

7. Auflage

Dirk W. HOFFMANN

# GRUNDLAGEN DER TECHNISCHEN INFORMATIK



Im Internet: Lösungen zu den Übungsaufgaben

HANSER





### **Bleiben Sie auf dem Laufenden!**

Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter:

**[www.hanser-fachbuch.de/newsletter](http://www.hanser-fachbuch.de/newsletter)**





Dirk W. Hoffmann

# Grundlagen der Technischen Informatik

7., aktualisierte Auflage

HANSER

*Prof. Dr. Dirk W. Hoffmann*

Fakultät für Informatik und Wirtschaftsinformatik,  
Hochschule Karlsruhe – Technik und Wirtschaft

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt geprüft und getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Werk enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Weise aus der Benutzung dieser Informationen – oder Teilen davon – entsteht.

Ebenso wenig übernehmen Autor und Verlag die Gewähr dafür, dass die beschriebenen Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt also auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Werkes, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Einwilligung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung – mit Ausnahme der in den §§53, 54 URG genannten Sonderfälle –, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2023 Carl Hanser Verlag GmbH & Co. KG, München, <http://www.hanser-fachbuch.de>

Lektorat: Brigitte Bauer-Schiewek

Satz: Dirk W. Hoffmann

Titelmotiv: © shutterstock.com/Olga Zhurba und Gorodenkoff

Covergestaltung: Max Kostopoulos

Coverkonzept: Marc Müller-Bremer, [www.rebranding.de](http://www.rebranding.de), München

Druck und Bindung: Beltz Grafische Betriebe, Bad Langensalza

Printed in Germany

Print-ISBN: 978-3-446-47779-7

E-Book-ISBN: 978-3-446-47813-8



# Vorwort

---

Die Computertechnik hat in wenigen Jahrzehnten eine Entwicklung vollzogen, die in ihrer Geschwindigkeit und Intensität einzigartig ist. Setzten sich die ersten Computer noch aus vergleichsweise wenigen Schaltkreisen zusammen, so verrichten in jedem modernen Arbeitsplatzrechner, Tablet-PC oder Smartphone Abermillionen von Transistoren ihren Dienst und führen in jeder Sekunde Milliarden von Berechnungen aus. Doch so rasant die Entwicklung der letzten Jahrzehnte auch war: Vergleichen wir die Maschinen der Pionierzeit mit unseren modernen Rechenboliden, so lassen sich eine Reihe von Grundprinzipien identifizieren, die sich im Laufe der Zeit zwar weiterentwickelt, aber im Kern nicht verändert haben. Diese Grundprinzipien, zusammen mit ihren modernen Ausprägungen, formen das Gebiet der technischen Informatik und sind Gegenstand des vorliegenden Buchs.

Geschrieben habe ich das Buch für Bachelor-Studenten der Fachrichtungen Informatik, Elektrotechnik, Informationstechnik und verwandter Studiengänge. Inhaltlich habe ich mich dabei an den typischen Lehrinhalten orientiert, die im Grundstudium an Hochschulen und Universitäten vermittelt werden. Neben dem Grundlagenwissen aus den Gebieten der Halbleitertechnik, der Zahlendarstellung und der booleschen Algebra werden die Entwurfsprinzipien kombinatorischer und sequenzieller Hardware-Komponenten bis hin zur Beschreibung moderner Prozessor- und Speicherarchitekturen vermittelt. Damit spannt das Buch den Bogen von den mathematischen Grundlagen digitaler Schaltelemente bis hin zu den ausgefeilten Hardware-Optimierungen moderner Hochleistungscomputer.

Es ist mir ein besonderes Anliegen, den Stoff anwendungsorientiert und didaktisch ansprechend zu vermitteln. Damit das Buch sowohl vorlesungsbegleitend als auch zum Selbststudium eingesetzt werden kann, werden die Lehrinhalte aller Kapitel durch zahlreiche Übungsaufgaben komplementiert. Des Weiteren habe ich etliche Anwendungsbezüge mit aufgenommen, um eine enge Verzahnung zwischen Theorie und Praxis zu erreichen.

Seit dem Erscheinen der letzten Auflage habe ich wieder zahlreiche Zuschriften erhalten, über die ich mich sehr gefreut habe. Namentlich bedanken möchte ich mich bei Herrn Prof. Dr. Michael Wiehl für wertvolle Hinweise im Bereich der MOS-Schaltungstechnik. Inzwischen erscheinen die *Grundlagen der technischen Informatik* in der siebten Auflage, und ich bin weiterhin jedem aufmerksamen Leser für Hinweise zu Verbesserungsmöglichkeiten oder Fehlern dankbar.

---

## Symbolwegweiser



Definition



Satz, Lemma, Korollar



Leichte Übungsaufgabe



Mittelschwere Übungsaufgabe



Schwere Übungsaufgabe

---

## Lösungen zu den Übungsaufgaben

In wenigen Schritten erhalten Sie die Lösungen zu den Übungsaufgaben:

1. Gehen Sie auf die Seite [www.dirkwhoffmann.de/TI](http://www.dirkwhoffmann.de/TI)
2. Geben Sie den neben der Aufgabe abgedruckten Webcode ein
3. Die Musterlösung wird als PDF-Dokument angezeigt





# Inhaltsverzeichnis

---

<b>1</b>	<b>Einführung</b>	<b>11</b>
1.1	Was ist technische Informatik? . . . . .	11
1.2	Vom Abakus zum Supercomputer . . . . .	13
1.3	Wohin geht die Reise? . . . . .	30
<b>2</b>	<b>Halbleitertechnik</b>	<b>33</b>
2.1	Halbleiter . . . . .	34
2.1.1	Atommodell von Bohr . . . . .	34
2.1.2	Reine Halbleiter . . . . .	37
2.1.3	Dotierte Halbleiter . . . . .	39
2.2	Integrierte Schaltelemente . . . . .	41
2.2.1	Halbleiterdioden . . . . .	41
2.2.2	Bipolartransistoren . . . . .	42
2.2.3	Feldeffekttransistoren . . . . .	46
2.3	Chip-Fertigung . . . . .	51
2.3.1	Produktion integrierter Schaltkreise . . . . .	51
2.3.2	Integrationsdichte . . . . .	57
2.4	Übungsaufgaben . . . . .	58
<b>3</b>	<b>Zahlendarstellung und Codes</b>	<b>59</b>
3.1	Zahlensysteme . . . . .	60
3.2	Rechnerinterne Zahlenformate . . . . .	67
3.2.1	Darstellung natürlicher Zahlen . . . . .	67
3.2.2	Darstellung rationaler Zahlen . . . . .	73
3.3	Zahlencodes . . . . .	80
3.3.1	Tetraden-Codes . . . . .	80
3.3.2	Fehlererkennende Codes . . . . .	84
3.4	Übungsaufgaben . . . . .	86
<b>4</b>	<b>Boolesche Algebra</b>	<b>89</b>
4.1	Axiomatisierung nach Huntington . . . . .	90
4.1.1	Mengenalgebra . . . . .	91
4.1.2	Schaltalgebra . . . . .	93
4.2	Boolesche Ausdrücke und Aussagen . . . . .	95
4.2.1	Abgeleitete Operatoren . . . . .	97
4.2.2	Erfüllbarkeit und Äquivalenz . . . . .	100
4.2.3	Strukturelle Induktion . . . . .	102

4.2.4	Dualitätsprinzip . . . . .	105
4.3	Rechnen in booleschen Algebren . . . . .	109
4.3.1	Abgeleitete Umformungsregeln . . . . .	109
4.3.2	Vereinfachung boolescher Ausdrücke . . . . .	111
4.3.3	Vollständige Operatorensysteme . . . . .	117
4.4	Normalformdarstellungen . . . . .	119
4.4.1	Konjunktive und disjunktive Normalform . . . . .	119
4.4.2	Reed-Muller-Normalform . . . . .	122
4.4.3	Binäre Entscheidungsdiagramme . . . . .	125
4.5	Übungsaufgaben . . . . .	133
<b>5</b>	<b>Schaltnetze</b>	<b>139</b>
5.1	Grundlagen der Digitaltechnik . . . . .	140
5.1.1	Schaltkreisfamilien . . . . .	140
5.1.2	MOS-Schaltungstechnik . . . . .	145
5.1.3	Lastfaktoren . . . . .	155
5.2	Schaltungssynthese . . . . .	156
5.2.1	Zweistufige Schaltungssynthese . . . . .	157
5.2.2	BDD-basierte Schaltungssynthese . . . . .	158
5.2.3	FDD-basierte Schaltungssynthese . . . . .	159
5.3	Formelsynthese . . . . .	161
5.3.1	Funktionale Formelsynthese . . . . .	161
5.3.2	Relationale Formelsynthese . . . . .	163
5.3.3	Definitorische Formelsynthese . . . . .	164
5.4	Komplexitätsanalyse . . . . .	167
5.5	Zeitverhalten digitaler Schaltungen . . . . .	169
5.5.1	Signalausbreitung und -verzögerung . . . . .	169
5.5.2	Störimpulse . . . . .	171
5.6	Übungsaufgaben . . . . .	175
<b>6</b>	<b>Minimierung</b>	<b>181</b>
6.1	Minimierungsziele . . . . .	182
6.2	Karnaugh-Veitch-Diagramme . . . . .	186
6.2.1	Minimierung partiell definierter Funktionen . . . . .	190
6.2.2	Konstruktion Hazard-freier Schaltungen . . . . .	194
6.2.3	Minimierung mehrstelliger Funktionen . . . . .	196
6.3	Quine-McCluskey-Verfahren . . . . .	197
6.4	Übungsaufgaben . . . . .	201
<b>7</b>	<b>Standardschaltnetze</b>	<b>205</b>
7.1	Motivation . . . . .	206
7.2	Multiplexer und Demultiplexer . . . . .	206
7.3	Komparatoren . . . . .	213
7.4	Präfix-Logik . . . . .	215

7.5	Addierer . . . . .	218
7.5.1	Halb- und Volladdierer . . . . .	218
7.5.2	Carry-ripple-Addierer . . . . .	220
7.5.3	Carry-look-ahead-Addierer . . . . .	221
7.5.4	Conditional-Sum-Addierer . . . . .	224
7.5.5	Präfix-Addierer . . . . .	227
7.5.6	Carry-save-Addierer . . . . .	229
7.6	Inkrementierer . . . . .	232
7.7	Subtrahierer . . . . .	233
7.8	Multiplizierer . . . . .	234
7.8.1	Matrixmultiplizierer . . . . .	235
7.8.2	Carry-save-Multiplizierer . . . . .	238
7.8.3	Wallace-Tree-Multiplizierer . . . . .	241
7.8.4	Dadda-Tree-Multiplizierer . . . . .	246
7.9	Barrel-Shifter . . . . .	249
7.10	Arithmetisch-logische Einheit . . . . .	251
7.11	Programmierbare Logikbausteine . . . . .	253
7.12	Übungsaufgaben . . . . .	256
<b>8</b>	<b>Schaltwerke</b>	<b>265</b>
8.1	Digitale Speicherelemente . . . . .	266
8.1.1	Asynchrone Speicherelemente . . . . .	267
8.1.2	Taktzustandgesteuerte Speicherelemente . . . . .	271
8.1.3	Taktflankengesteuerte Speicherelemente . . . . .	274
8.1.4	Bevorrechtigte Eingänge . . . . .	281
8.1.5	CMOS-Implementierung . . . . .	282
8.2	Vom Flipflop zum Schaltwerk . . . . .	285
8.2.1	Endliche Automaten . . . . .	286
8.2.2	Schaltwerksynthese . . . . .	289
8.3	Übungsaufgaben . . . . .	293
<b>9</b>	<b>Standardschaltwerke</b>	<b>299</b>
9.1	Register . . . . .	300
9.1.1	Auffangregister . . . . .	300
9.1.2	Schieberegister . . . . .	302
9.1.3	Universalregister . . . . .	304
9.1.4	Akkumulatoren . . . . .	305
9.2	Zähler . . . . .	308
9.2.1	Synchrone Binärzähler . . . . .	309
9.2.2	Asynchrone Binärzähler . . . . .	313
9.2.3	Mischzähler . . . . .	314
9.2.4	Instruktionszähler . . . . .	316
9.3	Hauptspeicher . . . . .	318
9.3.1	SRAM-Speicher . . . . .	318

9.3.2	DRAM-Speicher . . . . .	320
9.3.3	Fehlererkennung und -korrektur . . . . .	327
9.4	Übungsaufgaben . . . . .	330
<b>10</b>	<b>Register-Transfer-Entwurf</b>	<b>335</b>
10.1	Entwurf komplexer Systeme . . . . .	336
10.1.1	Operationswerksynthese . . . . .	338
10.1.2	Steuerwerksynthese . . . . .	340
10.2	Mikroprogrammierung . . . . .	343
10.3	Übungsaufgaben . . . . .	349
<b>11</b>	<b>Mikroprozessortechnik</b>	<b>351</b>
11.1	Elemente eines Mikrorechners . . . . .	352
11.1.1	Von-Neumann-Architektur . . . . .	352
11.1.2	Aufbau der CPU . . . . .	356
11.2	Ein einfacher Modellprozessor . . . . .	360
11.3	Übungsaufgaben . . . . .	374
<b>12</b>	<b>Rechnerstrukturen</b>	<b>377</b>
12.1	Rechnerklassifikation nach Flynn . . . . .	378
12.2	Instruktionsarchitekturen . . . . .	379
12.2.1	CISC-Prozessoren . . . . .	380
12.2.2	RISC-Prozessoren . . . . .	385
12.3	Methoden zur Leistungssteigerung . . . . .	389
12.3.1	Pipelining . . . . .	389
12.3.2	Cache-Speicher . . . . .	394
12.4	Leistungsbewertung . . . . .	400
12.4.1	Maßzahlen zur Leistungsbewertung . . . . .	400
12.4.2	Benchmarks . . . . .	403
12.5	Übungsaufgaben . . . . .	406
<b>A</b>	<b>Notationsverzeichnis</b>	<b>411</b>
<b>B</b>	<b>Abkürzungsverzeichnis</b>	<b>413</b>
<b>C</b>	<b>Glossar</b>	<b>415</b>
	<b>Literaturverzeichnis</b>	<b>433</b>
	<b>Namensverzeichnis</b>	<b>437</b>
	<b>Sachwortverzeichnis</b>	<b>439</b>

# 1 Einführung

---

*„The first microprocessor only had 22 hundred transistors. We are looking at something a million times that complex in the next generations – a billion transistors. What that gives us in the way of flexibility to design products is phenomenal.“*

Gordon E. Moore, Intel Corporation

## 1.1 Was ist technische Informatik?

Blicken wir auf die Entwicklung der letzten hundert Jahre zurück, so hat keine andere technische Innovation unser Leben mehr verändert als die Erfindung des Computers, wie wir ihn heute kennen. Die Geschwindigkeit, mit der die digitale Revolution immer größere Bereiche unseres täglichen Lebens erobert und umgestaltet hat, ist nicht nur in der Retrospektive atemberaubend. Die Auswirkungen sind heute tief bis in unser kulturelles und gesellschaftliches Leben zu spüren. Ob wir wollen oder nicht: Wir stehen heute an der Schwelle des *ubiquitären Computerzeitalters* und haben sie in manchen Bereichen auch schon überschritten. Mit der Fortsetzung der kontinuierlich voranschreitenden Miniaturisierung und der zunehmenden Vernetzung verschiedenster Geräte ist der Computer von morgen allgegenwärtig und in vielen Fällen nicht einmal mehr als solcher zu erkennen.

Hand in Hand mit der sich rasant entwickelnden Computertechnik wuchs gleichermaßen die Bedeutung der Informatik, die sich in kürzester Zeit von einer Nischendisziplin zu einer eigenständigen Wissenschaft entwickeln konnte (vgl. Abbildung 1.1). Eine ihrer Kernsäulen ist die *technische Informatik*, die sich grob gesprochen mit dem *Entwurf, der logischen Struktur und der technischen Realisierung von Computer-Hardware* beschäftigt.

Ausgehend von der elementaren Hardware-Komponente des *Logikgatters* beschäftigt sich die technische Informatik mit der Konstruktion

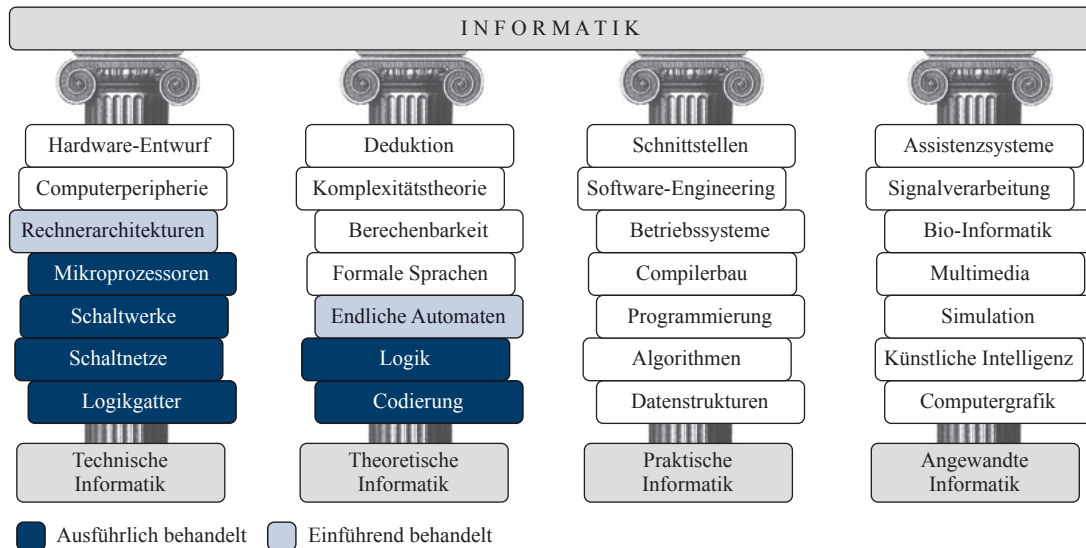


Abbildung 1.1: Die vier Säulen der Informatik

komplexer Digitalschaltungen. Hierzu gehören einfache *Schaltnetze* genauso wie komplexe, mit *Speicherelementen* angereicherte *Schaltwerke*. Durch das wechselseitige Zusammenspiel von Millionen von Schaltelementen sind wir heute in der Lage, Systeme mit einer Komplexität zu konstruieren, die noch vor nicht allzu langer Zeit als unmöglich erachtet wurde. Nichtsdestotrotz lässt sich selbst das komplexeste System stets auf die gleichen Grundprinzipien zurückführen. Die folgenden Kapitel werden diese in ein helleres Licht rücken und den Leser praxisnah in die technische Funktionsweise moderner Computersysteme einführen.

