

Inhaltsverzeichnis

1	Grundlegende Ideen, Technologien und Komponenten	1
1.1	Einführung	1
1.1.1	Klassen der Computeranwendungen und deren Eigenschaften	3
1.1.2	Was Sie in diesem Buch lernen können	5
1.2	Das verbirgt sich hinter einem Programm	9
1.3	Unter der Gehäuseabdeckung	12
1.3.1	Anatomie einer Maus	14
1.3.2	Ein Blick hinter die Mattscheibe (<i>computer display</i>)	16
1.3.3	Öffnen des Gehäuses	17
1.3.4	Ein sicherer Ort für Daten	21
1.3.5	Die Kommunikation mit anderen Computern	24
1.3.6	Technologien für die Herstellung von Prozessoren und Speichern	25
1.4	Leistung	27
1.4.1	Leistung definieren	27
1.4.2	Leistung messen	30
1.4.3	CPU-Leistung und ihre Faktoren	32
1.4.4	Befehlsleistung	33
1.4.5	Die klassische Gleichung für die CPU-Leistung	35
1.5	Die Hürde des Stromverbrauchs	38
1.6	Eine grundlegende Veränderung: Der Wechsel von Einzelprozessoren zu Multiprozessoren	41
1.7	Fallstudien: Herstellung und Benchmarking des AMD Opteron X4	45
1.7.1	SPEC CPU-Benchmark	48
1.7.2	SPEC-Leistungs-Benchmark	50
1.8	Fallstricke und Fehlschlüsse	52
1.9	Schlussbetrachtungen	55
1.10	Historische Perspektiven und Literaturhinweise	57
1.11	Aufgaben	58

2	Befehle: Die Sprache des Rechners	73
2.1	Einführung	73
2.2	Operationen der Rechnerhardware	74
2.3	Operanden der Rechnerhardware	78
2.3.1	Speicherooperanden	79
2.3.2	Konstante oder Direktoperanden	83
2.4	Vorzeichenbehaftete und nicht vorzeichenbehaftete Zahlen . .	84
2.5	Darstellung von Befehlen im Rechner	90
2.5.1	Die Felder im MIPS-Befehlsformat	92
2.6	Logische Operationen	97
2.7	Befehle zum Treffen von Entscheidungen	100
2.7.1	Schleifen	102
2.7.2	Case-/switch-Anweisung	105
2.8	Unterstützung von Prozeduren durch die Rechnerhardware . .	106
2.8.1	Verwendung weiterer Register	108
2.8.2	Geschachtelte Prozeduren	110
2.8.3	Zuordnen von Speicherplatz für neue Daten im Keller	113
2.8.4	Zuordnen von Speicherplatz für neue Daten auf der Halde	114
2.9	Kommunikation mit Menschen	116
2.9.1	Zeichen und Zeichenfolgen in Java	120
2.10	Umgang mit 32-Bit-Direktoperanden und 32-Bit-Adressen . .	121
2.10.1	32-Bit-Direktoperanden	122
2.10.2	Adressbildung bei Verzweigungen und Sprüngen . . .	123
2.10.3	MIPS-Adressierungsarten – eine Übersicht	126
2.10.4	Entschlüsseln der Maschinensprache	126
2.11	Parallelismus und Befehle: Synchronisierung	130
2.12	Übersetzen und Starten eines Programms	133
2.12.1	Compiler	134
2.12.2	Assembler	135
2.12.3	Binder	136
2.12.4	Lader	139
2.12.5	Dynamisch gebundene Bibliotheken (DLLs, Dynamically Linked Libraries)	140
2.12.6	Starten eines Java-Programms	142
2.13	Zusammenfassung am Beispiel eines Sortierprogramms in C .	144
2.13.1	Die Prozedur swap	144
2.13.2	Die Prozedur sort	146
2.14	Felder und Zeiger im Vergleich	152
2.14.1	Die Version von clear mit Feldern	153
2.14.2	Die Version von clear mit Zeigern	154
2.14.3	Vergleich der beiden Versionen von clear	155
2.15	Fortgeschrittener Stoff: C-Compiler und Java-Interpreter . . .	156
2.16	Fallstudien: ARM-Befehle	156
2.16.1	Adressierungsmodi	157
2.16.2	Vergleichen und bedingte Verzweigung	158
2.16.3	Spezielle Funktionen von ARM	159

