise zum Anschluss von externen Festplatten und digitalen Camcordern verwendet werden. Am Mac werden die 6-poligen Stecker verwendet, die auch die Stromversorgung enthalten.

FireWire-Geräte sind meist mit zwei Schnittstellen – und einem sogenannten Repeater, der die Signale weiterleitet – ausgestattet, sodass mehrere Geräte hintereinander in einer Kette betrieben werden können. Im Gegensatz zu USB 2 können die Geräte dabei auch direkt miteinander kommunizieren, der Computer muss die Kommunikation lediglich anstoßen.

## FireWire 800



Mit FireWire 800 wurde die Geschwindigkeit der FireWire-Schnittstelle noch einmal verdoppelt. Dabei werden aber auch andere (9-polige) Stecker verwendet. FireWire-400-Geräte können jedoch mit einem Adapter angeschlossen werden.



USB-Anschlussstecker:  
B-Form (bei einigen Peripheriegeräten verwendet),  
A-Form (für Anschlussbuchse am Mac),  
Mini 5-polig (beispielsweise bei Digitalkameras)

## ExpressCard/34



Ältere MacBook Pro besitzen eine ExpressCard-Schnittstelle. ExpressCard kombiniert eine PCIe-Schnittstelle mit einer USB-2-Schnittstelle in einem Steckplatz. Im MacBook Pro können lediglich die schmalere ExpressCard/34-Karten verwendet werden, nicht die breiteren ExpressCard/54-Karten. ExpressCard ist nicht kompatibel zu PC-Card.

## Thunderbolt



Die neuesten Mac-Modelle besitzen eine Thunderbolt-Schnittstelle. Diese überträgt über die Schnittstelle des Mini-Displayport zusätzlich PCIe-Signale zum Anschluss verschiedener Peripherie-Geräte. Bis zu sechs Geräte können in einer Kette angeschlossen werden, darunter – je nach verwendeter Grafikhardware – ein oder zwei Displays.

## SD-Card



Einige neuere Intel-Macs haben eine eingebaute Schnittstelle für SD-Speicherkarten (Secure Digital Memory Card). In dieser können Speicherkarten im Standard-SD-Format gelesen werden und mit Adapter auch Mini-SD- und Micro-SD-Karten.



FireWire-Anschlussstecker:  
FireWire 400 (4-polig), FireWire 400 (6-polig) und  
FireWire 800

► 214  
Praxis,  
Datenaustausch mit  
dem Handy  
über Bluetooth

## Bluetooth



Bluetooth ist ein Kurzstreckenfunk, der eigentlich für die Kommunikation zwischen einem Mobiltelefon und Komponenten konzipiert ist. Hier können aber auch andere Peripheriegeräte angeschlossen werden. Die neuesten Macs haben Bluetooth eingebaut, ältere können mit einem Adapter nachgerüstet werden, der an die USB-Schnittstelle gesteckt wird. Bluetooth bietet eine Übertragungsgeschwindigkeit von maximal 723 kbit, mit der neueren Version 2.0 wurde die Geschwindigkeit verdreifacht.

► 363  
Referenz,  
Kontrollfeld  
Monitore

## Monitor



Monitore werden je nach Alter des Macs an verschiedene Schnittstellen angeschlossen. Sie sind jedoch alle elektrisch miteinander kompatibel. Monitore mit einer der anderen Schnittstellen können mittels Adaptern angeschlossen werden. Monitore melden sich über DDC am Mac an; so erkennt Mac OS X automatisch den Monitortyp.

An älteren Modellen wird die **DVI-Schnittstelle** verwendet. Hierbei wird die Konfiguration DVD-I verwendet, die neben den Pins für das digitale Monitorsignal auch Pins für analoges **SVGA**-Monitorsignal enthält. Für den Anschluss eines analogen Monitors liegt meist ein Adapter bei.

Einige Modelle besitzen eine **Mini-DVI-Schnittstelle**, an die DVI- oder VGA-Monitore mit einem Adapter angeschlossen werden können.

In neueren Modellen setzt Apple statt dessen auf den **Mini-DisplayPort**. Dieser überträgt kein analoges Signal mehr. Zum Anschluss von DVI- oder VGA-Monitoren oder Fernsehgeräten mit HDMI-Eingang werden entsprechende Adapter benötigt.

An den **HDMI**-Ausgang können moderne Fernsehgeräte ohne Adapter angeschlossen werden.



## iSight bzw. FaceTime-Kamera

iMacs und Notebooks sind mit einer integrierten Kamera ausgestattet, die bei Apple iSight, bzw. neuerdings FaceTime-Kamera genannt wird. Diese ist intern über USB angeschlossen. An Geräten, die keine eingebaute Kamera besitzen, können USB-Kameras anderer Hersteller angeschlossen werden. Für die Kamera stellt Mac OS X verschiedene Software-Funktionen zur Verfügung.



Monitor-Anschlussstecker: Mini DisplayPort, Mini-DVI, DVI, VGA und HDMI



## Ton



Schon der erste Mac hatte eine integrierte »Soundkarte«. Alle neueren Macs können mit ihrer Soundausstattung CD-Qualität produzieren. Alle Macs haben einen Line-Ausgang – manche Modelle auch zusätzlich einen Kopfhörerausgang und einen Line-Eingang – mit 3,5-mm-Miniklinken-Steckern. (Der Line-Eingang ist ein High-Level-Eingang, kein Mikrofoneingang. Die meisten Mikrofone brauchen hier einen Vorverstärker.)

Der Mac Pro verfügt außerdem über jeweils einen optischen digitalen Audioeingang und Audioausgang mit Toslink-Anschlüssen.

## Apple Remote (Infrarot-Schnittstelle)



Einige Intel-Macs sind mit einer Infrarot-Schnittstelle für die Fernbedienung »Apple Remote« ausgestattet. Diese ist nicht mit IrDA kompatibel.



Anschlüsse für Ton: Toslink und Miniklinke

## Netzwerk-Schnittstellen

### Ethernet



Schon seit vielen Jahren werden Macs mit einer Ethernet-Schnittstelle ausgerüstet. Für den Anschluss an das 10BaseT-, 100BaseT- bzw. 1000BaseT-Netz steht eine Buchse für RJ45-Stecker zur Verfügung.

### Wi-Fi



Macs haben mit der (je nach Modell optionalen oder serienmäßigen) Wi-Fi-Karte die Möglichkeit, Netzwerkverbindungen über Funk herzustellen. AirPort benutzt das Ethernet-Protokoll.

### FireWire



Bei Bedarf kann auch die FireWire-Schnittstelle zur Übertragung von Ethernet-Paketen verwendet werden.



Ethernet-Stecker

384f ▶

Referenz,  
Kontrollfeld  
Ton

333 ▶

Referenz,  
Audio-MIDI-  
Konfigura-  
tion

168ff ▶

Praxis,  
Netzwerk

365ff ▶

Referenz,  
Kontrollfeld  
Netzwerk

# Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als **persönliche Einzelplatz-Lizenz** zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschließlich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs und
- der Veröffentlichung

bedarf der **schriftlichen Genehmigung** des Verlags. Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: [info@pearson.de](mailto:info@pearson.de)

## Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. **Der Rechtsweg ist ausgeschlossen.**

## Hinweis

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website herunterladen:

**<http://ebooks.pearson.de>**