Die technische Informatik ist eng mit der theoretischen Informatik verzahnt. Viele der dort entwickelten Konzepte aus den Bereichen der Codierungstheorie, Logik und der endlichen Automaten dienen uns als das mathematische Fundament zur Beschreibung von Computer-Hardware. In entsprechender Weise werden wir uns auch in diesem Buch mit etlichen Teilaspekten dieser Disziplin beschäftigen. Doch bevor wir vollends in die Welt der Bits und Bytes eintauchen, wollen wir einen kurzen Streifzug durch die junge, aber bewegte Geschichte der Computertechnik wagen und uns mit der Frage beschäftigen, wohin die Reise in den nächsten Jahren führen wird.

## 1.2 Vom Abakus zum Supercomputer

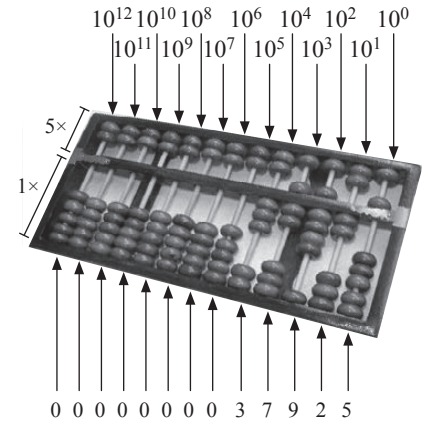
### Die ersten mechanischen Rechenhilfen

Wir beginnen unseren Streifzug durch die Geschichte im elften Jahrhundert vor Christus. Etwa zu dieser Zeit wird in China mit dem *Suan pan* die erste mechanische Rechenhilfe entwickelt – der sogenannte *Abakus*. Obwohl das auf den ersten Blick primitiv anmutende Rechenbrett nicht viel mit der heutigen Computertechnik verbindet, stellt der Abakus einen bedeutenden Schritt in Richtung des maschinellen Rechnens dar und ist mit seiner redundanten Zifferndarstellung ein willkommener Einstieg in die Thematik der Zahlensysteme.

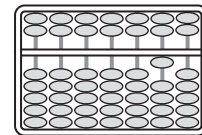
Abbildung 1.2 zeigt das Bild eines chinesischen Abakus, wie er noch heute auf vielen fernöstlichen Warenmärkten die uns vertraute elektronische Kasse ersetzt. Aufgebaut ist der *Suan pan* aus einer Reihe von Stäben, auf denen jeweils 7 bewegliche Kugeln in zwei unterschiedlichen Segmenten aufgefädelt sind. Das obere Segment – der *Himmel* – enthält jeweils 2 und das untere Segment – die *Erde* – die restlichen fünf Kugeln. Zur Zahlendarstellung verwendet der Abakus das uns geläufige arabische System. Jeder Stab repräsentiert eine einzelne Ziffer, deren Wert sich aus der Stellung und den Wertigkeiten der einzelnen Kugeln bestimmt. In die Berechnung des Ziffernwerts gehen ausschließlich diejenigen Kugeln ein, die nach *innen*, d. h. in Richtung der mittleren Querstrebe, geschoben wurden. Jede Kugel aus dem oberen Segment erhöht den Ziffernwerts dabei um 5 und jede Kugel aus dem unteren Segment um 1.

Abbildung 1.3 demonstriert die Darstellung der Zahl 10. Wie das Beispiel zeigt, ist die Zahlendarstellung im Gegensatz zum klassischen Dezimalsystem nicht eindeutig. Der Grund hierfür liegt in der Anzahl und der Wertigkeit der Kugeln, mit denen sich nicht nur die klassischen Dezimalziffern 0 bis 9, sondern auch die Werte 10 bis 15 darstellen lassen.

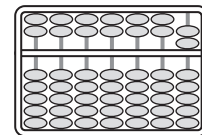
Der Aufbau des Abakus hat sich im Laufe der Zeit und durch den Einfluss verschiedener Kulturkreise in unterschiedliche Richtungen weiterentwickelt. So besitzt der japanische *Soroban* im Gegensatz zum chinesischen *Suan pan* nur noch 5 statt 7 Kugeln und der russische *Stschoty* kennt z. B. überhaupt keine Aufteilung in Himmel und Erde mehr.



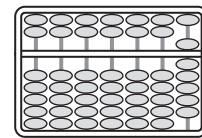
**Abbildung 1.2:** Der *Suan pan* (chinesischer Abakus)



$$10 = 1 \cdot 10$$



$$10 = 2 \cdot 5$$



$$10 = 1 \cdot 5 + 5 \cdot 1$$

**Abbildung 1.3:** Die Zahl 10 kann mit Hilfe des Abakus auf drei verschiedene Weisen dargestellt werden.

„Dasselbe, was Du auf rechnerischem Weg gemacht hast, habe ich kürzlich mechanisch versucht und eine aus 11 vollständigen und 6 verstümmelten Rädchen bestehende Maschine gebaut, welche gegebene Zahlen im Augenblick automatisch zusammenrechnet: addiert, subtrahiert, multipliziert und dividiert. Du würdest hell auflachen, wenn Du da wärest und sehen könntest, wie sie, so oft es über einen Zehner oder Hunderter weggeht, die Stellen zur Linken ganz von selbst erhöht oder ihnen beim Subtrahieren etwas wegnimmt.“



Wilhelm Schickard (1592 - 1635)

**Abbildung 1.4:** Schickard in einem Brief vom 20.9.1623 an Johannes Kepler



**Abbildung 1.5:** Die Schickard'sche Rechenuhr

## Die Schickard'sche Rechenuhr

Mit dem Abakus lassen sich Zahlen *darstellen* und mit etwas Übung selbst komplexe arithmetische Operationen durchführen. Gleichwohl erfolgen alle Berechnungen stets manuell. Weit mehr als tausend Jahre vergingen, bis der deutsche Astronom, Mathematiker und Theologe Wilhelm Schickard das erste mechanische Rechenwerk ersann, mit dessen Hilfe sich zwei Zahlen zum einen vollautomatisch addieren und subtrahieren und zum anderen halbautomatisch multiplizieren und dividieren ließen. Schickard, der am 22.4.1592 im schwäbischen Herrenberg geboren wurde und 1619 einem Ruf an die Universität Tübingen folgte, verband eine lebenslange Freundschaft mit dem kaiserlichen Mathematiker und Astronomen Johannes Kepler, der für seine umfangreichen Berechnungen zur Planetenbewegung bis dato auf Papier und Bleistift angewiesen war.

Die Schickard'sche Rechenuhr besteht aus drei Teilen, die übereinander angeordnet sind (siehe Abbildung 1.5). Der obere Teil entspricht dem *Multiplikationswerk*, der mittlere dem *Additionswerk* und der untere einem *Speicher*, der die Funktion einer Merkhilfe für Zwischenergebnisse übernimmt. Die Funktionsweise des Multiplikationswerks entspricht im Wesentlichen dem Prinzip der *Napierstäbchen*, die auf den schottischen Mathematiker Lord John Napier of Merchiston zurückgehen [32]. Das Multiplikationswerk und das Additionswerk verfügen über keinerlei mechanische Verbindung. Die verschiedenen bei der Produktberechnung entstehenden Teilsummen mussten deshalb manuell abgelesen und per Hand in das Addierwerk übertragen werden. Aus diesem Grund war die Multiplikation mit Hilfe der Schickard'sche Rechenuhr nur halbautomatisch möglich.

Von der Schickard'schen Rechenuhr wurden nur zwei Exemplare überhaupt gebaut, die in den Wirren des Dreißigjährigen Kriegs für immer verloren gingen. Anhand von Skizzen und Aufzeichnungen aus den Nachlässen von Schickard und Kepler konnte die Rechenuhr jedoch rekonstruiert und die Funktionstüchtigkeit in mehreren Nachbauten nachträglich unter Beweis gestellt werden.

Es war die Pest, die das Leben von Wilhelm Schickard und seiner Familie im sechzehnten Jahr des Dreißigjährigen Kriegs ein tragisches Ende nehmen ließ. Zuerst rafft der schwarze Tod im Jahre 1634 seine Frau und seine drei Töchter dahin. Ein Jahr später, am 24. Oktober 1635, stirbt auch Wilhelm Schickard – zwei Tage, bevor sein neunjähriger Sohn ebenfalls der Seuche erliegt.



## Die Rechenmaschinen des Charles Babbage

Weitere Meilensteine im Bereich des maschinellen Rechnens stellen die Rechenmaschinen des britischen Mathematikers und Ökonomen Charles Babbage dar, der am 26. Dezember 1791 in London das Licht der Welt erblickte [45]. Bereits mit 22 Jahren wird Babbage Mitglied in der Royal Society und nimmt 1823 unter Förderung der britischen Regierung die Arbeiten an der *Differenzenmaschine* auf. Im Gegensatz zur Schickard'schen Rechenuhr, die für die automatische bzw. halbautomatische Durchführung von primitiven arithmetischen Operationen konzipiert war, sollte die Differenzenmaschine in der Lage sein, ganze Wertetafeln komplexer Polynome selbstständig zu erzeugen.

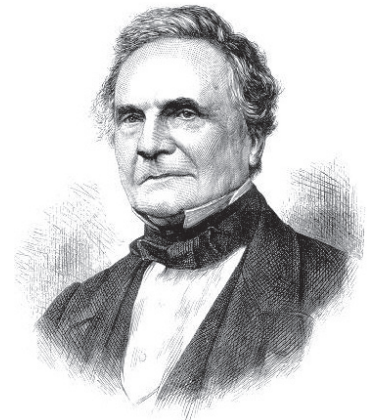
Das Prinzip der Maschine beruhte auf der Newton'schen *Differenzenmethode*, mit der solche Berechnungen auf trickreiche Weise unter der ausschließlichen Verwendung von Additionen und Subtraktionen durchgeführt werden können. Um zum Beispiel das Polynom

$$y = x^2 - 2x + 3$$

mit Hilfe der Differenzenmethode an den Stützstellen  $x = 0, 1, 2, 3, \dots$  auszuwerten, gehen wir in drei Schritten vor:

- Im ersten Schritt stellen wir die *initiale Differenzentabelle* auf. Für Polynome  $n$ -ten Grades tragen wir zunächst die ersten  $n + 1$  manuell berechneten Funktionswerte  $y(0)$  bis  $y(n)$  ein. Die nächsten Spalten werden durch sukzessive Differenzberechnung erzeugt. Dabei enthält die zweite Spalte die Differenzen der Elemente der ersten Spalte, die dritte Spalte die Differenzen der Elemente der zweiten Spalte und so fort. Insgesamt entsteht auf diese Weise die in Abbildung 1.7 (oben) dargestellte Tabelle.
- In der dritten Spalte erhalten wir als Differenz zweiter Ordnung den Wert 2. Egal um wie viele Stützstellen Sie die Tabelle nach unten ergänzen – die Elemente der dritten Spalte sind für unser Beispiel stets gleich. Aus funktionsanalytischer Sicht ist dieses Ergebnis auch nicht weiter verwunderlich, schließlich berechnen wir durch die  $n$ -malige Differenzenbildung nichts anderes als die diskrete Ableitung  $n$ -ter Ordnung und diese ist für Polynome  $n$ -ten Grades stets konstant. Unseren Überlegungen folgend können wir somit die dritte Spalte, wie in Abbildung 1.7 (Mitte) dargestellt, im zweiten Schritt beliebig nach unten erweitern.
- Im dritten Schritt füllen wir fehlende Einträge der Differenzentabelle von rechts nach links auf und können die gesuchten Funktionswerte

*„One evening I was sitting in the rooms of the Analytical Society, at Cambridge, my head leaning forward on the table in a kind of dreamy mood, with a table of logarithms laying open before me. Another member, coming into the room, and seeing me half asleep, called out, ‘Well Babbage, what are you dreaming about?’ to which I replied, ‘I am thinking that all these tables (pointing to the logarithms) might be calculated by machinery.’“ [95]*



Charles Babbage (1791 – 1871)

Abbildung 1.6: Charles Babbage

- Die initiale Differenzentabelle:

	$y(i)$	$\Delta(i)$	$\Delta\Delta(i)$
$i = 0$ :	3		
$i = 1$ :	2	1	
$i = 2$ :	3	-1	2

- Fortsetzen der letzte Spalte:

	$y(i)$	$\Delta(i)$	$\Delta\Delta(i)$
$i = 0$ :	3		
$i = 1$ :	2	1	
$i = 2$ :	3	-1	2
			2
			2
			2

- Ableiten weiterer Stützstellen:

	$y(i)$	$\Delta(i)$	$\Delta\Delta(i)$
$i = 0$ :	3		
$i = 1$ :	2	1	
$i = 2$ :	3	-1	2
$i = 3$ :	6	-3	2
$i = 4$ :	11	-5	2
$i = 5$ :	18	-7	2

**Abbildung 1.7:** Stützstellenberechnung mit Hilfe der Differenzenmethode am Beispiel des Polynoms  $y = x^2 - 2x + 3$

schließlich in der ersten Spalte ablesen. Die auf diese Weise vervollständigte Differenzentabelle ist für unser Beispiel in Abbildung 1.7 (unten) dargestellt.

Leider wird die Differenzenmaschine nie gebaut. Zum einen gestaltet sich die Fertigung vieler der geschätzten 25.000 Bauteile schwieriger als erwartet, zum anderen bringt Babbage stets neue Ideen und Verbesserungsvorschläge ein, die das Projekt zusätzlich verlangsamen. Als die Kosten schließlich vollends aus dem Ruder laufen, zieht sich die britische Regierung 1842 aus dem Projekt zurück.

In der Folgezeit ersann Babbage neben einer deutlich verbesserten *Differenzenmaschine 2* auch die *analytische Maschine*, die in ihrer Komplexität alle seine bisherigen Entwürfe nochmals weit übertrifft. Obwohl es sich bei der analytische Maschine um eine vollständig mechanische Konstruktion handelt, finden sich in deren Entwurf viele der klassischen Elemente wieder, die auch heute noch die Grundstrukturen moderner Computer prägen. So verfügt die analytische Maschine über einen Speicher und ein getrenntes Rechenwerk (*Mill*). Zur Ausgabe der Ergebnisse sah Babbage einen speziell konstruierten Drucker vor. Programmiert werden sollte die Maschine mit Hilfe von *Lochkarten* – ein Konzept, das zu dieser Zeit bereits bekannt war und erstmals 1805 von dem französischen Buchbinder Joseph-Marie Jacquard zur Automatisierung der Webtechnik eingesetzt wurde. Die analytische Maschine war so allgemein konzipiert, dass sie in der Lage gewesen wäre, selbst komplexe Kontrollstrukturen wie Schleifen, Unterprogrammaufrufe und bedingte Sprünge abzubilden. Leider war der Entwurf nicht nur konzeptionell, sondern auch in seiner Komplexität der damaligen Zeit weit voraus und die analytische Maschine wurde ebenfalls nie vollendet.

Trotz seiner unbestrittenen Erfolge in verschiedenen Gebieten der Wissenschaft war Babbage am Ende seines Lebens tief von dem Scheitern seiner drei Großprojekte gezeichnet. Mit der britischen Regierung, die jeder weiteren finanziellen Förderung eine Absage erteilte, ging er noch zu Lebzeiten hart ins Gericht. Am 18. Oktober 1871 starb Babbage in seinem Haus in London als enttäuschter und verbitterter Mann im Alter von 79 Jahren. 150 Jahre nach ihrer Erfindung schaffte zumindest die Differenzenmaschine 2 dann doch noch den Sprung vom Reißbrett in die Realität. Am Londoner Science Museum wurde die Maschine nach Originalplänen rekonstruiert. 1991 wurden die Arbeiten an der funktionsfähigen Maschine beendet – pünktlich zum 200. Geburtstag ihres Erfinders.

## Die elektrische Revolution

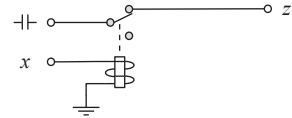
Die Verwendung elektromechanischer Bauteile stellt einen Quantensprung in der Computergeschichte dar. Mit der elektrischen Spannung als Informationsträger war auf einen Schlag die Konstruktion von Rechenmaschinen möglich, die in ihrer Komplexität und Zuverlässigkeit weit über das hinausgingen, was rein mechanisch arbeitende Apparaturen je zu leisten im Stande gewesen wären. Die ersten Rechenmaschinen dieser Art wurden Anfang der Vierzigerjahre gebaut und verwendeten zunächst *elektromechanische Relais* zur Daten- und Kontrollflusssteuerung. Das Relais, das schon kurz nach seiner Erfindung im Jahre 1835 durch Joseph Henry mit der Telegraphie das Kommunikationszeitalter einläutete, löste damit rund hundert Jahre später eine weitere technologische Revolution aus.

Zeitgleich mit dem Relais, das nur die zwei Zustände *offen* und *geschlossen* unterscheidet, hält das *Binärsystem* endgültig Einzug in die Computertechnik und bildet bis heute die Grundlage aller maßgebenden Computerarchitekturen. Abbildung 1.8 zeigt, wie sich die logischen Grundoperationen mit Hilfe konventioneller elektromechanischer Relais konstruieren lassen. In Kapitel 4 werden wir auf die verschiedenen Logikoperationen im Detail eingehen und in den darauf folgenden Kapiteln zeigen, wie sich unter alleiniger Verwendung der Elementarverknüpfungen NOT, AND und OR Hardware-Schaltungen mit nahezu beliebiger Komplexität in die Praxis umsetzen lassen.

