

Inhaltsverzeichnis

1	Abstraktionen und Technologien	1
1.1	Einführung	1
1.2	Sieben große Ideen in der Computerarchitektur	9
1.3	Was sich hinter einem Programm verbirgt	12
1.4	Unter der Haube	16
1.5	Prozessorherstellung und Speichertechnologien	25
1.6	Leistung	30
1.7	Die Hürde des Stromverbrauchs	42
1.8	Der Wechsel von Einzelprozessoren zu Multiprozessoren	45
1.9	Fallstudie: Benchmarking des Intel Core i7	49
1.10	Beschleunigung: Matrixmultiplikation in Python	51
1.11	Fallstricke und Trugschlüsse	53
1.12	Schlussbetrachtungen	56
1.13	Historische Perspektiven und Literaturhinweise	58
1.14	Fragestellungen für das Selbststudium	59
1.15	Aufgaben	62
2	Befehle: Die Sprache des Rechners	71
2.1	Einführung	71
2.2	Operationen der Rechnerhardware	74
2.3	Operanden der Rechnerhardware	76
2.4	Vorzeichenbehaftete und nicht vorzeichenbehaftete Zahlen	83
2.5	Darstellung von Befehlen im Rechner	91
2.6	Logische Operationen	98
2.7	Befehle zum Treffen von Entscheidungen	101
2.8	Unterstützung von Prozeduren durch die Rechnerhardware	107
2.9	Kommunikation mit Menschen	118
2.10	Umgang mit 32-Bit-Direktoperanden und 32-Bit-Adressen	124
2.11	Parallelität und Befehle: Synchronisierung	133
2.12	Übersetzen und Starten eines Programms	136
2.13	Zusammenfassung am Beispiel eines Sortierprogramms in C	146
2.14	Felder und Zeiger im Vergleich	155
2.15	Fortgeschrittener Stoff: C-Compiler und Java-Interpreter	159
2.16	Fallstudie: ARMv7-Befehle (32 Bit)	159

2.17	Fallstudie: ARMv8-Befehle (64 Bit)	163
2.18	Fallstudie: RISC-V-Befehle	164
2.19	Fallstudie: x86-Befehle	165
2.20	Beschleunigung: Matrixmultiplikation in C	176
2.21	Fallstricke und Trugschlüsse	177
2.22	Schlussbetrachtungen	179
2.23	Historische Perspektiven und Literaturhinweise	181
2.24	Fragestellungen für das Selbststudium	182
2.25	Aufgaben	185
3	Rechnerarithmetik	197
3.1	Einführung	197
3.2	Addition und Subtraktion	197
3.3	Multiplikation	202
3.4	Division	209
3.5	Gleitkommaarithmetik	217
3.6	Parallelität und Computerarithmetik: Subwort-Parallelität	246
3.7	Fallstudie: Streaming-SIMD und Vektorerweiterungen für x86	247
3.8	Beschleunigung: Subwort-Parallelität	249
3.9	Fallstricke und Trugschlüsse	251
3.10	Schlussbetrachtungen	255
3.11	Historische Perspektiven und Literaturhinweise	259
3.12	Fragestellungen für das Selbststudium	259
3.13	Aufgaben	262
4	Der Prozessor	271
4.1	Einführung	271
4.2	Konventionen für den Entwurf von Logikschaltungen	276
4.3	Aufbau eines Datenpfades	279
4.4	Eine einfache Implementierungsmethode	288
4.5	Eine Mehrtaktimplementierung	300
4.6	Übersicht über die Technik des Pipelinings	302
4.7	Pipelining von Datenpfad und Steuerwerk	317
4.8	Datenkonflikte: Forwarding vs. Stalling	335
4.9	Steuerkonflikte	349
4.10	Ausnahmebehandlung	359
4.11	Parallelität auf Befehlsebene	366
4.12	Fallstudie: Intel Core i7 6700 und ARM Cortex-A53	382
4.13	Beschleunigung: Parallelität auf Befehlsebene	391
4.14	Fortgeschrittener Stoff: Einführung in den Schaltungsentwurf	393
4.15	Fallstricke und Trugschlüsse	393
4.16	Schlussbetrachtungen	395
4.17	Historische Perspektiven und Literaturhinweise	396
4.18	Fragestellungen für das Selbststudium	396
4.19	Aufgaben	398