2.17	Fallstudien: IA-32-Befehle	160
2.17.1	Die Entwicklung des Intel x86	161
2.17.2	x86-Integer-Operationen	166
2.17.3	x86-Befehlscodierung	169
2.17.4	x86 – Schlussbetrachtung	170
2.18	Fallstricke und Fehlschlüsse	171
2.19	Schlussbetrachtungen	173
2.20	Historische Perspektiven und Literaturhinweise	175
2.21	Aufgaben	175
3	Rechnerarithmetik	213
3.1	Einführung	213
3.2	Addition und Subtraktion	213
3.2.1	Multimedia-Arithmetik	216
3.2.2	Zusammenfassung	218
3.3	Multiplikation	219
3.3.1	Sequenzielle Version des Multiplikationsalgorithmus und der Multiplikationshardware	220
3.3.2	Multiplikation mit Vorzeichen	223
3.3.3	Schnellere Multiplikation	224
3.3.4	Multiplikation bei MIPS	225
3.3.5	Zusammenfassung	225
3.4	Division	226
3.4.1	Ein Divisionsalgorithmus und eine Divisionshardware	227
3.4.2	Division mit Vorzeichen	230
3.4.3	Schnellere Division	231
3.4.4	Division bei MIPS	231
3.4.5	Zusammenfassung	233
3.5	Gleitkommaarithmetik	234
3.5.1	Gleitkommadarstellung	235
3.5.2	Addition von Gleitkommazahlen	240
3.5.3	Multiplikation von Gleitkommazahlen	243
3.5.4	Gleitkommabefehle im MIPS-Befehlssatz	249
3.5.5	Genaue Arithmetik	256
3.5.6	Zusammenfassung	258
3.6	Parallelismus und Computerarithmetik: Assoziativität	260
3.7	Fallstudie: Gleitkommaarithmetik in der x86-Architektur	262
3.7.1	Die x86-Gleitkommaarchitektur	262
3.7.2	Die SSE2-Gleitkommaarchitektur von Intel (Streaming SIMD Extension 2)	263
3.8	Fallstricke und Fehlschlüsse	265
3.9	Schlussbetrachtungen	269
3.10	Historische Perspektiven und Literaturhinweise	273
3.11	Aufgaben	274

4	Der Prozessor: Datenpfad und Steuerwerk	289
4.1	Einführung	289
4.1.1	Eine einfache MIPS-Implementierung	290
4.2	Konventionen für den Entwurf von Logikschaltungen	294
4.3	Aufbau eines Datenpfads	297
4.3.1	Entwurf eines einfachen Datenpfads	302
4.4	Eine einfache Implementierungsmethode	306
4.4.1	Die ALU-Steuerung	306
4.4.2	Entwurf der Hauptsteuereinheit	308
4.4.3	Warum eine Eintaktausführung heute nicht verwendet wird	316
4.5	Übersicht über die Technik des Pipelinings	318
4.5.1	Entwurf von Befehlssätzen für das Pipelining	323
4.5.2	Pipeline-Hemmnisse	324
4.5.3	Zusammenfassung: Die Technik des Pipelinings	332
4.6	Pipelining des Datenpfads	334
4.6.1	Grafische Darstellung von Pipelines	344
4.6.2	Pipelining der Steuerung	348
4.7	Pipeline-Hemmnisse durch Datenabhängigkeit und die Tech- nik des Forwarding	353
4.7.1	Pipelinehemmnisse durch Datenabhängigkeit und Pi- pelineverzögerungen	362
4.8	Pipelinehemmnisse durch Kontrollflussabhängigkeiten	366
4.8.1	Annahme, dass Sprünge nicht ausgeführt werden	366
4.8.2	Reduktion der Verzögerung durch Sprünge	367
4.8.3	Dynamische Sprungvorhersage	369
4.8.4	Pipeline – Eine Zusammenfassung	374
4.9	Unterbrechungen	375
4.9.1	Die Verarbeitung von Unterbrechungen in der MIPS- Architektur	376
4.9.2	Unterbrechungen in einer Pipelined-Implementierung	378
4.10	Parallelismus und erweiterte Befehlsebenen-Parallelität	382
4.10.1	Das Prinzip der Spekulation	384
4.10.2	Statische Mehrfachzuordnung	385
4.10.3	Prozessoren mit dynamischer Mehrfachzuordnung	390
4.10.4	Energieeffizienz und fortgeschrittenes Pipelining	395
4.11	Fallstudie: Die Pipeline des AMD Opteron X4 (Barcelona)	396
4.12	Themenerweiterung: Eine Einführung in den Schaltungsent- wurf unter Verwendung einer Hardwaredesignsprache, um eine Pipeline zu beschreiben, und weitere Pipeline-Abbildungen	399
4.13	Fallstricke und Fehlschlüsse	399
4.14	Schlussbetrachtungen	401
4.15	Historische Perspektiven und Literaturhinweise	402
4.16	Aufgaben	402