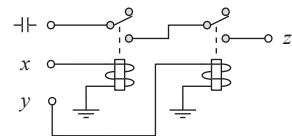
## Die legendäre Z3 des Konrad Zuse

Konrad Zuse konstruierte 1941 mit der Z3 den ersten voll funktionsfähigen Rechner dieser neuen Generation [96]. Im Gegensatz zu der noch weitgehend mechanisch arbeitenden Z1 und dem Versuchsmodell Z2, die aufgrund technischer Mängel beide nur unzuverlässig ihren Dienst verrichteten, verwendete Zuse in der Z3 ausschließlich elektromechanische Relais. Die Maschine war aus ca. 2000 Einzel-Relais aufgebaut und konnte mit einer Taktfrequenz von 5 bis 10 Hertz betrieben werden. Insgesamt kam die Z3 auf ein Gesamtgewicht von ca. 1000 kg und besaß eine Leistungsaufnahme von 4000 Watt. Nicht nur aus der Sicht der Ingenieurkunst war die Z3 bis dato einzigartig – sie war vor allem auch aus konzeptioneller Sicht ein Meisterwerk. Viele Konzepte, die sich heute in modernen Computerarchitekturen wiederfinden, waren bereits in der Z3 vorhanden, wenngleich auch in deutlich primitiverer Form:

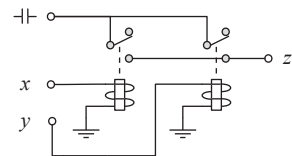
### ■ Die NOT-Verknüpfung:



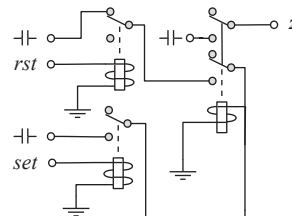
### ■ Die AND-Verknüpfung:



### ■ Die OR-Verknüpfung:



### ■ Der Relais-basierte Zustandsspeicher:



**Abbildung 1.8:** Die Basiskomponenten eines Relais-Rechners

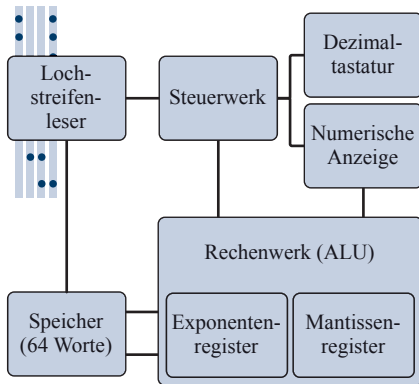


Abbildung 1.9: Blockschaltbild der Z3

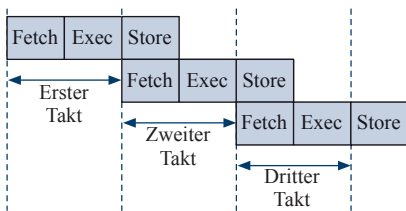


Abbildung 1.10: Die Pipeline-Architektur der Z3

Befehl	Bedeutung	Codierung
<b>Ein- und Ausgabe</b>		
<b>Lu</b>	Eingabe	01110000
<b>Ld</b>	Ausgabe	01111000
<b>Speichertransfer</b>		
<b>Pr z</b>	Lesen	11Adresse
<b>Ps z</b>	Schreiben	10Adresse
<b>Arithmetik</b>		
<b>Lm</b>	Multiplizieren	01001000
<b>Li</b>	Dividieren	01010000
<b>Lw</b>	Wurzelziehen	01011000
<b>Ls<sub>1</sub></b>	Addieren	01100000
<b>Ls<sub>2</sub></b>	Subtrahieren	01101000

Tabelle 1.1: Der Befehlssatz der Z3

■ Wie das Blockschaltbild in Abbildung 1.9 demonstriert, sind der Speicher und der Prozessor – bestehend aus Steuer- und Rechenwerk – klar voneinander getrennt. Diese Zweiteilung findet sich auch heute noch in den allermeisten Computerarchitekturen wieder.

■ Intern verwendet die Z3 das Gleitkommaformat zur Darstellung rationaler Zahlen. Über die numerische Tastatur eingegebene Operanden wurden automatisch in das interne Gleitkommaformat übersetzt und die Rechenergebnisse für die Anzeige ebenfalls vollautomatisch in das Dezimalsystem zurückkonvertiert. Damit war die Erfindung von Zuse auch hier anderen Maschinen weit voraus, die mit dem Festkommaformat eine wesentlich primitivere Art der Darstellung einsetzten. In Kapitel 3 werden wir uns intensiv mit den Unterschieden des Gleit- und Festkommaformats beschäftigen.

■ Das Prinzip der Mikroprogrammierung, das auch heute die Grundlage vieler moderner Mikroprozessoren bildet, findet sich bereits in der Z3 verwirklicht. Im Zentrum des Steuerwerks befindet sich hierzu der sogenannte Mikrosequenzer, mit dessen Hilfe sich komplexe Rechenoperationen in mehrere elementare Berechnungsschritte zerlegen und nacheinander ausführen ließen. Im Zusammenhang mit der CISC- und RISC-Architektur werden wir in Kapitel 12 auf dieses Prinzip der Ablaufsteuerung zurückkommen.

■ Zur Geschwindigkeitssteigerung führt die Z3 den Befehlsstrom überlappend aus, d. h. während des Zurückschreibens des Ergebnisses ist der Rechner in der Lage, bereits die nächste Instruktion einzulesen. Wie in Abbildung 1.10 gezeigt, folgte die Z3 damit einer dreistufigen Pipeline-Architektur, die mit einem einstufigen Versatz ausgeführt wurde. In Kapitel 12 werden wir uns auch mit diesem Architekturprinzip genauer beschäftigen.

■ Auch das verwendete Rechenwerk musste zu seiner Zeit keinen Vergleich scheuen. So setzte die Z3 zur Beschleunigung der Addition die Carry-look-ahead-Technik ein, die auch heute noch die Grundlage vieler Addierwerke bildet. In Kapitel 7 werden wir dieses Additionsprinzip im Detail kennen lernen.

Programmiert wurde die Z3 mit Hilfe eines 8-spurigen Lochstreifens. Jede Zeile codiert einen einzigen Befehl, zusammen mit den dazugehörigen Operanden. Wie die Befehlsübersicht in Tabelle 1.1 zeigt, lassen sich die insgesamt neun Befehle in die Kategorien Ein- und Ausgabe, Speichertransfer und Arithmetik unterteilen.

Mit Hilfe der Befehle **Lu** bzw. **Ld** wird eine Dezimalzahl über die numerische Tastatur eingelesen bzw. auf der numerischen Anzeige ausgegeben. Beide Befehle stoppen die Ausführung der Maschine, bis der Benutzer manuell die Fortsetzung veranlasst. Die Kommunikation mit dem Hauptspeicher wird über die Befehle **Pr** bzw. **Ps** gesteuert. Die anzusprechende Speicheradresse wird hierzu in den 8-Bit-Opcode der Befehle hineincodiert. Mit den 6 zur Verfügung stehenden Adressbits lassen sich insgesamt  $2^6 = 64$  Speicherworte adressieren. Die restlichen Befehle dienen zur Durchführung der vier arithmetischen Grundrechenarten sowie der Berechnung der Quadratwurzel.

Betrachten wir den Befehlssatz der Z3 genauer, so fällt auf, dass mit den vorhandenen Befehlen weder eine Verzweigung noch eine Schleife programmiert werden kann. Der Programmierung sind hierdurch enge Grenzen gesetzt und die Z3 stellt damit keinen Universalrechner im eigentlichen Sinne dar.

## Die Harvard Mark I

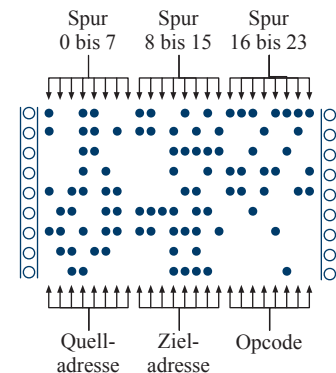
Fast zeitgleich mit dem Bau der Z3 wurde an den IBM Laboratories in New York unter der Regie von Howard H. Aiken der größte jemals fertiggestellte elektromechanische Rechner gebaut – der *Automatic Sequence Controlled Calculator* (ASCC) [17]. In einer feierlichen Zeremonie wurde die gigantische Maschine am 7. August 1944 an die Harvard University übergeben und offiziell in Betrieb genommen. Die Komplexität der Konstruktion, die fortan den Namen *Harvard Mark I* trug, lässt sich anhand weniger Kenndaten bereits erahnen. So bestand die gesamte Apparatur aus ca. 765.000 Einzelkomponenten, die über etliche Kilometer Kabel miteinander verbunden waren. Insgesamt füllte die Konstruktion mit 6 Metern Länge und über 2 Metern Höhe eine Werkhalle vollständig aus.

Die Harvard Mark I enthielt 72 *Akkumulatoren*, die jeder für sich eine dezimalcodierte Zahl mit einer Genauigkeit von 23 Ziffern speichern konnten. Neben einem weiteren Kurzzeit- sowie einem Konstantenspeicher verfügte der Rechner über separate Multiplikations- und Divisions-einheiten. Gesteuert wurde die Harvard Mark I über 24-spurige Lochstreifen, deren allgemeiner Aufbau in Abbildung 1.12 dargestellt ist. Jede Zeile codiert einen einzelnen Steuerbefehl, der sich aus drei Komponenten zusammensetzt. So definieren die Einstanzungen der ersten 8 Spuren die Quelladresse und die Einstanzungen der mittleren 8 Spuren die Zieladresse einer Operation. Die restlichen 8 Spuren definieren den Opcode des auszuführenden Befehls. Die Harvard Mark I konnte die so

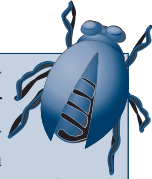


Konrad Zuse (1910 – 1995)

**Abbildung 1.11:** Bronze-Statue in Hünfeld zum Gedenken an Konrad Zuse



**Abbildung 1.12:** Aufbau des Lochstreifens zur Steuerung der Harvard Mark I



Dass wir heute im Software- und Hardware-Bereich Fehler gemeinhin als *Bugs* bezeichnen, wird in vielen Quellen in Zusammenhang mit der Harvard Mark I gebracht. Gerüchten zufolge geriet eine Motte in eines der zahlreichen elektrischen Relais des Rechners und brachte auf diese Weise die gesamte Maschine zum Erliegen. Die Motte wurde entfernt und der Begriff *Debugging* war geboren. In der Tat gehört die Geschichte zu den klassischen Mythen des Computerzeitalters. Die besagte Motte gab es wirklich, allerdings stammte sie nicht aus der Harvard Mark I, sondern von ihrem Nachfolger, der Harvard Mark II. Der Begriff des *Debuggings* wurde jedoch bereits viel früher geprägt, wie z. B. der folgende Ausschnitt aus einem Brief von Thomas Edison vom 18. November 1878 an Theodore Puskas verrät:

*„It has been just so in all my inventions. The first step is an intuition – and comes with a burst, then difficulties arise. This thing gives out and then that – ‘Bugs’ – as such little faults and difficulties are called – show themselves and months of anxious watching, study and labor are requisite before commercial success – or failure – is certainly reached“* [47]

Nichtsdestotrotz war die Motte der Harvard Mark II wahrscheinlich der erste richtige *Bug* der Computergeschichte. Howard Aiken selbst bezeichnete sie als den „*first actual case of bug being found*“.

codierten Befehle ausschließlich sequenziell abarbeiten – Befehle für Schleifen, Verzweigungen und Unterprogrammaufrufe sah die Rechnerarchitektur noch nicht vor. Damit ließen sich Endlosschleifen nur auf physikalischem Wege durch das Zusammenfügen des Lochstreifens zu einem endlosen Band realisieren.

Das bereits in der Harvard Mark I umgesetzte Konzept, Code und Daten in getrennten Speichern vorzuhalten, hat im Bereich moderner Computerarchitekturen in den letzten Jahren eine wahre Renaissance erlebt. So arbeiten z. B. moderne digitale Signalprozessoren fast ausschließlich nach diesem Prinzip, da sich im Vergleich zur klassischen Von-Neumann-Architektur deutlich höhere Übertragungsraten zwischen der Zentraleinheit und dem Speicher erreichen lassen. In Anlehnung an ihre historischen Wurzeln sprechen wir in diesem Zusammenhang heute immer noch von der *Harvard-Architektur*.

Die Harvard Mark I wurde erst 1959 außer Betrieb gestellt und anschließend zerlegt. Einige Komponenten sind heute noch im *Cabot Science Scenter* der Harvard University zu sehen, der Rest befindet sich im Besitz der IBM Laboratories in New York und des Smithsonian-Instituts in Washington, D.C.

## Die ENIAC

Die Frage, ob wir in der Z3 und der in etwa zeitgleich entwickelten Harvard Mark I die ersten richtigen *Computer* vor uns sehen dürfen, ist immer wieder Gegenstand hitziger Diskussionen und in Fachkreisen umstritten. Einige Experten sehen in der freien Programmierbarkeit und der konzeptionellen Untergliederung der Maschinen in Speicher, Rechenwerk und Steuerwerk die wesentlichen Grundzüge eines Computers verwirklicht. Andere Fachleute sind der Meinung, dass die *Universalität* des zu Grunde liegenden Berechnungsmodells das wesentliche Charakteristikum des Computers darstellt. Dieser Interpretation folgend, gelten sowohl die Z3 als auch die Harvard Mark I zwar als die ersten universellen Rechenmaschinen, nicht jedoch als Computer im modernen Sinne.

Den Bau der ersten voll funktionsfähigen Rechenmaschine, die nahezu allen Definitionen des modernen Computer-Begriffs standhält und daher von vielen Experten als der erste wirkliche Computer der Welt angesehen wird, datieren wir auf das Jahr 1946 – das Jahr, in dem die ENIAC<sup>1</sup> der Öffentlichkeit vorgestellt wurde [36, 86]. Der Rechnerko-

<sup>1</sup>Electronic Numerical Integrator And Computer

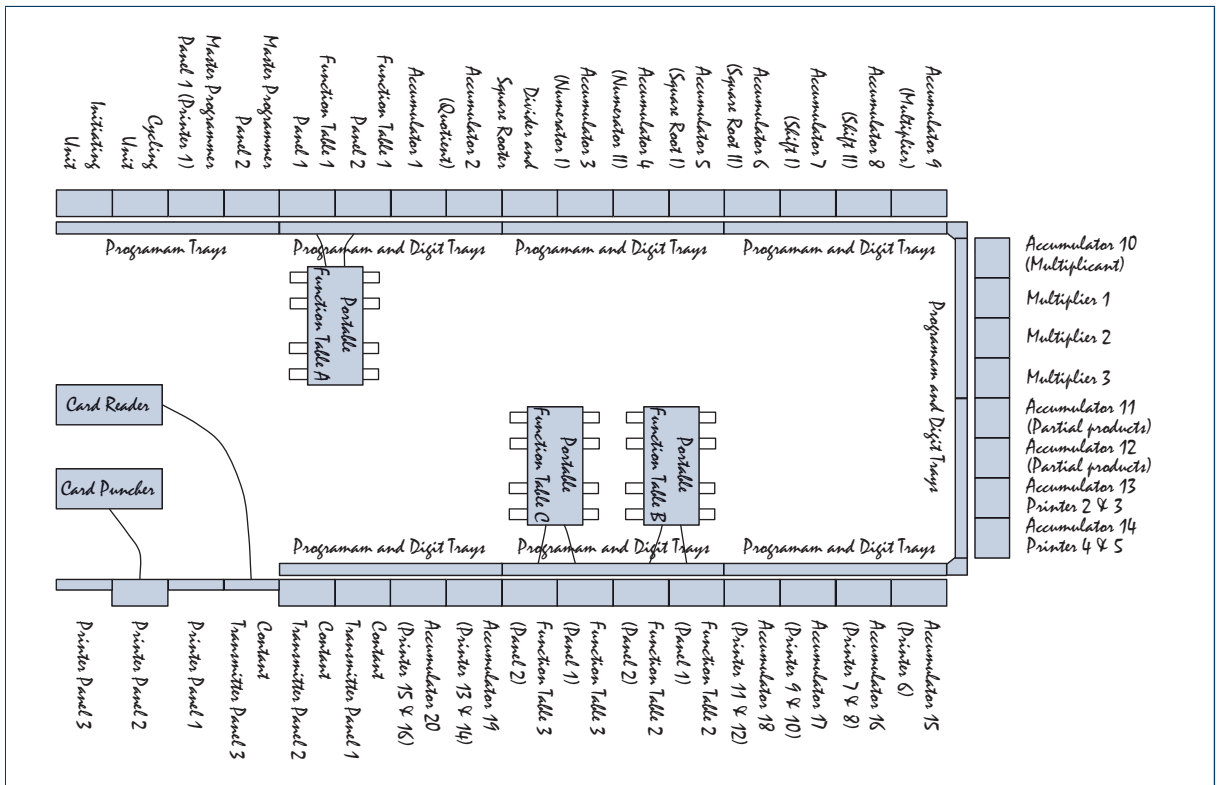


Abbildung 1.13: Der „Floorplan“ der ENIAC

loss wurde an der Moore School of Electrical Engineering der University of Pennsylvania unter der Leitung von J. Presper Eckert und John W. Mauchly gebaut und beeindruckte schon aufgrund seiner schieren Größe. Wie in Abbildung 1.13 dargestellt, bestand die ENIAC aus insgesamt 30 Einheiten, die U-förmig über den gesamten Raum verteilt angeordnet waren. Die gesamte Konstruktion kam auf ein Gesamtgewicht von knapp 30 Tonnen (Abbildungen 1.15 bis 1.17).