5	Groß und schnell: Ausnutzung der Speicherhierarchie	417
5.1	Einführung	417
5.2	Speichertechnologien	422
5.3	Grundlagen des Cachings	429
5.4	Cache-Leistung messen und verbessern	444
5.5	Zuverlässige Speicherhierarchie	465
5.6	Virtuelle Maschinen	471
5.7	Virtueller Speicher	476
5.8	Allgemeines Schema der Speicherhierarchien	506
5.9	Steuerung eines einfachen Caches	514
5.10	Parallelität und Speicherhierarchien: Cache-Kohärenz	519
5.11	Parallelität und Speicherhierarchie: RAID	524
5.12	Fortgeschrittener Stoff: Cache-Controller	524
5.13	Speicherhierarchien von ARM Cortex-A53 und Intel Core i7	524
5.14	Beschleunigung: Cache-Blocking	529
5.15	Fallstricke und Trugschlüsse	530
5.16	Schlussbetrachtungen	537
5.17	Historische Perspektive und Literaturhinweise	538
5.18	Fragestellungen für das Selbststudium	538
5.19	Aufgaben	543
6	Parallele Prozessoren: Vom Client zur Cloud	563
6.1	Einführung	563
6.2	Warum es schwierig ist, parallele Programme zu entwickeln	565
6.3	SISD, MIMD, SIMD, SPMD und Vektor	571
6.4	Hardwareseitiges Multithreading	579
6.5	Multiprozessoren mit gemeinsam genutztem Speicher	583
6.6	Grafikprozessoren (GPUs) – Einführung	587
6.7	Domänenspezifische Architekturen	595
6.8	Cluster und Warehouse Scale Computer	599
6.9	Einführung in Multiprozessor-Netztopologien	605
6.10	Kommunikation mit der Außenwelt: Cluster	609
6.11	Multiprozessor-Benchmarks und Performanzmodelle	609
6.12	Google TPuv3 und NVIDIA Volta	620
6.13	Beschleunigung: Multiple Prozessoren	630
6.14	Fallstricke und Trugschlüsse	633
6.15	Schlussbetrachtungen	636
6.16	Historische Perspektive und Literaturhinweise	639
6.17	Fragestellungen für das Selbststudium	639
6.18	Aufgaben	641

A	Assembler, Binder und SPIM-Simulator	657
A.1	Einführung	657
A.2	Assembler	664
A.3	Binder	672
A.4	Laden	673
A.5	Speichernutzung	674
A.6	Prozeduraufrufkonvention	676
A.7	Ausnahmen und Interrupts	686
A.8	Ein- und Ausgabe	690
A.9	SPIM	693
A.10	MIPS R2000 Assemblersprache	697
A.11	Schlussbetrachtungen	732
A.12	Aufgaben	733
B	Grundlagen des Logikdesigns	737
B.1	Einführung	737
B.2	Gatter, Wahrheitstabellen und logische Gleichungen	738
B.3	Kombinatorische Logik	742
B.4	Hardwarebeschreibungssprachen	754
B.5	Konstruktion einer einfachen Arithmetikeinheit	759
B.6	Schnellere Addition: Carry-Lookahead-Addierer	771
B.7	Taktsignale	781
B.8	Speicherelemente: Flip-Flops, Latches und Register	784
B.9	Speicherelemente: SRAMs und DRAMs	792
B.10	Endliche Automaten	802
B.11	Verfahren der Takterzeugung	807
B.12	Field Programmable Devices (FPD)	812
B.13	Schlussbetrachtungen	814
B.14	Aufgaben	814
	Fachbegriffe Deutsch – Englisch	825
	Englisch – Deutsch	825
	Deutsch – Englisch	828
	Stichwortverzeichnis	831