5	Speicherhierarchie	443
5.1	Einführung	443
5.2	Caches – Grundlagen	448
5.2.1	Zugriff auf einen Cache	452
5.2.2	Verarbeitung von Cache-Fehlzugriffen	458
5.2.3	Schreiboperationen verarbeiten	459
5.2.4	Ein Beispiel-Cache: Der Intrinsicity-FastMATH-Prozessor	461
5.2.5	Design des Speichersystems zur Unterstützung von Caches	463
5.2.6	Zusammenfassung	467
5.3	Cache-Leistung messen und verbessern	468
5.3.1	Reduzierung von Cache-Fehlzugriffen durch eine flexiblere Platzierung von Blöcken	472
5.3.2	Einen Block im Cache finden	476
5.3.3	Auswahl, welcher Block ersetzt werden soll	478
5.3.4	Reduzierung des Fehlzugriffsaufwands durch Cache-Speicherhierarchien	479
5.3.5	Zusammenfassung	484
5.4	Virtueller Speicher	485
5.4.1	Eine Seite platzieren und wieder finden	489
5.4.2	Seitenfehler	491
5.4.3	Schreiboperationen	495
5.4.4	Beschleunigung der Adressübersetzung: Der TLB	496
5.4.5	Integration von virtuellem Speicher, TLBs und Caches	501
5.4.6	Implementierung von Schutzmechanismen mit einem virtuellen Speicher	503
5.4.7	Verarbeitung von TLB-Fehlzugriffen und Seitenfehlern	505
5.4.8	Zusammenfassung	511
5.5	Ein gemeinsames Gerüst für Speicherhierarchien	513
5.5.1	Frage 1: Wo kann ein Block platziert werden?	513
5.5.2	Frage 2: Wie findet man einen Block?	514
5.5.3	Frage 3: Welcher Block soll bei einem Cache-Fehlzugriff ersetzt werden?	515
5.5.4	Frage 4: Was passiert bei einer Schreiboperation?	516
5.5.5	Die drei Cs: Ein intuitives Modell für ein Verständnis des Verhaltens von Speicherhierarchien	518
5.6	Virtuelle Maschinen	520
5.6.1	Anforderungen an einen VMM (Virtual Machine Monitor)	522
5.6.2	(Fehlende) ISA-Unterstützung für virtuelle Maschinen	522
5.6.3	Schutz und ISA	523
5.7	Mit einem endlichen Automaten einen einfachen Cache steuern	525
5.7.1	Ein einfacher Cache	525
5.7.2	Endliche Automaten	526
5.7.3	FSM für eine einfache Cache-Steuerung	528