Der Entschluss, einen Rechner dieses Ausmaßes überhaupt zu bauen, wurde aus der Not geboren. Die Intensität, mit der der Zweite Weltkrieg 1939 über die Welt hereinbrach, setzte die U.S.-Armee unter enormen Zeitdruck, sich auf den immer wahrscheinlicher werdenden Kriegseintritt vorzubereiten. Unter anderem experimentierten die U.S.-Streitkräfte mit großkalibrigen ballistischen Geschützen, die anhand von Trajektorien-Tabellen auf ihr Ziel ausgerichtet wurden. Das Aufstellen der Tabellen erforderte das Lösen von Differenzialgleichungen

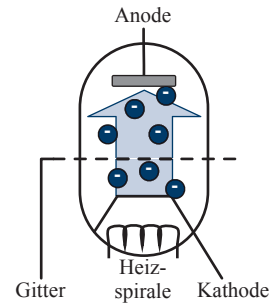
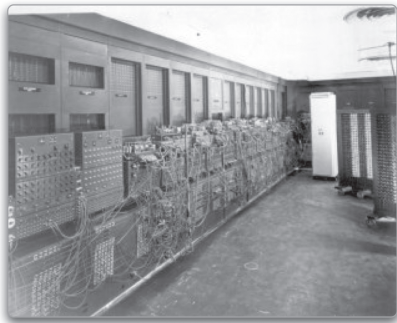
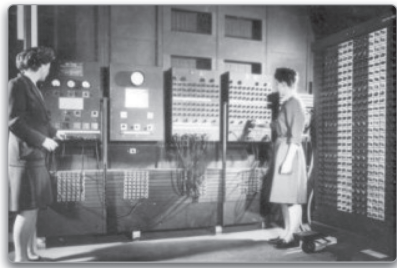


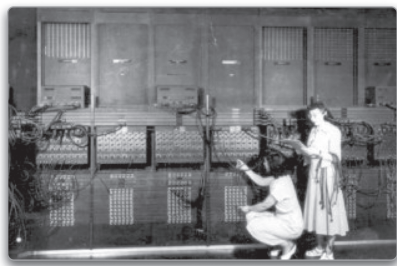
Abbildung 1.14: Die Trioden-Röhre als Schaltelement. Der durch die Heizspirale verursachte Stromfluss von der Kathode zur Anode kann durch das Anlegen einer Spannung an das Metallgitter blockiert oder verstärkt werden – ein einfaches Schaltelement entsteht.



**Abbildung 1.15:** Teilansicht der ENIAC (U.S.-Armee-Foto)



**Abbildung 1.16:** Das *Main Control Panel* der ENIAC (U.S.-Armee-Foto)



**Abbildung 1.17:** Programmiert wurde die ENIAC durch „Kabelstecken“ (U.S.-Armee-Foto)

zweiter Ordnung und damit Fähigkeiten außerhalb des Leistungsspektrums damaliger Rechenmaschinen. Die ENIAC, mit deren Hilfe die automatisierte Berechnung der Trajektorien-Tabellen möglich werden sollte, war in zweierlei Hinsicht revolutionär:

- Im Gegensatz zur Z3 oder der Harvard Mark I war es möglich, Verzweigungen und Schleifen zu programmieren. Damit unterscheidet sich das formale Berechnungsmodell der ENIAC deutlich von ihren Vorgängern und ist insbesondere mit dem heutiger Computer vergleichbar, auch wenn die Programmierung des Kolosses im Vergleich zur heutigen Technik kaum unterschiedlicher sein könnte.
- In der ENIAC wurde auf die Verwendung elektromechanischer Relais verzichtet und die Schaltlogik stattdessen mit Hilfe von *Vakuurröhren* implementiert. Zwar ist die ENIAC nicht der erste Röhrenrechner der Welt, aber der erste Großrechner, der die Vakuumröhre in Form der *Triode* konsequent als Schaltelement verwendete. Im Vergleich zum Relais zeichnet sich die Röhre durch eine um den Faktor 1000 bis 2000 gesteigerte Schaltgeschwindigkeit aus, so dass die ENIAC mit einer für damalige Verhältnisse beeindruckenden Taktfrequenz von 100 Kilohertz betrieben werden konnte. Eine einzige Multiplikation berechnete die ENIAC in nur knapp 3 Millisekunden.

Auf der negativen Seite brachte die ENIAC konstruktionsbedingt auch gravierende Nachteile mit sich. Ein wesentliches Element der Röhrentriode ist, wie in Abbildung 1.14 skizziert, die Heizspirale. Wird sie zum Glühen gebracht, so beginnt die Kathode Elektronen zu emittieren, die von der Anode angezogen und aufgefangen werden. Konstruktionsbedingt sind Vakuumröhren damit äußerst stromhungrige Bauelemente. Die Leistungsaufnahme der ca. 18.000 in der ENIAC verbauten Vakuumröhren summierte sich auf sagenhafte 174.000 Watt. Des Weiteren besitzen Vakuumröhren im Vergleich zu Relais eine sehr begrenzte Lebensdauer und das langwierige Auffinden und Austauschen defekter Röhren gehörte zur täglichen Arbeit eines ENIAC-Ingenieurs.

Der wohl gravierendste Nachteil betrifft jedoch die Handhabung der Maschine, die nicht auf eine flexible Programmierung ausgelegt war. Der Datenpfad wurde mit Hilfe von Steckverbindungen fest verdrahtet und ein Programm für die ENIAC war damit nichts anderes als ein Verbindungsplan, der die benötigten Steckverbindungen beschrieb. Hierdurch war ein flexibler Wechsel zwischen verschiedenen Programmen von vornherein ausgeschlossen.



Die Limitierungen der ENIAC inspirierten den in Budapest geborenen und später in die USA immigrierten Wissenschaftler John von Neumann zu einer bahnbrechenden Idee [34]. Sein Konzept sah vor, Programme und Daten beide in einem einzigen Speicher abzulegen und damit nicht mehr länger als getrennte Entitäten zu behandeln: Die Idee der *Von-Neumann-Architektur* war geboren (vgl. Abbildung 1.18). In Kapitel 11 werden wir dieses Prinzip, an dem sich auch die meisten modernen Computerarchitekturen heute noch orientieren, im Detail diskutieren.

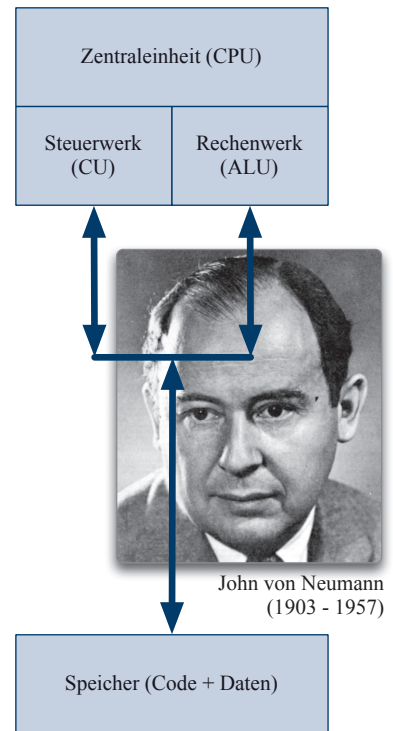
Trotz aller Limitierungen und Schwierigkeiten war die ENIAC ein eindrucksvoller Beweis für die Funktionstüchtigkeit der Röhrentechnik. In den Folgejahren wurden weitere Großcomputer gebaut, die dem alternen Koloss in ihrer technischen Konzeption und Handhabung in immer größerem Maße überlegen waren. Die ebenfalls legendäre, an der Manchester University entwickelte *Manchester Mark I* und der *Electronic Delay Storage Automatic Calculator (EDSAC)* der Cambridge University sind zwei Beispiele aus dieser Zeit. Wie die ENIAC basierten auch diese Rechner auf der Röhrentechnologie und begründeten zusammen die Computer der *ersten Generation*.

### Der Siegeszug des Transistors

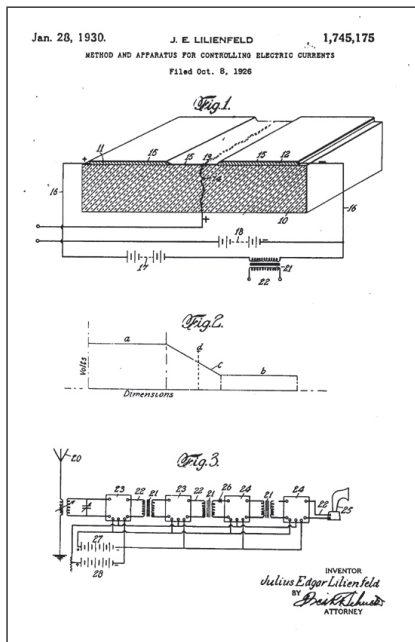
Der hohe Stromverbrauch und die begrenzte Lebensdauer der Vakuumröhre wurden mehr und mehr zum limitierenden Faktor bei der Entwicklung immer ausgefeilterer Rechnerkonstruktionen und für Forscher aus aller Welt zur treibenden Kraft bei der Suche nach neuen Technologien. Am 1. Juli 1948 verkündeten die Bell Laboratories in New York offiziell den Durchbruch und stellten den ersten praxistauglichen *Transistor* vor, der die bis dato dominierende Vakuumröhre vollständig ersetzen sollte:

*„An amazingly simple device, capable of performing efficiently nearly all the functions of an ordinary vacuum tube, was demonstrated for the first time yesterday at Bell Telephone Laboratories where it was invented. Known as the Transistor, the device works on an entirely new physical principle discovered by the Laboratories in the course of fundamental research into the electrical properties of solids. Although the device is still in the laboratory stage, Bell scientists and engineers expect it may have far-reaching significance in electronics and electrical communication.“*

*Bell Laboratories, Press Release [9]*



**Abbildung 1.18:** John von Neumann begründete die grundlegende Rechnerarchitektur, die auch heute noch die Basis moderner Computer bildet.



**Abbildung 1.19:** Julius Lilienfeld patentiert als Erster die Idee des Feldeffekt-Transistors. Die Abbildung zeigt einen Auszug aus der zweiten Patentschrift aus dem Jahre 1928 [56, 57].

An der Entwicklung des ersten technisch verwertbaren, voll funktionsfähigen Transistors waren die Ingenieure William Shockley, John Bardeen und Walter Brattain maßgeblich beteiligt und erhielten am 1. Dezember 1956 als Anerkennung ihrer bahnbrechenden wissenschaftlichen Leistung den Nobelpreis [6, 80]. Das Ergebnis der Arbeiten von Shockley, Bardeen und Brattain gilt heute als der erste technisch verwertbare und voll funktionsfähige Transistor – die Idee desselben war jedoch schon weit früher bekannt. So wurde das Prinzip des Transistors bereits 1928 von Julius Edgar Lilienfeld zum Patent angemeldet (Abbildung 1.19).

Der erste Computer auf Transistor-Basis wurde an der Manchester University Anfang der Fünfzigerjahre gebaut. Dem 1953 fertiggestellten Prototyp folgte eine deutlich erweiterte Variante, die zwei Jahre später in Betrieb genommen werden konnte. Aufgrund der noch sehr jungen Transistortechnik war der Rechner den modernen Röhren-Computern von damals sowohl in seiner Leistungsfähigkeit, aber auch in seiner Zuverlässigkeit deutlich unterlegen. Sahen einige Experten den Transistor gegenüber der Röhre zu Anfang im Nachteil, so konnte die neue Technologie den beginnenden Wettstreit jedoch alsbald für sich entscheiden. Mit der zunehmenden Verfeinerung der Technik stand mit dem Transistor ein Schaltelement zur Verfügung, das der Röhrentriode in puncto Leistungsaufnahme, Geschwindigkeit und Zuverlässigkeit haushoch überlegen war.

Der Übergang von der Röhre zum Transistor läutete das Zeitalter der Computer der *zweiten Generation* ein und war von weiteren wichtigen Entwicklungen begleitet. Steuer- und Rechenwerke wurden deutlich komplexer und bis dato fortschrittliche Konzepte wie die Verwendung indizierbarer Register oder der Einsatz von Gleitkomma-Hardware gehörten schnell zum Stand der Technik. Speicherseitig hielt der *Ferritkernspeicher* Einzug, der Daten im Kilobyte-Bereich speichern konnte, allerdings in mühevoller Handarbeit gefertigt werden musste. Auch die ersten Programmiersprachen datieren auf diese Zeit. 1957 wurde der erste FORTRAN-Compiler ausgeliefert und 1960 die Programmiersprache COBOL verabschiedet. Beide Sprachen sind heute immer noch im Einsatz – wenngleich sich deren Verwendungszweck nur noch auf wenige Spezialanwendungen konzentriert.

Der Siegeszug des Transistors begann im Jahre 1958, als es dem für Texas Instruments arbeitenden Ingenieur Jack Kilby gelang, den ersten integrierten Schaltkreis (engl. *Integrated Circuit*, kurz IC) herzustellen. Konnten Transistoren bis zu diesem Zeitpunkt ausschließlich als diskretes Bauelement verbaut werden, war es mit Kilbys Technik nunmehr möglich, mehrere Transistoren auf einem kleinen Stück Silizium zu in-

tegrieren. Kilbys Entdeckung, für die er im Jahr 2000 – fünf Jahre vor seinem Tod am 20.5.2005 – mit dem Nobelpreis geehrt wurde, war vielleicht der entscheidendste Durchbruch der Computergeschichte und der Auslöser einer bis heute anhaltenden Leistungsexplosion, die sich zur damaligen Zeit niemand jemals hätte erträumen können.

Bereits ein Jahr nach der Entwicklung des integrierten Schaltkreises gelang es dem Halbleiterhersteller Fairchild Semiconductor, mit der *Planartechnik* die Grundlage der Massenfertigung zu schaffen. Im Folgejahr brachte Fairchild nahezu zeitgleich mit Texas Instruments den ersten kommerziellen IC auf den Markt. Weitere drei Jahre später begann Fairchild 1963 mit der Produktion des Modells 907, der bereits zwei vollständige Logikgatter in einem einzigen Chip vereinte und damit die SSI-Technologie begründete. SSI ist die Abkürzung für *Small-Scale Integration* und bezeichnet eine von mehreren Komplexitätsklassen, in die sich integrierte Schaltklassen einordnen lassen.

Tabelle 1.2 fasst die verschiedenen Komplexitätsklassen zusammen, denen sich integrierte Schaltkreise zuordnen lassen. Zwischen den einzelnen Klassen existiert keine scharfe Trennung und die numerischen Angaben zur Gatteranzahl sind als Orientierungswerte zu verstehen. So unterscheiden sich auch die in der Literatur angegebenen Gattergrenzen mitunter erheblich und insbesondere die Abgrenzung im Bereich der VLSI- und ULSI-Klassen wurde im Laufe der Zeit mit zunehmender Chip-Komplexität immer wieder nach oben korrigiert oder durch neu eingeführte Klassen (SLSI, XLSI, GSI) ergänzt.

Der *Micromosaic*, ebenfalls von Fairchild Semiconductor auf den Markt gebracht, integrierte 1967 bereits 150 Logikgatter und ist aus historischer Sicht in zweierlei Hinsicht interessant. Zum einen gehörte er mit der Anzahl integrierter Gatter zu den ersten ICs der MSI-Technik (*Medium-Scale Integration*), zum anderen ist er der Vorläufer programmierbarer Logikbausteine, die wir in Kapitel 7 genauer betrachten werden.

Für die Herstellung des *Micromosaics* wurde ein regelmäßig aufgebaute Chip verwendet und die Transistorverbindungen erst später durch das Einbringen einer kundenspezifischen Belichtungsmaske erzeugt. Auf diese Weise war es erstmals möglich, eine speziell auf einen Kunden zugeschnittene Schaltung mit vergleichsweise geringem Aufwand und Kosten zu produzieren.

Eines der ersten vollständigen Computersysteme, das sich die integrierte Schaltkreistechnologie zu Nutze machte, war das legendäre *System/360*, das am 7. April 1964 der Öffentlichkeit vorgestellt wur-

Small-Scale Integration
<ul style="list-style-type: none"> <li>■ kurz SSI</li> <li>■ weniger als 100 Gatter</li> </ul>
Medium-Scale Integration
<ul style="list-style-type: none"> <li>■ kurz MSI</li> <li>■ 100 bis 1000 Gatter</li> </ul>
Large-Scale Integration
<ul style="list-style-type: none"> <li>■ kurz LSI</li> <li>■ 1000 bis 10.000 Gatter</li> </ul>
Very-Large-Scale Integration
<ul style="list-style-type: none"> <li>■ kurz VLSI</li> <li>■ 10.000 bis 100.000 Gatter</li> </ul>
Ultra-Large-Scale Integration
<ul style="list-style-type: none"> <li>■ kurz ULSI</li> <li>■ 100.000 bis 1.000.000 Gatter</li> </ul>
Super-Large-Scale Integration
<ul style="list-style-type: none"> <li>■ kurz SLSI</li> <li>■ 1.000.000 bis 10.000.000 Gatter</li> </ul>
Extra-Large-Scale Integration
<ul style="list-style-type: none"> <li>■ kurz ELSI oder XLSI</li> <li>■ 10.000.000 – 100.000.000 Gatter</li> </ul>
Giga-Scale Integration
<ul style="list-style-type: none"> <li>■ kurz GSI</li> <li>■ mehr als 100.000.000 Gatter</li> </ul>

**Tabelle 1.2:** Klassifizierung integrierter Schaltkreise

Erste Generation		Zweite Generation	Dritte und vierte Generation
1940 – 1954		1955 – 1964	ab 1965
Relais	Vakuümrohre	Transistor	Integrierter Schaltkreis
Schaltzeit: $10^{-1}$ s	Schaltzeit: $10^{-4}$ s	Schaltzeit: $10^{-6}$ s	Schaltzeit: $< 10^{-9}$ s

The images below the table show four examples of switching elements in house-shaped frames:

- 1. A mechanical relay with multiple contacts and a coil.
- 2. A vacuum tube (thermionic valve) with a glass envelope and metal pins.
- 3. A small, cylindrical transistor with three leads.
- 4. An integrated circuit (IC) chip, specifically a 7414N K8424, with multiple pins.