5.8	Parallelismus und Speicherhierarchien: Cache-Kohärenz . . .	530
5.8.1	Grundlegende Vorgehensweisen für die Erzwingung der Kohärenz	531
5.8.2	Snooping-Protokolle	532
5.9	Fortgeschrittener Stoff: Implementierung von Cache-Steuerun- gen	534
5.10	Fallstudien: Speicherhierarchien des AMD Opteron X4 (Bar- celona) und des Intel Nehalem	534
5.10.1	Die Speicherhierarchien des Nehalem und Opteron . .	535
5.10.2	Techniken zur Reduzierung des Fehlzugriffsaufwands	536
5.11	Fallstricke und Fehlschlüsse	539
5.12	Schlussbetrachtungen	543
5.13	Historische Perspektive und Literaturhinweise	544
5.14	Aufgaben	545
6	Sekundärspeicher, Netzwerke und andere Peripheriegeräte	563
6.1	Einführung	563
6.2	Verlässlichkeit, Zuverlässigkeit und Verfügbarkeit	566
6.3	Festplattenspeicher und Zuverlässigkeit	568
6.4	Flash-Speicher	574
6.5	Verbindungen von Prozessoren, Speicher und Ein-/Ausgabege- räten	576
6.5.1	Verbindungsgrundlagen	577
6.5.2	Die Ein-/Ausgabeverbindungen der x86-Prozessoren .	579
6.6	Schnittstellen von den Ein-/Ausgabe-Geräten zu Prozessor, Speicher und Betriebssystem	581
6.6.1	Befehle an Ein-/Ausgabe-Geräte übermitteln	582
6.6.2	Kommunikation mit dem Prozessor	584
6.6.3	Interrupt-Prioritäten	585
6.6.4	Übertragung der Daten zwischen Gerät und Speicher .	587
6.6.5	DMA und das Speichersystem	589
6.7	Leistungsmaße für die Ein-/Ausgabe: Beispiele aus Festpla- tten- und Dateisystemen	590
6.7.1	Ein-/Ausgabe-Benchmarks für die Transaktionsverar- beitung	591
6.7.2	Benchmarks für Dateisysteme und Web-Ein-/Ausgaben	592
6.8	Entwurf eines Ein-/Ausgabe-Systems	593
6.9	Parallelismus und Ein-/Ausgabe: RAID	594
6.10	Fallstudie: Sun Fire x4150 Server	602
6.11	Fortgeschrittene Themen: Netzwerke	608
6.12	Fallstricke und Fehlschlüsse	609
6.13	Schlussbetrachtungen	614
6.14	Historische Perspektiven und Literaturhinweise	615
6.15	Aufgaben	616

7 Multikernprozessoren, Multiprozessoren und Cluster	627
7.1 Einführung	627
7.2 Die Schwierigkeit, parallel arbeitende Programme zu entwickeln	630
7.3 Multiprozessoren mit gemeinsam genutztem Speicher	634
7.4 Cluster und andere Multiprozessoren mit Nachrichtenaustausch	636
7.5 Hardware-Mehrfädigkeit (Multithreading)	640
7.6 SISD, MIMD, SIMD, SPMD und Vektor	644
7.6.1 SIMD in x86: Multimedia-Erweiterungen	645
7.6.2 Vektor	646
7.6.3 Vektor im Vergleich zu Skalar	648
7.6.4 Vektor im Vergleich zu Multimedia-Erweiterungen	649
7.7 GPUs (Graphics Processing Units) – Einführung	650
7.7.1 NVIDIA GPU-Architektur – Einführung	653
7.7.2 GPUs in ihrer Umgebung	656
7.8 Einführung in Multiprozessor-Netzwerktopologien	657
7.9 Multiprozessor-Benchmarks	660
7.10 Roofline: Ein einfaches Leistungsmodell	663
7.10.1 Das Roofline-Modell	665
7.10.2 Vergleich von zwei Opteron-Generationen	667
7.11 Fallstudien: Benchmarks für vier Multicores unter Verwen- dung des Roofline-Modells	672
7.11.1 Dünn besetzte Matrix (Sparse Matrix)	676
7.11.2 Strukturiertes Gitter	677
7.12 Fallstricke und Fehlschlüsse	680
7.13 Schlussbetrachtungen	683
7.14 Historische Perspektive und Literaturhinweise	684
7.15 Aufgaben	685
Fachbegriffe Deutsch – Englisch	701
Englisch – Deutsch	701
Deutsch – Englisch	704
Sachverzeichnis	707