**Tabelle 1.3:** Die Entwicklung der Schaltelemente in der Übersicht

de [74]. Mit dieser Modellreihe begründete die Firma IBM die Ära der *Mainframe-Computer* und läutete gleichermaßen das Ende der Pionierzeit der Computertechnik ein. Der Aufwand, den IBM in die neue Idee investierte, war gigantisch. Zu Hochzeiten arbeiteten rund 50.000 Mitarbeiter an dem Projekt, dessen Gesamtkosten auf über 5 Milliarden US-Dollar geschätzt werden. Das System/360 brachte einige technische Neuerungen mit sich, die wir noch heute in modernen Computersystemen verwenden. So nutzt die S/360 beispielsweise das *Zweierkomplement* zur Darstellung negativer Zahlen und führt das Prinzip der byteweisen Speicheradressierung ein. Auf beide Konzepte kommen wir in Kapitel 3 im Detail zurück. Neben der Entwicklung von Großrechnern hat die zunehmende Verfügbarkeit leistungsfähigerer und billigerer SSI-, MSI- und LSI-Komponenten auch den Bau immer kleinerer und kostengünstigerer Computer ermöglicht. Die um 1970 gebauten PDP-Rechner (PDP = Programmable Data Processor) der Digital Equipment Corporation (DEC) sind Beispiele solcher *Minicomputer*.

Nach dem elektromagnetischen Relais, der Röhre, und dem Transistor in Form eines diskreten Bauelements ist der *integrierte Schaltkreis* bereits der vierte revolutionäre Wechsel der Basistechnologie, den die Computertechnik in ihrer vergleichswisen jungen Geschichte erfahren hat. Genau wie die Relais- und Röhrentechnologie stellvertretend für die Computer der ersten Generation und die diskrete Transistortechnik für die Computer der zweiten Generation steht, so werden die frühen Rechner auf Basis integrierter Schaltkreise als Computer der dritten Ge-

neration bezeichnet. Bis zum heutigen Tag arbeiten Computer nach diesem Prinzip, wenngleich sich hochintegrierte Schaltkreise der letzten Generationen bezüglich ihrer Leistungsdaten kaum noch mit den ersten integrierten Schaltkreisen vergleichen lassen. Tabelle 1.3 stellt die Basistechnologien der verschiedenen Computergenerationen gegenüber.

Auch die Computertechnik der dritten Generation wurde durch einhergehende Entwicklungen weiter geprägt. Der Ferritkernspeicher hatte ausgedient und wurde nach und nach durch integrierte Speicherchips ersetzt. Diese waren deutlich kleiner und schneller als ihre Vorgänger und ermöglichten hierdurch drastisch höhere Speicherkapazitäten zu geringeren Kosten. Mehr und mehr Lochkartenleser mussten interaktiven *Text-Terminals* weichen und so ging auch die Ära der Lochkarte beständig ihrem Ende zu. Auch die ersten Betriebssysteme entstanden auf Computern dieser Generation. Bereits mit Hilfe der ersten Betriebssysteme war es möglich, verschiedene Programme gleichzeitig auszuführen und zu überwachen. Die Produktivität der Software-Entwicklung nahm hierdurch erst richtig an Fahrt auf.

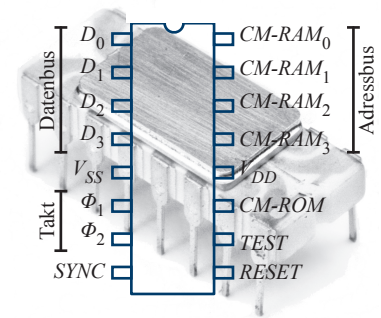


Abbildung 1.20: Der 4004-Mikroprozessor

## Der erste Mikroprozessor

1968 verließen Robert Noyce, Gordon Moore und kurze Zeit später Andrew Grove die Firma Fairchild Semiconductor und gründeten die Integrated Electronics Corporation (Intel) im kalifornischen Santa Clara [46]. Bereits zwei Jahre später stellte Intel mit dem *1103* den ersten DRAM-Speicherbaustein vor. Die eigentliche Sternstunde schlug für Intel jedoch im Jahre 1971, als es dem Ingenieur Federico Faggin als Erstem gelang, alle Komponenten eines Prozessors gemeinsam auf einem einzigen Stück Silizium zu integrieren. Bereits kurze Zeit später brachte Intel mit dem *Intel 4004* den ersten Mikroprozessor der Welt auf den Markt und leitete damit den Beginn der vierten und bis dato letzten Computergeneration ein.

Der in einer Strukturbreite von 10  $\mu\text{m}$  gefertigte Prozessor bestand aus 2250 Transistoren und lief mit einer Taktfrequenz von 740 kHz. Wie in Abbildung 1.20 gezeigt, wird das Taktsignal über zwei separate Leitungen zugeführt. Eine steigende Taktflanke im Signal  $\phi_2$  bewirkte den eigentlichen Zustandswechsel, während das zeitversetzt angelegte Signal  $\phi_1$  zur internen Steuerung eingesetzt wurde. Die beiden Pins  $V_{SS}$  und  $V_{DD}$  dienten der Stromaufnahme und versorgten den Prozessor mit einer Spannung von 5 V bzw.  $-10$  V. Intern arbeitete der Prozessor, wie in Abbildung 1.21 skizziert, mit einer Bitbreite von gerade einmal 4 Bit und verfügte neben 16 Datenregistern über 4 Stapelregister zur Spei-

Sprungbefehle	
<b>JUN</b>	Direkter Sprung
<b>JIN</b>	Indirekter Sprung
<b>JCN</b>	Bedingte Verzweigung
<b>JMS</b>	Unterprogrammaufruf
<b>BBL</b>	Unterprogrammende
Ladebefehle	
<b>FIM</b>	Register laden (direkt)
<b>FIN</b>	Register laden (indirekt)
<b>LD</b>	Register $\rightarrow$ Akkumulator
<b>XCH</b>	Register $\leftrightarrow$ Akkumulator
<b>LDM</b>	Wert $\rightarrow$ Akkumulator
Arithmetikbefehle	
<b>INC</b>	Inkrementieren
<b>ISZ</b>	<b>INC</b> + bedingter Sprung
<b>ADD</b>	Addition
<b>SUB</b>	Subtraktion
Sonstige Befehle	
<b>NOP</b>	Keine Operation

Tabelle 1.4: Die Grundoperationen des Intel 4004-Prozessors

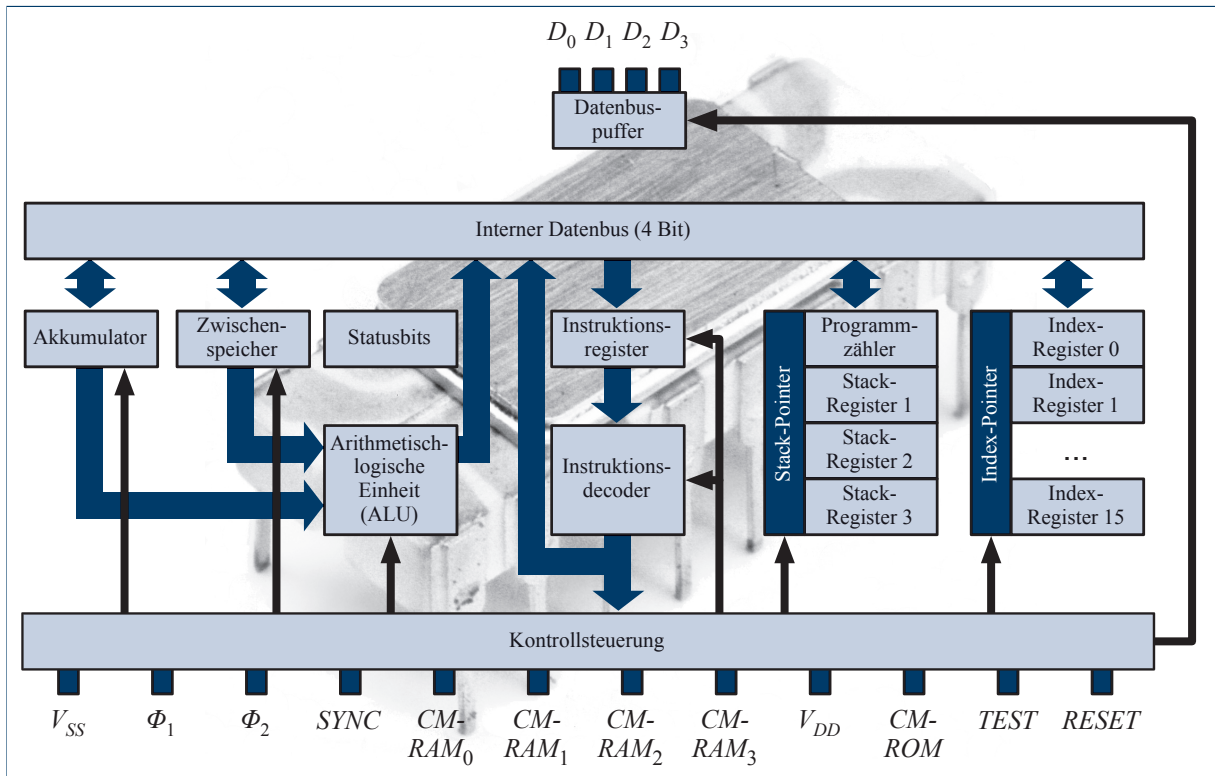


Abbildung 1.21: Blockschaltbild des 4004-Mikroprozessors

cherung des Programmzählers und der Rücksprungadressen von Unterprogrammaufrufen.

Um die mindestens 8 Bit langen Befehlsörter mit Hilfe des 4 Bit breiten Bus des 4004 trotzdem in einem einzigen Takt einlesen zu können, arbeitete der Prozessor im sogenannten *Multiplexing-Modus*. Das bedeutet, dass der Bus mit doppelter Taktfrequenz betrieben wurde, so dass die ersten 4 Bit in der ersten Hälfte und die restlichen 4 Bit in der zweiten Hälfte einer Taktperiode eingelesen werden konnten. Obwohl die Multiplexing-Logik die interne Architektur des 4004 deutlich verkomplizierte, vereinfachte sie erheblich die Produktion. Der Prozessor passte vollständig in ein 16-Pin-Gehäuse.

Tabelle 1.4 vermittelt einen detaillierteren Eindruck über die Grundbefehle des 4004-Prozessors. Sehen wir von der NOP-Instruktion ab, die ausschließlich dazu dient, die Befehlsausführung um einen einzi-

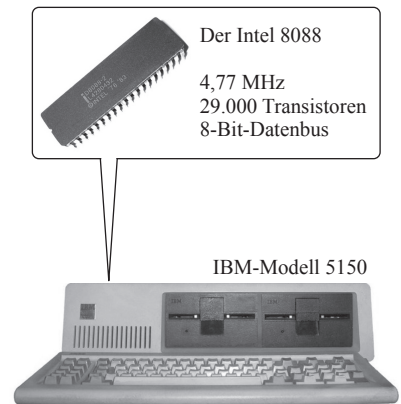
gen Takt zu verzögern, lassen sich die Grundbefehle in drei Kategorien einteilen. Die erste enthält eine Ansammlung von Sprungbefehlen zur Steuerung des Kontrollflusses. Die zweite dient zum Datentransfer zwischen ROM, Register und Akkumulator und die dritte umfasst alle Arithmetikbefehle. Neben diesen Grundbefehlen verfügt der 4004 über etliche spezialisierte Befehle, die in der Tabelle nicht aufgeführt sind und dem Lesen und Beschreiben des RAM-Speichers dienen. Wie die Liste der Grundbefehle zeigt, besaß der 4004 nur begrenzte Arithmetikfähigkeiten. Befehle zum Multiplizieren oder Dividieren zweier Datenwörter sind ebenso wenig vorhanden wie Befehle zur Durchführung logischer Verknüpfungen. Ebenfalls fehlen Instruktionen zur Interrupt-Steuerung – ein Prinzip, das der 4004 noch gar nicht kannte.

## Die Entwicklung bis heute

Die Innovationskraft, die der erste Mikroprozessor mit sich brachte, wurde Anfang der Siebzigerjahre durchaus skeptisch betrachtet und selbst Intel maß dem 4004 zum Zeitpunkt seines Erscheinens bei weitem nicht die Bedeutung zu, die er rückblickend verdient. Trotzdem setzte sich das Prinzip des Mikroprozessors durch und entwickelte sich schließlich zur treibenden Kraft der gesamten Computerbranche.

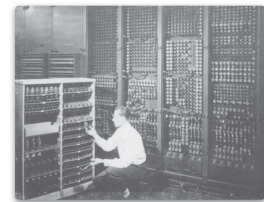
Die Evolution, die der Mikroprozessor in wenigen Jahren vollzog, lässt sich anhand der Entwicklung der Intel-Prozessoren deutlich nachzeichnen. Obwohl bereits der 4004 in etwa die Leistung einer ENIAC bot, war er mit seiner geringen Bitbreite und dem limitierten Befehlssatz noch nicht für den universellen Einsatz geeignet. Dies änderte sich 1975 mit dem Erscheinen des 8-Bit-Prozessors 8080, den viele Experten zu den ersten universell einsetzbaren Mikroprozessoren zählen. 1978 begründete Intel mit dem 16-Bit-Prozessor 8086 und kurze Zeit später mit dem auf 8 Bit reduzierten 8088 die x86-Architektur. Als kleiner Bruder des 8086 feierte der 8088 im ersten *Personal Computer (PC)* der Firma IBM – dem in Abbildung 1.22 gezeigten IBM-Modell 5150 – sein Debüt. Mit dem beispiellosen Markterfolg der PCs wurde die x86-Architektur gleichermaßen zu einem Massenprodukt.

1982 ersetzte IBM den 8086 durch den 80286, der zwar immer noch mit einer Registerbreite von 16 Bit arbeitete, die Funktionalität des 8086 jedoch durch zahlreiche neue Konzepte, wie das Schützen bestimmter Speicherbereiche (*protected mode*), deutlich übertraf. Der erste Intel-Prozessor mit einer 32-Bit-Architektur erschien drei Jahre später in Form des 80386-Prozessors. 1989 wurde der 80386 durch den deutlich schnelleren 80486 abgelöst, der unter anderem den vormals externen



**Abbildung 1.22:** Das Modell 5150 der Firma IBM begründet das Zeitalter der PCs.

170.000 Watt  
5000 Operationen/s  
18.000 Röhren



ENIAC  
1946

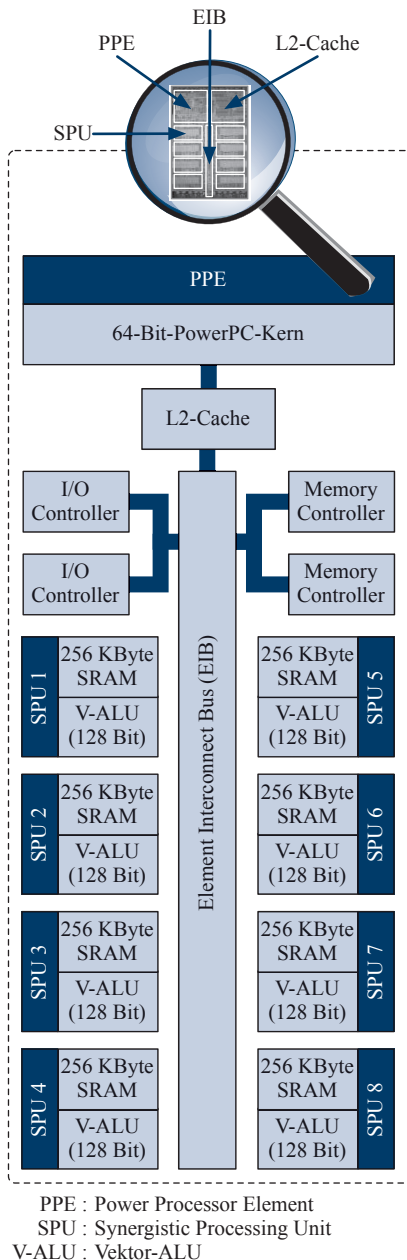
(U.S.-Armee-Foto)

Tablet-PC  
2023



< 5 Watt  
> 15 Billionen Operationen/s  
> 15 Milliarden Transistoren

**Abbildung 1.23:** Computer der heutigen Generation übertreffen die Rechenleistung ehemaliger Großcomputer um ein Vielfaches.



**Abbildung 1.24:** Architektur des Cell-Prozessors von IBM

Coprozessor direkt auf dem Prozessorchip integrierte. Der Nachfolger des 80486 kam 1993 auf den Markt – aus patentrechtlichen Gründen nicht als 80586, sondern unter dem schutzfähigen Namen Pentium. In den Jahren folgten die Modelle Pentium Pro, Pentium II, Pentium III sowie diverse Varianten des Pentium 4. Im Jahr 2006 verabschiedete sich Intel schließlich vom Namen *Pentium* und leitete mit der *Core microarchitecture* die Ära der Mehrkernprozessoren ein. Kurz danach begann Intel, Prozessoren der Core-2-Serie zu vertreiben, und im Jahr 2008 wurde daraus die Intel-Core-i-Serie.

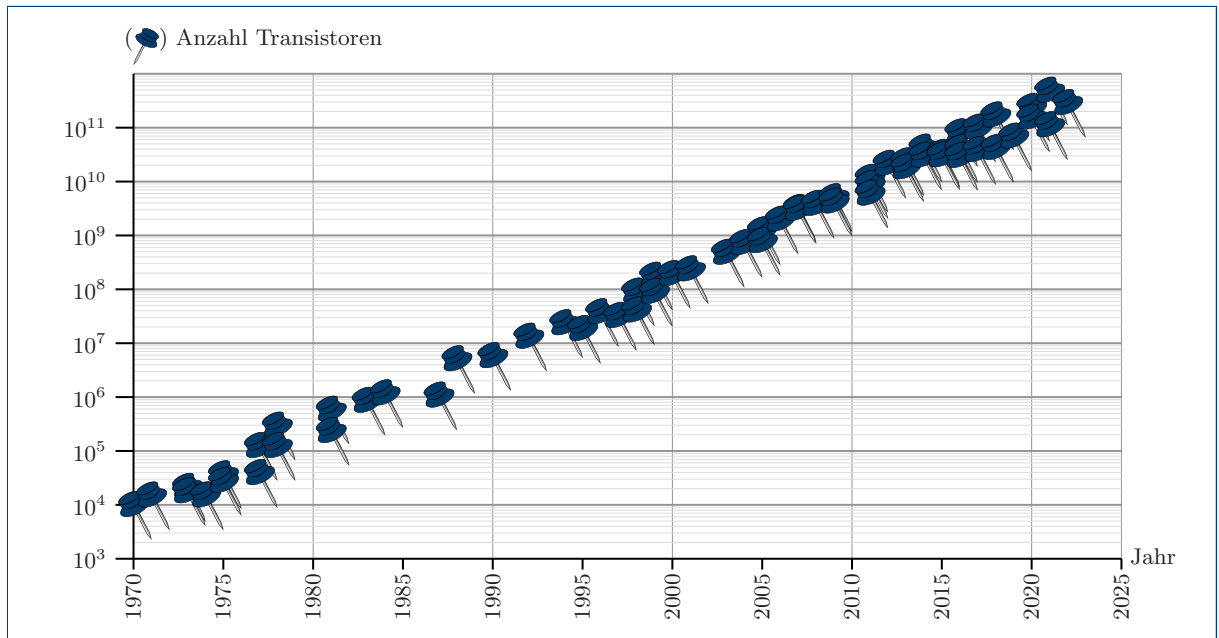
Verglichen mit dem 4004 hat sich die Architektur moderner Prozessoren zwar an unzähligen Stellen weiterentwickelt, jedoch nie grundlegend verändert. In puncto Leistung haben beide trotzdem nicht mehr viel gemein. In einem typischen Arbeitsplatzrechner, Tablet-PC oder Smartphone operieren heute mehrere parallel arbeitende Kerne im Gigahertzbereich, die jeder für sich die Leistung früherer Großrechner um ein Vielfaches übertreffen (Abbildung 1.23). Auch die Anzahl der Transistoren spricht eine deutliche Sprache. Kommt der 4004-Prozessor noch mit 2250 Transistoren aus, so liegt die Anzahl bei modernen Prozessoren heute jenseits der Milliardengrenze.

Als Beispiel für die Topologie einer Mehrkernarchitektur ist in Abbildung 1.24 das Blockschaltbild des *Cell-Prozessors* der Firma IBM dargestellt, der nicht zuletzt durch den Einsatz in der Playstation 3 einen hohen Bekanntheitsgrad erlangte. Der größte Teil der über 230 Millionen Transistoren verteilt sich auf 8 separate *Synergistic Processing Units* (SPUs), die mit ihren SIMD-Vektoreinheiten für die effiziente Ausführung arithmetischer Operationen optimiert sind. Angesteuert werden die SPU's über einen Prozessorkern auf PowerPC-Basis, der sich direkt neben dem L2-Cache befindet und die Kontrolle über den gesamten Chip besitzt. Mit der Einführung der Playstation 4 hat sich Sony von der Cell-Architektur verabschiedet und sich stattdessen für eine x86-CPU der Firma der AMD entschieden. Die Grafikberechnung übernimmt in der neuen Konsole eine GPU (*Graphics Processing Unit*) der Radeon-Familie, ebenfalls von der Firma AMD.

### 1.3 Wohin geht die Reise?

Blicken wir auf die bewegte Geschichte der Computertechnik zurück, so drängt sich unweigerlich die Frage auf, welchen Weg die Entwicklung in Zukunft einschlagen wird. Wie vorsichtig wir mit voreiligen Schlussfolgerungen sein müssen, lehrt uns abermals die Vergangenheit.

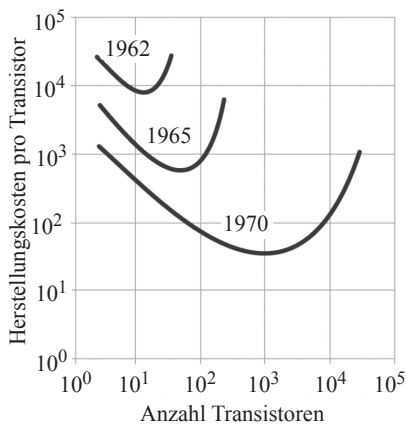




**Abbildung 1.25:** Die Entwicklung der Hardware-Komplexität am Beispiel der Desktop-Mikroprozessoren. Werden die einzelnen Messpunkte zu einer Linie verbunden, so entsteht eine fast perfekte Gerade. Durch die logarithmische Skala auf der y-Achse lässt sich hieraus ein exponentielles Wachstum der Hardware-Komplexität ableiten.

So hätte in den Pioniertagen der Computertechnik niemand auch nur zu träumen gewagt, dass sich die Informationstechnologie zu dem entwickelt, was sie heute ist.

Der folgende, von Howard Aiken im Jahre 1947 formulierte Satz bringt mit wenigen Worten den Zeitgeist der Pioniertage auf den Punkt: „*Only six electronic digital computers will be required to satisfy the computing needs of the entire United States*“. In die gleiche Kerbe schlägt das folgende Zitat, das ebenfalls um diese Zeit herum datiert wird: „*I think there is a world market for maybe five computers*“. Zugeschrieben wird dieses Zitat keinem anderen als Thomas Watson Sr., dem Gründer der Firma IBM. Ob es wirklich Watson war, der die legendären Worte formulierte, ist jedoch heftig umstritten und konnte bis heute nie zweifelsfrei belegt werden. Doch welchen Nutzen können wir aus den Prognosen der Vergangenheit ziehen, die nicht unzutreffender hätten sein können? Am Ende ist es die Erkenntnis, dass wir von *langfristigen* Vorhersagen zur Entwicklung der Computertechnik absehen sollten. Die rasante Entwicklung der letzten Jahrzehnte barg vielerlei Überraschungen und wir dürfen davon ausgehen, dass weitere folgen.



„The complexity for minimum component costs has increased at a rate of roughly a factor of two per year ... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer.“ [69]

**Abbildung 1.26:** Im Jahre 1965 prognostizierte Gordon Moore die exponentielle Zunahme der Chip-Komplexität. In leicht abgewandelter Form ist das Moore'sche Gesetz bis heute gültig.

Gleichwohl lassen sich in den vielen Unstetigkeiten der Vergangenheit auch konstante Entwicklungen erkennen, mit deren Hilfe wir zumindest kurz- und mittelfristige Trends mit einer gewissen Wahrscheinlichkeit prognostizieren können. Eine dieser Entwicklungen betrifft die Zunahme der Hardware-Komplexität, gemessen an der Anzahl der Transistoren, die ein einzelner Silizium-Chip integriert. Betrachten wir die Entwicklung über die Jahre hinweg, so zeigt sich ein beeindruckend stabiles Bild.

Seit dem Erscheinen des Intel-4004-Prozessors hat sich die Anzahl der Transistoren, die auf ein einziges Stück Silizium integriert wurden, ca. alle 20 Monate verdoppelt. Dieser Trend setzt sich bis heute fort und zumindest mittelfristig ist kein Ende in Sicht. Die exponentielle Zunahme der Hardware-Komplexität hat einen berühmten Namen und wird heute als das *Moore'sche Gesetz* bezeichnet (*Moore's law*). Gordon Moore hatte das geschilderte Phänomen im Jahre 1965 vorausgesagt, wenngleich mit einer etwas zu optimistischen Geschwindigkeitsannahme (Abbildung 1.26).

Die Bedeutung, die das Moore'sche Gesetz für die Entwicklung der Computertechnik besitzt, ist selbst in Expertenkreisen bis heute umstritten. Für einige spiegelt sich in der Gesetzmäßigkeit schlicht der kontinuierliche Fortschritt eines ganzen Technologiesektors wider. Andere sehen im Gesetz von Moore eine treibende Kraft, die großen Teilen der Hardware-Industrie die Schlaggeschwindigkeit regelrecht aufzwingt. Ganz von der Hand zu weisen ist eine solch aktive Rolle nicht. Nahezu die gesamte Computerindustrie vertraute früher wie heute auf die Gültigkeit dieser Wachstumsprognose und richtete ihre mittel- und langfristige Planung darauf aus. Einige Experten sehen in *Moore's law* deshalb keinen empirischen Zufall, sondern eine Gesetzmäßigkeit, die im Sinne einer sich selbst erfüllenden Prophezeiung aus eigener Kraft wahr wurde.

Wann der Tag kommen wird, an dem *Moore's law* seine Gültigkeit verliert, ist umstritten. Dass der Tag kommen wird, gilt als sicher – schon aus rein fundamentalen Überlegungen sind exponentielle Wachstumsraten nicht über beliebig lange Zeiträume durchzuhalten. In der Tat wurde durch die gesamte Computergeschichte hindurch immer wieder postuliert, das Ende des Machbaren sei erreicht und jedes Mal waren es findige Tüftler und Ingenieure, die sämtliche Kritiker Lügen strafte.

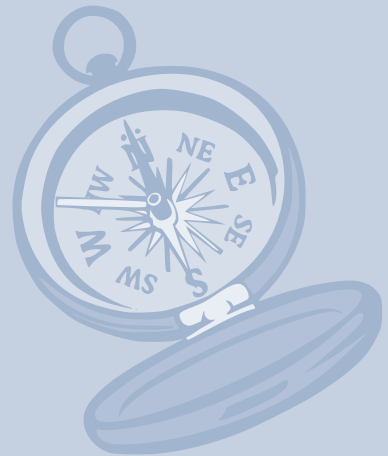
Damit ist es an der Zeit, hinter die Kulissen zu blicken und die Methoden und Techniken genauer zu beleuchten, die eine technische Entwicklung dieses Ausmaßes überhaupt erst möglich machten.

## 2 Halbleitertechnik

---

### In diesem Kapitel werden Sie . . .

- die elektrischen Eigenschaften von Halbleiterkristallen verstehen,
- mit der Diode und dem Transistor zwei grundlegende Halbleiterelemente kennen lernen,
- die Produktion integrierter Schaltkreise nachvollziehen.



Der dänische Physiker Niels Henrik David Bohr wurde 1885 in Kopenhagen geboren und zählt zu den bedeutendsten Physikern des zwanzigsten Jahrhunderts. Im Jahre 1922 wurde Bohr für seine Verdienste um die Erforschung der Struktur der Atome und der von ihnen ausgehenden Strahlung mit dem Nobelpreis ausgezeichnet. Zu seinen wichtigsten Hinterlassenschaften gehört zweifelsfrei das Bohr'sche Atommodell, das neue Erkenntnisse in den inneren Aufbau von Atomen gewährte und zugleich als Wegbereiter für die sich später entwickelnde Quantenmechanik angesehen wird. Das Modell von Bohr ist eine Weiterentwicklung des Rutherford'schen Modells, das Elektronen wie winzige, den Atomkern umkreisende Planeten interpretiert und damit im Widerspruch zur klassischen Elektrodynamik steht. Diese sagt aus, dass bewegende elektrische Ladungen einen Teil ihrer Energie als elektromagnetische Welle abstrahlen. Genau eine solche bewegliche Ladung stellt das Elektron dar. Aufgrund seiner Rotation um den Atomkern müsste dieses permanent an Energie verlieren und schließlich in den Atomkern stürzen. Bohr postulierte in seinem Modell die Existenz von *Schalen*, auf denen sich die Elektronen mit unterschiedlichen Energieniveaus verlustfrei bewegen können. Mit Hilfe dieses Modells war es erstmals möglich, viele chemische Reaktionen auf erstaunlich einfache Weise physikalisch zu erklären. Heute gilt das Bohr'sche Atommodell als veraltet, da es im Widerspruch zu den modernen Erkenntnissen der Quantenmechanik steht. Andere Modelle, wie z. B. das Orbitalmodell, nehmen gegenwärtig dessen Platz ein. Nichtsdestotrotz beschreibt das Bohr'sche Atommodell die Eigenschaften von Atomen und Elektronen so genau, dass sich damit unter anderem das elektrische Verhalten von Halbleiterkristallen präzise erklären lässt.

## 2.1 Halbleiter

Der historische Rückblick in Kapitel 1 hat aufgezeigt, dass die Entwicklung der Computertechnik eng mit den Fortschritten im Bereich der integrierten Schaltungstechnik verbunden ist. Ohne die beeindruckenden Erfolge im Bereich der Hochintegration in der zweiten Hälfte des vorherigen Jahrhunderts wäre die Konstruktion von elektronischen Geräten, wie wir sie heute kennen und fast schon als selbstverständlich erachten, niemals Realität geworden. Im Detail betrachtet, setzen sich moderne Mikrochips aus mehreren Millionen winziger Verknüpfungsglieder zusammen, die in einem komplizierten Fertigungsprozess dicht gepackt auf ein kleines Stück Silizium aufgebracht werden. Dass wir heute technisch in der Lage sind, mikroskopisch kleine Schaltelemente zu erzeugen, die zudem um viele Größenordnungen schneller schalten als die lange Zeit dominierende Röhrentriode, haben wir den chemischen und physikalischen Eigenschaften einer ganz bestimmten Stoffgruppe zu verdanken – den *Halbleitern*.

Halbleiter sind der Grundstoff mikroelektronischer Schaltungen und die daraus gefertigten Transistoren spielen in der Computertechnik die gleiche Rolle wie die Nukleotide in der Genetik. Aufgrund ihrer immensen Bedeutung wollen wir in diesem und den nächsten Abschnitten einen genaueren Blick auf die Grundbausteine wagen, aus denen sich sämtliche modernen Hardware-Schaltungen zusammensetzen. Insbesondere werden wir die Frage klären, was Halbleiterelemente so einzigartig macht und wie wir ihre besonderen Eigenschaften für die Konstruktion komplexer Mikrochips nutzen können.

Im nächsten Abschnitt werden wir zunächst einen kleinen, aber unabdingbaren Ausflug in die Chemie unternehmen und zunächst auf atomarer Ebene klären, wie sich Stromflüsse durch die Bewegung einzelner Elektronen im Detail erklären lassen. Auf den erworbenen Grundkenntnissen aufbauend werden wir in den Abschnitten 2.2.1 bis 2.2.3 mit der Halbleiterdiode und dem Transistor die zentralen Bausteine kennen lernen, die in einem komplexen Zusammenspiel das Verhalten von Mikrochips definieren, mit denen wir tagtäglich hundertfach in Berührung kommen.

### 2.1.1 Atommodell von Bohr

Nach dem Bohr'schen Atommodell setzt sich ein einzelnes Atom aus *Protonen*, *Neutronen* und *Elektronen* zusammen. Protonen tragen eine

positive, Elektronen eine negative Ladung. Beide weisen die exakt identische Ladungsmenge auf, so dass ein Proton durch jeweils ein Elektron kompensiert wird. Der dritte atomare Baustein – das Neutron – ist ladungsneutral und trägt ausschließlich zur Masse des Atoms bei. In allen chemischen Elementen ist die Beschaffenheit der drei Grundbausteine gleich. Einzig die zahlenmäßige Zusammensetzung von Protonen und Neutronen im Atomkern entscheidet, welches chemische Element wir letztendlich vor uns haben.

Im Normalzustand ist ein Atom nach außen ladungsneutral – es besitzt genauso viele Protonen wie Elektronen. Weicht die Zahl der Elektronen von der Zahl der Protonen ab, so sprechen wir von einem *Ion*. Ein Ion ist stets negativ oder positiv geladen, je nachdem, ob die Anzahl der Elektronen die Anzahl der Protonen übersteigt oder umgekehrt.

Abbildung 2.2 demonstriert den schematischen Aufbau eines Heliumatoms im Bohr'schen Atommodell. Während je zwei Protonen und Neutronen den *Atomkern* bilden, befinden sich die beiden Elektronen in der *Atomhülle*. Diese besteht aus mehreren Schalen, auf denen sich die Elektronen um den Kern bewegen. Beachten Sie, dass die Skizze in Abbildung 2.2 bei weitem nicht maßstabsgetreu gezeichnet ist. In Wirklichkeit ist der Radius der Atomhülle rund 10.000 mal größer als der des Kerns. Trotzdem trägt der Atomkern fast die komplette Masse eines Atoms. Obwohl Protonen und Neutronen eine unglaublich kleine Masse von  $1\,6725 \times 10^{-24}$  g bzw.  $1\,6748 \times 10^{-24}$  g besitzen, sind sie immer noch knapp 2000 mal schwerer als ein Elektron.

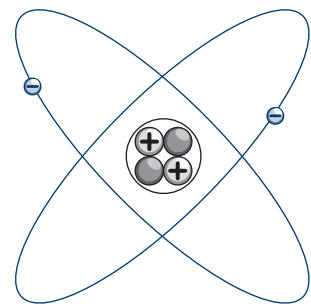
Ein wesentliches Merkmal des Bohr'schen Atommodells betrifft die Abstände, in denen einzelne Elektronen den Atomkern umkreisen können. Anders als z. B. im Rutherford'schen Modell sind diese nicht beliebig. Wie weiter oben angedeutet, wird der Atomkern durch mehrere Schalen eingehüllt, auf denen sich die verschiedenen Elektronen bewegen. Auf welcher Schale sich ein einzelnes Elektron befindet, wird einzig und alleine durch sein Energieniveau bestimmt. Elektronen auf den inneren Schalen besitzen ein niedrigeres, Elektronen auf den äußeren Schalen ein höheres Niveau. Das Energieniveau eines Elektrons ist keine kontinuierliche Größe und kann nur ganz bestimmte, diskrete Werte annehmen. Dies ist der Grund, warum sich ein Elektron immer nur auf einer bestimmten Schale, nie jedoch dazwischen befinden kann.

Zur besseren Unterscheidung wurden die verschiedenen Schalen im Bohr'schen Atommodell mit Buchstaben versehen. Die innerste wird als K-Schale, die zweite als L-Schale, die dritte als M-Schale usw. bezeichnet. Die verschiedenen Schalen eines Atoms unterscheiden sich in der Anzahl der Elektronen, die sich zur gleichen Zeit darauf befinden

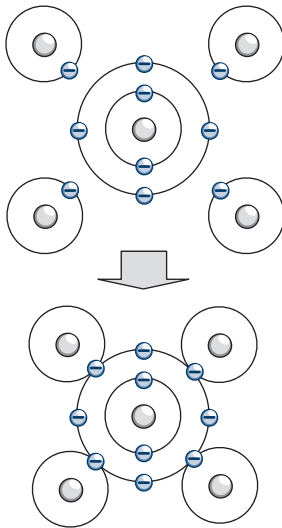


Niels Bohr (1885 – 1962)  
Albert Einstein (1879 – 1955)

**Abbildung 2.1:** Niels Bohr (links) im Zwiegespräch mit Albert Einstein im Jahre 1925.



**Abbildung 2.2:** Das Heliumatom. Im Kern befinden sich je zwei Protonen und Neutronen, die im Bohr'schen Atommodell von zwei Elektronen umkreist werden.



**Abbildung 2.3:** Vier Wasserstoffatome und ein Kohlenstoffatom im Bohr'schen Atommodell. Über die freien Valenzelektronen vereinigen sich die Atome zu einem stabilen Methanmolekül ( $\text{CH}_4$ ).

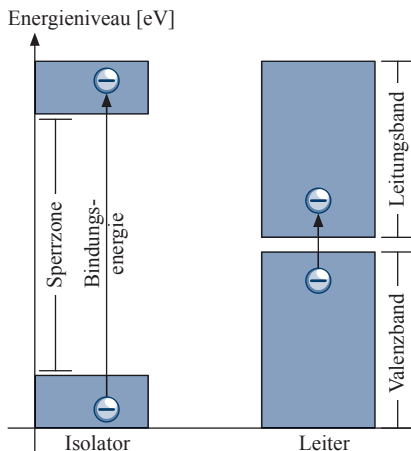
können. So kann die K-Schale nur 2, die L-Schale 8 und die M-Schale bereits 18 Elektronen aufnehmen.

Abbildung 2.3 demonstriert die Elektronenanordnung am Beispiel von Kohlenstoff- und Wasserstoffatomen. In beiden ist die äußere Schale nicht vollständig besetzt. Die Elektronen auf der ungesättigten Schale werden *Valenzelektronen* genannt und bestimmen maßgeblich, wie sich ein Atom im Zuge einer chemischen Reaktion verhält. Atome sind stets bestrebt, den energetisch ärmsten Zustand anzunehmen und werden demnach versuchen, ihre äußeren Schalen zu vervollständigen. Im Falle von Kohlenstoff und Wasserstoff führt dies dazu, dass jeweils vier Wasserstoffatome ihr freies Valenzelektron mit einem Kohlenstoffatom teilen und diesem dadurch zu einer gesättigten Schale verhelfen. Als Ergebnis entsteht ein Methan-Molekül (Erdgas,  $\text{CH}_4$ ). Wie bei allen anderen chemischen Reaktionen auch, bleiben die inneren Atomshalen durch diese Vorgänge gänzlich unberührt.

Weiter oben haben wir herausgearbeitet, dass einzig und allein das Energieniveau bestimmt, auf welcher Schale sich ein Elektron befindet. Im Umkehrschluss bedeutet dieses Ergebnis, dass ein Elektron durch die Aufnahme oder die Abgabe von Energie zwischen den Schalen hin- und herwechselt. Auf atomarer Ebene finden diese Vorgänge in der Tat fortwährend statt. Elektronen, die z. B. aufgrund thermischer Erhitzung Energie aufnehmen, bewegen sich in Richtung der äußeren Schalen. Wird ein gewisses Energieniveau überschritten, so verliert das Elektron gänzlich seine Bindung und kann sich frei im Atomverbund bewegen. Aus dem ehemaligen Valenzelektron ist jetzt ein freies *Leitungselektron* geworden.

Um ein Elektron aus dem Atom zu lösen, muss die sogenannte *Bindungsenergie* aufgebracht werden. Diese unterscheidet sich erheblich zwischen den verschiedenen chemischen Substanzen. In klassischen Isolatoren wie z. B. Hartgummi, ist die aufzubringende Energiemenge so groß, dass selbst bei hohen Temperaturen nur wenige Elektronen diesen Sprung schaffen. Kurzum: Ein Stromfluss kommt so gut wie nicht zustande. In elektrischen Leitern wie z. B. Kupfer oder Silber, reicht hingegen eine sehr geringe Energiemenge aus, um freie Leitungselektronen zu erzeugen.

Besonders deutlich wird der Unterschied zwischen Isolatoren und Leitern im *Bändermodell*. Tragen wir die möglichen Energieniveaus eines Elektrons, wie in Abbildung 2.4 gezeigt, auf der y-Achse eines Diagramms auf, so lässt die Unterscheidung in Leitungs- und Valenzelektronen eine natürliche Zweiteilung entstehen. Elektronen hohen Energieniveaus befinden sich im *Leitungsband*, während sich Elektronen nied-



**Abbildung 2.4:** Bändermodell. Die verschiedenen Energieniveaus von Leitungs- und Valenzelektronen führen zu einer natürlichen Zweiteilung auf der y-Achse.

rigen Niveaus im *Valenzband* befinden. Je mehr Bindungsenergie für die Freisetzung eines Elektrons aufgebracht werden muss, desto weiter klaffen das Leitungs- und das Valenzband auseinander. Zwischen beiden Bändern entsteht eine Sperrzone, die keine einnehmbaren Energieniveaus enthält und mit zunehmender Größe von immer weniger Elektronen überwunden werden kann. Wie zu erwarten, zeichnen sich gute Isolatoren durch eine große Sperrzone aus, während das Leitungs- und das Valenzband in guten Leitern fast nahtlos ineinander übergehen.

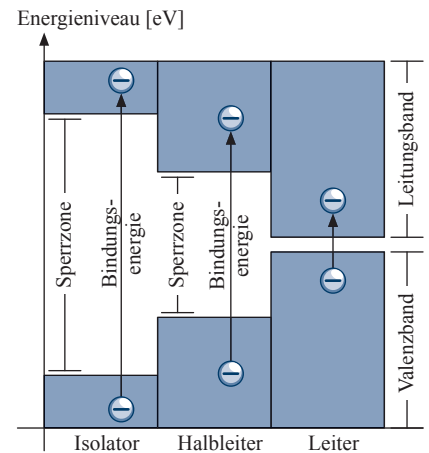
### 2.1.2 Reine Halbleiter

*Halbleiter* sind spezielle Festkörper, die gleichsam als Isolator wie auch als elektrischer Leiter auftreten können. Ihre spezielle Eigenschaft haben diese Stoffe der Energiemenge zu verdanken, die zur Freisetzung eines Valenzelektrons aufgebracht werden muss. Die Bindungsenergie liegt in einem Bereich, der groß genug ist, um das Material bei geringen Temperaturen zu einem Isolator werden zu lassen, gleichzeitig aber klein genug ist, um bei mäßigen Temperaturen von einer größeren Anzahl von Elektronen überwunden zu werden. So reicht z. B. für den Halbleiter Germanium eine Temperatur von ca. 50 C aus, um eine gute elektrische Leitfähigkeit zu erreichen.

Abbildung 2.5 zeigt die Einordnung von Halbleitern in das Bändermodell. Die eingezeichnete Sperrzone befindet sich zwischen derer von Isolatoren und jener von elektrischen Leitern. Eine exakte Einteilung von Materialien in Leiter, Halbleiter und Isolatoren ist über deren *spezifischen Widerstand*  $\rho$  möglich (vgl. Tabelle 2.1). Materialien mit einem Wert kleiner als  $10^{-6} \Omega\text{m}$  bzw. größer als  $10^{10} \Omega\text{m}$  gelten als Leiter bzw. als Isolator. Halbleiter besitzen einen spezifischen Widerstand, der zwischen diesen beiden Werten liegt.

Für den Bau elektronischer Schaltungen spielen insbesondere die Halbleiterelemente Silizium (*Si*) und Germanium (*Ge*) eine Rolle. Beide Elemente sind so beschaffen, dass ihre Leitfähigkeit durch äußere Einflüsse vergleichsweise einfach beeinflusst werden kann. Silizium steht an Position 14 des Periodensystems und ist nach dem Sauerstoff das zweithäufigste Element in der Erdkruste. Die 14 Elektronen eines Siliziumatoms verteilen sich auf insgesamt 3 Schalen. Die innerste ist mit 2 und die zweite mit 8 Elektronen vollständig gefüllt. Die M-Schale ist mit 4 Valenzelektronen dagegen ungesättigt.

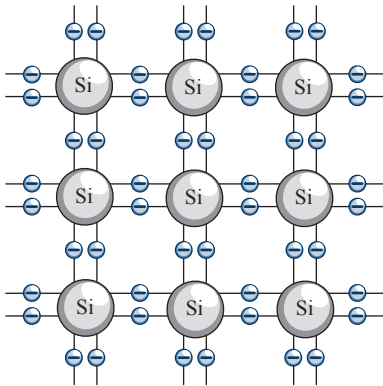
Im Verbund ordnen sich die Siliziumatome, wie in Abbildung 2.6 skizziert, in Form eines *Kristallgitters* an. Um jedes Atom gruppieren sich 4



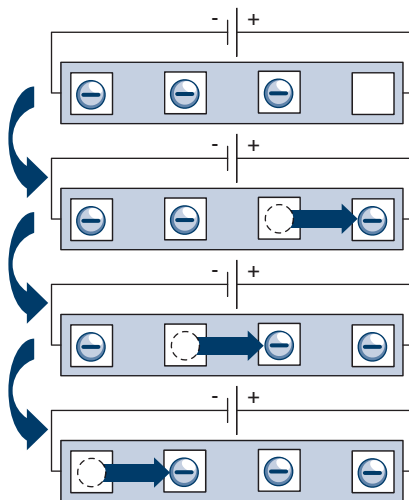
**Abbildung 2.5:** Einordnung von Halbleitern im Bändermodell

Isolatoren	
Material	Widerstand
Hartgummi	$10^{16} \Omega\text{m}$
Bernstein	$10^{14} \Omega\text{m}$
Halbleiter	
Material	Widerstand
Silizium (rein)	$10^2 \Omega\text{m}$
Germanium (rein)	$10^0 \Omega\text{m}$
Germanium (dotiert)	$10^{-4} \Omega\text{m}$
Leiter	
Material	Widerstand
Platin	$10^{-7} \Omega\text{m}$
Silber	$10^{-8} \Omega\text{m}$

**Tabelle 2.1:** Über den spezifischen Widerstand lassen sich Materialien in Isolatoren, Halbleiter und Leiter klassifizieren.



**Abbildung 2.6:** Struktur des Siliziumkristalls. Jedes Atom ist von 4 weiteren Atomen umgeben, die über jeweils zwei gemeinsam genutzte Valenzelektronen eine stabile Verbindung herstellen.



**Abbildung 2.7:** Eigenleitung im Halbleiterkristall. Die freigesetzten Leitungselektronen richten sich im elektrischen Feld aus und wandern in Richtung der positiven Spannungsquelle. Die gleichzeitig entstehenden Elektronenlöcher bewegen sich in entgegengesetzter Richtung auf den Minuspol zu.

weitere, die über je 2 gemeinsam genutzte Valenzelektronen eine feste Bindung eingehen und die M-Schalen jedes Siliziumatoms damit auf effektiv 8 Elektronen auffüllen. Aufgrund der regulären Struktur der Atomanordnung sprechen wir in diesem Zusammenhang auch von einem *Einkristall (single crystal)*. Anders als in der schematischen Zeichnung suggeriert, ordnen sich die einzelnen Atome in der Realität dreidimensional an. Bei der skizzierten Struktur handelt es sich um ein vereinfachtes zweidimensionales Modell, das für unsere Anschauungszwecke jedoch völlig ausreichend ist.

Die Struktur des Siliziumkristalls entspricht jener des Diamanten. Beide Materialien unterscheiden sich jedoch erheblich in der Bindungsenergie, die zum Heraustrennen eines Elektrons benötigt wird. Um ein Elektron des Siliziumkristalls in das Leitungsband zu heben, ist eine Energie von ca. 1,1 eV ausreichend. In der wesentlich stabileren Diamantstruktur muss mit 6 eV bereits mehr als die fünffache Energie aufgebracht werden, um ein Elektron von einem einzelnen Kohlenstoffatom zu trennen. Diese hohe Energiemenge macht den Diamant zu einem exzellenten Isolator – wenn auch zu einem sehr kostspieligen.

Durch die *Brown'sche Molekularbewegung* besitzt jedes Elektron eine gewisse Schwingungsenergie, die bei Raumtemperatur im Bereich von 0,025 eV liegt. Der exakte Wert variiert zwischen den einzelnen Atomen, so dass es dem ein oder anderen Elektron immer wieder gelingt, die Bindungsenergie zu überwinden und in das Leitungsband zu gelangen. Steigt oder sinkt die Temperatur, so nimmt mit der thermischen Energie auch die Elektronendichte im Leitungsband kontinuierlich zu bzw. ab. In einem Siliziumkristall befinden sich bei 50 °C bereits  $10^{10}$  freie Elektronen pro  $\text{cm}^3$  im Leitungsband. Verglichen mit den klassischen elektrischen Leitern ist diese Elektronendichte trotzdem gering. Hier kommt im Durchschnitt auf jedes der ca.  $10^{22}$  Atome pro  $\text{cm}^3$  ein freies Leitungselektron.

Die Freisetzung von Elektronen ist der Schlüssel für die elektrische Leitfähigkeit des Halbleiterkristalls. Für jedes herausgetrennte Elektron entsteht eine *Bindungslücke*, die auch als *Elektronenloch* oder als *Defektelektron* bezeichnet wird. Die gleichzeitige Entstehung von Leitungselektronen und Löchern wird mit dem Begriff der *Paarbildung* umschrieben. Diese hat einen entscheidenden Einfluss auf die Ladungsverteilung innerhalb des Kristalls. Wechselt ein Elektron in das Leitungsband, so hinterlässt es ein positiv geladenes Ion, das seinerseits anziehend auf die umliegenden Elektronen wirkt. Entsprechend häufig wird das entstandene Loch entweder durch ein freies Leitungselektron oder durch ein benachbart freigesetztes Elektron aufgefüllt. Wir sprechen in diesem Fall von einer *Rekombination*.



Legen wir an den Halbleiterkristall ein elektrisches Feld an, so findet ein gerichteter Stromfluss statt. Wie in Abbildung 2.7 gezeigt, werden die freigesetzten Elektronen durch das elektrische Feld in Richtung des Pluspols getrieben. Da wir die Elektronenlöcher ohne Probleme als positive Ladungsträger interpretieren können, erzeugen diese einen *Löcherstrom*, der dem *Elektronenstrom* entgegengesetzt ist. Obwohl die positiv geladenen Ionen in Wirklichkeit fest in das Kristallgitter integriert sind und sich selbst nicht von der Stelle bewegen, ist es für die Anschauung vollkommen korrekt, sich den Löcherstrom als einen Fluss positiv geladener Ladungsträger vorzustellen (vgl. Abbildung 2.8).

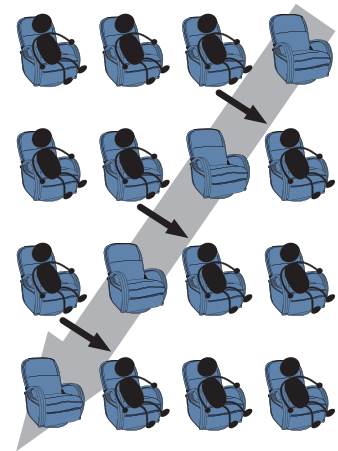
Die Fähigkeit eines Halbleiters, über die Bildung freier Elektronen und Defektelektronen einen Stromfluss zu erzeugen, wird als *Eigenleitung* bezeichnet. Wie oben angedeutet, nimmt die Paarbildung mit zunehmender thermischer Energie stark zu, so dass Halbleiter mit wachsender Temperatur zu einem immer besseren Leiter werden. Von den klassischen elektrischen Leitern kennen wir diese Eigenschaft nicht. Hier stehen sich die Leitungselektronen bei zunehmender Molekularbewegung gegenseitig wie Hindernisse im Weg und sorgen dafür, dass die Leitfähigkeit mit zunehmenden Temperaturen kontinuierlich sinkt.

### 2.1.3 Dotierte Halbleiter

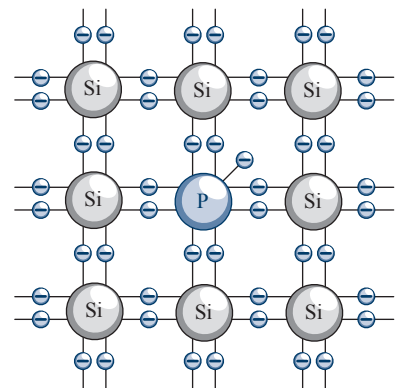
In einem reinen Siliziumkristall kommen Elektronen und Defektelektronen in gleicher Anzahl vor. Wird dieses Gleichgewicht durch eine gezielte Verunreinigung des Trägermaterials gestört, so lässt sich die elektrische Leitfähigkeit des Kristalls erheblich verbessern. Der Vorgang der Verunreinigung wird als *Dotierung* und die entstehende Kristallstruktur als *dotierter Halbleiter* bezeichnet.

Als erstes dotiertes Halbleitermaterial betrachten wir den *Elektronenüberschussleiter*. Dieser entsteht, indem Fremdatome in das Kristallgitter eingebaut werden, die über ein zusätzliches Valenzelektron verfügen. Wie ein solches Gitter aussehen kann, demonstriert Abbildung 2.9 am Beispiel eines mit Phosphor verunreinigten Siliziumkristalls. Das Phosphoratom besitzt 5 Valenzelektronen in der M-Schale und damit eines mehr als das Siliziumatom (vgl. Abbildung 2.10). Von den 5 Valenzelektronen werden nur 4 für den Einbau in das Kristallgitter benötigt. Das überschüssige fünfte Elektron ist nur schwach eingebunden. Geringe Energiemengen reichen aus, um es zu lösen und zu einem freien Ladungsträger werden zu lassen.

Aufgrund ihrer elektronenspendenden Funktion werden die künstlich eingebauten Phosphoratome als *Donatoren* bezeichnet. Da in einem sol-



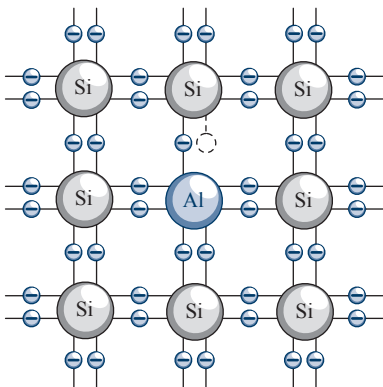
**Abbildung 2.8:** Eigenleitung einmal anders. Die Bewegung der Ladungsträger lässt sich bildlich mit dem Auffüllen einer Kinoreihe vergleichen. Während die Besucher (Elektronen) von links nach rechts durch die Reihe rücken, scheinen die freien Sitze (Elektronenlöcher) trotz ihrer stationären Montage von rechts nach links zu wandern.



**Abbildung 2.9:** Struktur eines Elektronenüberschussleiters (n-Leiter). Durch den gezielten Einbau von Phosphoratomen werden zusätzliche Valenzelektronen in das Siliziumgitter eingebracht. Diese können sich nahezu ungehindert durch die Kristallstruktur bewegen.

IA	IIA	IIIB	IVB	VB	VIB	VIIIB	VIIIIB			IB	IIB	IIIA	IVA	VA	VIA	VIIA	VIIIA
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1 H																	2 He
3 Li	4 Be											5 B	6 C	7 N	8 O	9 F	10 Ne
11 Na	12 Mg											13 Al	14 Si	15 P	16 S	17 Cl	18 Ar
19 K	20 Ca	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr
37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe
55 Cs	56 Ba		72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl	82 Pb	83 Bi	84 Po	85 At	86 Rn
87 Fr	88 Ra		104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Uub						
			57 La	58 Ce	59 Pr	60 Nd	61 Pm	62 Sm	63 Eu	64 Gd	65 Tb	66 Dy	67 Ho	68 Er	69 Tm	70 Yb	71 Lu
			89 Ac	90 Th	91 Pa	92 U	93 Np	94 Pu	95 Am	96 Cm	97 Bk	98 Cf	99 Es	100 Fm	101 Md	102 No	103 Lr

**Abbildung 2.10:** Das Periodensystem. Die Elemente der dritten, vierten und fünften Hauptgruppe spielen für die Halbleitertechnik eine zentrale Rolle. Um die gewünschte elektrische Leitfähigkeit zu erreichen, werden die Trägersubstanzen Silizium (Si) bzw. Germanium (Ge) mit Elementen der dritten bzw. der fünften Hauptgruppe gezielt verunreinigt (dotiert). Die Dotierung mit Bor (B), Indium (In), Aluminium (Al) oder Gallium (Ga) lässt ein p-Gebiet entstehen, während die Verunreinigung mit Phosphor (P), Arsen (As), Antimon (Sb) oder Bismut (Bi) ein n-Gebiet erzeugt.



**Abbildung 2.11:** Struktur eines Elektronenmangelleiters (p-Leiter). Durch den gezielten Einbau von Aluminiumatomen entstehen künstliche Elektronenlöcher, die wie positive Ladungsträger wirken.

chen Halbleiterkristall mehr Elektronen als Elektronenlöcher vorhanden sind, spielen erstere die Rolle des *Majoritätsträgers* und letztere die des *Minoritätsträgers*. Aufgrund des negativen Vorzeichen des Majoritätsträgers sprechen wir im Falle eines Elektronenüberschussleiters auch von einem *n-Leiter*.

Wird der Siliziumträger nicht mit Elementen aus der fünften, sondern mit Elementen aus der dritten Hauptgruppe des Periodensystems dotiert, so entsteht ein *Elektronenmangelleiter*. Die in Frage kommenden Elemente Bor, Indium, Aluminium oder Gallium besitzen mit 3 Valenzelektronen in der M-Schale ein Elektron weniger als der Silizium- oder der Germaniumträger. Wie in Abbildung 2.11 demonstriert, erzeugt der Einbau der Fremdatome künstliche Löcher innerhalb des Kristallgitters. Diese können Elektronen an sich ziehen und werden aufgrund dieser Eigenschaft als *Akzeptoren* bezeichnet. Anders als im Falle des n-Leiters spielen jetzt nicht mehr die Elektronen, sondern die Elektronenlöcher dank ihrer zahlenmäßigen Überlegenheit die Rolle des Majoritätsträgers. Gemäß ihrer positiven Ladung sprechen wir jetzt von einem *p-Leiter*.

## 2.2 Integrierte Schaltelemente

### 2.2.1 Halbleiterdioden

*Dioden* sind spezielle Schaltelemente, die den Stromfluss richtungsabhängig begrenzen. Während sie sich in *Durchlassrichtung* neutral verhalten, wirken sie in *Sperrrichtung* als Isolator.

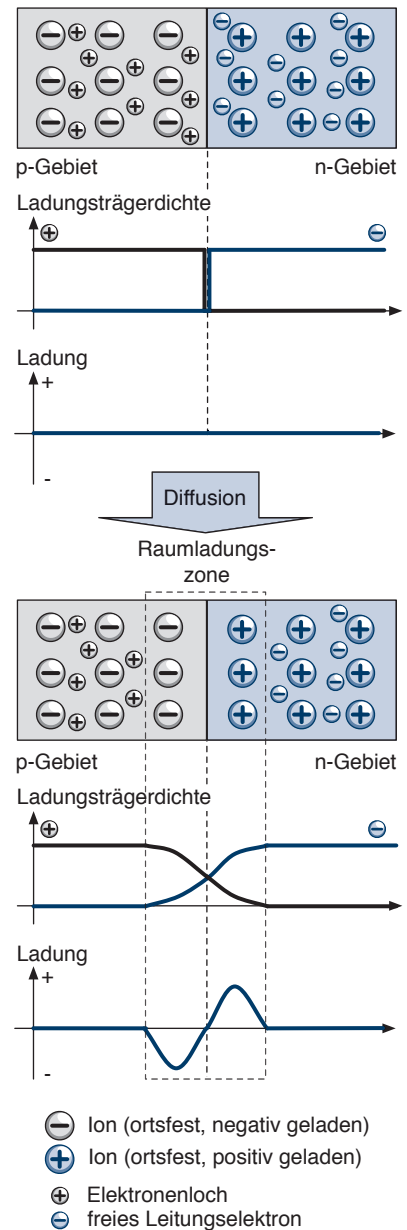
Wie in Abbildung 2.12 skizziert, lässt sich das Schaltverhalten einer Diode durch den Zusammenschluss zweier komplementär dotierter Halbleiterkristalle erzeugen. Um die physikalischen Vorgänge innerhalb der Kristallverbindung im Detail zu verstehen, betrachten wir zunächst, wie sich der Zusammenschluss eines p-Gebiets und eines n-Gebiets im Ruhezustand verhält. Danach werden wir erarbeiten, wie sich das Anlegen einer Spannung auf die Leitfähigkeit des Kristallverbunds auswirkt.

Die elektrischen Eigenschaften der Halbleiterdiode werden maßgeblich durch die physikalischen Vorgänge an der Grenzschicht des p-Gebiets und des n-Gebiets – dem *pn-Übergang* – bestimmt. Während das p-Gebiet aufgrund seiner speziellen Dotierung viele Elektronenlöcher aufweist, befindet sich innerhalb des n-Gebiets eine große Anzahl freier Leitungselektronen. Werden beide Halbleitermaterialien aneinandergesetzt, so beginnen die freien Leitungselektronen in den p-Leiter zu drängen. Auf diese Weise bildet sich um den pn-Übergang eine Grenzschicht aus, die nur noch wenige freie Ladungsträger enthält. Im n-Gebiet lässt der Elektronenabfluss positive Ionen zurück und vermehrt zudem die Anzahl negativer Ionen innerhalb des p-Gebiets. Diese stoßen die nachdrängenden Elektronen in zunehmendem Maße ab. Nach kurzer Zeit stellt sich zwischen den wirkenden Kräften ein Gleichgewicht ein und die Diffusion kommt zum Stillstand.

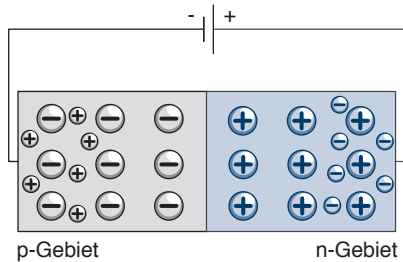
Die entstandene Zone wird als *Raumladungszone* bezeichnet. Zusammenfassend zeichnet sich diese durch zwei wesentliche Eigenschaften aus. Zum einen wird sie durch die Abstoßungskräfte der entstehenden Ionen räumlich begrenzt. Zum anderen weist sie aufgrund der sich zahlreich ereignenden Rekombinationen nur noch wenige freie Ladungsträger auf und besitzt dadurch eine isolierende Wirkung.



Im Bereich eines pn-Übergangs führt die Rekombination von Elektronen und Elektronenlöchern zu einer Verarmung der freien Ladungsträger. Die entstehende Raumladungszone besitzt eine isolierende Wirkung.



**Abbildung 2.12:** Aufbau einer Halbleiterdiode. Durch die Diffusion der Ladungsträger bildet sich im pn-Gebiet die *Raumladungszone* aus. Diese wirkt wie eine isolierende Grenzschicht.



**Abbildung 2.13:** Anlegen einer Spannung in Sperrrichtung. Der Minuspol wird mit der p-Schicht und der Pluspol mit der n-Schicht verbunden. Die Ladungsträger werden in Richtung der Spannungspole weggezogen und bewirken eine Vergrößerung der Sperrschicht. Der Stromfluss wird unterbunden.

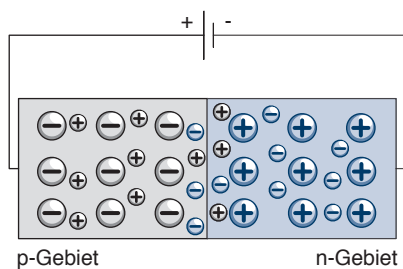
Im Folgenden wollen wir erarbeiten, wie sich die Grenzschicht verhält, wenn wir an die Halbleiterdiode eine elektrische Spannung anlegen. Wir unterscheiden zwei Fälle:

#### ■ Anlegen einer Spannung in Sperrrichtung

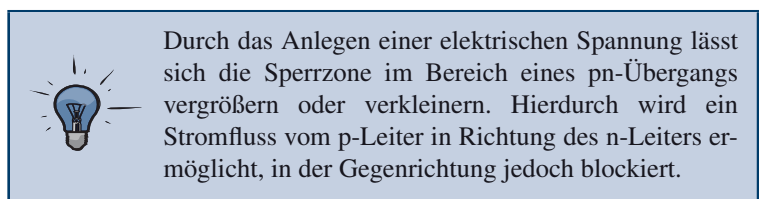
Wir schließen die Halbleiterdiode so an eine Spannungsquelle an, dass die p-Schicht mit dem Minuspol und die n-Schicht mit dem Pluspol verbunden ist (Abbildung 2.13). In diesem Fall werden die freien Elektronen des n-Leiters in Richtung des Pluspols und die Elektronenlöcher des p-Leiters in Richtung des Minuspols gezogen. Die Abwanderung der Ladungsträger führt zu einer Vergrößerung der isolierenden Grenzschicht, die jetzt von kaum einem Elektron mehr überwunden werden kann. Kurzum: Die Halbleiterdiode wirkt in der angelegten Spannungsrichtung als Isolator, der den Stromfluss effektiv unterbindet.

#### ■ Anlegen einer Spannung in Durchlassrichtung

Drehen wir die Polarität der angelegten Spannung um, indem wir die p-Schicht mit dem Pluspol und die n-Schicht mit dem Minuspol verbinden, so werden zusätzliche Elektronen in den n-Leiter und Defektelektronen in den p-Leiter gedrängt (Abbildung 2.14). Die Zunahme der freien Leitungsträger im Bereich des pn-Übergangs führt jetzt zu einer Verkleinerung der Sperrschicht. Übersteigt die angelegte Spannung eine bestimmte Schwelle, so können freie Ladungsträger in großer Anzahl in die Sperrschicht eindringen und rekombinieren. Die Halbleiterdiode verliert ihre isolierende Wirkung und ermöglichen einen nahezu ungehinderten Stromfluss.



**Abbildung 2.14:** Anlegen einer Spannung in Durchlassrichtung. Der Minuspol wird mit der n-Schicht und der Pluspol mit der p-Schicht verbunden. Die freien Ladungsträger bewegen sich aufeinander zu und rekombinieren in der Sperrschicht. Strom fließt.



## 2.2.2 Bipolartransistoren

Im vorherigen Abschnitt haben wir mit der Diode ein klassisches Halbleiterelement eingeführt und ihre Funktionsweise auf atomarer Ebene begründet. In diesem Abschnitt werden wir die Überlegungen in der eingeschlagenen Richtung vertiefen und uns dem Herzstück aller modernen Computer zuwenden: dem *Transistor*.

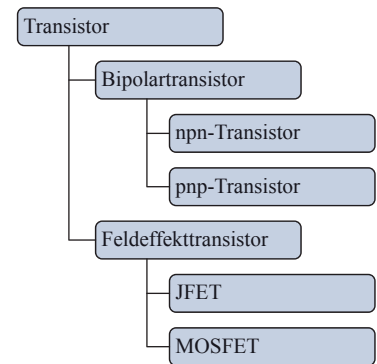
Das Wort Transistor ist ein Kunstprodukt aus den Begriffen *Transfer* und *Resistor* und bedeutet so viel wie „steuerbarer Widerstand“. In seiner grundlegenden Funktion entspricht der Transistor der in Kapitel 1 eingeführten Röhrentriode und kann dazu verwendet werden, ein elektrisches Signal zu verstärken bzw. im Falle einer digitalen Ansteuerung ein- oder auszuschalten. So sehr sich die beiden Bauelemente funktional gleichen, so unterschiedlich sind jedoch ihre physikalischen Kenndaten. Der Transistor benötigt nur einen Bruchteil der Energie einer Röhrentriode und lässt sie auch in Bezug auf die Schaltgeschwindigkeit um mehrere Größenordnungen hinter sich.

Wie in Abbildung 2.15 skizziert, lassen sich Transistoren auf unterschiedliche Weise realisieren. Im nächsten Abschnitt werden wir zunächst die verschiedenen Spielarten des Bipolartransistors einführen und anschließend zeigen, wie sich diese von den heute meistgebräuchlichen Feldeffekttransistoren unterscheiden.

## npn-Transistor

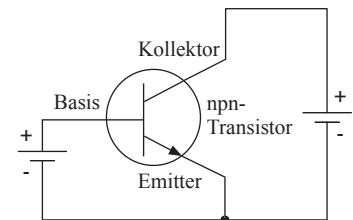
Abbildung 2.16 (oben) zeigt das Symbol und die typische Beschaltung eines npn-Transistors. Genau wie die klassische Triodenröhre besitzt der Transistor drei Anschlüsse. Der *Emitter* und der *Kollektor* dienen dem Zufluss bzw. dem Abfluss von Elektronen. Der Steueranschluss *B* wird als *Basis* bezeichnet und regelt den Stromfluss zwischen Emitter und Kollektor. Analog zur Funktionsweise der Triodenröhre besitzt der Steueranschluss eine verstärkende Wirkung: Eine geringe Änderung des Stromflusses auf der *Emitter-Basis-Strecke* bewirkt eine große Änderung des Stromflusses auf der *Emitter-Kollektor-Strecke*.

Im Folgenden wollen wir genauer betrachten, welche physikalischen Vorgänge zu diesem Verhalten führen. Werfen wir einen Blick auf den internen Aufbau eines Transistors, so werden Parallelen zu der weiter oben eingeführten Halbleiterdiode sichtbar. Schematisch unterscheidet sich der npn-Transistor einzig durch das Vorhandensein einer dritten Halbleiterschicht. Wie in Abbildung 2.16 (unten) gezeigt, besitzt der Transistor im Inneren einen p-dotierten Kern, der von zwei n-dotierten Schichten eingerahmt wird. Die Basis ist mit dem p-Leiter verbunden, während Emitter und Kollektor an die beiden n-Leiter angeschlossen werden. Im Gegensatz zur Halbleiterdiode sind die eingesetzten Materialien unterschiedlich stark dotiert. So besitzt der p-Leiter eine deutlich niedrigere Dichte an Fremdatomen als die beiden n-Leiter. Diese unterscheiden sich untereinander ebenfalls sowohl im Grad der Dotierung als auch in ihrer Größe, so dass die Emitter- und Kollektoranschlüsse nicht miteinander vertauscht werden dürfen.

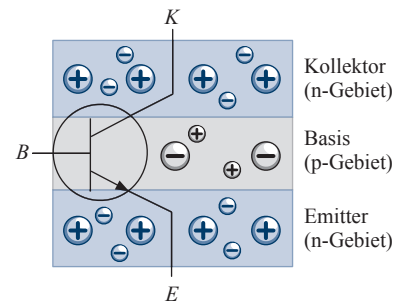


**Abbildung 2.15:** Transistoren lassen sich auf verschiedene Arten realisieren. Auf der obersten Ebene werden Bipolartransistoren und Feldeffekttransistoren unterschieden.

### ■ Schaltsymbol



### ■ Interner Aufbau



**Abbildung 2.16:** Beschaltung und Aufbau eines npn-Transistors. Die Basis wird durch ein schwach dotiertes p-Gebiet gebildet, das von zwei stark dotierten n-Gebieten (Kollektor und Emitter) eingerahmt wird.